Structural Health Monitoring Meets Data Mining

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 16 December 2014
klokke 12.30 uur

door

Shengfa Miao

geboren te Shandong, China
in 1981

Promotiecommissie

Promotor:	prof. dr. J.N. Kok	
Co-promotor:	dr. A.J. Knobbe	
Overige leden:		
prof. dr. A.P.J.M. Siebes		(University Utrecht)
prof. dr. E.A.B. Koenders		(Technische Universität Darmstadt)
dr. J. Hollmén		(Aalto University)
dr. W.A. Kosters		
prof. dr. A. Plaat		
prof. dr. T. Bäck		
dr. A. Koopman		(ASML)

Cover photos: The Hollandse Brug (bridge) and a Correlation Matrix. The photo of the Hollandse Brug is taken by Gouwenaar, published under the Creative Commons CC0 1.0 Universal Public Domain Dedication; the Correlation Matrix is derived from correlations among 145 signals collected with 145 sensors installed on the Hollandse Brug.

This research is financially supported by the Chinese CSC and the Dutch funding agency STW, under project number 10.970 (the Infrawatch project).

Contents

Chapter 1

Introduction

1.1 Background

Over the last decade, assessing the service-life of concrete civil structures is a theme that has gained a lot of interest. Despite concrete being a construction material that can last several decades to centuries, it has become clear that external influences may substantially (and often unexpectedly) shorten the service-life of concrete structures. More in detail, the factors that affect the service-life of civil structures have various origins, such as traffic load, varying climate conditions as well as the natural degradation of the material involved, notably the concrete and the reinforcement bars.

The traditional way to assess the actual condition of infrastructural assets is based on visual inspection or portable instruments, an approach which suffers from the following drawbacks:

- It is fairly subjective and difficult to quantify.
- It requires a lot of manpower, material and equipment.
- It may have blind spots, and completeness cannot be guaranteed.
- Its inspection period is long and inefficient.
- It interferes with the normal flow of traffic.

According to a recent survey from the US Federal Highway Commission [1], on average 56% of the assessments made by visual inspection are inappropriate. Driven by these drawbacks, the field of *Structural Health Monitoring* (SHM) is emerging, which is an interdisciplinary field, including civil engineering, signal processing, sensor technology, material sciences, data management and mining. The SHM process can be approached from a Statistical Pattern Recognition paradigm [2, 3], which employs an array of sensors to periodically collect the dynamic response of the monitored structure, and assesses the system's health, with damage-sensitive features extracted from these measurements.

In this thesis, we discuss results from a Dutch SHM project, the so-called *InfraWatch* project. The project is one of the key projects of a Dutch STW perspective program, called *Integral Solutions for Sustainable Construction* (IS2C). The IS2C program is composed of nine research projects, aiming to enforce new innovations in the state-of-the-art service-life assessment and to set a new standard for sustainable construction. The InfraWatch project covers the aspects of sensing, monitoring and degradation mechanisms, and is a joint research project between Leiden University and Delft University of Technology. The data used for this project is captured by a monitoring system that is installed at a major highway bridge in the Netherlands, called the *Hollandse Brug*. The computational data analysis and data mining has been conducted by researchers at Leiden University, while the physical interpretation and matching with structural analysis models was conducted at Delft University of Technology.

1.2 Objectives and Scope

According to Farrar and Sohn's approach [2, 3], the SHM process can be broken down into four parts:

- Part 1: Operational Evaluation: damage definition, life-safety, economic justification, operational and environmental conditions and limitations are considered in this step.

- Part 2: Data Acquisition, Fusion and Cleansing: excitation methods, structural response and data transmission are considered in this step.
- Part 3: Feature Extraction and Information Condensation: this step focuses on selecting features that indicate the health of the structure, such as natural frequencies, damping ratios and mode shapes.
- Part 4: Statistical Model Development for Feature Discrimination: this step aims to design algorithms to distinguish between features from the undamaged and damaged structures.

Part 1 and Part 4 are beyond the scope of this thesis. The bridge in our project was closed for renovation in 2007, and since then a sensor network has been installed on the bridge. InfraWatch started two years after the renovation and the sensor network installation, so the operational evaluation step is skipped in this work. Since the installation of the network, three years of data has been collected. During this period, it is reasonable to assume that the bridge has not suffered any major damage, so the damage identification in Part 4 is not covered in this thesis.

In this thesis, we focus on Part 2 and Part 3 in the above paradigm. We are interested in understanding the specifics of each sensor type and individual sensors, and the dependencies between sensors of different types. Each dataset collected with an individual sensor produces a time series, which is sensitive to several external factors, most notably daily traffic and temperature variations. Some of these factors are useful, like truck events, which help to excite (bring in motion) the bridge, while some of them are interference factors, like temperature influence, which hinder the proper analysis. To select and extract reliable features from massive datasets, we employ a number of signal processing and data mining techniques, which are briefly introduced in the following subsections.

1.2.1 Data Acquisition and Signal Processing

In SHM, one can distinguish two general excitation methods [3] for data acquisition: the *forced* and the *ambient* excitation method. With the forced excitation

method, the input forces are controllable and measurable (such as with an impact hammer, a shaker, or a controlled load such as a weighed truck), so it is easy to obtain a clear and interpretable signal. This method is usually adopted in laboratory tests or to obtain short-term data from a real structure in the field. In contrast, with the ambient excitation method, it is hard to measure the input forces accurately, because these forces are usually varying and occur at random intervals. However, this method is suitable for long-term monitoring of structures.

Our InfraWatch project is based on the ambient excitation method for data acquisition. There are three sensor types involved in the sensor network, measuring strain, vibration and temperature. The strain sensors indirectly measure the load of the bridge by measuring strain experienced parallel to the bridge, in two horizontal directions. Strain measurements are sensitive not only to the experienced load of the bridge (which will make the structure bend), but also to a large extent to temperature effects. The vibration sensors measure vertical shaking of the bridge, caused by the impact and passing of traffic. Contrary to strain gauges, they are hardly sensitive to temperature changes. The temperature sensors measure the local temperature of the bridge, at the exact point where they are attached to the bridge. This temperature may vary a bit, depending on the location of the bridge.

On the Hollandse Brug, we can generally recognise three different phenomena at different time scales. This shows up as three different components in the strain signal: a low-frequency component, a medium-frequency component and a high-frequency component. The low-frequency component includes effects such as the daily temperature fluctuations or traffic jams. These effects will show up in the strain signal as a drifting baseline. Removing the low-frequency component from the signal is known as *baseline correction* [4]. A good baseline correction method helps to study useful patterns hidden in the raw time series. The medium-frequency component consists of normal traffic events, which are considered useful patterns in this work. The high-frequency component consists of noise (for example caused by the rolling tires of vehicles or measurement noise), which can be eliminated using smoothing methods.

Although the strain signal seems more informative than the vibration signal when considering the time domain, it turns out to be less useful in the frequency domain. Especially when considering details of the vibrations caused by (heavy) vehicles, which is captured to varying degrees by both sensors, the vibration signal is more clear, and not affected by the baseline drift. By combining the strain and the vibration signals, we succeeded in developing a method to select high-quality data for feature extraction.

1.2.2 Feature Extraction

The integrated performance of the bridge can be studied through so-called *modal parameters*: natural frequencies, damping ratios and mode shapes [5, 6, 7, 8]. In this thesis, we are interested in the following topics:

- The selection of high-quality datasets.
- Modal analysis methods.
- The influence of temperature on modal parameters.
- The influence of traffic mass on modal parameters.

Based on the understanding of the different behaviour of the strain and vibration sensors, we select the vibration sensor as our target sensor type for modal analysis. To get rid of the influence of traffic mass, we select datasets during so-called *free-vibration periods* as our target datasets. The free-vibration period is the period right after a vehicle has passed, and before a next vehicle appears on the bridge. The reason for choosing this period is that the bridge is put in motion by the vehicle, but the actual weight does not influence the frequency of vibration after the vehicles has disappeared, nor do any other vehicles.

A number of methods have been developed for modal analysis, such as structural calculation methods (Finite Element Method (FEM)), the *Peak-Picking* method (PP) [9, 10] and the *Stochastic Subspace Identification* method (SSI) [11, 12, 13, 14, 15]. We extract modal parameters by combing these modal

analysis methods, and pay special interest to the relationship between natural frequencies and temperature.

1.3 Thesis Outline

This thesis is composed of eight chapters. Most of these chapters are based on published papers by the author. The following provides a brief description of each chapter.

Chapter 2 presents some basic concepts that play a role in the remainder of the thesis.

Chapter 3 presents an introduction to the InfraWatch project. In this chapter, we introduce the bridge and the sensor network in detail. Parts of the content in this chapter were previously published in the following paper:

> Veerman R., Miao S., Koenders E., and Knobbe A. *Data-Intensive Structural Health Monitoring in the InfraWatch Project.* In Proceedings of the 6th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII6), Hongkong, 2013.

Chapter 4 explores sensor dependencies among multiple sensor types. All the sensors in the sensor network are sensitive to related aspects of the measured system, that is to say there are certain dependencies. To gain insight into these dependencies, and how the placement and location of sensors influences them, we employ linear regression, convolution, envelope and band pass filters to model signals in both the time and the frequency domain, and then utilise Subgroup Discovery [16, 17, 18] to further analyse the obtained models. This work was published in the following paper:

> Miao S., Vespier U., Vanschoren J., Knobbe A., and Cachucho R. *Modeling Sensor Dependencies between Multiple Sensor Types.* In Proceedings of BeneLearn, Nijmegen, 2013.

Chapter 5 looks into the problem of baseline drift. To separate the influence of normal traffic events from other environmental factors, we propose a novel baseline correction method, the *Most-Crossing* method, which is a piece-wise method, based on probability theory. The method assumes that patterns of the same scale follow the same probability distribution, so that patterns of different scales can be distinguished based on their probability distributions. In strain signals of the sensor network, the probability distribution of environmental factors, which contribute to baseline, is different from that of traffic events. Based on this observation, we propose the Most-Crossing method to extract the baseline from strain signals. This work was previously published in the following paper:

> Miao S., Koenders E., and Knobbe A. *Automatic Baseline Correction of Strain Gauge Signals*. In Structural Control and Health Monitoring 22 (1), pp. 36-49, 2015.

Chapter 6 covers the topic of *predefined pattern detection*. Given a pattern (template), we can characterise it as a combination of landmarks and constraints. Landmarks are remarkable points in the pattern, e.g., local extrema. Constraints are composed of local constraints and global constraints. The former focus on properties of individual landmarks, and the latter focus on relationships between properties of different landmarks within the pattern. If the prior knowledge is given to us by domain experts, the pattern detection procedure can be addressed as a predefined pattern detection issue. Predefined pattern detection has its advantage in processing huge datasets collected from a specific domain. It will be extremely expensive to detect patterns with traditional pattern detection methods, which work through all possible pattern lengths. What's more, most of the existing pattern detection methods focus on full sequence matching, that is, sequences with clearly defined beginnings and endings, where all data points contribute to the match. These methods will become ineffective when deformations appear in both temporal and amplitude dimensions. This work has been submitted to the journal of Information Sciences:

> Miao S., Vespier U., Meeng M., Cachucho R., and Knobbe A. *Predefined Pattern Detection in Large Time Series*. revised to Information Sciences, 2014.

Chapter 7 presents modal analysis of the bridge. Changes in the integrity of the material and/or structural properties of structures are known to adversely affect their performance, which can be observed from structures' dynamic response. We propose a procedure to select high-quality datasets, and employ two modal analysis methods to extract modal parameters from them. We also look into the influence of environmental factors, such as traffic mass and temperature, on modal parameters. This work was published in the following papers:

> Miao S., Veerman R., Koenders E., and Knobbe A. *Modal Analysis of a Concrete Highway Bridge — Structure Calculations and Vibration-Based Results*. In Proceedings of the 6th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII6), Hongkong, 2013.
>
> Miao S., Knobbe A., Koenders E., and Bosma C. *Analysis of Traffic Effects on a Dutch Highway Bridge*. In Proceedings of the International Association for Bridge and Structural Engineering (IABSE), Rotterdam, 2013.

Chapter 8 concludes the research involved in this thesis, and presents a number of recommendations for further work.

Chapter 2

Preliminaries

2.1 Background

In this chapter, we will review and explain a number of concepts to help better understand subsequent chapters. We begin by giving definitions related to the sensors involved in the sensor network, and then introduce concepts related to datasets collected with sensors.

In the InfraWatch project, a sensor network consisting of 145 sensors is employed. These sensors are placed along three cross-sections of a single span of the bridge[1]. Each of the sensors is either embedded in the concrete, or attached to the outside of the deck and girders. To measure the forces in different directions on the bridge, we utilize sensors of different types: *vibration sensors* measure vertical motion of the bridge, and *strain gauges* measure horizontal strain caused by deflection of the bridge. To measure the temperature of different parts of the bridge, we also employ multiple *temperature sensors*. To formalise this placement, we define each sensor as follows:

Definition 1 (Sensor) *A sensor is a tuple* $(type, x, y, e, o)$*, where* $type \in \{St, Vi, Te\}$ *indicates the sensor type (strain, vibration, and temperature, respectively),*

[1]The bridge has multiple spans, but they are identical in design, and independently constructed.

x and y are its coordinates on the bridge, $e \in \{embed, attach\}$ *indicates whether the sensor is embedded or attached to the concrete, and* $o \in \{X\text{-}axis,\ Y\text{-}axis\}$ *indicates the orientation of the sensor.*

The sensor network is collecting data at 100 Hz, and each sensor in the network produces a sequence of data-points, which forms a *time series* of measurements. We define a time series as:

Definition 2 (Time Series) *A time series* $\boldsymbol{T}$ *is an ordered sequence of n real values*

$$\boldsymbol{T} = (t_1, t_2, \ldots, t_n), \quad t_i \in \mathbb{R}$$

in which $t_i \in \mathbb{R}$ stands for the i^{th} item in the sequence collected by a sensor. In this thesis, we will often also refer to data produced by a single sensor over time as a *signal*. When we speak of a signal, we are typically interested in general aspects of the data (such as the main frequency), whereas when we speak of a time series, it typically refers to a specific sequence of data measured over a specific interval of time.

Instead of the whole time series, in some cases, we are more interested in part of a time series. For example, given a time series recording traffic events of one whole day, we may just want to know the traffic situation during rush hour. Here we define a piece of a given time series as a *subsequence*, and define it formally as:

Definition 3 (Subsequence) *Given a time series* $\boldsymbol{T} = (t_1, \ldots, t_n)$ *of length n, a subsequence* $\boldsymbol{S}$ *of* $\boldsymbol{T}$ *is a series of length* $m \leq n$ *consisting of contiguous data points from* $\boldsymbol{T}$

$$\boldsymbol{S} = (t_k, t_{k+1}, \ldots, t_{k+m-1}), \quad 1 \leq k \leq n - m + 1$$

In the following sections, we will introduce some operations related to the concepts mentioned above.

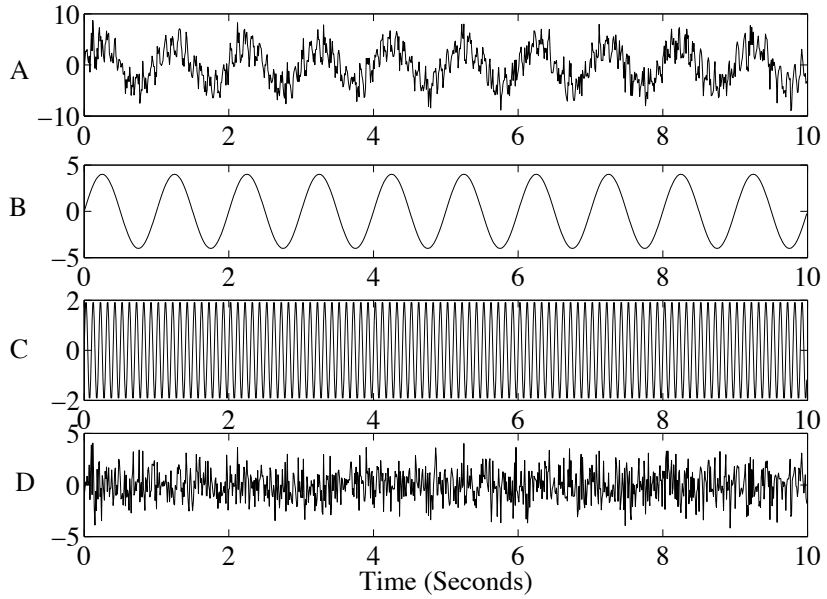

Figure 2.1: A signal in the time domain - The pictures illustrate a simulated signal (A) and its components (B, C, D) in the time domain.

2.2 Fourier Transform

The top picture (A) in Fig. 2.1 shows a signal in the time domain, which is sampled at 100 Hz (100 data points each second). As mentioned in the previous chapter, the signal can be generally decomposed into three components: a low frequency component (B), a high frequency component (C) and random noise (D). One method that is used to convert a signal from the time domain to the frequency domain, essentially extracting spectral information from the signal, is the Discrete Fourier Transform (DFT) [19]. The transform is defined as:

Definition 4 (Discrete Fourier Transform (DFT)) *Given a sequence of N samples $\{x_0, x_1, \ldots, x_{N-1}\}$, the Discrete Fourier Transform is defined as:*

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \quad k \in \mathbb{Z}$$

in which $\mathbb{Z}$ are integers [20].

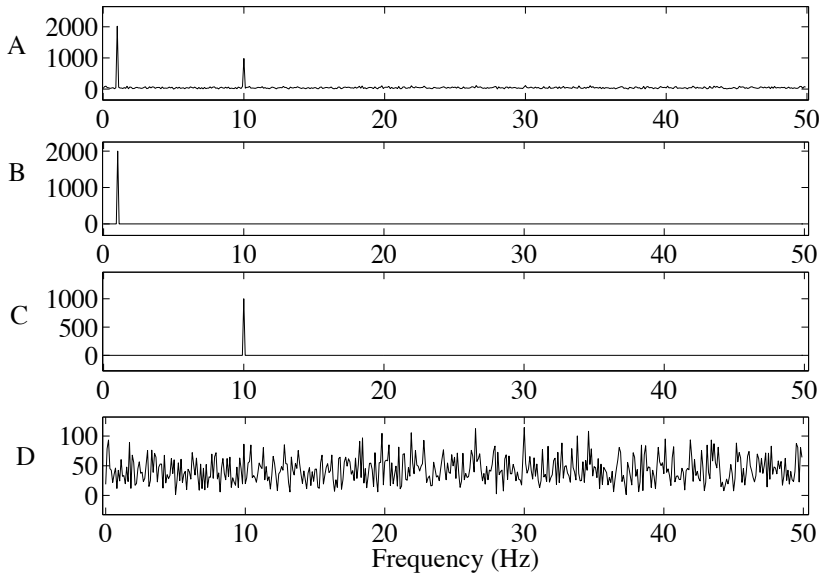

Figure 2.2: A signal in the frequency domain - The pictures illustrate a simulated signal (A) and its components (B, C, D) in the frequency domain.

An efficient way to implement DFT is the Fast Fourier Transform (FFT) [21]. With the FFT, the signals in Fig. 2.1 can be transformed into four spectra, as shown in Fig. 2.2. Spectrum A corresponds to signal A, which consists of two dominant peaks, corresponding to the two components present. The components B and C are noise-free signals, so the corresponding spectra consist of single peaks. Component D is a non-periodic signal, so there are no clear peaks in its spectrum D.

2.3 Convolution

Apart from the subtle degradation of the structure which we ignore for the moment, a bridge can be viewed as a *Linear Time-Invariant* (LTI) system [22]. Here, *time-invariant* indicates that the nature of the response of the system does not change over time. LTI systems are *linear* because their 'output' is a linear combination of the 'inputs'. The behaviour of an LTI system with single input signal $x(n)$ and single output signal $y(n)$ can be represented as a discrete convolution

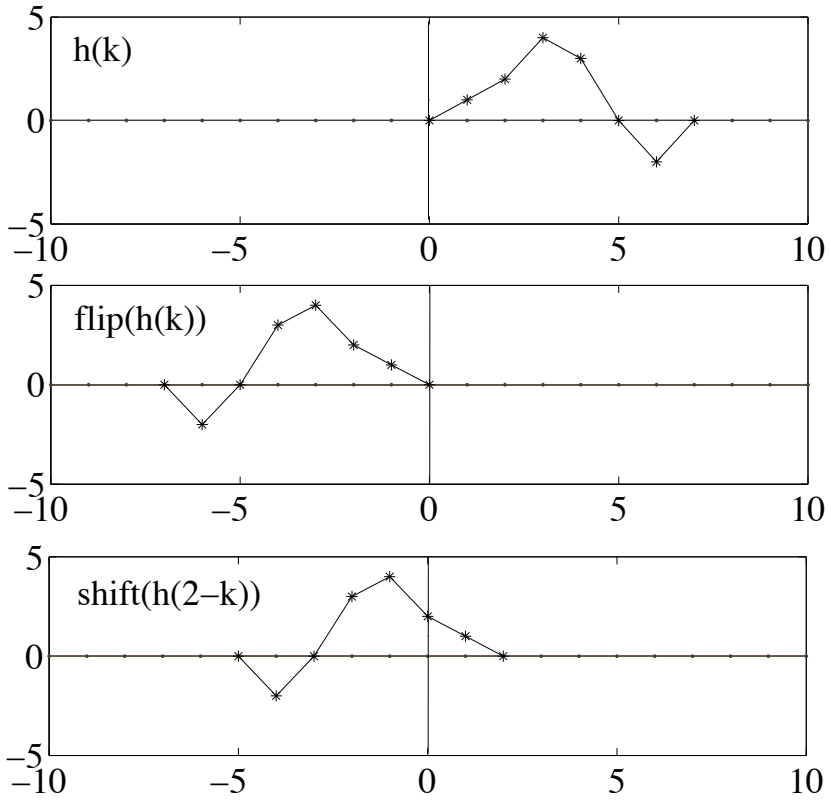

Figure 2.3: Graphical illustration of convolution - The top picture is the impulse response signal. The middle picture illustrates the flip operation. The bottom picture demonstrates the shift operation, which is obtained by shifting the flipped impulse response signal by 2 to the right.

integral [23], which is defined as:

$$y(n) = (x * h)(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.1}$$

in which $h(n-k)$ is impulse response signal. The convolution summation can be graphically interpreted as a combination of two operations: a flip and shift. For a given impulse response signal $h(k)$, shown as the top picture in Fig. 2.3, we first flip the impulse response signal (the middle picture in Fig. 2.3), then shift the flipped impulse response signal with n data points (the bottom picture in Fig. 2.3 is obtained by shifting the flipped $h(k)$ by 2 to the right).

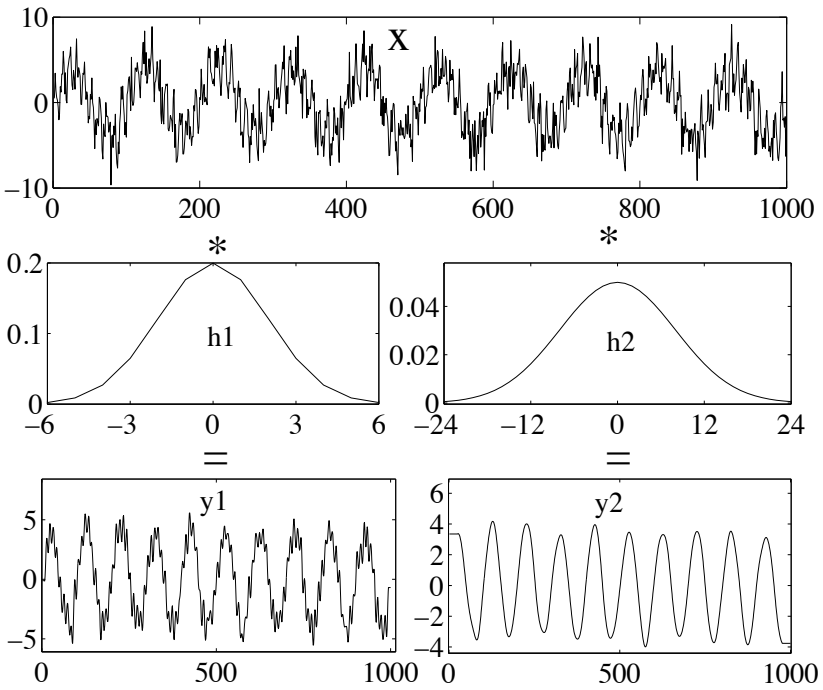

Figure 2.4: Convolution for low-pass filtering - The top picture is the input signal. The middle left curve is a small Gaussian kernel with $\sigma = 2$. The middle right curve is a big Gaussian kernel with $\sigma = 8$. The bottom graphs show the corresponding resulting signals using the small and big kernel. Note how the big kernel has a larger influence on the resulting signal, and most of the high-frequency component has been removed from the signal.

If the system is considered to be a filter, the impulse response signal is called a filter kernel. One of the widely used kernels is the *Gaussian* kernel, which resembles a "bell curve". The normalised *Gaussian* kernel is defined as:

$$\mathbf{G}(\sigma, x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \tag{2.2}$$

where the parameter σ controls the width of the "bell".

The Gaussian kernel can be used for low-pass filtering. Shown as the pictures in Fig. 2.4, the input signal X is the same signal as the top picture in Fig. 2.1. When the input signal X is convolved with a small Gaussian kernel $h1$ ($\sigma = 2$), the output signal $y1$ preserves both the low and middle frequency components,

while suppressing the influence of high frequency noise. When the input signal X is convolved with a big Gaussian kernel $h2$ ($\sigma = 8$), the output signal $y2$ only preserves the low frequency component.

2.4 Similarity Measurement

Given two time series (or subsequences) of interest, we may want to know how similar they are. A number of similarity measurements have been proposed [24], of which the Euclidean Distance (ED) [25, 26] is the most common [27, 28]. Given two time series $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, their ED-based similarity can be obtained by comparing local point values. The ED between P and Q can be defined as:

$$D(P, Q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

The advantage of the ED is that it is simple and efficient. When the given time series are well aligned, like the left picture in Fig. 2.5, the ED works well. However, the ED is sensitive to scaling, and when shifting and temporal distortions exist in the given time series, like the right picture in Fig. 2.5, it is proven to be ineffective [28] as a similarity measure.

Time series might still be intuitively similar even though each series might be subject to a certain scaling. In this situation, the similarity can still be captured with the so-called *correlation measures*. The most well-known, Pearson's correlation coefficient [29] is defined as:

$$r = \frac{n \sum p_i q_i - \sum p_i \sum q_i}{\sqrt{n \sum p_i^2 - (\sum p_i)^2} \sqrt{n \sum q_i^2 - (\sum q_i)^2}}$$

The correlation coefficient r is always between -1 and 1, where 1 means that the two series are a strict linear function of one another, 0 means that they are completely uncorrelated, and -1 means that they are perfect opposites. Although this correlation also suffers from temporal shifting and distortions, it is invariant to scaling and translation (in the domain of measurement).

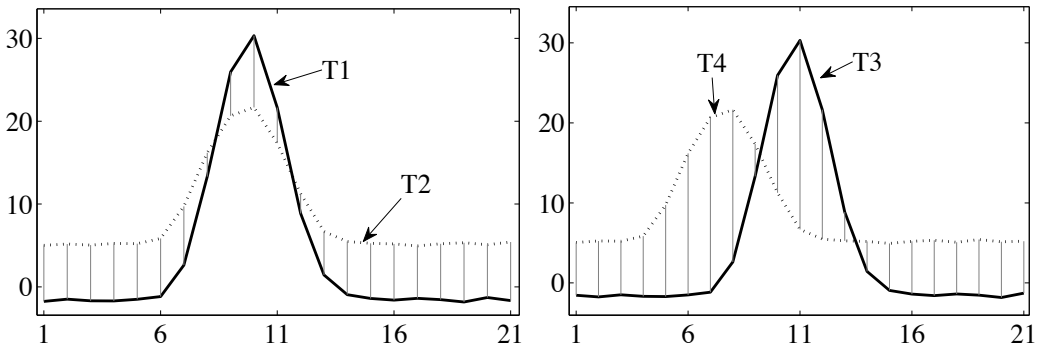

Figure 2.5: Euclidean Distance. - The left picture demonstrates the ED between two aligned time series T1 and T2; the right picture illustrates the ED between two shifted time series T3 and T4, where ED fails to capture the intuitive similarity.

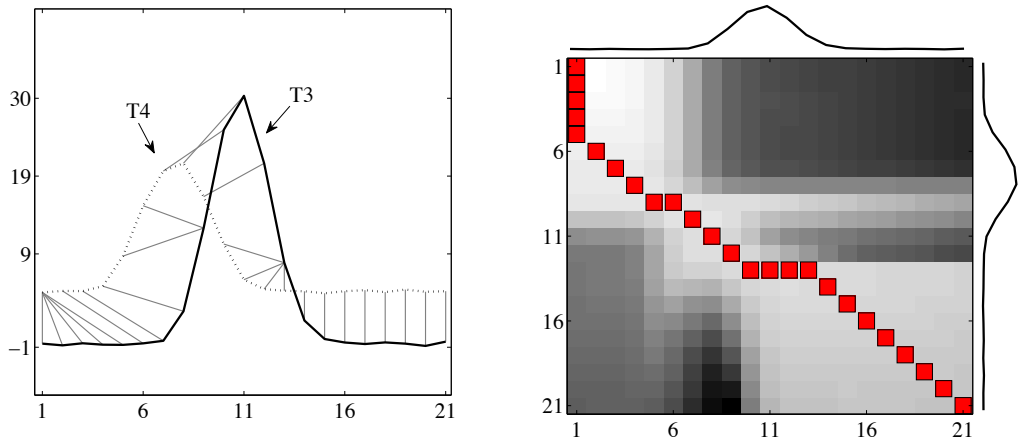

Figure 2.6: Dynamic Time Warping. - The left picture shows the DTW between two shifted time series T3 and T4; the right picture illustrates the accumulated distance matrix and the optimal matching path (a square-chain going through light grey cells), between time series T3 and T4.

To handle shifting, stretching and compression along the temporal dimension, Dynamic Time Warping (DTW) [30] was proposed, which achieves an optimal temporal alignment, shown as the left picture in Fig. 2.6, through detecting the shortest warping path in an accumulated distance matrix [24, 31, 32, 33], shown as the right picture in Fig. 2.6. Given two time series $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_m)$, the element $r(i, j)$ in the accumulated distance matrix [34] is defined as:

$$r(i,j) = \begin{cases} d(p_1, q_1) & i = 1, j = 1 \\ d(p_i, q_1) + r(i-1, 1) & j = 1, 1 < i \leq n \\ d(p_1, q_j) + r(1, j-1) & i = 1, 1 < j \leq m \\ d(p_i, q_j) + \min\{r(i-1, j-1), r(i-1, j), r(i, j-1)\} & 1 < i \leq n, 1 < j \leq m \end{cases}$$

where $d(i, j)$ is the distance found in the current cell, which can be chosen from several metrics, such as p-norms [35] (when p is 2, the distance becomes the Euclidean distance), and $r(i, j)$ is the cumulative distance of $d(i, j)$ and the minimum cumulative distances from three adjacent cells.

Finding the shortest warping path is a non-trivial problem, whose computation complexity is $O(n^2)$. To speed up the computation of DTW, some lower bounding constraints, such as LB_Keogh [32, 36] and the Ratanamahatana-Keogh Band [37], have been introduced to prune expensive computations, which can reduce the complexity to $O(n)$.

In Chapter 6, we propose a novel pattern detection method, based on landmarks and constraints. The method is capable of extracting predefined patterns efficiently, and is robust to temporal and magnitude deformations.

Chapter 3

The Infrawatch Project

The full name of the InfraWatch project is "Data Management of Large Systems for Monitoring Infrastructural Performance", which aims to design, develop and optimise a data management system for measuring and reporting the actual performance of large infrastructural projects. The datasets involved in this project are collected with a sensor network installed on a Dutch highway bridge. In this chapter, we will introduce some background information about the bridge and the sensor network, and give an overview of the specific focus of each sensor type.

3.1 Description of the Bridge

The bridge in this project, shown as Fig. 3.1, is called Hollandse Brug, which is a concrete bridge, built in the late sixties, and opened in 1969. This bridge forms the motorway connection between Amsterdam and the north-east of the Netherlands. The bridge is composed of 7 spans, with a total length of 354 meters. As shown in Table 3.1, each span contains 9 pre-stressed prefab girders, which are connected with in situ concrete and reinforcement steel in transverse direction. In addition, two in situ post-tensioned cross girders are present to reduce rotation and torsion of the girders. Dilatation joints are installed between the girders in

Figure 3.1: The Hollandse Brug - This is a bridge between the Flevoland and Noord-Holland provinces and is located at the place where the Gooimeer joins the IJmeer.

longitudinal direction. Due to this connection, the girders can deform freely, and imposed deformations do not influence the internal stresses.

In the last decades, the condition of the Hollandse Brug decreased dramatically, and after an inspection of TNO (a Dutch organisation for applied scientific research) in 2007, the bridge was considered 'unsafe'. Heavy traffic was blocked from the bridge until a necessary renovation was finished. During renovation, the width of the bridge was also increased with extra girders. Due to these girders, an extra traffic lane in both directions could be realised. In addition to the renovation and the extra girders, a sensor network was installed underneath the first span of the bridge.

3.2 The Sensor Network

The initial goal of the system was very much short-term, with an emphasis on monitoring the curing of the new concrete before re-opening of the bridge, and providing evidence for the renewed safety of the bridge in that period. As the

Table 3.1: Some Parameters of the Hollandse Brug.

Parameters	Value	Unit
Weight of the girders	2,820	kg/m
Number of girders	9	–
Weight of the bridge deck	500	kg/m
Width of bridge deck	34	m
Total bridge deck weight	42,380	kg/m
Elastic modulus	38,500	MPa
Total bending stiffness	$5.91 \cdot 10^{11}$	Nm^2
Girder length	50.55	m

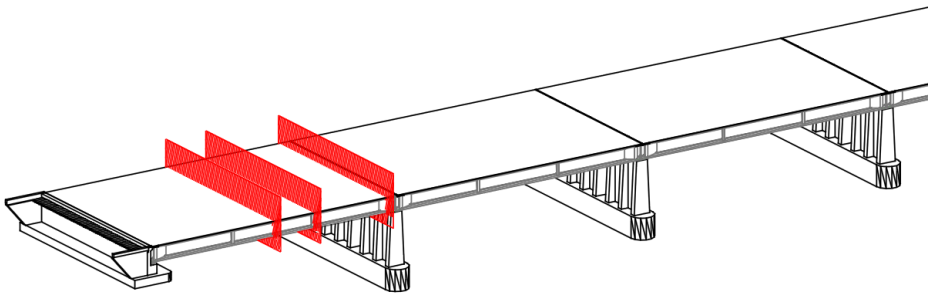

Figure 3.2: The layout of the sensor network on the Hollandse Brug - sensors are installed at three cross-sections within one span.

monitoring system was a major investment, it was then decided to make the system available to the publicly funded InfraWatch project, allowing research into the monitoring of infrastructure and the ageing of concrete bridges. The sensor system has been handed over to the TU Delft and its collection of historic data is available within InfraWatch and the IS2C program as a whole.

The sensor system is installed on one of the spans of the bridge, measuring over 50 meters in length. A total of 145 sensors are placed along the width of the bridge, at three cross-sections of the span. Furthermore, a weather station and a camera were installed. At each cross-section, sensors of various types are placed at a variety of locations, for example attached to the bottom of the deck, under a girder, or embedded in the deck. Furthermore, sensors (especially the strain

Figure 3.3: Sensor locations of one of the three cross-sections - Sensors are either attached or embedded to the deck and girders of the bridge.

gauges) are placed in various orientations, such that the total network senses the bridge in many different dimensions and from a range of perspectives, shown as pictures in Fig. 3.2 and Fig. 3.3. The sensor network features a total of 91 strain gauges, 44 of which are embedded, and 47 are attached. Furthermore, there are 34 vibration sensors, as well as 20 temperature sensors. The sensors are connected to five data-collection computers, sampling data at 100 Hz, and this data is finally recorded on-site in a central computer.

3.3 The Specific Focus of each Sensor Type

The loadings on the bridge not only contain vehicles with various weights, lengths, speeds, and directions, but also include environmental factors such as wind, temperature, rain and so on. The duration of the loadings varies from a few seconds to a couple of hours, or even longer. To show the specific focus of each sensor type, we choose two datasets of different scales: 5 minutes and 24 hours.

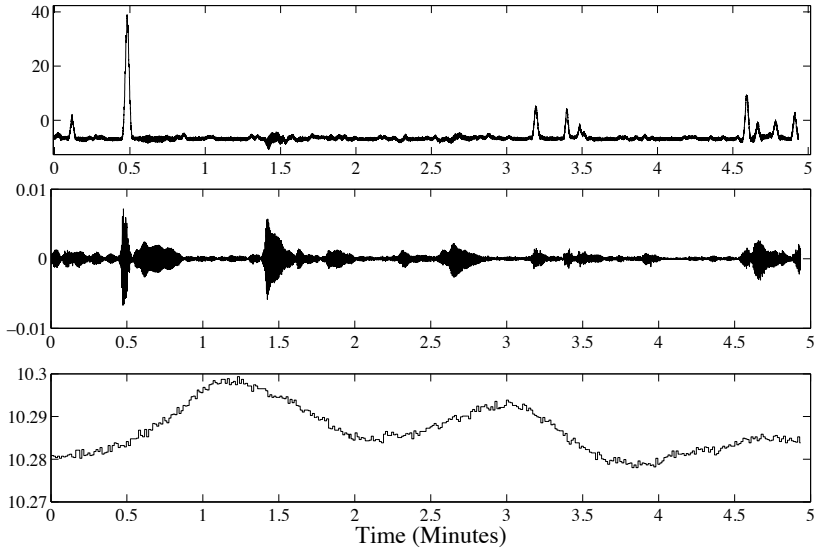

Figure 3.4: Signals of 5 minutes collected with sensors installed on the left side of the bridge - Top: a strain signal; middle: a vibration signal; bottom: a temperature signal.

3.3.1 The Specific Focus of Small Scale

To explore the specific focus of each sensor type on small scales, we choose a dataset of 5 minutes, including 30,000 data points, with a sampling rate of 100 Hz. Fig. 3.4 shows a group of signals collected with a group of sensors installed on the left side of the bridge, and Fig. 3.5 shows a group of signals collected with a group of sensors installed on the right side of the bridge.

The top pictures in the figures mentioned above are strain signals, collected with two different strain sensors, in which small peaks are caused by light vehicles, and big peaks are caused by heavy vehicles. We also notice that strain sensors are more sensitive to vehicles passing by lanes on the same side. For example, there is a group of peaks around 1.5 minutes in the strain signal of Fig. 3.5 (on the right side), but during the same period, we fail to detect any peaks in the strain signal of Fig. 3.4 on the left side.

The middle pictures in the figures mentioned above are vibration signals, collected with two different vibration sensors, which are sensitive to vehicles passing on

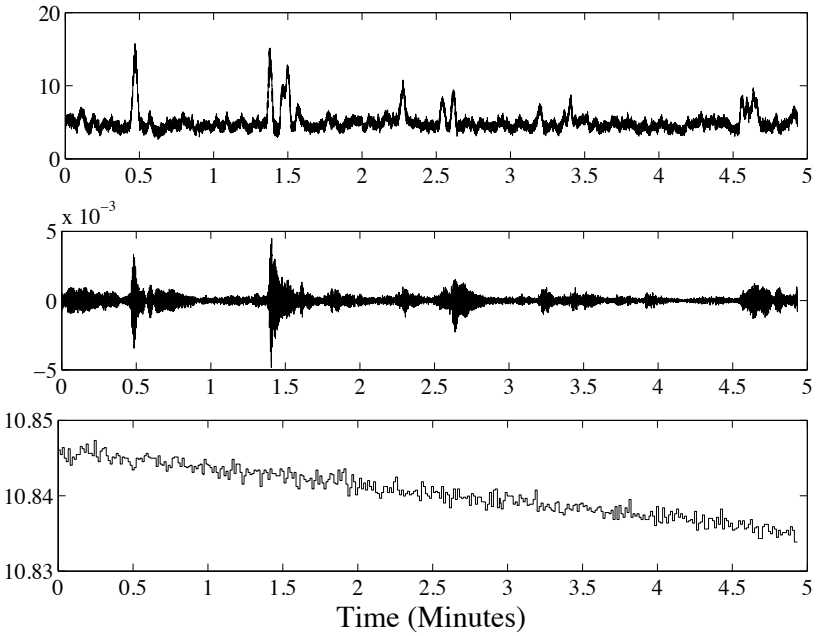

Figure 3.5: Signals of 5 minutes collected with sensors installed on the right side of the bridge - Top: a strain signal; middle: a vibration signal; bottom: a temperature signal.

both sides of the bridge. Compared with peaks in strain signals, the fluctuations in vibration signals last longer, and are capable of catching free vibrations of the bridge.

The bottom pictures of the figures mentioned above are temperature signals, collected with two different temperature sensors, which indicate local temperature changes of the bridge. The temperature sensor in Fig. 3.4 is embedded in the bridge surface, and the temperature sensor in Fig. 3.5 is attached to the bottom of the deck. We notice that temperature measurements vary with locations, and they are insensitive to traffic events.

3.3.2 The Specific Focus of Big Scale

To show the specific focus of each sensor type on large scales, we choose a dataset of 24 hours, composed of 8,640,000 data points, with a sampling rate of 100 Hz.

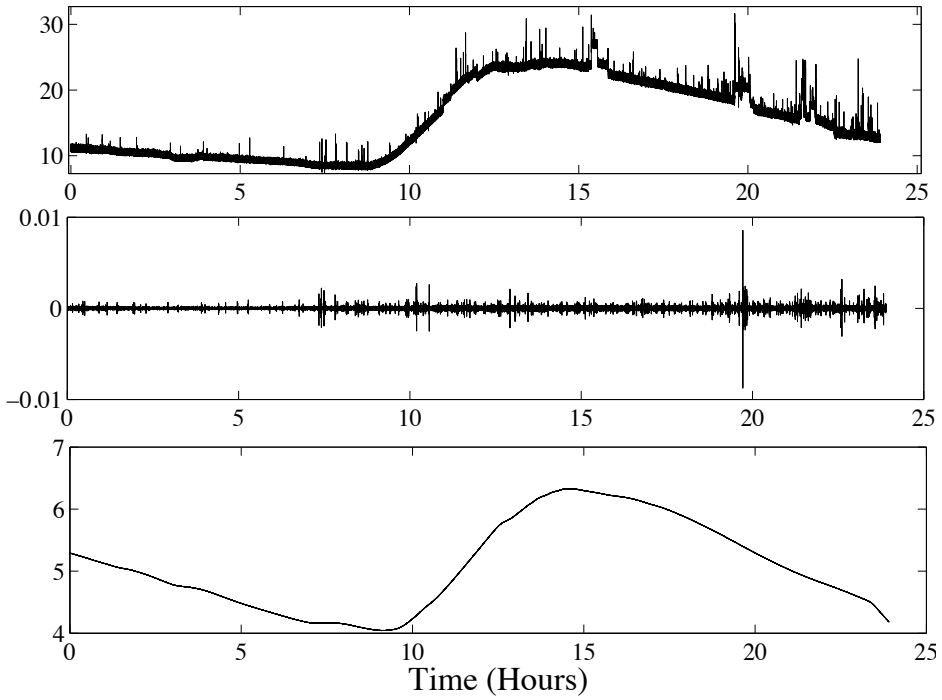

Figure 3.6: Sensor signals of big scale - Top: the strain signal of one day, in which tiny spikes are normal traffic events, jumps are traffic jams, and the baseline drift is temperature influenced; middle: the vibration signal of one day, in which small spikes are vehicles; bottom: the temperature signal of one day.

Fig. 3.6 shows a group of signals of different sensor types.

The signal in the top picture of Fig. 3.6 is a strain signal, which is composed of events of different scales. The tiny spikes in the strain signal represent normal traffic events, such as trucks and cars; the temporary jumps in the strain signal are caused by traffic jams; the slow big drift in the strain signal is caused by temperature changes, which is similar to the temperature variations shown in the bottom picture of Fig. 3.6.

The signal in the middle picture of Fig. 3.6 is a vibration signal, which is sensitive to normal traffic events, but is insensitive to long-term changes such as those due to traffic jams or temperature.

The signal in the bottom picture of Fig. 3.6 is a temperature signal, which rep-

resents temperature variations within 24 hours, and is insensitive to any traffic events.

Chapter 4

Sensor Dependencies among Multiple Sensor Types

With the rapidly decreasing prices for sensors, data gathering hardware and data storage, monitoring physical systems in the field is becoming a viable option for many domains. In fields such as civil engineering, windmills and aviation, so-called Structural Health Monitoring (SHM) systems are becoming popular to understand the actual workings of the system in situ, as well as to monitor the system for any developing faults. More and more, sensor networks consisting of multiple sensor types are being employed in these environments, and large quantities of data are being collected. New methods are required to deal with the proper analysis and interpretation of such data collections.

When dealing with multiple sensors measuring a physical system, each individual sensor will be sensitive to some aspects of the system, based on the specific characteristics of the type of sensor and on which part of the system the sensor is placed. This is clearly the case for sensors of different types (such as vibration and temperature sensors), but also for identical sensors attached differently to the system. If two sensors are measuring in each other's vicinity, they will likely show some dependency, but in most cases, this dependency will be non-trivial, depending on the location, the orientation and the attachment. As an example, consider an SHM-system employed on an aircraft. In order to measure stresses

on a wing, and potential metal fatigue on the wing attachment, strain gauges are fitted to the wing attachment. During high-g-force manoeuvres, the strain gauges will measure high values of strain on the attachment. Other sensors might be placed at the tip of the wing, to measure vibrations caused by turbulence for example. These vibration sensors however, will not be sensitive to sustained bending of the wing, as the sensor simply moves along with the wing, and is only sensitive to rapid changes in the location of the wing. As such, strain gauges are sensitive to different aspects of the dynamics than vibration sensors, although some overlap exists in the physical phenomena captured by either type.

In this chapter, we provide some examples of modelling the dependencies between (pairs of) sensors on the Hollandse Brug, specifically where multiple sensor types are involved. One of the main challenges here is to understand the specific focus of each sensor type, and to model any relationships across types. Having such a model may help, for instance, to remove certain phenomena measured by one sensor type from the signal of another sensor type. Specifically, we will consider the effect of temperature changes on the strain measurements at various locations on the bridge. As such, we can correct for this temperature effect.

Modelling dependencies between sensors also helps to remove redundancies in the data. Being able to infer the measurements of a particular sensor from the remaining sensor may suggest a smaller, and thus cheaper monitoring set-up. Finally, any modelling over the collection of sensors is beneficial for tracking the health of the bridge over longer periods. Changes in the value of a single sensor will often indicate transient effects, such as traffic or weather, but changes in the *models* of the bridge data indicate structural changes to the actual bridge, warranting further investigation.

A further issue we will be investigating is the effect that location and placement of sensors has on their usefulness within the network. For example, if we wish to understand the effect of temperature on strain measurements, it will be relevant to know where and how these two parameters are being measured. By investigating the dependencies between all pairs of sensors from two types (in this case strain and temperature), we hope to discover practical guidelines for the optimal placement of sensors. In Section 4.4, we use a meta-learning approach based on

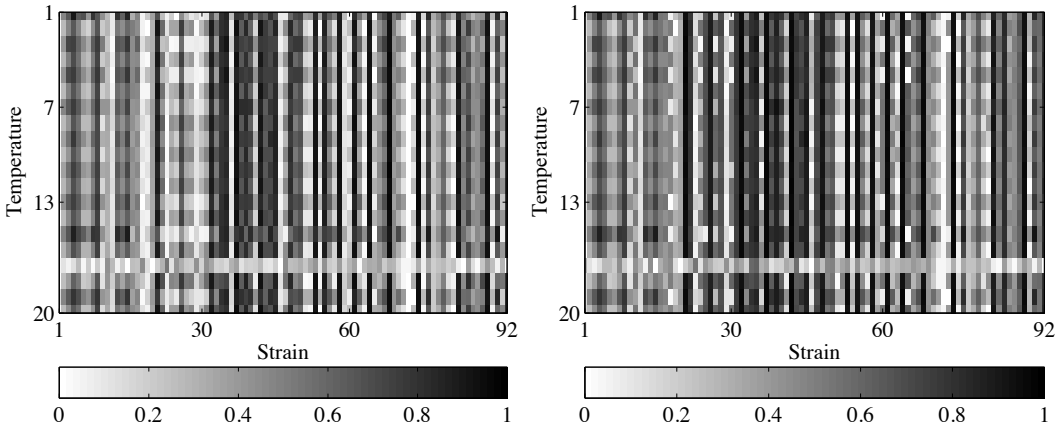

Figure 4.1: Correlation matrices between strain and temperature sensor types. - The numbers on the axes indicate the sensor number; the gray scale indicates the correlation between two sensors, 0 means no dependency, and 1 means strong dependency; St-Te before exponential moving average (left) and after exponential moving average (right).

subgroup discovery to find key characteristics of sensors in terms of their type, location, mode of attachment and orientation.

4.1 The Dependency between Strain and Temperature Sensors

In this section, we study the relationship between two types of sensor: strain and vibration. The sensor network features a total of 91 strain sensors, 44 of which are embedded (14 along X-axis, 28 along Y-axis), and 47 are attached (34 along X-axis, 13 along Y-axis). Of the 20 temperature sensors, one half is embedded in the surface of the deck, and the other half is attached to the underside of the deck.

In Fig. 4.1 on the left, the absolute correlation coefficients between strain and temperature vary from 0 to 0.97. For these sensor pairs with high correlation coefficients, we can simply employ a linear model that assumes the measured strain

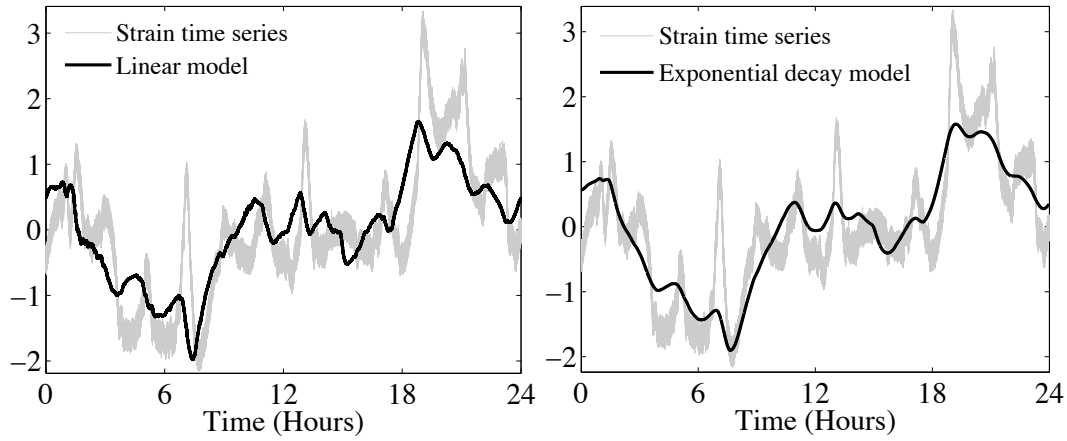

Figure 4.2: Models between strain and temperature time series. - The left picture shows the linear model between strain and temperature time series of 24 hours (100 Hz), in which there are a number of peak delays; The model in the right picture is obtained with an exponential decay model, which improves the delays in the linear model.

is directly influenced by the temperature of one of the temperature sensors:

$$S = a \cdot T + b \tag{4.1}$$

In this model, the coefficients a and b translate between the temperature scale (in Celsius) and the micro-strain scale (in $\mu m/m$). The left graph of Fig. 4.2 shows the effect of this model applied to two time series that are only moderately related, with $a = -3.288$ and $b = 27.547$ obtained through linear regression over a period of 24 hours. The correlation coefficient for this example is $r = 0.776$, which indicates that the selected pair of sensors are moderately correlated. However, when considering the time series in more detail, one can note that there is a dependency of the strain signal on the temperature measurements, but this relation is non-trivial: it involves a degree of delay: the upward and downward movement of the signal appear to be shifted by several hours.

The linear model fails to capture the complete effect of temperature on the strain, because the temperature sensor does not actually measure the bridge temperature, but rather the outside temperature. The temperature of the bridge is of

course mostly influenced by the outside temperature, but this influence is spread over time, and the bridge temperature will follow changes of outside temperature with a delay. The amount of delay depends on the size and material of the structure, with larger structures (such as the bridge in question) being less sensitive to sudden changes of outside temperature. In other words, a large concrete bridge has a large capacity to store heat, which is mirrored in a slow response of the strain signal.

In the systems analysis field, systems with a capacity are often modelled as a *Linear Time-Invariant* (LTI) system [22]. *Time-invariant* indicates that the response of the system does not change over time, which is a reasonable assumption for a bridge, if subtle deterioration of the structure is ignored. LTI systems are *linear* because their 'output' is a linear combination of the 'inputs'. In terms of the bridge, the temperature of the bridge is modelled as a linear combination of the outside temperature over a certain period of time (typically the recent temperature history):

$$T_{bridge}(t) = \sum_{m=0}^{\infty} h(m)T(t-m) \tag{4.2}$$

where $T_{bridge}(t)$ is the internal temperature and h is an impulse response (to be defined below). Note that this is a special case of convolution, which is defined in Chapter 2. Of the many impulse response functions h, which include for example the well-known moving average operation, we decide to model the delayed effect of the outside temperature using the exponential decay function $h_e(m) = e^{-\lambda m}$ (for $m \geq 0$). In this function, λ is the decay factor, which determines how quickly the effect of past values reduces with time. Note that the resulting equation

$$S = a \cdot h_e * T + b \tag{4.3}$$

where $h_e(m) = e^{-\lambda m}$, is the solution to a linear differential equation that is known as *Newton's law of cooling*, which states that the change in temperature of the bridge is proportional to the difference between the temperature of the bridge and its environment:

$$\frac{dT_{bridge}}{dt} = -r \cdot (T_{bridge}(t) - T(t)) \tag{4.4}$$

This is a somewhat simplified representation of reality, in that it assumes that the systems consists of two 'lumps', the bridge and the environment, and that within each lump the distribution of heat is instantaneous. Although in reality this is clearly not the case, it turns out that this model performs fairly well.

For a given pair of sensors and the associated data, we will have to choose optimal values for a, b and λ. It turns out that λ behaves very decently, with only a single optimum for given a and b, such that simple optimisation with a hill-climber will produce the desired result. For Equation 4.3, we obtain a fitted model for the selected sensor pair as shown in Fig. 4.2 on the right, which clearly demonstrates that the exponential decay model has removed the apparent delay in the data. The fitted coefficients were $a = -12.147, b = 30.463$, and $\lambda = 3 \cdot 10^{-5}$, with a correlation coefficient $r = 0.867$. Considering every possible pair of sensors from St and Te, we find that the correlation coefficients of 47.4% of sensor pairs are improved by the exponential decay model. Indeed, the successful modelling of the dependency for a given pair of sensors still depends on the location and placement of either sensor. In Section 4.4, we look into the question of finding suitable pairs of sensors in more detail, when we apply meta-learning to the modelling of St-Te sensor pairs.

4.2 The Dependency between Strain and Vibration Sensors

Our sensor network contains 34 vibration sensors, 15 of which are attached to the bridge deck, while the remaining 19 sensors are attached to the bridge girders. As mentioned in Chapter 3, both vibration and strain sensors are used to measure the dynamic stresses acting on the bridge. In theory, there should thus be some degree of correlation. However, we failed to detect a strong linear dependency between any pair. As illustrated in Fig. 4.3 (left), the correlations between most sensor pairs are quite weak, the highest one for this data being 0.1557. To demonstrate what types of modelling can be done for these two types of sensors, we selected one pair of sensors with a moderate correlation coefficient, as shown in the time

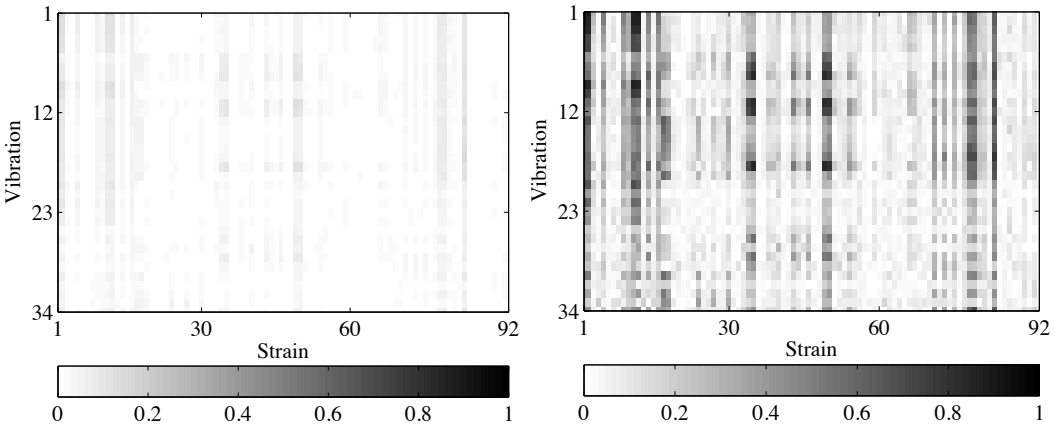

Figure 4.3: Correlation matrices between strain and vibration sensor types. - The correlation matrix in the left picture is obtained with raw strain and vibration time series of 5 minutes (100 Hz), in which the correlations are fairly weak; the correlation matrix in the right picture is obtained by applying a bandpass filter to the vibration time series, in which the correlations between several strain-vibration pairs are improved significantly.

domain in Fig. 4.4 (left top). The graphs show that the vibration sensor is a symmetric signal, while the strain sensor time series is not. However, the peaks in both occur consistently, which indicates that they are related. Using a simple correlation, this effect is hidden by the symmetric nature of the vibration signal.

In order to extract the amplitude of the vibration signal, which should correspond to the magnitude of the strain on the bridge, we apply an *envelope* operation. In the simplified situation where a signal consists of a single frequency s_f, modulated by another signal e as $s = s_f \cdot e$, we can simply obtain this envelope e by dividing s and s_f. However, in the presence of complex signals and noise, the envelope will have to be approximated by detecting peaks and interpolating between them. In our method, we define a peak as the maximum between two consecutive zero-crossings of the signal. To suppress the influence of noise, we first smooth the vibration time series with a small Gaussian kernel, and then interpolate adjacent peaks with piece-wise linear approximation. After being processed with the envelope operation, the vibration signal, shown in Fig. 4.4 (left bottom), shows

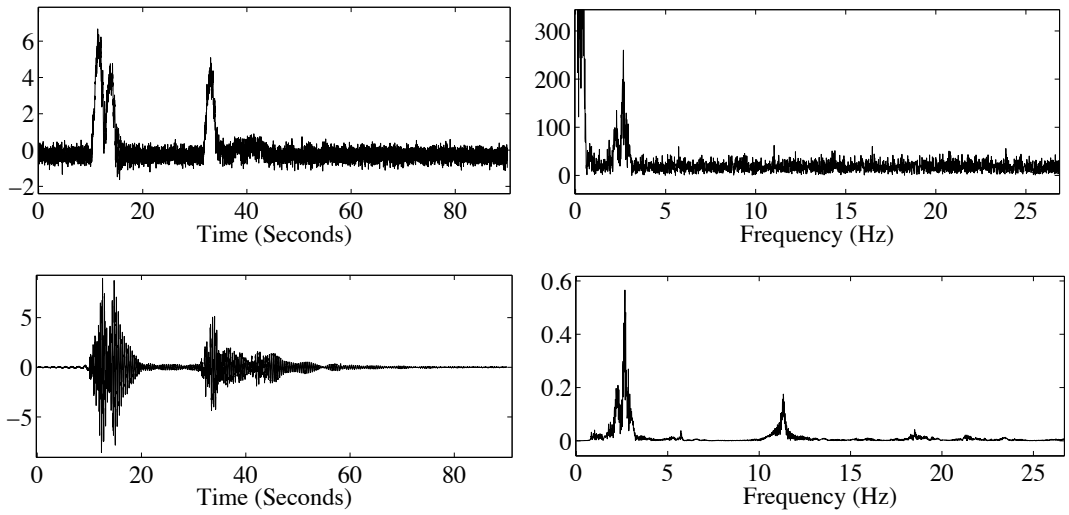

Figure 4.4: Strain and vibration time series in the time and frequency domain. - The left top picture is a strain time series of 90 seconds in the time domain; the right top picture is the spectrum of the strain time series; the left bottom picture is a vibration time series in the time domain; the right bottom picture is the spectrum of the vibration time series.

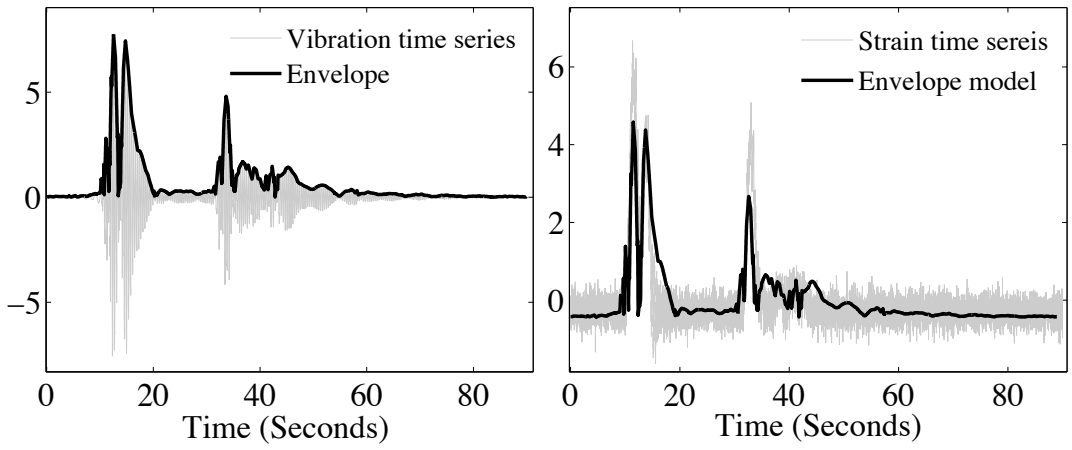

Figure 4.5: Envelope model - The envelope in the left picture is obtained by detecting peaks from vibration time series and interpolating them; the envelope model is obtained by shifting the envelope in the left picture by 105 data points, and then modelling it with the strain time series.

a better correlation with strain signal, improved from −0.16 to 0.44, as demonstrated in the left picture of Fig 4.5. By aligning the envelope, derived from the vibration time series, with the strain time series, we can obtain an improved envelope model, with a correlation of 0.80. We can also detect the dependencies between strain and vibration time series in the frequency domain.

Fig. 4.4 (right), which features the spectra obtained for the two signals, shows that despite a lack of a direct relation in the time domain, the signals are actually fairly similar in parts of the spectrum, notably where frequencies above 1 Hz are concerned. Note the big peak around 2.8 Hz in both spectra. In fact, what is missing in the vibration spectrum are the lower frequencies, which correspond to slower bridge movements. In other words, the vibration sensors are not sensitive to gradual changes in the deflection of the bridge, as the sensors themselves simply move along with the bridge. The strain gauges, on the other hand, *are* sensitive even to the slowest changes in bridge deflection. However, both sensors measure shaking of the bridge (frequencies above 1 Hz) in a similar fashion.

Based on these observations, an obvious way to relate St to Vi is to focus on a fairly specific range of frequencies. To obtain the boundaries of each well-matched spectrum range, we first transfer the spectrum of strain and vibration time series into one spectrum ratio signal by diving the strain spectrum with the vibration spectrum. Shown as the left picture in Figure 4.6, there are some "flat" ranges, which indicate that the spectral components of the strain and the vibration time series are similar during these ranges. We propose an approach to detect these flat spectrum ranges, based on sliding windows and the standard deviation. The standard deviation is used to measure the amount of variation from the average. A low standard deviation indicates a flat spectrum range. Given a time series $T = (t_1, t_2, \ldots, t_n)$, the standard deviation σ can be represented as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i - \mu)}, \text{ where } \mu = \frac{1}{n}\sum_{i=1}^{n} t_i$$

We employ a sliding window of a small length, which is 0.1 Hz (9 data points) in this work. Each sliding window can be featured with two parameters: the

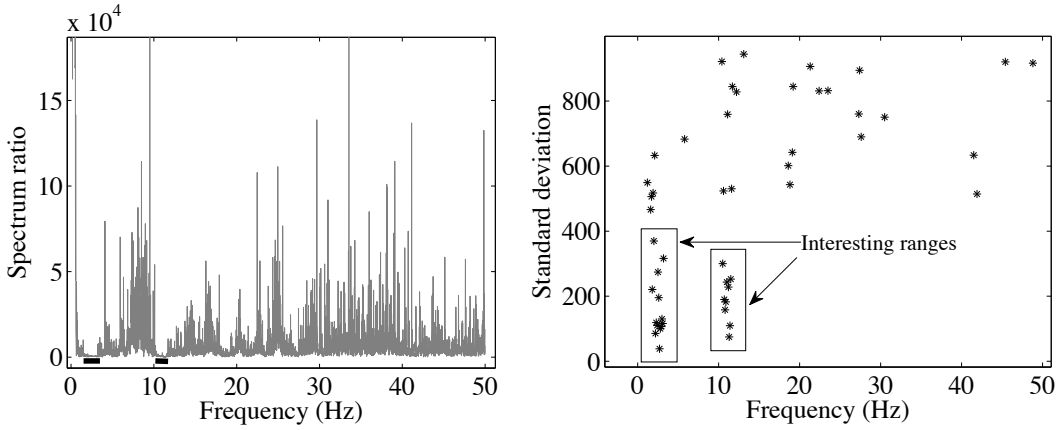

Figure 4.6: Interesting spectrum ranges detection - The left picture shows the spectrum ratio between the strain and the vibration time series. The right picture illustrates the spectrum ratio distribution, which is obtained by calculating the standard deviation of a sliding widow of 0.1 Hz (9 data points).

mean frequency and standard deviation. Given a vibration time series of 9,000 data points, we obtain 1,000 pairs of features. The scatter plot of these features is shown as the right picture in Fig. 4.6, in which there are two interesting spectrum ranges, whose standard deviations are below 400. The first spectrum range is between 2.0 Hz and 3.2 Hz, and the second one is between 10.5 Hz and 11.5 Hz.

In our experiments, we employ a *bandpass filter* to remove all spectral components outside the interesting ranges. The linear model between the strain and vibration time series then becomes:

$$\sum_{i=1}^{n} BPF_i(S) = a \cdot \sum_{i=1}^{n} BPF_i(V) + b \tag{4.5}$$

in which BPF_i stands for the band-pass filter operation on the ith frequency range, and n is the total number of interesting spectral ranges. After applying the band-pass filter operation to both St and Vi, the correlation coefficient improves from -0.16 to -0.90, as is shown in Fig. 4.7.

The model we achieved through the band-pass filter operation works well for a small selection of sensor pairs. In Fig. 4.3 on the right, information is displayed

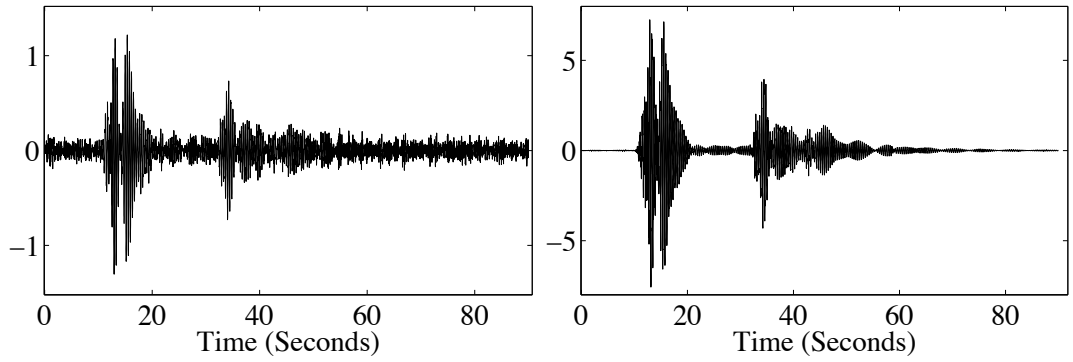

Figure 4.7: Bandpass filter model. - The left picture is the bandpass filter model derived from the strain signal. The right picture is the bandpass filter derived from the vibration signal.

on which sensor pairs specifically gain from this operation. Note that some strain gauges correspond well to most of the vibration sensors (dark columns in the matrix). These sensors are primarily located on the right-hand side of the bridge. The few exceptions are located on the girder entirely on the other side of the bridge. We look into such observations in more detail in the coming meta-learning section (Section 4.4).

4.3 The Dependency between Vibration and Temperature Sensors

As mentioned in the previous section, the vibration spectrum shows little activity in the range below 1 Hz, which happens to be where all of the temperature changes occur (for example due to the daily difference between day and night). For this reason, we do not expect significant dependencies between the sensors from Vi and Te. However, the vibration of the bridge does depend on the temperature. It is well known that bridges tend to oscillate at specific frequencies, and that these frequencies are determined by the stiffness of the structure, which in turn is influenced by changes in the temperature of the material. In a simplified model of a span of the bridge [38], the *natural frequency* f_n of the span is computed as

follows:

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{4.6}$$

In this equation, m refers to the mass of the bridge (including the possible load on the bridge), and k is a stiffness coefficient that depends on several factors such as material, humidity, corrosion, etc., but also on temperature. Note that an increasing temperature leads to a decreasing stiffness k, and hence a decrease in frequency, such that we expect a negative relationship between Vi and Te sensors.

The effect of temperature on natural frequencies is widely studied [39, 40]. After external excitation, for example traffic or wind, a bridge can vibrate in different *modes* [41]. Each mode stands for one way of vibration, which can be vertical, horizontal, torsional or more complicated combinations thereof, and there is one natural frequency corresponding to each. We will introduce a number of modal parameter extraction methods in Chapter 7, and look into the dependencies between temperature derived from temperature time series and frequencies derived from vibration time series.

4.4 Meta-learning

As mentioned at the ends of Section 4.1 and Section 4.2, we can accurately model some of the strain sensor signals using temperature sensor signals, and correlate some vibration sensors with strain sensors. However, the models we obtained are not universal for every pair of sensors. To further look into why some sensor pairs work well and others not, we analysed them in a meta-learning setting, which takes various sensor properties such as location and orientation into account.

Table 4.1: Example of the data that was used in the meta-learning experiment.

Strain							Temperature						corr.
sensor	x	y	embed.	orient.	lane	layer	struct.	sensor	x	y	embed.	lane	layer	struct.	
St1	14	0	attach	X-axis	right	girder	girder	Te1	13	7	embed	right	top	deck	0.139
St1	14	0	attach	X-axis	right	girder	girder	Te2	13	5	attach	right	bottom	deck	0.024
St1	14	0	attach	X-axis	right	girder	girder	Te3	9	7	embed	middle	top	deck	0.068
...															
St2	14	2	attach	X-axis	right	girder	girder	Te1	13	7	embed	right	top	deck	0.277
St2	14	2	attach	X-axis	right	girder	girder	Te2	13	5	attach	right	bottom	deck	0.472
...															

Table 4.2: The $d \leq 2$ results for the St-Te models ($\mu_0 = 0.533$).

Rules	Coverage%	Quality	Average
St vertical = inside deck & St horizontal ≤ 7	11.0	18.2	0.89
St vertical = inside deck & St orientation = Y-axis	9.9	17.8	0.90
St vertical = inside deck	16.5	16.1	0.79
St vertical = inside deck & Te horizontal ≤ 9	13.2	15.9	0.82
St vertical = inside deck & Te horizontal ≥ 5	13.2	14.1	0.79
St vertical = inside deck & Te embedding = attach	8.5	12.2	0.81
St vertical = inside deck & Te horizontal ≤ 5	6.6	11.2	0.82
St vertical = inside deck & Te embedding = embed	8.2	10.6	0.77
St embedding = embed	47.3	10.3	0.63

The software we used to conduct the meta-learning is called Cortana [42], which is a generic toolbox for Subgroup Discovery tasks, including the regression setting that is required here.

Table 4.1 shows the structure of our data in the meta-learning procedure. We represent each sensor pair and their properties, including the correlation of the best model, in one row. In the St-Te model, we have $91 \cdot 20 = 1,820$ rows, and $91 \cdot 34 = 3,094$ rows for the St-Vi model. The sensor locations are represented using x and y coordinates, but we also introduced several intervals in both dimensions to group sensors based on the part of the bridge they are placed.

St-Te models In meta-learning for the strain and temperature sensors, we take the absolute correlation value of each sensor pair as the primary target, and employ *z-score* as quality measure. The first 9 subgroups (sets of pairs of St-Te sensors) with search depth $d \leq 2$ are shown in Table 4.2. The average correlation over the entire set of pairs is $\mu_0 = 0.533$.

This table shows 2 subgroups of depth one and 7 subgroups of depth two. The depth-one subgroups indicate that the interesting vertical position for strain sensors is inside the deck, and that embedded strain sensors are influenced more than attached ones. The depth-two subgroups show more detailed information. For the strain sensors inside the deck, their interesting horizontal position is the

Table 4.3: The $d \leq 2$ results for the St-Vi models ($\mu_0 = 0.139$).

Rules	Coverage %	Quality	Average
St vertical = girder	17.4	31.3	0.36
Vi vertical = girder & St vertical= girder	10.2	28.0	0.40
St vertical = girder & St horizontal = right	6.5	24.8	0.43
St embedding = attach & St orientation = X-axis	38.0	17.7	0.21
St vertical = girder & Vi vertical = under deck	7.2	15.3	0.31
sensor = St1 & Vi vertical = girder	0.6	12.8	0.62
St vertical = girder & St horizontal = left	6.5	12.2	0.28
sensor = St83 & Vi vertical = girder	0.6	11.4	0.56
sensor = St11 & Vi horizontal = right	0.4	11.4	0.68

left side of the bridge, and the orientation is along the Y-axis. The attached temperature sensors perform slightly better than those embedded in the bridge. The horizontal position of the temperature sensors is also important, which indicates that the strain sensors inside the deck correspond well with temperature sensors located on the middle and left side of the bridge.

St-Vi models In meta-learning for the strain and vibration models, our target is the improvement of correlation after band-pass filtering. Table 4.3 presents the top-9 subgroups. From these results, it is clear that the interesting vertical locations for both strain and vibration sensors are the girders. For vibration sensors, placement under the deck is also interesting. The interesting horizontal locations for strain sensors are either side of the bridge, rather than the middle lanes. The strain sensors on the right-hand side measure more variation than those on the left. This is explained by checking the video stream: there was more traffic on the right lanes during this period of time. We also identified several specific strain sensors located on both sides of the bridge, which are consistent with the other subgroups in Table 4.3. Note that these selected sensors correspond to the three darkest columns in the correlation matrix in Fig. 4.3 (right).

4.5 Conclusion

We have demonstrated the use of a number of key data mining and signal processing techniques to model dependencies among multiple sensor types. We have built a linear model to correlate strain and temperature readings, and improved this model through convolution with an exponential response function. In the frequency domain, we used band-pass filters to detect the correlated spectra between strain and vibration sensor time series. Finally, we conducted meta-learning on the models obtained in Section 4.1 and Section 4.2, and extracted subgroups to explain the effects of sensor placement. The extracted rules can be used as guidelines for designing more (cost-)effective networks on future Structural Health Monitoring installations.

Chapter 5

Baseline Correction

5.1 Introduction

With recent advances in monitoring capabilities and hardware solutions, more and more civil structures are being fitted with a sensor network. Based on the collected data, a number of Structural Health Monitoring (SHM) methods have been developed to assess the condition of structures. Most of these methods assume that damage and degradation will affect the physical properties of the structure, such as their mass and stiffness [43]. These fundamental changes in the structure will manifest themselves in important parameters of the structure, notably resonance frequencies, mode shapes, and damping ratios [44, 45]. However, in practical applications, modal parameters are also subject to varying operational and environmental conditions such as traffic, humidity, wind [7, 46], solar radiation and, most importantly, temperature [7, 47, 48, 49].

Considerable research effort has been devoted to distinguishing changes caused by the environmental variability from those due to structural damage or degradation [3, 43, 45, 50, 51, 52], but unfortunately, investigations studying the operational variability (the effect of varying traffic load on key parameters) have been mostly lacking. Even for environmental influences, for example the temperature-effect on strain measurements, one can model in detail the response to temperature

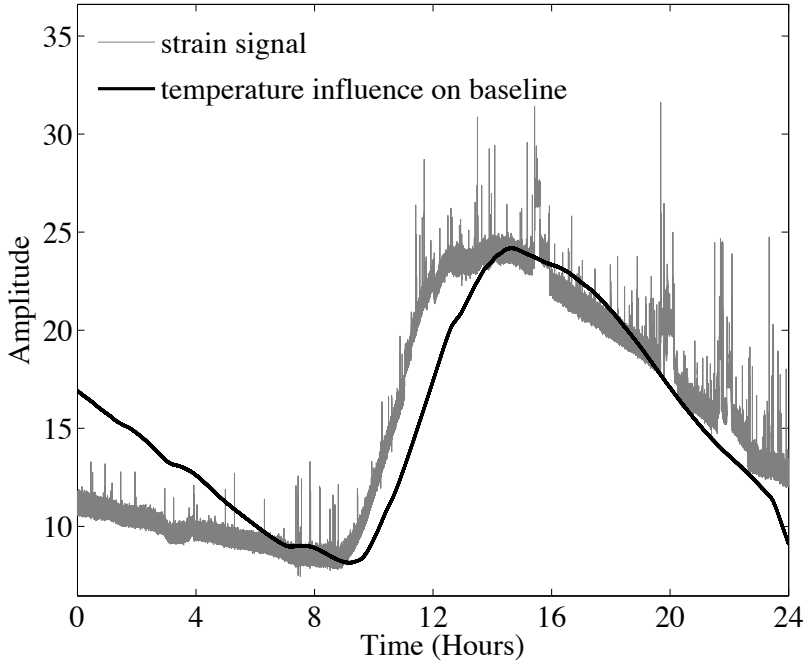

Figure 5.1: The influence of temperature on the strain signal - A linear model between the strain and temperature signals with a length of one day.

changes, but not to a sufficient degree for some applications. For reliable performance of SHM systems, it is of vital importance to filter out the effects of both environmental and operational influences.

The approach we take in this chapter, is to identify two components of the signal: a slowly fluctuating baseline due to gradual environmental effects, and a rapidly changing signal superimposed on the baseline that is due to short-term, transient effects, such as traffic. As was demonstrated before, the baseline in the strain signal is strongly dependent on the daily temperature effects, as well as some medium-term events such as traffic jams (recognisable as temporary jumps in strain). Superimposed on this gradual effect are the peaks that represent individual vehicles. For various SHM applications, identifying the baseline, or simply removing it, is a crucial step. For the basic operation of traffic event identification, for example to compile daily traffic load statistics, recognising peaks over a baseline is an essential step. But also for more sophisticated applications, such as extracting modal parameters from free-vibration periods (the several seconds

of unloaded shaking after heavy traffic has passed), require exact identification of the baseline [53]. Note that especially modern SHM systems need to deal with the long-term baseline drift, as they tend to monitor structures around the clock, if not around the calendar, such that baseline correction will require considerable attention.

Baseline correction A baseline is not a fixed physical phenomenon, but rather something that depends on the application, and therefore subject to definition. The most common way to define what constitutes the *baseline*, and what the *signal*, is in terms of time scale. Essentially, any long-term effect belongs to the baseline, and any short-term effect to the signal.

In the example data of Fig. 5.1, most of the undesirable drift in the signal is caused by changes in outside temperature, as indicated by the black line (scaled in this picture to match the strain signal). Clearly, the strain gauge has captured the response of the bridge to this temperature change, but the effect of outside temperature (and in fact all other weather parameters) is non-trivial, such that we cannot simply remove this effect from the strain signal. Another source of disturbance in the bridge case is the occasional traffic jam (for example around 4 and 8 PM), which temporarily shifts the signal upwards, in response to the increased weight on the bridge. Note that traffic jams are often only on one side of the bridge, so that traffic in the opposite direction still is showing up as peaks in the signal.

For a range of SHM applications, including traffic identification and modal analysis, strain gauge measurements are a vital resource [53, 54, 55]. However, as Fig. 5.1 demonstrates, strain signals are subject to large baseline fluctuations not directly relevant to such applications. In fact, in most cases the range of fluctuations that can be considered part of the baseline is often substantially larger than the actual short-term dynamic behaviour that the strain gauges are designed to capture. For that reason, any non-trivial application will first need to deal with identification of the baseline, and correction thereof.

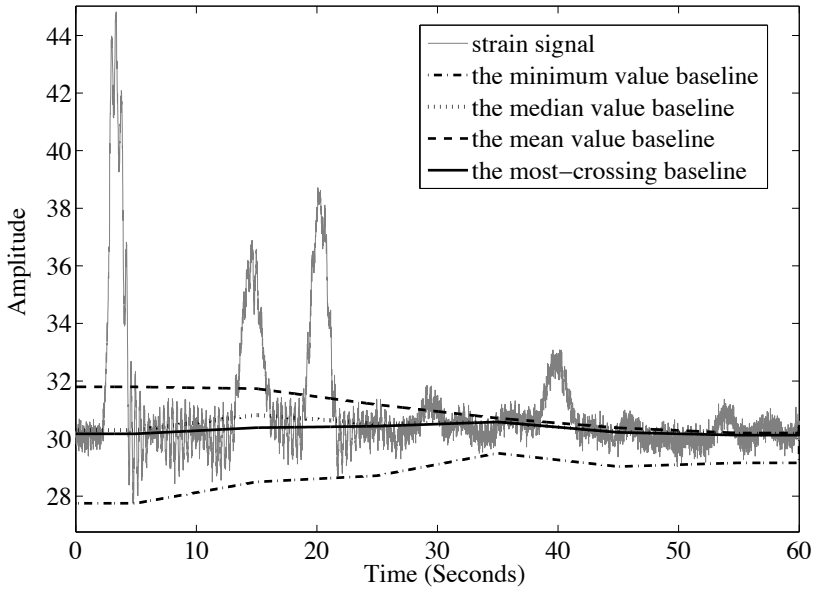

Figure 5.2: Comparison of several piece-wise baseline correction methods - The length of each window is 1000 data points (10 seconds). The baseline of each window is assumed to be a constant value, which is either obtained by calculating the mean value (the dashed line), the minimum (the dash-dotted line), the median value (the dotted line), or the most-crossing value (the solid black line). Baselines of two adjacent windows are connected using linear interpolation.

Significant work has been done with nuclear magnetic resonance (NMR) signals as well as for standardising electrocardiogram (ECG) signals, but to the best of our knowledge, it has received little attention in the civil engineering domain. In this chapter, we present a novel baseline correction method, the *most-crossing* method, for processing strain signals in civil SHM applications. The most-crossing method has only a few manual parameters, and can be used automatically for real-time baseline correction. This method is designed to extract useful peaks from signals under conditions of high frequency noise and baseline drift. It can deal with peaks of irregular shapes and random distributions.

In the coming sections, we will first present the procedure of the most-crossing method, and then apply this method to practical signals and compare its performance with some other popular methods.

5.2 The Most-Crossing Method

The proposed most-crossing method is a *piece-wise* method, which employs a sliding window, like all piece-wise baseline correction methods. The sliding window is an interval in time of size L that is slid over the time series. The size L is determined by the actual application. Within a sliding window, we can assume the baseline to be a constant value. What defines a specific piece-wise method is how this constant value is determined from the data within the window. There are several common choices for this value, such as using the mean, the median or the minimum value. These solutions may work well with simple signals, but cannot process complex signals, like the strain signal shown in Fig. 5.2. The mean and median value method weigh each measurement equally, whether part of a peak or not, so the detected baseline is unstable in heavy traffic. The minimum value method is useful when all the peaks are upward, but it will cause distortion if the direction of peaks is mixed. Motivated by the disadvantages of these choices, we introduce the *most-crossing* method to extract the baseline.

The most-crossing method is based on the probability density function (PDF). The method is a four-step procedure: baseline recognition, baseline modelling, traffic jam detection and baseline removal.

5.2.1 Baseline Recognition

We assume that the data points within a sliding window are composed of two kinds of data points: 'noise points' and 'peak points'. A peak point is defined as a data point that corresponds to dynamic excitation of the structure, in our case traffic events. The remaining data points are noise points, which contribute to the baseline of the sliding window. Normally, the probability distribution of these two kinds of data points are different, so we can use the PDF for baseline recognition.

The PDF of a continuous random variable is a function that describes the relative likelihood for this random variable to take on a given value. The PDF is non-negative everywhere, and its integral over the entire space is equal to one [56].

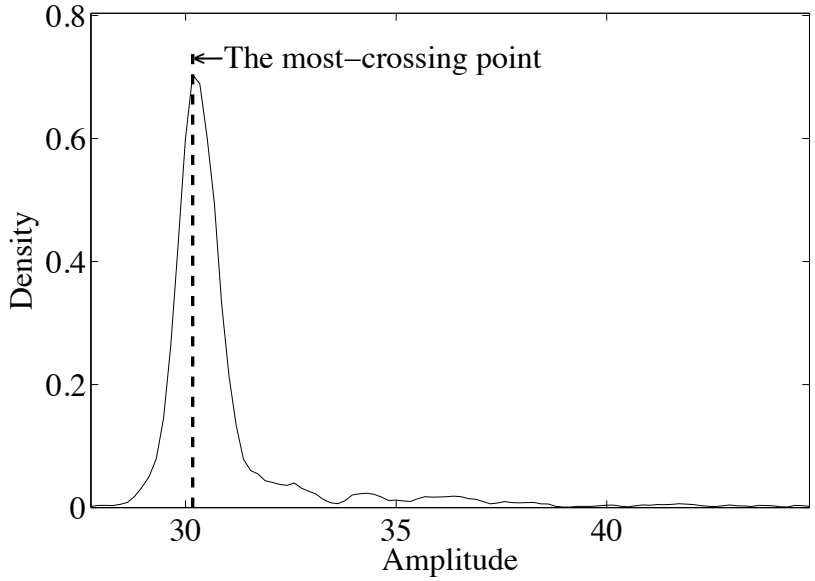

Figure 5.3: The kernel smoothed probability density function - The PDF is derived from the same dataset as Fig. 5.2; the most-crossing point is the first peak of the kernel smoothed PDF.

For discrete variables, such as sensor readings, the PDF is often estimated by a histogram. To construct a histogram, we first compute the range for the data set, and then divide it into a number of equal intervals, also known as 'bins'. The PDF is estimated by counting the number of points that fall within each interval. Although a histogram is a simple way to estimate the density, it is known to depend a lot on exact parameter choices and is sensitive to artefacts. To alleviate these problems, we adopt the more sophisticated *kernel density estimation* (KDE).

The KDE ($\hat{f}_h(x)$) is a non-parametric way to estimate the PDF ($f(x)$), which can be represented as Eq. 5.1.

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right) \tag{5.1}$$

where $K(\cdot)$ is a kernel function that integrates to 1; h is a smoothing parameter called the bandwidth; x_i is the ith point in the equally spaced amplitude interval;

n is the number of portions used to divide the amplitude interval.

In the KDE, there are two important parameters: the kernel function and the bandwidth. There is a range of kernel functions, including Gaussian, uniform, biweight, etc. Due to the convenient mathematical properties, Gaussian kernels are the most often adopted. The bandwidth of the kernel exhibits a strong influence on the KDE. The optimal bandwidth is the one that minimises the mean integrated squared error (MISE). Under the asymptotic conditions, the MISE can be approximated as follows [57, 58]:

$$MISE(h) \approx \frac{1}{nh} \int K(x)^2 dx + \frac{h^4}{4} \left(\int x^2 K(x) dx \right)^2 \int f''(x)^2 dx \tag{5.2}$$

By replacing $MISE(h)$ with zero, we can obtain a solution to the equation of (5.2), which is the optimal bandwidth. To obtain a concrete value for the optimal bandwidth, we must replace the unknown density f with an estimate. The data points contributing to the baseline are dominated by random noise and free vibration waves, so we empirically estimate the PDF f with the normal distribution $N(\mu, \sigma^2)$. The optimal bandwidth $\hat{h}_{opt}$ can be represented as Eq. 5.3, which is known as Silverman's rule of thumb [58]:

$$\hat{h}_{opt} = \left(\frac{4\hat{\sigma}^5}{3n} \right)^{\frac{1}{5}} \approx 1.06 \hat{\sigma} n^{-1/5} \tag{5.3}$$

where $\hat{\sigma}$ is the standard deviation of the samples.

Based on the optimal bandwidth and the assumed normal distribution, we can obtain a kernel-smoothed PDF, shown as the picture in Fig. 5.3. There are several peaks in the PDF of the selected signal, and each peak stands for the density distribution of one kind of signal component. The first peak in the PDF corresponds to values of the baseline (in this case around 31 micro-strain). We take the *most-crossing point*, the maximum value of the first peak, as the value of the baseline.

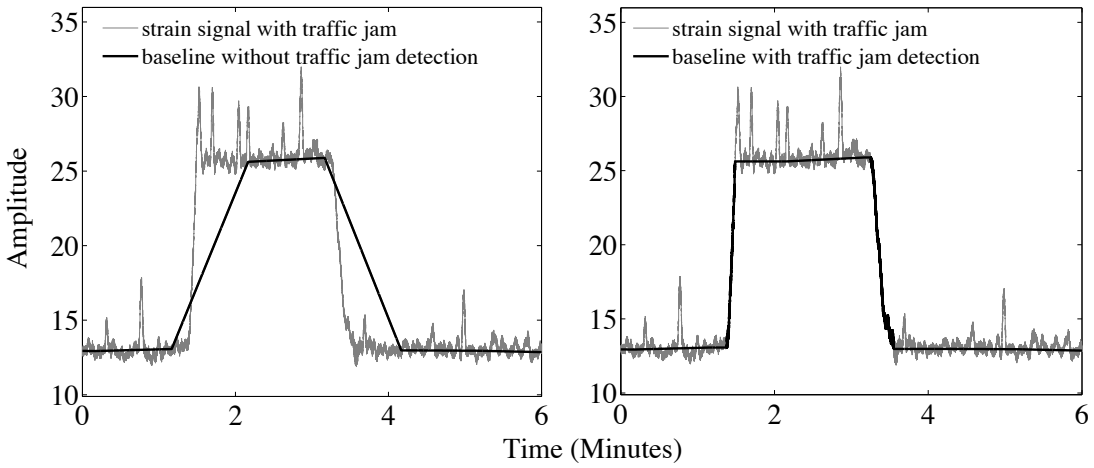

Figure 5.4: The reason for traffic jam detection - The baseline without traffic jam detection (left) and the baseline with traffic jam detection (right).

5.2.2 Baseline Modeling

By moving the sliding window point by point, we can obtain the baseline for the whole signal immediately. But this method is too time consuming and unnecessary in most situations. To detect the baseline more efficiently, we move the sliding window with a user-defined overlap. The downside of this process is that it may cause discontinuities. To solve this problem, we employ linear interpolation to modify the last part of one sliding window baseline and the first part of the next sliding window baseline. This modelling method makes no assumption about the shape or functional form of the baseline, but works well even when the SNR is high. The baseline obtained by such a procedure is called a raw baseline, because traffic jams have not been considered in this step.

5.2.3 Traffic Jam Detection

When a traffic jam occurs, we expect a baseline that looks as Fig. 5.4 (right), which catches the boundaries of traffic jam well. In practice however, the baseline obtained with the procedure mentioned above often looks like the left figure in

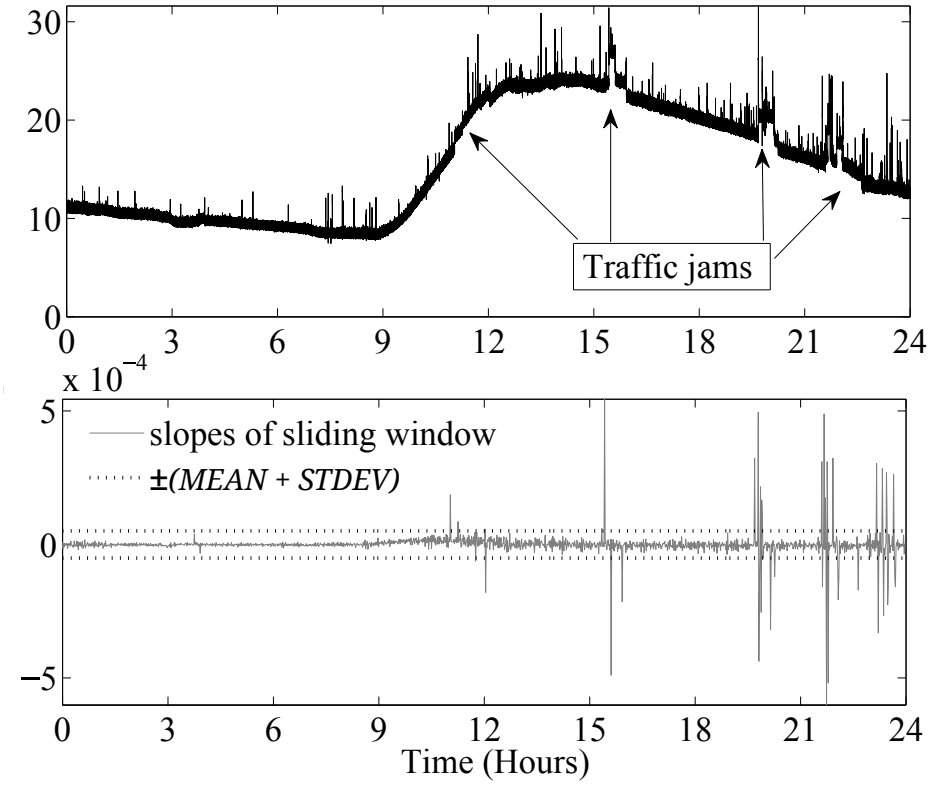

Figure 5.5: The traffic jam detection - The threshold for triggering the traffic jam detection procedure, which is set as the sum of the mean value ($MEAN$) and standard deviation ($STDEV$) of sliding window slopes.

Fig. 5.4, which has the problem of representing boundaries well. We solve this boundary problem with the aid of slopes of two successive windows.

When the traffic on the bridge is normal, the baseline of the strain signal varies only slightly, and the absolute slope values of sliding windows are also relatively small. However, when a traffic jam occurs, the baseline of the strain signal will jump to a higher value within a short time period, shown as the right part of the top picture of Fig. 5.5. If we plot slope values against time (shown as the bottom picture of Fig. 5.5), the traffic jam will cause a slope peak between two sliding windows. If the absolute value of a peak is above a certain threshold, a traffic jam detection procedure will be triggered (see Fig. 5.6). The threshold is dependent on the target data set. Here, for one day's dataset collected at 100 Hz, we set the

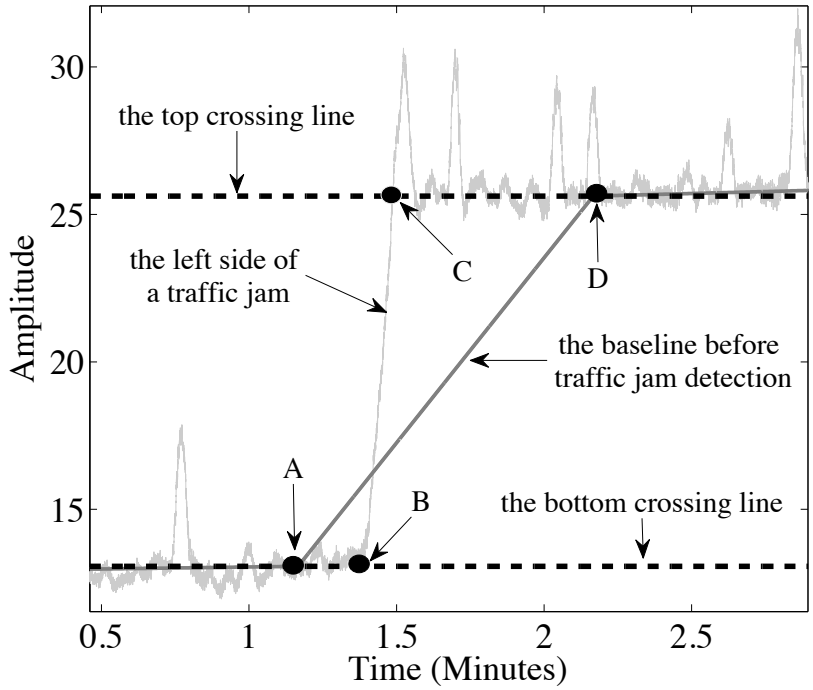

Figure 5.6: The traffic jam boundary detection - A and D are middle points of two successive sliding windows. The bottom-crossing line is a horizontal line across the middle point A. The top-crossing line is a horizontal line across the middle point D. The turning point B is the last intersection between the bottom-crossing line and the strain signal. The turning point C is the first intersection between the top-crossing line and the strain signal.

threshold as the mean plus one standard deviation of all slope values.

The boundary problem happens between the points A and D (in Fig. 5.6), which are middle points of two successive windows. We draw a bottom-crossing line across the middle point A = (x_a, y_a), and a top-crossing line across the middle point D = (x_d, y_d). The traffic jam turning point B is now defined as (x_b, y_a), where x_b is the last time between A and D that the signal crosses the horizontal line defined by $y = y_a$. The baseline between the turning points B and C is now simply made to follow the actual signal. The baseline between A and B is obtained with the normal most-crossing method. Point C and the associated baseline between C and D are produced in analogous fashion.

5.2.4 Baseline Removal

This step is quite straightforward. We just need to subtract the obtained baseline from the original signal.

5.3 Experimental Evaluation

We apply our most-crossing method to the InfraWatch strain signal to remove the baseline, and compare its performance to the first derivative method and the iterative polynomial fitting method. As discussed, strain gauges are not only sensitive to vehicles, but also to temperature and traffic jams. We employ a dataset with a length of 24 hours (8.64 million measurements), which is informative enough to include all important events. The dataset is the same as the one used in the top picture of Fig. 5.5, in which the baseline wander is caused by temperature changes, the small spikes stand for vehicles and the big jumps are caused by traffic jams.

In Fig. 5.7, we first present an overview of three different baseline correction methods on the selected dataset: the black solid line in the left picture shows the baseline obtained with our most-crossing method, which fits the baseline drift quite well. The black solid line in the middle picture stands for the baseline derived from the first derivative method (Dietrich's method [59]), in which outliers are detected through checking their adjacent points. This is insufficient for detecting outliers in our strain signal. The last picture illustrates the baseline obtained with a 20-order polynomial fitting, which moderately fits the baseline drift caused by temperature changes, but fails to catch the drift induced by traffic jams. In the coming sections, we will look into some detailed performances of these methods.

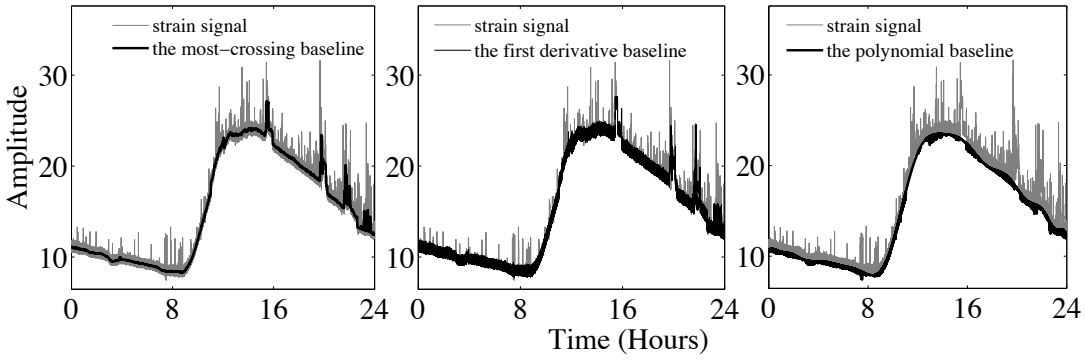

Figure 5.7: The comparison between three different baseline removal methods - The most-crossing baseline (left) is derived from a sliding window with length of 1 minute (6,000 data points). The first derivative baseline is obtained with a classification threshold of $MEAN + 3 \cdot STDEV$, and a false baseline segments threshold of 150 data points. The polynomial baseline is obtained by a 20-order polynomial fitting.

5.3.1 Baseline Removal over a Short Period Signal

For a detailed analysis, we select a dataset of 1 minute (6,000 data points) around midnight, when the traffic is not too heavy. The selected interval includes one truck and several cars.

The most-crossing method Within such a small dataset, we can simply choose the window size the same as the length of the dataset. The minimum strain is 10.84 micro-strain, the maximum strain is 19.04. The strain interval $[10.84, 19.04]$ is divided equally into 100 bins, for estimating the density of strains. The optimal bandwidth $\hat{h}_{opt}$ is 0.153. Based on Eq. 5.1, we obtain an estimator of the signal PDF. The most-crossing value 12.35 is then taken as the baseline (Fig. 5.8 (left)). After subtracting the baseline from the signal, we obtained a signal that preserves all the useful peaks but has a more meaningful centering on the Y-axis (Fig. 5.8 (right)).

The first derivative method We carry out a similar analysis with the first derivative method introduced by Dietrich et al. [59]. We first apply a Gaussian

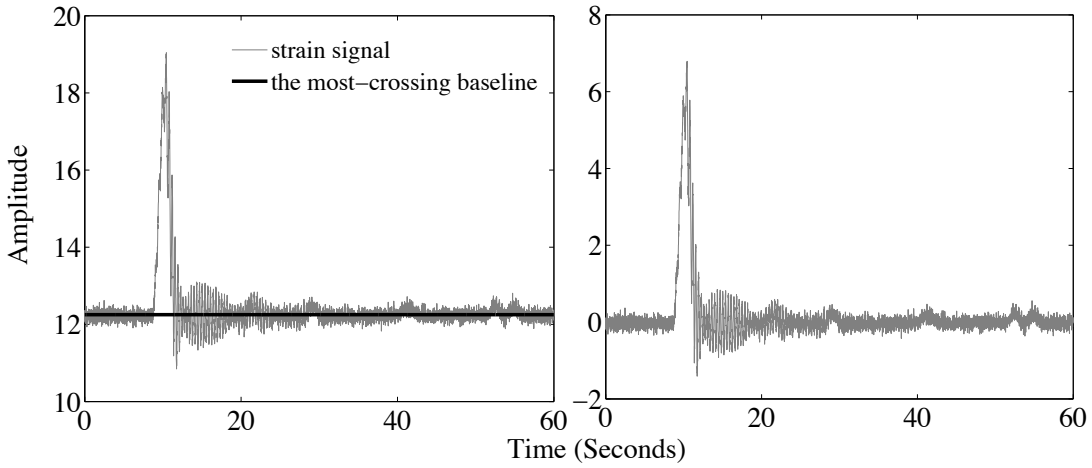

Figure 5.8: The most-crossing baseline over a short period signal - The baseline derived from the most-crossing method (left) and the baseline removed signal (right).

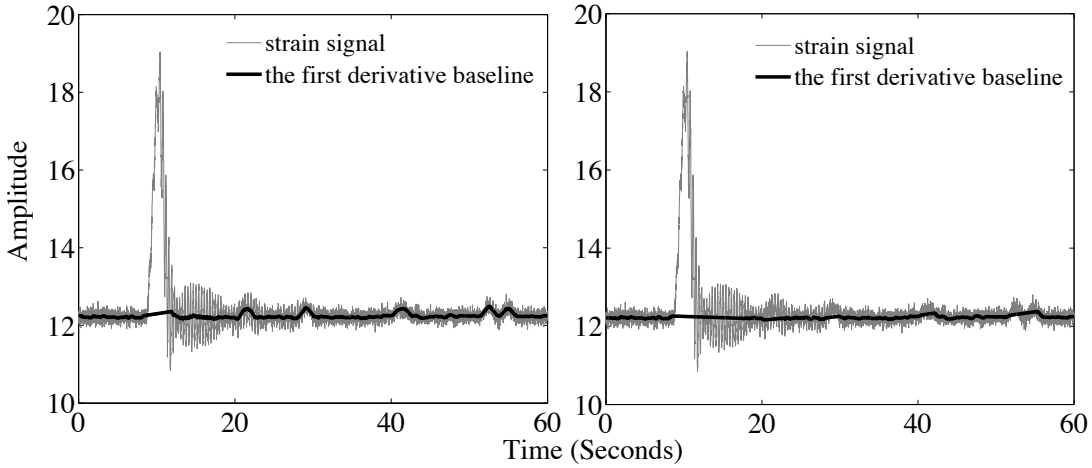

Figure 5.9: The first derivative baseline over a short period signal - The baseline obtained by just checking adjacent points (left) and the baseline obtained by correcting noise segments whose lengths are less than 150 data points.

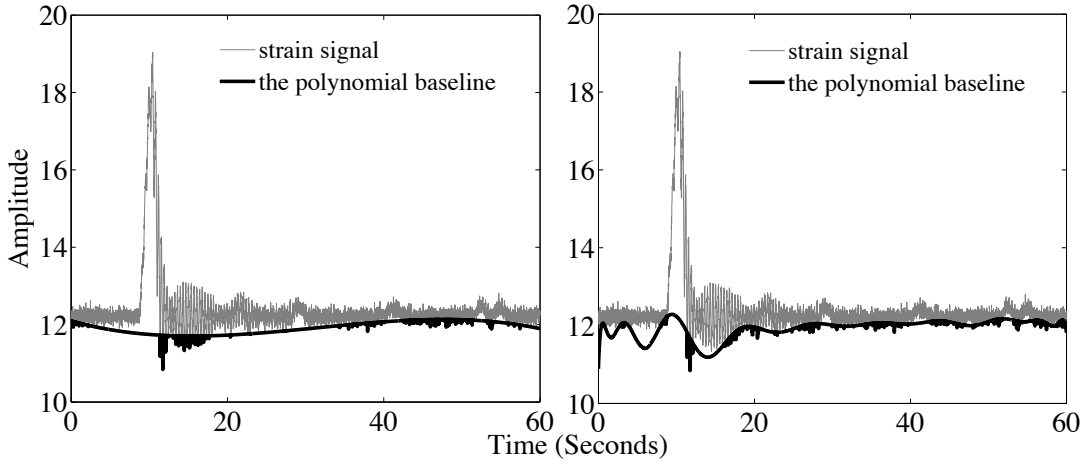

Figure 5.10: The polynomial baseline over a short period signal - The baseline derived from a 3-order polynomial fitting (left) and the baseline derived from a 20-order polynomial fitting (right).

filter to smooth the original signal, and then calculate the derivative by replacing every point in the signal with the difference between this point and the next point. The automatic threshold used to classify data points is set as $MEAN+3\cdot STDEV$. For the outlier detection step, by just checking two neighbours of a data point, we obtain the baseline shown in the left picture of Fig. 5.9, from which we can see that most of useful peaks are assigned to the baseline. We improve this result by correcting short noise segments into peak segments. By changing noise segments of less than 150 data points into peak segments, we obtain an improved baseline, shown as the solid black line in the right picture of Fig. 5.9. The improved baseline is good for processing signals with sharp peaks, but still performs moderately with broad and overlapping peaks.

The iterative polynomial method We also apply the improved iterative polynomial fitting method [60] to the same dataset. We assume the initial fitting result equals to the original signal, and employ a low order (3) polynomial (left picture of Fig. 5.10) to fit the original signal with the least-squares criterion. If the elements in the original signal are bigger than the elements in the obtained fitting result, then we replace them with the latter. The original signal is truncated

iteratively until the criterion of convergence, shown as Equation 5.4, is reached. We repeat the same procedure with a 20-order polynomial. The fitting result is shown in the right picture of Fig. 5.10.

$$\frac{\|b_k - b_{k-1}\|}{b_{k-1}} < 0.001 \tag{5.4}$$

where b_k and b_{k-1} are polynomial fitting results at the kth and $(k-1)$th iteration, respectively. At iteration 0, b_0 is the original signal y_0.

For a given order, the iterative polynomial method aims to generate an optimal fitting with the least-squares criterion, which considers all the data points in the dataset equally. From the results in Fig. 5.10, we can clearly see that neither a low nor a high-order polynomial can fit the baseline well.

5.3.2 Baseline Elimination for Traffic Jams

In this section, we will consider the baseline elimination during traffic jams. Traffic jams, which may last from a few minutes to a couple of hours, typically happen during rush hour. In most cases, traffic jams happen just on one side of the bridge, while on the other side of the bridge, traffic flow is normal. So the sensors on the bridge may collect information about traffic jams and traffic events at the same time. The dataset for this section, which covers 1 hour (360,000 data points), contains a traffic jam of about 10 minutes on one side of the bridge.

The most-crossing method We employ a sliding window to move along the selected dataset. The window size is also set as 1 minute (6,000 data points), with no overlap between successive windows. Without traffic jam detection, false traffic peaks (boundary problems) will occur around the boundaries of the traffic jam, shown as the left picture of Fig. 5.11. By empirically setting the traffic jam threshold as $MEAN + STDEV$ of all slope values within this period as described in Section 5.2, we solved the boundary problem (shown as the right picture of Fig. 5.11).

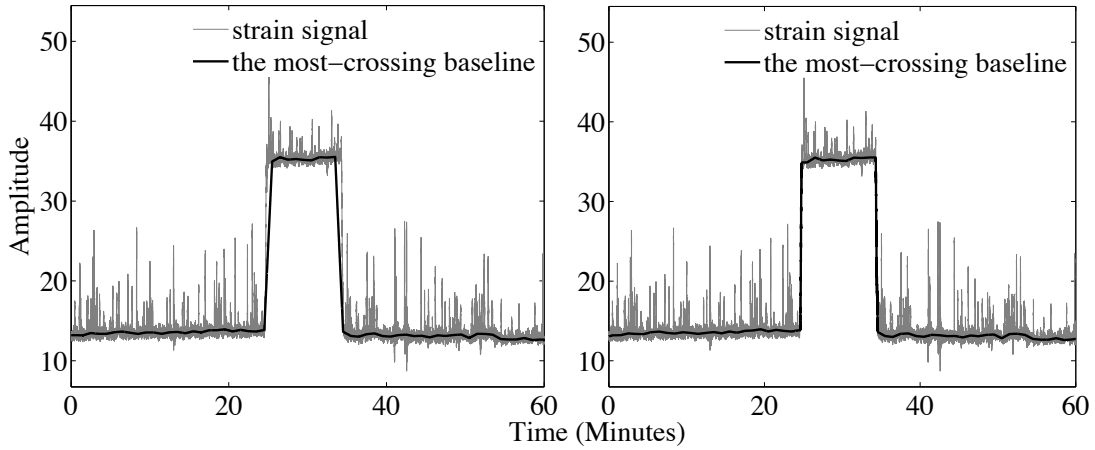

Figure 5.11: The most-crossing baseline for traffic jam signal - The traffic jam baseline before solving the boundary problem (left), and the baseline after traffic jam detection (right).

The first derivative method We process the traffic jam signal with the same first derivative method mentioned above. The automatic threshold used to classify data points is set as $MEAN + 3 \cdot STDEV$. We first detect the baseline with Dietrich's method, which eliminates outliers through just checking two neighbours of a data point. The obtained result, shown as the left picture of Fig. 5.12, can catch the traffic jam moderately, but it still suffers from broad peak and traffic jam boundary problems. We then improve the result by correcting the false noise segments (the lengths of which are less than 150 data points). The improved result, shown as the right picture of Fig. 5.12, can substantially reduce the problems mentioned above, but cannot overcome them completely.

The iterative polynomial method For the iterative polynomial method, the most critical parameter is the order of the polynomial. The higher order we use, the more detail can be caught. To show two extremes, we employ a low order (1 degree) polynomial and a high order (25 degree) polynomial to iteratively fit the traffic jam signal. As shown in Fig. 5.13, the low order polynomial can catch part of the baseline of the normal traffic periods, but fails to detect the traffic jam, and the high order polynomial cannot deal with the traffic jam either.

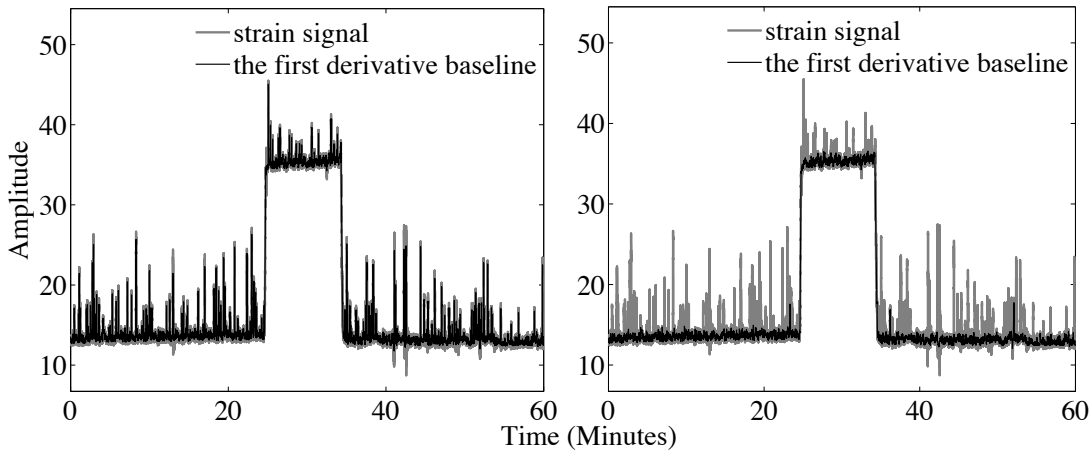

Figure 5.12: The first derivative baseline for a traffic jam signal - The first derivative-based baseline obtained by just checking adjacent points (left) and the baseline obtained by correcting noise segments whose lengths are less than 150 data points (right).

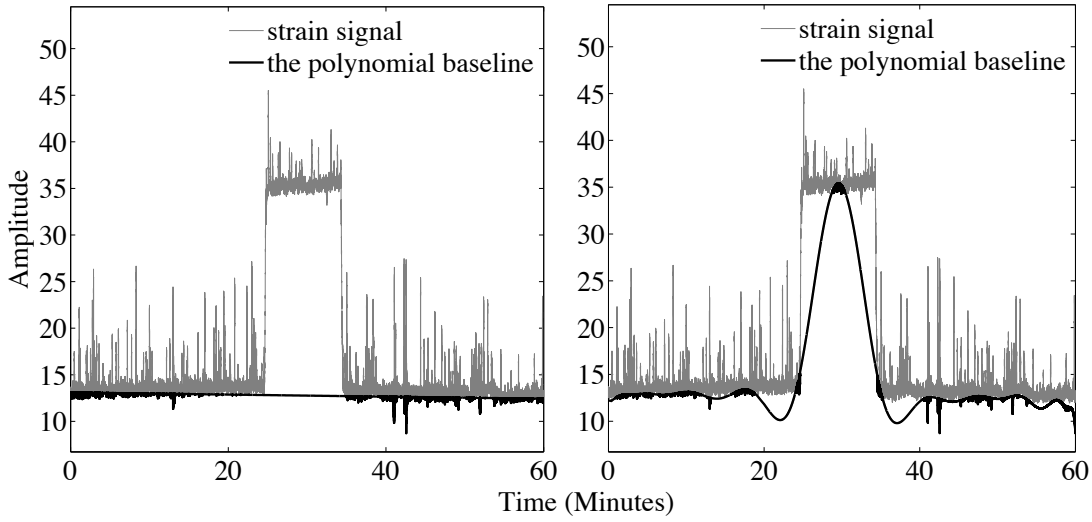

Figure 5.13: The polynomial baseline for traffic jam signal - The baseline derived from the 1 degree polynomial (left) and the 25 degree polynomial (right).

Table 5.1: Vehicle information of 7 days.

Day	Car	Van	Truck	Total
Monday	2,647	987	265	3,899
Tuesday	2,611	1,023	324	3,958
Wednesday	2,610	1,021	302	3,933
Thursday	2,725	1,073	292	4,090
Friday	2,742	1,088	290	4,120
Saturday	2,750	303	24	3,077
Sunday	2,389	124	12	2,525

5.4 Baseline Correction Applied to Traffic Counting

Traffic event statistics on a bridge are of vital importance in assisting bridge managers to evaluate the condition of the bridge and implement a maintenance plan. The top picture of Fig. 5.14 shows the strain signal of 7 days, during the period from Monday Dec 8, 2008 to Sunday Dec 14, 2008, based on which we will estimate the traffic load for this period. A dataset of 7 days sampling at 100 Hz means a huge computational burden. To make it work on our PC, we down-sample the dataset to 1 Hz, which will not affect the statistical result, because traffic events are low frequency components of the strain signal (below 1 Hz).

Traffic events appear as peaks in the strain signal, with varying amplitudes and durations (depending on weight and speed of the vehicles). To extract these features, we need to get rid of the moving baseline first. Since the signal is sampled at 1 Hz, we employ a sliding window of length 60 data points (1 minute). The traffic jam trigger threshold is set as the sum of the mean value and 3 times the standard deviation of slope values, as shown in the middle picture of Fig. 5.14. When the absolute slope value of two sliding windows is above the threshold, the traffic jam detection procedure is fired, and the traffic jam is recognised as part of the baseline. The baseline-free signal in the bottom picture of Fig. 5.14 is obtained

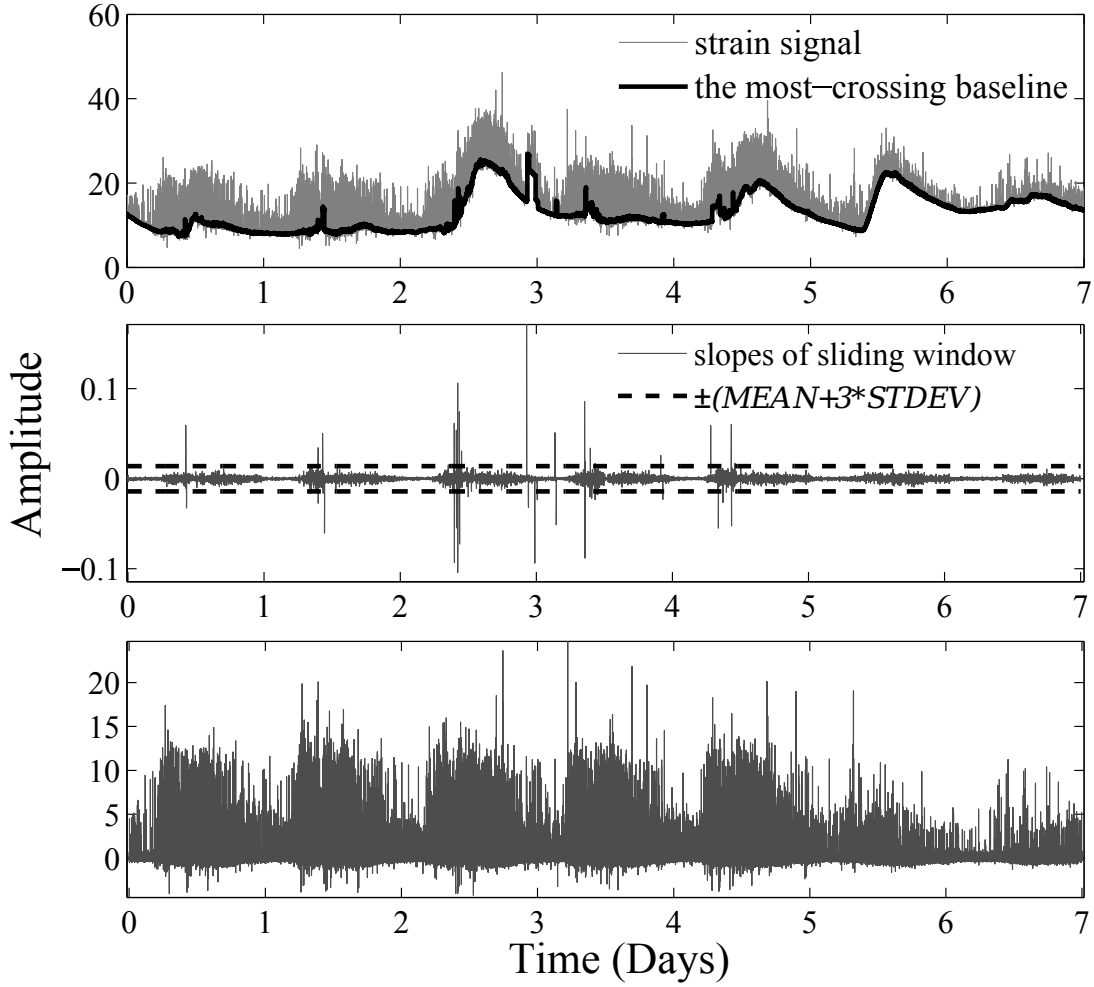

Figure 5.14: Traffic event statistics of 7 days (During the period between Dec 8, 2008 and Dec 14, 2008) - Top picture: the strain signal of 7 days at 1 Hz and its baseline obtained with the most crossing method. Middle picture: the slope values of adjacent sliding windows (with length 60 data points) and the threshold lines for triggering traffic jams, which are set as the mean plus 3 times standard deviation. Bottom picture: the strain signal without baseline drift.

Table 5.2: The traffic jam statistics of 7 days.

Traffic jam	Start (Hour)	Duration (Minute)	Day
1	10:12	7.4	Monday
2	9:32	1.2	Tuesday
3	10:19	19.9	Tuesday
4	9:31	2.2	Wednesday
5	9:55	1.4	Wednesday
6	10:07	1.8	Wednesday
7	10:27	1.9	Wednesday
8	10:48	72.1	Wednesday
9	22:16	113.2	Wednesday/Thursday
10	3:18	13.6	Thursday
11	8:31	4.8	Thursday
12	9:31	14.3	Thursday
13	21:56	22.4	Thursday
14	6:45	129.4	Friday
15	10:22	2.7	Friday

by subtracting the baseline in the top picture from the original strain signal. With the traffic event identification method presented in our previous work [53], we obtain 108,161 peaks from the baseline-free signal, with location, amplitude and duration. We assume that, based on the peak amplitude, these peaks can be divided into 4 categories: noise, car, van and truck, and the last three categories are interesting for us, which are mentioned as useful peaks. The clustering method employed in this work is the k-means method [61], which aims to divide all the obtained peaks into k clusters. The k-means uses squared Euclidean distances, and the distance between two objects within the same cluster is smaller than that of two objects in different clusters. By setting k as 4, 25,602 peaks are classified as useful peaks, and the remaining 82,559 peaks are classified as noise. The detailed information of useful peaks is listed in Table 5.1.

Based on the vehicle statistics results, we learn that the number of vehicles on

work days is considerably more than that of weekends; within one day, cars form the majority of traffic events. During the weekends, the number of vans and trucks is reduced sharply, while the number of cars is only slightly reduced.

As shown in the Table 5.2, we recognised 15 traffic jams (there are also 15 traffic jams existing in the video data of this period), the durations of which range from 1.19 minutes to 129.40 minutes. All the traffic jams occur on weekdays, and weekends are traffic jam free. Most traffic jams happen during rush hour of the workday, but there are also exceptions, like the 9th traffic jam, which lasted nearly two hours around midnight. Through checking the video record, we found out that the bridge was under substantial maintenance during this period.

5.5 Related Work

Baseline correction techniques have been extensively discussed in the literature since the 1970's [62]. Schulze et al. [4] conducted an excellent literature review and comparison of various baseline-removal methods. Most of the techniques can be divided into two groups: time-domain methods and frequency-domain methods. In the frequency domain, the baseline is assumed to be represented by the low frequency components. The peaks of interest belong to the medium frequency components, and the independent noise is usually distributed among medium and high frequency components. The *wavelet transform* [63] and the *Fourier transform* [64] are two common methods in this domain. When the spectral components are complicated, it is difficult to differentiate the baseline from others with a Fourier transform. Utilising the wavelet transform, we have to make great efforts to choose a mother wavelet, decomposition level and coefficients to remove. Improper selection may lead to baseline extraction failure.

There are more baseline correction methods developed in the time domain. The *median filter* method was first introduced by Friedrichs [65] to deal with the baseline drift in nuclear magnetic resonance (NMR) spectra. This method takes the median value in a sliding window as the baseline. Through properly choosing the window size, the median filter will ignore the peaks of interest, and just focus

on the points in the baseline. As shown in Fig. 5.2, this method works well with low signal-to-noise ratio (SNR) spectra with narrow peaks, but cannot handle broad peaks or high SNR spectra.

The *iterative polynomial fitting* method [60, 66] assumes that the baseline can be estimated by a low order polynomial. Under a given polynomial order, a suitable polynomial is obtained by fitting the original signal with the least squares criterion. The fitted polynomial can be used as automatic threshold to truncate the original signal. Iterative processes are implemented on the truncated signal until the criterion of convergence is reached. One drawback of this method is that the order of the fitted polynomial should be chosen appropriately. If the order is too small, the baseline cannot be detected correctly. On the other hand, if the order is too large, the peaks of interest may be fitted into the baseline, which can also lead to distortions.

Since the slopes, the differences of successive points, of the baseline are generally lower than those of useful peaks, we can employ the *first derivative* [59, 64] or the *second derivative* method [67] to get rid of the baseline. The first derivative method first uses a moving average filter to suppress the high-frequency noise in the original signal, and then calculates the derivative by replacing every point in the signal with the difference between this point and the next point. The sum of the mean value plus three times the standard deviation is chosen as a threshold to iteratively divide the data points in the signal into two groups: baseline and peaks, until no data points change groups. According to this method, if one single data point belongs to the baseline, and both of its neighbours do not, then this point is put back to the baseline. The advantages of the derivative methods are that they are fast and suitable for automation. But they can be unstable when peaks are broad or overlap happens.

5.6 Conclusion

In this work, we proposed the most-crossing method as a method for detecting the baseline in sensor data from civil engineering applications. The most-crossing

method combines the notion of a sliding window with the probability density function. Within one window, the random noise and traffic events cannot be treated equally, because just the former contributes to the baseline. Traditional baseline correction methods (like the polynomial or first derivative method) consider all the data points in the window equally, so they are unsuitable for baseline correction in the civil engineering domain. The most-crossing method is also capable of processing traffic events of bigger scales, like traffic jams, which is of vital importance for engineers or bridge owners to study the dynamic loads on the bridge. We have evaluated the most-crossing method on datasets of multiple scales, and compared its performance with existing popular baseline correction methods. The results indicate that the most-crossing method is superior in dealing with baselines of strain signals in the civil engineering domain. At the end of the work, we apply the most-crossing method to a big data set of one week, and succeed in obtaining the traffic events distribution during that period.

Chapter 6

Predefined Pattern Detection

6.1 Introduction

This chapter focuses on the problem of detecting instances of predefined patterns in large time series data [28, 68]. While most pattern detection algorithms in time series deal with discovering previously unknown, frequently recurring regularities in the streaming data, here we assume that one or more example sequences (the *templates*) are provided by a domain expert, and instances of these need to be identified in the actual data. During this detection, one needs to allow for a certain degree of difference between the template and the instances, for example because the instance is somewhat longer or shorter in duration, the magnitude of the signal is different, or parts of the signal are either stretched or compressed in time (so-called warps).

Li Wei et al. [68] mention a number of use-cases that motivate the predefined pattern detection problem. For example, in ECG monitoring, a cardiologist may observe some interesting pattern that he or she wants to annotate, and flag any future occurrences, to be investigated by the cardiologist or fellow experts. Alternatively, in insect pest control, one would like to observe specific cases of harmful insects, as identified by specific patterns of audio signal (wing beats). In our application to infrastructure monitoring, the predefined pattern detection problem is relevant for specifying and detecting known disturbances in the data, that can

then be removed from the signal, or accounted for in subsequent modelling steps. For example, when monitoring the structural health of a bridge, the measured signal is dominated by recurring and understandable peaks due to vehicles crossing the bridge and traffic jams. One can imagine an expert providing a template for each of theses phenomena, after which all instances should be identified, regardless of the speed and weight of the vehicles (influencing the width and height of the hump in the signal), or the duration of the traffic jam.

When matching a predefined phenomenon (a template [31, 33, 69]) with the time series under investigation, it is not always required to involve every individual measurement in the selected interval and in the template. In fact, when a certain level of fuzzy matching is required, it makes sense to somehow simplify the signal, or extract some key features that are characteristic for the sequence in question. This condensed representation can then be used to compare the time series with the template, both effectively (the matching is only based on the characteristic aspects) and efficiently (no computation is wasted on insignificant details). Specifically when large time series with high sampling rates are concerned, and the matching is nontrivial due to warps, efficient representation methods can be helpful. A considerable number of such methods have been proposed in the past, including Symbolic Aggregate approXimation (SAX) [70], bit-level approximation [71], and Piecewise Aggregate Approximation (PAA) [72][1]. In this chapter specifically, we focus on the representation of time series by means of *landmarks* [73] (also referred to as key-points [74], break-points [75] and change-points [76]), which can be thought of as those points in the time series that are obviously remarkable (peaks, valleys, inflection points, ...). Rather than matching every detail of the data and the template, only the landmarks will be matched, and subsequent landmarks will be checked for their relationship to one another.

We match the given template to the actual data in three steps. The first step involves transforming the time series into a landmark sequence, which preserves all the prominent features. The second step is landmark subsequence selection, which is based on the constraints over the landmarks occurring in the template. The third step is landmark model construction, which introduces *trust feature*

[1] A comprehensive list of representation methods for time series is given in Section 6.7.

and *trust region* to model the time series segments corresponding to the selected landmark subsequence. Unlike most of the representation and similarity methods, which are designed mainly for full sequence matching [28], our proposed approach is capable of processing both full sequence and subsequence matching of various length, while being less sensitive to noise, and being able to handle deformations in both magnitude and temporal dimensions.

One of the challenges when extracting landmarks from actual data is the noise and high-frequency vibrations that are included. An obvious step to get rid of such distractions and to produce a set of meaningful landmarks is to convolve the signal with a smoothing kernel. The question now becomes what level of smoothing is appropriate for the template in question. Too much smoothing may cause one to miss characteristic landmarks in the data, and too little smoothing will cause an abundance of landmarks at every little disturbance in the data. We propose an MDL-based solution to this challenge, that picks the correct smoothing level. Minimum Description Length (MDL) [77, 78, 79] is an information-theoretic model selection framework that selects the best model according to its ability to *compress* the given data.

The contributions of this chapter are summarised as follows:

- It provides a general definition of a template for time series, which can be represented by a landmark vector.
- It proposes the use of landmarks: a triple involving temporal, magnitude and type information.
- It takes the relationship between landmarks within a landmark sequence as constraints for landmark subset selection.
- It introduces the concept of a trust region from the image processing domain [80] to time series to build a reliable template model, which could help to detect the precise location of landmarks.
- It employs MDL [77, 78, 79] for selection of the right smoothing level for landmark extraction.

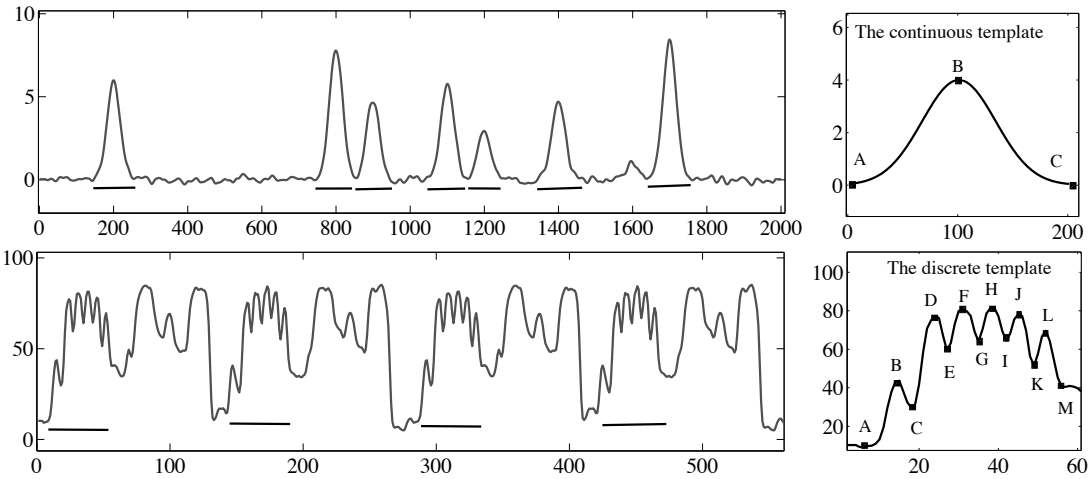

Figure 6.1: The continuous template and the discrete template - The signal in the top left picture is the time series. The curve in the top right picture is a continuous template (more specifically a Gaussian), which is marked with landmarks A, B, and C. The bottom left picture represent bird songs. The curve in the bottom right picture is a discrete template, corresponding to one of the selected subsequences, marked with landmarks A, B, . . . , M.

The rest of this chapter is organised as follows. Section 6.2 gives the definitions of template and landmark, and specifies the task of predefined pattern detection. Section 6.3 introduces the concept of landmark constraints. Section 6.4 introduces landmark model construction based on continuous and discrete templates. Section 6.5 uses MDL to select the optimal smoothing scale. Section 6.6 evaluates the proposed method by applying it to artificial and real datasets. Section 6.7 gives a literature review of related work, followed by a conclusion in Section 6.8.

6.2 Preliminaries

In order to specify the exact predefined temporal pattern one hopes to find in the time series, we define a *template* in one of two ways. In the first, *continuous*, way, we assume that a temporal pattern is defined by a function that specifies the shape of the pattern with infinite precision. In the second way, the *discrete*

one, a temporal pattern is defined by a sequence of values, for example obtained by averaging a number of selected subsequences of interest.

Definition 5 *A* continuous template $\boldsymbol{H}_c$ *is a function that can serve as a model for subsequences of a time series*

$$\boldsymbol{H}_c(x) = f_A(x)$$

where x is an integer, and f is a given function with coefficients A (for example $\{\mu, \sigma\}$ in the case of a Gaussian curve).

We demonstrate this type of template using an artificial dataset, shown as the curve in the top left picture of Fig. 6.1. The shape of the recurring subsequence can be modeled faithfully with a Gaussian function, an instance of which is shown as the curve in the top right picture of Fig. 6.1. The matching subsequences are identified by the bars below the graph.

Definition 6 *A* discrete template $\boldsymbol{H}_d$ *is a time series that can serve as a model*

$$\boldsymbol{H}_d = (h_1, h_2, \ldots, h_k), \quad h_i \in \mathbb{R}$$

where k specifies the size of the subsequence. The recurring subsequences in the bottom left picture of Fig. 6.1 (depicting bird songs [81]) are more complicated than the patterns in the top left picture of Fig. 6.1. We could choose one subsequence from the smoothed time series as a template, such that the discrete template becomes the one shown on the bottom right.

6.2.1 Landmark Extraction

Although we expect the user to specify the predefined pattern in terms of a template (be it discrete or continuous), we will not be matching the template directly to the given time series. Rather, we intend to extract important *landmarks* [73] from both the template and the time series, and use these to match more efficiently and effectively. A landmark is defined as follows:

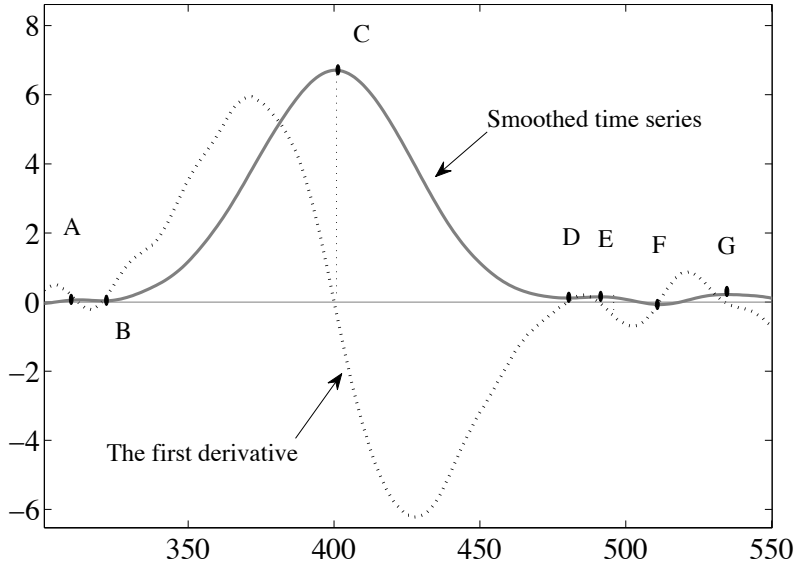

Figure 6.2: Landmark extraction - The dark curve is part of the smoothed time series. The dotted line is the scaled first derivative of the smoothed time series. The points marked with letters are landmarks.

Definition 7 *Given a time series* $\boldsymbol{T} = (t_1, t_2, \ldots, t_n)$*, a* landmark *is a remarkable point in* $\boldsymbol{T}$*, specified by a triple* l*:*

$$l = (id, m, type), \quad id \in \mathbb{N}, m \in \mathbb{R}$$

where id is the index of the landmark in the time series $\mathbf{T}$: the landmark is located at t_{id}. m is the magnitude of the landmark, *type* is the peak type indicator, which can be local extreme, inflection point or some other notable characteristic of the time series at this point.

In later sections, we will be introducing *landmark extraction* methods, which produce a sequence $\mathbf{L}$ of landmarks from a given time series. Such a method, generally identified as a function E, can be applied to obtain a sequence of remarkable points from a given time series, but equally, it can be used to produce such points from a (discrete) template, as that is essentially a time series also.

Landmark extraction methods are typically application dependent. In general, local extrema of a smoothed time series are good landmark candidates. They

are found by considering the zero-crossings of the first derivative of the series. These zero-crossings (roots) correspond to the extrema in the time series, which we assume to be of interest. The *inflection points* derived from the extrema in the first derivative time series can also be considered landmark candidates. Such landmarks can be found by looking at the zero-crossings of the second derivative. A landmark sequence preserves the main features of the time series, but significantly reduces its representation size. As shown in Fig. 6.2, the time series segment with a length of 250 can be compressed to a landmark sequence of only 7 elements. Note the importance of convolution with a smoothing kernel (e.g. a Gaussian) in order to get rid of the noise, which would produce an over-abundance of landmarks.

For each sequence of landmarks, there is a time series *segment* corresponding to it, which is defined as:

Definition 8 *Given a landmark sequence* $\boldsymbol{L} = (l_1, l_2, \ldots, l_k)$ *of length* k*, a* landmark segment $\boldsymbol{S}$ *of* $\boldsymbol{L}$ *is defined as a subsequence of time series* $\boldsymbol{T}$*:*

$$\boldsymbol{S} = (s_1, s_2, \ldots, s_m) = (t_{start}, \ldots, t_{end}),$$

where t_{start} and t_{end} are the data points indicated by indexs (id) of l_1 and 1_k.

6.2.2 Predefined Pattern Detection

With the definitions of templates and landmarks now established, we can proceed by formally specifying the main task that we are concerned within this chapter, as follows:

Definition 9 *The task of* Predefined Pattern Detection *takes as input a time series* $\boldsymbol{T}$*, a (discrete or continuous) template* $\boldsymbol{H}$ *and a landmark extraction method* E_σ*, and produces a sequences of matches* $\boldsymbol{M} = (m_0, \ldots, m_k)$*, where each* m_i *is an index in* $\boldsymbol{T}$ *where a match is found between the template and the subsequence starting at* m_i*.*

Note the role of E_σ in this definition. As mentioned, we are not matching the template to the time series directly, but rather extracting landmarks from both

first, using E_σ. An important parameter in E is σ, which determines the level of smoothing applied to both the template and the time series. By smoothing, we prevent noise from playing a role in determining what constitutes a landmark. Of course, the level of noise (as opposed to the actual signal) depends on the application, so for the moment we assume this as simply a parameter of the task. In Section 6.5.1, we will describe how the MDL principle can be employed to decide on a proper choice of σ.

6.3 Landmark Constraints

In theory, for a given template landmark sequence of length n and a time series landmark sequence of length m, there are $m - n + 1$ candidate landmark subsequences. Compared with the subsequence candidates from the original time series, the number has already been reduced a lot. However, there are still many ways in which landmarks in the template can be matched with those in the time series. In this section, we introduce landmark constraints to break the landmark sequence into a number of meaningful landmark subsequences.

For a given template, the landmarks in its landmark sequence signify more than just several data points obtained with landmark extraction methods. There are two levels of constraint existing in the landmark sequence of the template:

1. The first level is local constraints. As defined in Section 6.2, each landmark has three properties: index, magnitude and type indicator. We can set constraints based on each landmark property.

2. The second level is global constraints. The number of landmarks within the template landmark sequence determines the length of interesting landmark subsequences: the relationship between properties of different landmarks could form an even richer constraint-set.

For example, based on the template landmark sequence $\mathbf{L}_H = \{\text{A, B, C}\}$ in the top right picture of Fig. 6.1, the constraint-set can be set as follows:

- The length of landmark subsequences should be 3.

- The first landmark A and the third landmark C should be valley points.
- The second landmark B should be a peak point.
- The magnitude of landmark B should be higher than the magnitude of A and C.
- The relative magnitude $B_m - A_m$ should be higher than 0.8.
- The peak duration $C_{id} - A_{id}$ should be longer than 50.

The thresholds of the above constraints should be general enough to include all the potentially interesting patterns, and at the same time, they should be strict enough to filter out false patterns.

6.4 Fitting Templates to the Data

In Section 6.2, a template is defined as either a continuous or discrete template. According to the type of the template, we build two types of landmark models: the continuous landmark model and the discrete landmark model.

6.4.1 Continuous Landmark Model

For a given continuous template $\mathbf{H}_c$, and a landmark segment $\mathbf{S}$, the continuous landmark model $\mathbf{LM}_c$ of the landmark segment $\mathbf{S}$ is an instance of the continuous template $\mathbf{H}_c$. The coefficient set A of the continuous template $\mathbf{H}_c$ can be obtained by extracting features from the landmark subsequence of $\mathbf{S}$.

For example, a Gaussian function is chosen as the continuous template $\mathbf{H}_c = ae^{-(x-\mu)^2/(2\sigma^2)}$ to model the bell-like landmark segments in the middle and right pictures of Fig. 6.3. An instance of the template is shown as the curve in the left picture of Fig. 6.3, which is marked with landmarks A, B and C. The template is characterized with three features: σ, the peak location μ and the magnitude a. The peak location is derived from landmark B. σ can be derived from the temporal difference between any pair of landmarks, so there are 3 combinations

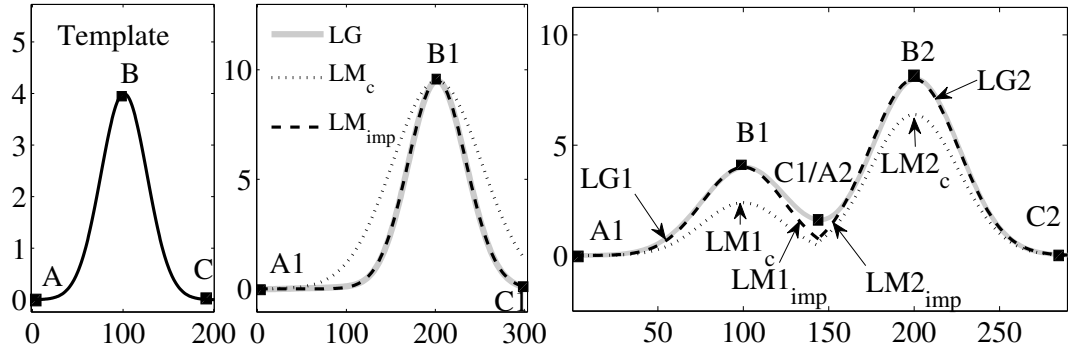

Figure 6.3: The continuous landmark model - The curve in the left picture an instance of the given continuous template, whose landmarks are A, B and C; the dotted curves in the middle and right pictures are false continuous landmark models; the dashed curves in the middle and right pictures are the improved continuous landmark models based on trust features.

for the template landmark sequence. The magnitude a can be obtained from the magnitude difference between the landmarks B and A, or that between B and C.

Incorrect feature choices may lead to false landmark models: the dotted curve LM_c in the middle picture of Fig. 6.3 is a false landmark model caused by incorrect choice of σ (the temporal difference between landmarks C1 and A1 divided by 2); the dotted curves in the right picture of Fig. 6.3 are false landmark models caused by incorrect choice of magnitudes (the magnitude difference between landmarks B1 and C1 for $\mathrm{LM1}_c$, and that between landmarks B2 and A2 for $\mathrm{LM2}_c$).

Trust feature To overcome the limitations existing in the continuous landmark models mentioned above, we introduce the notion of trust feature. A feature is considered to be a trust feature if it reflects the true characteristics of a given landmark segment. Shown as the dashed curve in the middle picture of Fig. 6.3, the improved landmark model LM_{imp} is obtained by computing σ from the temporal difference between landmarks C1 and B1, which can be further improved by selecting more reliable landmarks, such as inflection points. Similarly, we improve the landmark models in the right picture of Fig. 6.3 by employing the magnitude

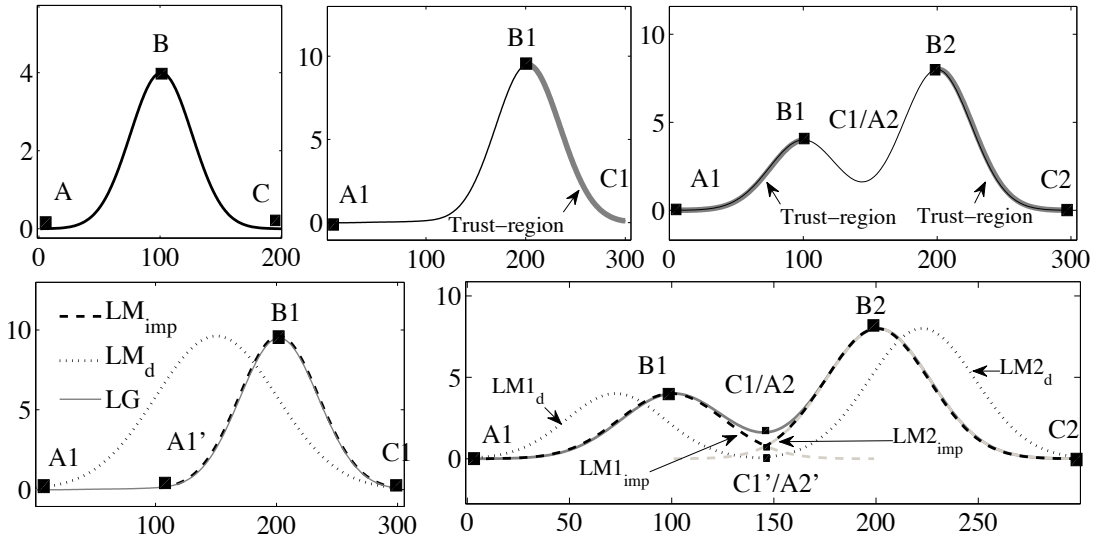

Figure 6.4: The discrete landmark model - The curve in the left picture is the given discrete template, which is marked with landmarks A, B and C; the segment between the landmark B1 and C1 in the top middle picture is chosen as the trust region; in the top right picture, the segment between the landmark A1 and B1 is the trust region of the first landmark segment, and that between the landmark B2 and C2 is the trust region of the second landmark segment; the dotted curves in the bottom left and right pictures are discrete landmark models simply obtained with transformations of the given template; the dashed curves in the bottom left and right pictures are improved landmark models based on the trust regions in the top middle and right pictures.

difference between landmarks B1 and A1 for LM1_{imp}, and that between landmarks B2 and C2 for LM2_{imp}, shown as the dashed curves.

6.4.2 Discrete Landmark Model

For a given discrete template $\mathbf{H}_d$ of length n and a landmark segment $\mathbf{S}$ of length m, we model $\mathbf{S}$ through scaling $\mathbf{H}_d$ in both temporal and magnitude dimensions. The temporal scale operation *X-scale* results in a new sequence $\mathbf{LM}_X$, whose length is the same as that of the landmark segment.

$$\mathbf{LM}_X = \textit{X-scale}(\mathbf{H}_d, m, n), \quad m, n \in \mathbb{N}$$

$$X\text{-}ratio = m/n$$

If *X-ratio* > 1, the *X-scale* is an up-sampling operation, and if *X-ratio* < 1, the *X-scale* is a down-sampling operation, otherwise, the $\mathbf{LM}_X$ is the same as the template $\mathbf{H}$.

We continue to process the obtained model $\mathbf{LM}_X$ with the magnitude scale operation *Y-scale*, and obtain a landmark model $\mathbf{LM}_Y$, which can be taken as a primary approximation of the landmark segment $\mathbf{S}$.

$$\mathbf{LM}_Y = Y\text{-}scale(\mathbf{H}_d, \mathbf{LM}_X)$$

$$Y\text{-}ratio = (\max(\mathbf{S}) - \min(\mathbf{S}))/(\max(\mathbf{H}_d) - \min(\mathbf{H}_d))$$

where $\mathbf{LM}_Y$ is obtained by first scaling $\mathbf{LM}_X$ with a ratio *Y-ratio*, resulting in a temporary model $\mathbf{LM}_{XY}$, then shifting along magnitude dimension by $\min(\mathbf{S}) - \min(\mathbf{LM}_{XY})$.

Based on the transformations mentioned above, we can define the discrete landmark model as:

Definition 10 *Given a landmark segment* $\boldsymbol{S} = (s_1, s_2, \ldots, s_k)$ *of length* k*, and a discrete template* $\boldsymbol{H}_d$*, the* discrete landmark model $\boldsymbol{LM}_d$ *of* $\boldsymbol{S}$ *is a sequence derived from transformations of the discrete template* $\boldsymbol{H}_d$*:*

$$\boldsymbol{LM}_d = Transf(\boldsymbol{S}, \boldsymbol{H}_d)$$

where *Transf* are the transformations defined above (*X-scale* and *Y-scale*).

The dotted curve $\mathbf{LM}_d$ in the bottom left picture of Fig. 6.4 is a discrete landmark model obtained by simply transforming the given template shown as the curve in the top left picture of Fig. 6.4. The obtained landmark model indicates that the left boundary of the landmark segment is incorrect, which is caused by the false landmark A1. The gray curves in the bottom right picture of Fig. 6.4 are two overlapping landmark segments, whose boundaries are correctly caught, but their landmark models, shown as the dotted curves $\mathrm{LM1}_d$ and $\mathrm{LM2}_d$, are still incorrect. This is caused by the complex structure of the time series, which has not been covered by the given template. To overcome all these limitations, we introduce the notion of trust region.

6.4.2.1 Trust Region

By assuming part of the landmarks within a landmark subsequence are reliable, we can define the corresponding segment between these landmarks as the trust region. Trust region is a concept borrowed from the field of image processing. We introduce the concept into time series (or to be more precise, into the discrete landmark model), and define it as:

Definition 11 *Given a landmark segment* $\boldsymbol{S} = (s_1, s_2, \ldots, s_n)$ *of length* n*, and its landmark sequence* $\boldsymbol{L} = (l_1, l_2, \ldots, l_k)$ *of length* k*, and assuming the segments between landmarks* l_i *and* l_j *(*$1 \leq i < j \leq k$*) are influenced less by noise, then the* trust region *of* $\boldsymbol{S}$ *is defined as:*

$$\boldsymbol{S}_{trust} = (s_a, \ldots, s_b), \quad 1 \leq a < b \leq n$$

where a is the index of landmark l_i in the landmark segment $\mathbf{S}$, and b corresponds to the index of landmark l_j. The dashed curves in the bottom left and right pictures of Fig. 6.4 are discrete landmark models obtained with the trust regions given in the top middle and right pictures, which indicate that the trust regions help to achieve improved discrete landmark models and updated landmark segments. The detailed procedure is illustrated in Algorithm 1.

The inputs of the algorithm are: a template $\mathbf{H} = (h1, \ldots, hn)$ of length n, a landmark segment $\mathbf{S} = (s1, \ldots, sm)$ of length m, and their landmark sequences ($\mathbf{L}_H = (lh_1, \ldots, lh_k)$ and $\mathbf{L}_S = (ls_1, \ldots, ls_k)$, respectively, both of length k). The outputs of the algorithm are an improved landmark model and an updated landmark segment.

A candidate trust region $\mathbf{H}_{cand}$ (identified by *index* and *len*) of the template and the landmark segment $\mathbf{S}_{cand}$ can be obtained with the $Region(\mathbf{S}, \mathbf{H}, index, len)$ operation, in which *index* is the index of the first landmark in $\mathbf{L}_H$ and $\mathbf{L}_S$, and *len* is the length of the subsequence. Based on these two parameters, we need to obtain the indexes of the first and last data points of the candidate trust regions in $\mathbf{S}$ and $\mathbf{H}$. Assuming the corresponding indexes of $\mathbf{H}$ are a and b, and that of $\mathbf{S}$ are c and d, the trust region of the template can be represented as

Algorithm 1 The landmark model.

Require: a template $\mathbf{H}$, and its landmark sequence $\mathbf{L}_H$ of length k, a landmark segment $\mathbf{S}$, and its landmark sequence $\mathbf{L}_S$ of length k.

Ensure: an improved landmark model $\mathbf{LM}_{imp}$ and an updated landmark segment $\mathbf{S}_{up}$

```
S_cand = S, H_cand = H
[LM_imp, S_up] = Model(S, S_cand, H, H_cand)
sim_max = Similarity(LM_imp, S_up)
for len = 2 to k do
  for index = 1 to k − len + 1 do
    [H_cand, S_cand] = Region(S, H, index, len)
    [LM_temp, S_temp] = Model(S, S_cand, H, H_cand)
    sim = Similarity(LM_temp, S_temp)
    if sim > sim_max then
      sim_max = sim
      LM_imp = LM_temp
      S_up = S_temp
    end if
  end for
end for
```

$\mathbf{H}_{cand} = (h_a, \ldots, h_b)$, and that of the landmark segment can be represented as $\mathbf{S}_{cand} = (s_c, \ldots, s_d)$, respectively.

Based on the candidate trust regions $\mathbf{H}_{cand}$ and $\mathbf{S}_{cand}$, the associated landmark model $\mathbf{LM}_{temp}$ and the updated landmark segment $\mathbf{S}_{temp}$ can be obtained with the $Model(\mathbf{S}, \mathbf{S}_{cand}, \mathbf{H}, \mathbf{H}_{cand})$ operation. The procedure of this operation is illustrated as follows:

- Step 1: based on the candidate trust regions $\mathbf{S}_{cand}$ and $\mathbf{H}_{cand}$, we can obtain a new sequence $\mathbf{LM}_X$ with the *X-scale* operation:

$$\mathbf{LM}_X = X\text{-}scale(\mathbf{H}, d - c + 1, b - a + 1)$$

- Step 2: following the same *Y-scale* operation procedure as mentioned above,

we obtain a landmark model $\mathbf{LM}_Y$:

$$\mathbf{LM}_Y = \textit{Y-scale}(\mathbf{H}, \mathbf{LM}_X)$$

- Step 3: pruning the obtained landmark model $\mathbf{LM}_Y$ or fixing the landmark segment $\mathbf{S}$, we finally achieve a landmark model $\mathbf{LM}_{temp}$ and landmark segment $\mathbf{S}_{temp}$.

  This step takes the segment, derived from the region $\mathbf{S}_{cand}$, in the temporary landmark model $\mathbf{LM}_Y$ as reference. If $\mathbf{LM}_Y$ is longer than the landmark segment $\mathbf{S}$, we need to prune the temporary model, otherwise, fix the landmark segment.

Finally, the fit of each candidate landmark model is evaluated with the *Similarity*() operation, which can be any of the choices as outlined in Chapter 2, most typically a similarity function based on the Euclidean Distance.

6.5 Determining the Smoothing Level

As noted, the Predefined Pattern Detection task requires the specification of a smoothing level, in order for the landmark detection to work effectively. When smoothing a time series for landmark extraction, there is clearly a trade-off at play. Smoothing too little will produce a time series that shows too many landmarks, and smoothing too much will remove too much of the interesting signal, such that important landmarks may be overlooked (see Fig. 6.5). In this section, we tackle the challenge of setting an appropriate value for the smoothing scale σ in E_σ.

Our solution to this challenge employs the Minimum Description Length principle [77]. The MDL principle states that, when choosing between several different candidate models of the data, the one that produces the cheapest encoding is the most desirable. In this context, the different candidate models are produced by the different choices of smoothing scale σ. In a nutshell, we consider a range of values for σ, applying landmark extraction E_σ to the smoothed time series. The

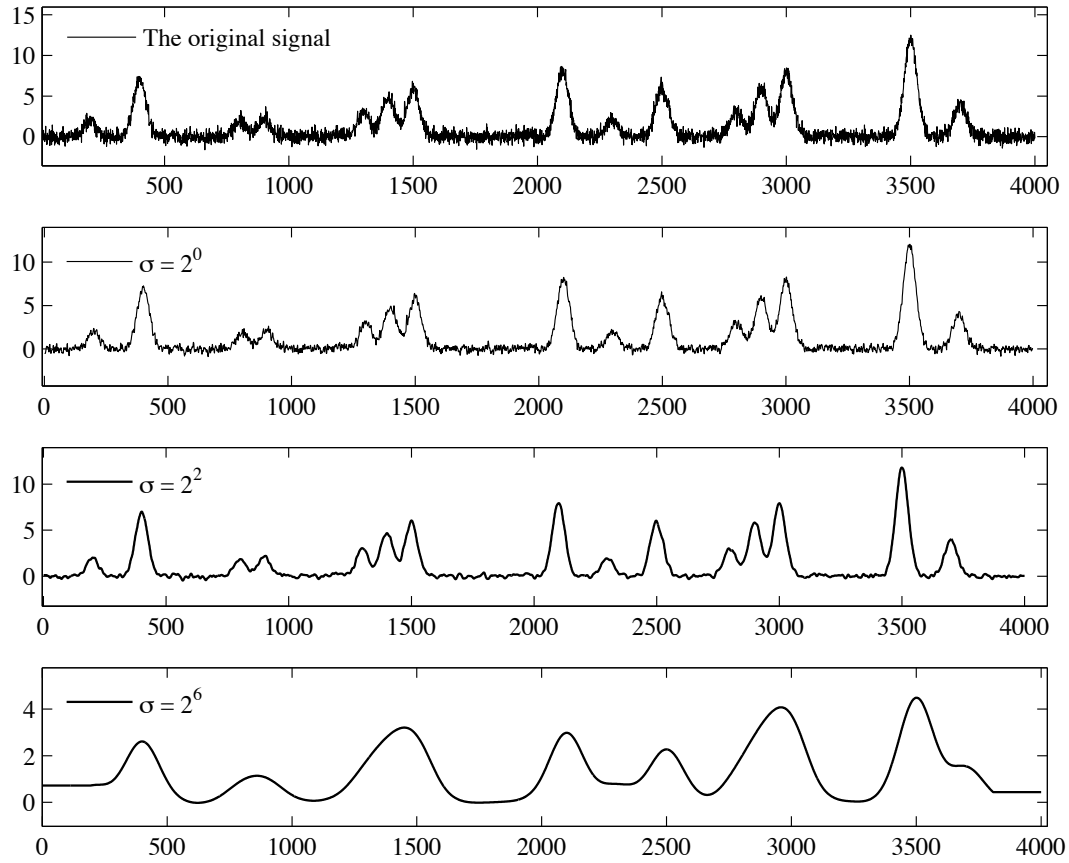

Figure 6.5: Various levels of smoothing - The first picture shows the original time series without any convolution; the second picture shows the smoothed time series with a scale of $\sigma = 2^0$, which still contains considerable noise; the third picture shows the smoothed time series with $\sigma = 2^2$, which suppresses the noise and preserves the interesting patterns; the fourth picture presents $\sigma = 2^6$, which suppresses both the noise and some of the interesting features.

idea of using MDL as a guiding principle to model various aspects of time series data has been introduced before in [79, 82], but not with the specific intent of selecting an appropriate choice of σ.

6.5.1 Minimum Description Length

We concentrate on the two-part version of the MDL principle, which states that the best landmark model LM to describe the time series $\mathbf{T}$ is the one that minimises the sum $L(LM) + L(\mathbf{T} \mid LM)$, where

- $L(LM)$ is the cost, in bits, of the landmark model derived from the given template.
- $L(\mathbf{T} \mid LM)$ is the length, in bits, of the description of the time series when encoded with the help of the landmark model LM, that is the residual information not represented by LM.

A good, detailed model that catches most features of the target dataset leads to a low cost of $L(\mathbf{T} \mid LM)$, but a good model also means a higher cost compared with a simple model. Therefore, a trade-off between model fit and its complexity is guaranteed by considering the size of the encoding. This property prevents the MDL method from overfitting.

Before we calculate $L(LM)$ and $L(\mathbf{T} \mid LM)$, we first need to discretise the landmark segment. We assume that the values t_i of the input time series $\mathbf{T}$ have been quantised to a finite number of symbols by employing the function defined below:

$$Q(t_i) = \left\lfloor (t_i - \min(\mathbf{T}))/(\max(\mathbf{T}) - \min(\mathbf{T})) \cdot N \right\rfloor - N/2$$

where N, assumed to be even, is the number of bins to use in the discretisation while $\min(\mathbf{T})$ and $\max(\mathbf{T})$ are respectively the minimum and maximum value in $\mathbf{T}$. Throughout the rest of the chapter, we assume $N = 256$, in correspondence with similar work on MDL in time series [79, 82]. One question that might arise is if such a quantisation removes meaningful information from the time series. In [82], the authors show that the effect of quantisation is rather modest on several time series from various domains.

6.5.1.1 Encoding of the Model

We will first discuss the encoding of the landmark model LM, which is derived from a given template. In the time series, the cost for encoding the landmark model is composed of two parts: the index and the model parameters. The location of any landmark segment candidate is less than the total length of the time series $\mathbf{T}$, so it can be encoded with $\log_2 n$ bits. Assuming there are m parameters for each model (for a continuous landmark model, m stands for the number of coefficients; for a discrete landmark model, m is the cost of transformations), and each parameter can be modelled with b bits, the total cost can be obtained by summing up these two parts:

$$L(LM) = k \cdot (\log_2 n + mb)$$

where k is the number of landmark segments in the time series that meet the landmark constraints.

6.5.1.2 Encoding the Data

The second part of MDL, $L(\mathbf{T} \mid LM)$, represents the residual information after subtracting the landmark model LM from the time series $\mathbf{T}$. To encode this part, we first need to introduce the notion of entropy.

Definition 12 *The* entropy *of a time series $\boldsymbol{T}$, discretised according to a set of values D, is defined as below*

$$Entropy(\boldsymbol{T}) = -\sum_{v \in D} P(t_i = v) \log_2 P(t_i = v)$$

where t_i stands for the i_{th} element in the time series $\mathbf{T}$, $P \log_2 P = 0$ in the case of $P = 0$, and $P(t_i = v)$ indicates the fraction of points in the time series which has value v.

Given the definition of entropy, we can define the description length of the second part of MDL as follows:

Definition 13 *Given a time series* $\boldsymbol{T}$ *of length* n*, the* description length *of* $L(\boldsymbol{T} \mid LM)$ *(in bits) is given by*

$$L(\boldsymbol{T} \mid LM) = n \cdot Entropy(\boldsymbol{T} \mid LM)$$

6.5.2 Smoothing Level Selection

For assessing a candidate smoothing level with parameter σ, we can simply take the corresponding smoothed landmark segments, which meet the landmark constraints, as landmark models, and obtain a residual by subtracting the obtained models from the original time series. We need two parameters (the indexes of the first and last data points of a landmark segment) to identify a landmark model. Assuming there are r interesting landmark segments under a smoothing scale σ, the landmark model cost $L(LM)$ becomes:

$$L(LM) = 2r \cdot log_2 n$$

The second MDL part $L(\mathbf{T} \mid LM)$, according to Def. 13 is represented as:

$$L(\mathbf{T} \mid LM) = n \cdot Entropy(\mathbf{T} - \sum_{i=1}^{i=r} \mathbf{Ts}_i)$$

where $\mathbf{Ts}_i$ is the i^{th} smoothed landmark segment.

For a given template and time series, we assume that the optimal degree of smoothing is the one that leads to the minimal total MDL cost.

6.6 Experiments

To show the effectiveness of the proposed method, we apply it to three different datasets, which ranges from artificial dataset to real-life datasets (traffic and ECG signals). We divide each dataset into two parts: training dataset and test dataset. The training dataset is used to detect the right smoothing scale and landmark constraints, then these obtained parameters will be applied on the test dataset.

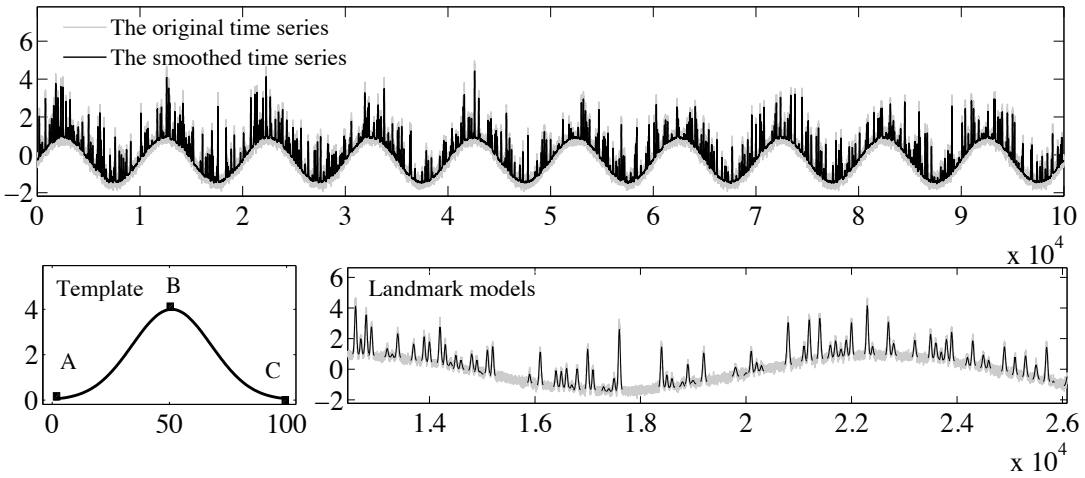

Figure 6.6: Predefined pattern detection from an artificial dataset - The grey curve in the top picture is an original artificial dataset composed of 100,000 data points; the black curve in the top picture is the time series smoothed with the smoothing scale $\sigma = 2^3$; the black curve in the bottom left picture is an instance of the selected template, which is a Gaussian function, marked with landmarks A, B, C; the black peaks in the bottom right picture are the detected patterns in a fragment of the whole dataset.

6.6.1 Artificial Dataset

We begin with testing the method on an artificial dataset, which is obtained by combining Gaussian peaks, random high-frequency noise and a slowly fluctuating baseline. The Gaussian peaks are the interesting patterns we want to detect. Shown as the grey curve in the bottom right picture of Fig. 6.6, the artificial dataset is composed of 100,000 data points, including 481 useful Gaussian peaks. We take the first 20,000 data points as the training dataset, which includes 104 interesting patterns, and take the remaining 80,000 data points as the testing dataset, which includes 377 interesting patterns.

As we have a function to describe the pattern of interest, we use a continuous template (Gaussian function) here, an instance of which is shown as the bottom left picture of Fig. 6.6. The template can be marked with 5 landmarks, in which A and C are the begin and end points; B is the data point that has the maximum

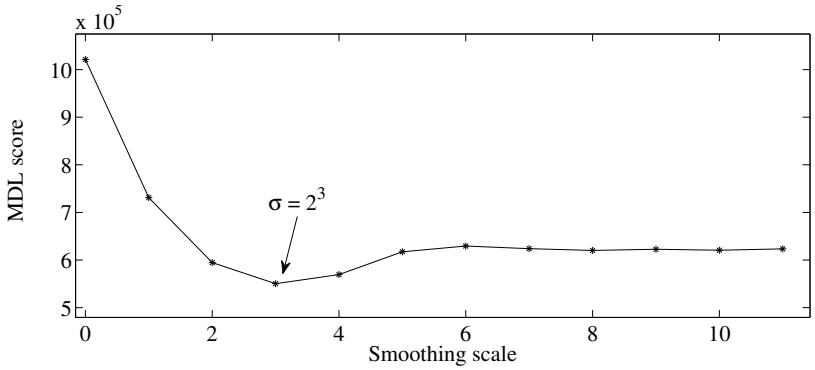

Figure 6.7: The smoothing scale for an artificial dataset - This picture shows a scatter plot between smoothing scale and MDL score, in which the fourth smoothing scale corresponds to the minimal MDL score.

magnitude in the template. We utilise the first-derivative method to extract landmarks from the artificial dataset. Because the first-derivative method is sensitive to noise, we cannot apply it directly to the raw dataset. We first smooth the raw dataset with the convolution method mentioned in Section 6.5.1, and then transform the smoothed dataset to a landmark sequence.

We calculate MDL scores based on a smoothing scale candidate set $\{2^0, 2^1, 2^2, \ldots, 2^{11}\}$, and choose the scale with the minimal MDL score as the right smoothing scale, which is 2^3 in this case, shown as Fig. 6.7. The landmark constraints that succeed in identifying all the 104 interesting patterns in the training dataset are chosen as the target landmark constraints, as follows:

- The length of landmark subsequences should be 3.
- The first and last landmarks should be valley points.
- The second landmark should be a peak point.
- The peak magnitude should be no less than 0.15.

Based on the obtained smoothing scale and landmark constraints, we detect 380 landmark segments, 377 of which are true peaks, and the remaining 3 landmark segments are caused by noise. The precision of the continuous landmark model is thus 99.2%, and the recall is 100%.

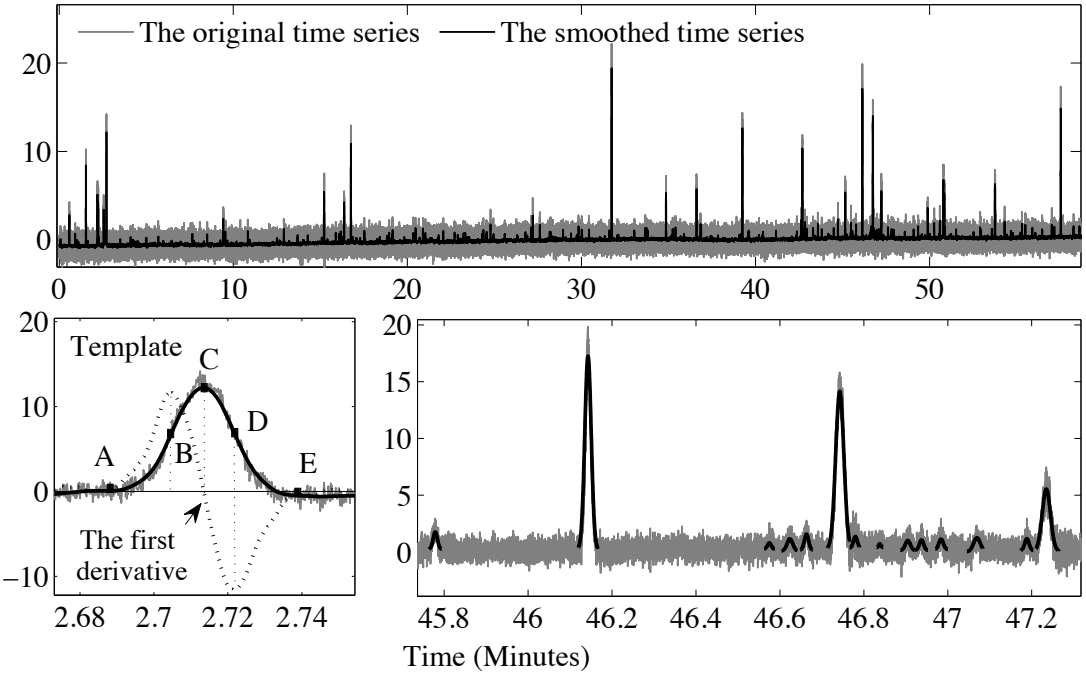

Figure 6.8: Traffic event detection from a traffic dataset - The grey curve in the top picture is the raw strain signal collected from one strain sensor installed on a highway bridge; the black curve in the top picture is the strain signal smoothed with the smoothing scale $\sigma = 2^4$; the black curve in the bottom left picture is the selected discrete template, marked with landmarks A, B, C, D and E; the black peaks in the bottom right picture are landmark models.

6.6.2 Real-life Traffic Dataset

In this section, we apply a discrete template to a real-life traffic dataset collected at the Hollandse Brug. We select a piece of strain signal at 100 Hz of 1 hour at 3:00 AM to detect traffic events, which is composed of 360,000 data points. Using the video record of this period, we label the traffic events as cars or trucks. We take the first 60,000 data points as the training dataset, including 23 cars and 6 trucks, and the remaining 300,000 data points as the test dataset, including 150 car events and 14 truck events.

Following a similar MDL-based procedure as in the previous experiment, the smoothing scale is determined as $2^4 = 16$, shown as Fig. 6.9. The black curve in

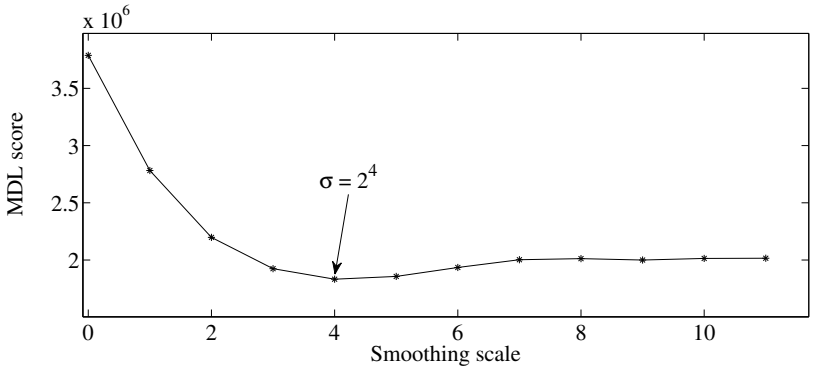

Figure 6.9: The smoothing scale for a traffic dataset - This picture shows a scatter plot of MDL score as a function of smoothing scale, in which the fifth smoothing scale corresponds to the minimal MDL score.

the top picture of Fig. 6.8 is the smoothed curve under this optimal smoothing scale. We select the 6 trucks in the smoothed training dataset for template construction. In order to select the most representative sequence to act as the template, we actually employ MDL as a model selection framework. The truck data that leads to the minimal MDL score (on the training data) is chosen as the discrete template, shown as the peak in the bottom left picture of Fig. 6.8.

The testing dataset is smoothed with a $\sigma = 16$ Gaussian kernel and the landmark constraints are set as:

- The length of landmark subsequences should be 5.
- The first and last landmarks should be valley points.
- The second and fourth landmarks should be inflection points.
- The third landmark should be a peak point.
- The peak magnitude should be no less than 0.44.

Based on the obtained smoothing scale and landmark constraints, we manage to catch 170 landmark segments in the test data: 14 of which correspond to truck events (as validated through the video of the bridge), 150 of which correspond to car events, and the remaining 6 landmark segments are caused by noise.

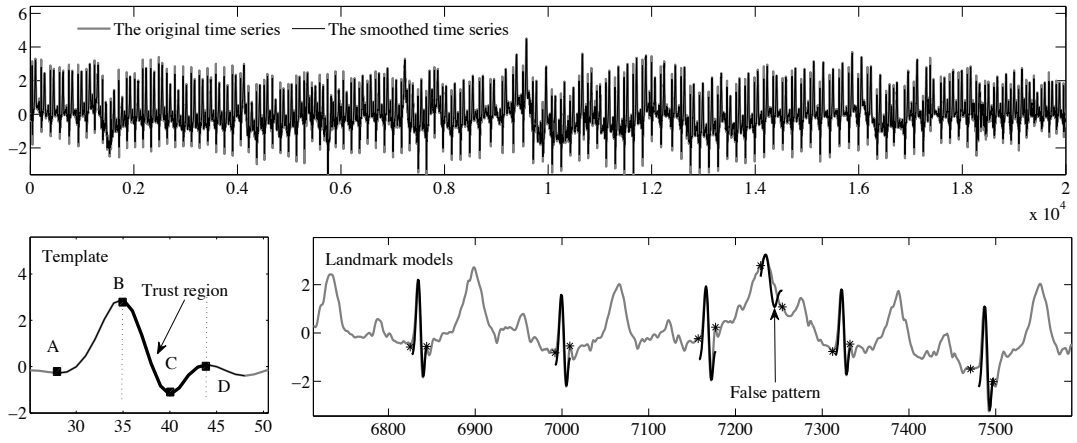

Figure 6.10: The QRS complex detection from an ECG dataset - The grey curve in the top picture is an ECG dataset taken from a real patient, the black curve is a smoothed time series; the curve in the bottom left picture is a discrete template of the QRS complex, which is composed of a R wave and a S wave, and can be marked with four landmarks A, B, C and D; the thick black segment between landmarks B and D is chosen as the trust region; the black peaks in the bottom right picture are landmark models, one of which is a false pattern.

The precision of the continuous landmark model is thus 96.5%, and the recall is 100%.

Our algorithm is quite fast. To give an indication of the efficiency on this relatively large dataset, the total running time was 4.47 seconds on the test set (2.51 seconds for landmark subsequence selection and 1.96 seconds for the landmark models).

6.6.3 ECG Signal

The electrocardiogram (ECG) signal is used to measure the electrical activity of the (human) heart [83]. A single heart beat is typically composed of 5 deflections, called the P, Q, R, S and T wave, in which the Q, R and S waves are often considered together as the QRS complex, because they are closely linked. Note that not every QRS complex contains all the three wave elements, and any combination of

these waves can also be referred to as a QRS complex [84]. Accurately recognising the QRS complex and distinguishing them from the other noise sources such as P and T waves is a critical technology for many clinical instruments [85].

In this section, we choose an ECG dataset of 20,000 data points from [86], which is collected at a frequency of 250 Hz. We take 2,000 data points as the training dataset, which consists of 13 QRS complexes, and the remaining 18,000 data points as the test dataset, containing 106 QRS complexes. Since there is only little noise in the original time series, the optimal smoothing scale comes out as $\sigma = 2^0$ (see also the top picture of Fig. 6.10).

We take the 13 QRS complexes in the training dataset for template construction. The curve in the bottom left picture of Fig. 6.10 is selected as the discrete template. The template can be represented using four landmarks, which are extracted with the first-derivative method. The landmarks B and D are the least sensitive to disturbances, so the landmark segment between these two landmarks is selected as the trust region, shown as the bottom left picture in Fig. 6.10. Based on the training dataset and smoothing scale, the landmark constraints are set as:

- The length of each landmark subsequence should be 4.
- The first and third landmarks should be valley points.
- The second and the fourth landmarks should be peak points.
- The magnitude of the second landmark should be the highest one in the landmark subsequence.
- The magnitude of the third landmark should be the lowest one in the landmark subsequence.
- The temporal difference between the last and the first landmark should be less than 25.
- The magnitude difference between the second and the third landmark should be less than 2.

Based on the obtained smoothing scale and landmark constraints set above, we manage to catch 107 landmark segments: 103 of which are true QRS complexes, 4 of which are false QRS complexes, and 3 true QRS complexes are missing. The precision of the continuous landmark model is thus 96.3%, and the recall is 97.2%. Fig. 6.10 shows one instance of such a false detection, between time points 7,200 and 7,300.

This figure also demonstrates the purpose of landmark models. Note that the detection of predefined patterns (the core of our work) produces a list of consecutive landmarks for each instance of the template detected. When visualising an instance in the actual data, pointing out the landmarks in question is of limited interest. By means of the landmark models, the matching of the template to the actual data can be determined (including unreliable segments of the data), such that the transformed instance of the template can be overlain on the time series, as is demonstrated in the figures in this section.

6.7 Related Work

In this chapter, we have presented three concepts that have been extensively used in image matching fields: templates [87, 88], landmarks [88, 89] and trust regions (or trust features) [80].

Template matching can be used for face detection [90], duplicate document detection [91] and motion classification [92]. The concept of template has been introduced to time series to detect specific patterns or shapes [31, 33, 69, 93, 94]. Frank et al. [94] propose Geometric Template Matching (GeTeM) which uses time-delay embeddings for building models from segments of time series and compares the reconstructed dynamical systems in terms of their state space as well as their dynamics. In [93], a novel and flexible approach is proposed based on segmental semi-Markov models. In [31, 33, 69], meaningful templates are constructed with shape-based averaging algorithms, such as Prioritized Shape Averaging (PSA) [69] and Accurate Shape Averaging (ASA) [31]. Wei et al. propose the Atomic Wedgie method "that exploits the commonality among the predefined patterns

to allow monitoring at higher bandwidths, while maintaining a guarantee of no false dismissals" [68]. Most of the proposed methods are mainly designed for full sequence matching, which are ineffective in detecting predefined patterns from streaming time series.

Landmarks can be used to break time series into meaningful segments, and a template can be featured by a vector of landmarks. Landmarks are also referred to as key-points [74], break-points [75] and change-points [76]. Perng et al. [73] propose a feature-based technique called the landmark model, which uses landmarks instead of the raw data for processing. A two-level representation [74] is proposed to recognise gestures, using both local and global features. In practice, the reliability of each landmark varies with its location. To the best of our knowledge, this hasn't been mentioned in the literature.

It has been pointed out by researchers that some unspecified portions of streaming time series should be ignored [81, 95] to achieve a better result, which means some data points have nothing to do with predefined patterns, and should be filtered out. Ye and Keogh [96] propose a new time series primitive, time series shapelets, for time series classification. The shapelets are informally defined as the subsequences that are in some sense maximally representative of a class. This method is interpretable and accurate in classifying static time series [97], but is ineffective in handling streaming time series. Inspired by these works, we introduce trust region (trust feature) into streaming time series to obtain a more reliable landmark model.

A number of representation methods have been developed in the literature to reduce the dimensionality of time series, such as Discrete Fourier Transform (DFT) [25], Single Value Decomposition (SVD) [98], Discrete Wavelet Transform (DWT) [99]. There are also some researchers who employ symbolic representations, such as Symbolic Aggregate approximation (SAX) [70] and bit-level approximation [71]. Features extracted from time series carry summarized information of the time series [27, 100], which can represent the original time series concisely [101], and are less sensitive to noise [102], so the feature extraction operation can also be used to reduce dimensionality (reduce the size of the data), such as Amplitude-Level Features (ALF) [103], characteristic-based clustering (CBC) [100]. Some

representations are based on piecewise techniques, such as Piecewise Linear Approximation (PLA) [75], Piecewise Aggregate Approximation (PAA) [72], Adaptive Piecewise Constant Approximation (APCA), Derivative Time Series Segment Approximation (DSA) [101, 104] and Piecewise Vector Quantized Approximation (PVQA) [105, 106]. Some representations aim to keep both local and global information about the original time series, such as Multi-resolution Vector Quantization (MVQ) approximation [107] and multi-resolution PAA (MPAA) [108].

Next to the representation methods, a number of similarity measures have been proposed [24], of which the Euclidean Distance (ED) [25, 26] is the most common [27, 28]. However, when shifting and temporal distortions exist in the given subsequences, the ED is proven to be ineffective [28]. To handle stretching and compression along the temporal dimension, Dynamic Time Warping (DTW) [30] was proposed, which achieves an optimal temporal alignment through detecting the shortest warping path in a distance matrix [24, 31, 32, 33]. Finding the shortest warping path is a non-trivial problem, whose computation complexity can reach $O(n^2)$, where n is the number of data points. To speed up the computation of DTW, some lower bounding constraints, like LB_Keogh [32, 36] and the Ratanamahatana-Keogh Band [37], have been introduced to prune expensive computations, which can reduce the complexity to $O(n)$. There are also some other edit-based methods proposed to handle outliers and noise [24], such as Longest Common Subsequence (LCSS) [109], Edit Distance with Real Penalty (ERP) [110] and Edit Distance on Real sequence (EDR) [111]. However, most of the proposed methods focus mainly on temporal deformations [93], which are inadequate in dealing with shifting and scaling in the amplitude dimension [28]. Consequently, Spatial Assembling Distance (SpADe) [28] is proposed to handle shifting and scaling in both the temporal and amplitude dimensions.

6.8 Conclusion

Predefined pattern detection from streaming time series is a quite challenging topic, because it is not only sensitive to noise, but also sensitive to temporal and

magnitude deformations. A number of representation and similarity measure methods have been proposed to approximate interesting subsequences, but most of them are mainly designed for full sequence matching, and are ineffective when the disturbances mentioned above exist. Based on MDL, we smooth the streaming time series with a reasonable scale, and construct a template from the smoothed training time series. The template stands for the pattern of interest that we want to extract from the streaming time series. To carry out this task, we proposed a three-stage representation method, which first transfers the time series into a landmark sequence, and then utilizes the constraints within a template landmark sequence to select promising landmark subsequences of interest patterns, finally employing the trust region (or trust feature) to model candidate patterns. Most of the existing feature-based methods just focus on the quality of models, and pay little attention to the reliability of candidate patterns. Our landmark model overcomes this problem by transferring the template in both temporal and magnitude dimensions according to trust regions (or trust features).

Chapter 7

Modal Analysis

7.1 Background

In the Structural Health Monitoring field, damage detection methods are based on the premise that global modal parameters (natural frequencies, mode shapes and damping ratios) are functions of physical properties (mass, damping distribution, and stiffness) [45, 49, 112]. Changes in physical properties will cause changes in the modal parameters [40, 43, 49].

Modal analysis is a procedure that extracts modal parameters of a structure from its measured response data. Modal analysis was originally used for Experimental Modal Analysis (EMA), primarily applied to aerospace and mechanical structures, where the structures are excited by controlled dynamic forces. The responses to these forces are then recorded, and the modal parameters are obtained based on both input and output measurements [8]. Due to improvements in computing capacity, technological advances and developments in sensors and data acquisition systems, these analysis techniques can also be applied in SHM systems for civil infrastructures. In SHM, modal analysis is often applied as a form of Operational Modal Analysis (OMA) [14]. The major difference between OMA and EMA is that the input forces of OMA are unknown, and only the output measurements are available. Considering a highway bridge under normal

in-service conditions, the input forces may include various vehicles and environmental effects, such as wind and temperature changes, influences which are difficult to measure or quantify. Unfortunately, various techniques upon which EMA relies are invalid for OMA.

Driven by the demand for assessing the health of civil structures, a number of powerful techniques for OMA have been developed. Some common techniques are the Peak-Picking (PP) method [113, 114], the Auto Regressive-Moving Average Vector model [115], the Natural Excitation Technique (Next) [5, 116], the Random Decrement Technique [117], the Frequency Domain Decomposition [6] and the Stochastic Subspace Identification (SSI) [11, 114, 118]. The SSI algorithm is known as one of the most robust methods for OMA measurements, and has already been successfully applied to infrastructures under operational conditions, such as bridges [50, 119], towers [114, 120], and buildings [121, 122]. In this chapter, we will employ both the PP and the SSI methods for modal analysis.

In reality, modal parameters are not only sensitive to structural damage and degradation, but also to varying operational and environmental loadings, such as traffic, humidity, wind and most importantly, temperature [43, 49]. The modal changes caused by these factors can be much larger than those caused by real structural damage or degradation [112]. For reliable modal analysis, we must distinguish the abnormal changes caused by operational or environmental inputs from normal changes due to damage and degradation [45, 50]. In this chapter, we will take the influence of temperature and vehicle mass into account.

In this chapter, we begin with introducing the procedure of data selection in Section 7.2, then apply two modal analysis methods: the PP method and the SSI method, to extract modal parameters from the selected dataset in Section 7.3, and finally analyse the influence of temperature and vehicle mass on modal parameters in Section 7.4.

7.2 Data Selection

The accuracy of modal analysis relies on the quality of the utilised datasets. To extract modal parameters correctly, we first need to select some datasets of high-quality. In this section, we illustrate the data selection task with a three-step procedure: sensor selection, traffic event detection and free vibration periods extraction.

7.2.1 Sensor Selection

In the sensor network, both strain and vibration signals respond to traffic events. The left two pictures in Fig. 7.1 illustrate one truck event in the time domain, for either sensor type. From these pictures, it is easy to see that the truck event in the strain signal is represented as a peak, which occurs when the vehicle is actually on the measured span, and disappears rapidly when the vehicle passes. The truck event in the vibration signal produces oscillations, which will last for a long period after the truck has passed, if it is not disturbed by subsequent vehicles. Based on this observation, it is reasonable to select the strain signal to recognise traffic events [54, 55]. To monitor and evaluate the health of the bridge, spectral analysis is one of the widely used methods [3]. The right two pictures of Fig. 7.1 (right) illustrate the spectrum of both the strain and vibration signal, which are produced by a Discrete Fourier Transform (DFT). It is clear that the spectrum of the vibration signal is more informative than that of the strain signal. So both the strain and vibration signals are employed in our experiments: first, we use the strain signal to detect traffic events, then conduct spectral analysis on the corresponding vibration signal.

Since there are 91 strain sensors and 34 vibration sensors in our sensor network, which sensors are suitable? One simple standard of choosing strain sensors is that they can clearly represent traffic events. That is to say, the peak of the selected strain signal should have a strong amplitude. We choose one truck event on each side of the bridge as excitation, look into the response of all of the strain sensors, and finally choose one sensor on each side of the bridge as target. The selection

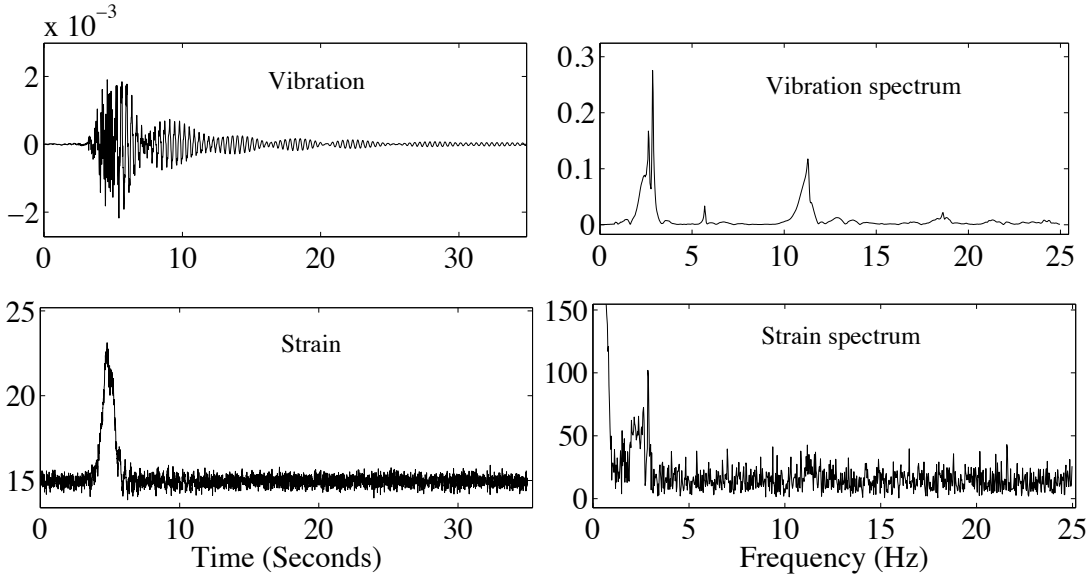

Figure 7.1: The strain and vibration signal in the time and frequency domain - The top left picture is a vibration signal; the top right picture is the spectrum of the vibration signal; the bottom left picture is a strain signal; the bottom right picture is the spectrum of the strain signal.

of the vibration sensors is based on the strain and vibration correlation matrix as mentioned in Chapter 4. We choose the vibration sensors that have strong correlations with the selected strain sensors as the target vibration sensors.

7.2.2 Traffic Event Detection

Following the procedure mentioned above, we obtain a pair of strain and vibration sensors installed on each side of the bridge, reducing the sensor candidates from 125 $(91 + 34)$ to 4 $(2 + 2)$. However, extracting traffic events from the reduced sensor signals is still a challenging task, because the bridge is a complex system, which responds to various inputs. As mentioned in previous chapters, some inputs play disturbing roles. What's more, even a useful input, such as a car or a truck, on one side of the bridge could not be detected or mis-detected in the signal collected with sensors on another side of the bridge. To prepare some high-quality datasets for modal analysis, we propose a procedure to extract traffic events. We

prefer that each target dataset just contains a single truck event. The procedure is listed as follows:

- **Step 1**: Find baseline. The baseline of the strain signal is influenced a lot by temperature and traffic jams. To measure the amplitudes of peaks correctly, we must find the baseline first.
- **Step 2**: Remove baseline. Baseline removal is quite straightforward. It is obtained by subtracting the baseline from the original strain signal.
- **Step 3**: Find peaks. Using the zero-crossing and the local maximum methods, we detect a number of peaks, with amplitude, duration and area under the peak as peak descriptive features.
- **Step 4**: Label peaks. Based on the video stream, we hand-label each peak as either of noise, car on lane 1, truck on lane 1, car on lane 4 or truck on lane 4. This will be our supervised training data (lane 1 and lane 4 stand for two different traffic directions).
- **Step 5**: Classify peaks. Based on the obtained peak features and labels, we try to find the boundaries between each class, by means of classification techniques from the Data Mining field [123].
- **Step 6**: Extract truck events. One whole traffic event is composed of the traffic-free period before the traffic peak, the actual peak and the traffic-free period after the traffic peak. We should look into the traffic events on both lanes to catch all traffic.

We choose a dataset of one hour at 3:00 AM (100 Hz) as the training dataset. The traffic during this time is not too heavy, and most of time there is just a single lane on either side in use. The baseline correction method mentioned in Step 1 and Step 2 is the most-crossing method, which was proposed in Chapter 5. After removing baselines from the selected strain signals, we continue to process the obtained signals with zero-crossing and local maximum methods in Step 3, achieving a number of peaks, with amplitude, duration and area under the peak as peak features. In Step 4, we hand-label these detected peaks according to the video taken during this period on the bridge. All the peaks are given one of five

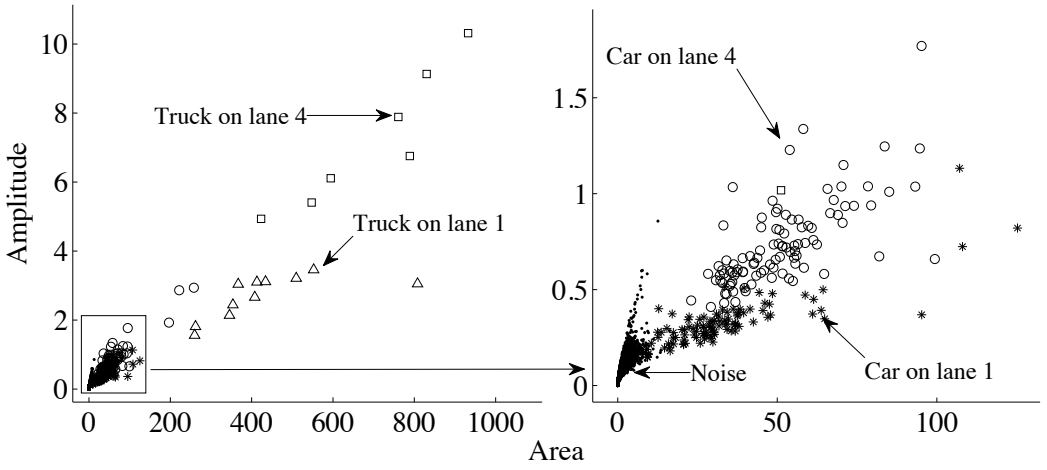

Figure 7.2: Peak classification - All peak labels within one hour (left), based on two peak features: area and amplitude; the right picture shows the details in the bottom left corner of the left picture, shown as the rectangular box in the left picture.

categories: noise, car on lane 1, truck on lane 1, car on lane 4 and truck on lane 4. The scatter plot based on area and amplitude of the strain peaks on lane 4 is illustrated as Fig. 7.2.

From the labels in Fig. 7.2, we can see that truck events on either lane are easy to distinguish, but the boundaries between car events on opposite lanes and the boundaries between the noise and car events on opposite lanes are blurry. When cars on an opposite lane are not heavy enough, they are easily mistaken as noise in the strain signal of the current lane. But the vibration sensor is much more sensitive to traffic events than the strain sensor, which can catch a small car event on another lane. To detect the complete free vibration period according to the strain signal, we must make the boundaries as clear as possible.

We processed our labeled peaks with Weka [123], a powerful Data Mining tool. A decision tree (C4.5) was applied to the labeled dataset (peak features), which takes *area* and *amplitude* on lane 4 as attributes.

The labeled dataset (derived from the training dataset) is composed of 7,169 instances, of which 7,137 (99.55%) instances are correctly classified. The confusion

Table 7.1: The confusion matrix.

truck 4	truck 1	car 4	car 1	noise	
7	0	1	0	0	truck 4
1	10	0	0	0	truck 1
0	2	97	4	0	car 4
0	0	3	98	4	car 1
0	0	2	15	6,925	noise

matrix is shown as Table 7.1.

The result, with a few minor mistakes, is already quite good, but can be further improved by combining the traffic events on the lane of opposite traffic direction (lane 1). We applied this model to a bigger dataset (the test dataset), which was obtained by selecting one hour per day at 3:00 AM for 45 days. We succeeded to catch 17,220 traffic events (of which 852 are trucks) on lane 1 and 13,064 traffic events on lane 4 (of which 768 are trucks). Truck events are usually featured with high amplitudes and peak areas, which indicate that the bridge is well excited, so we prefer to utilise the datasets caused by them for modal analysis. However, not every truck event is interesting to us. We expect that the vibrations caused by one truck should be long enough for modal analysis, without being disturbed by other traffic events. To meet this expectation, we explore truck events with long free vibration periods.

7.2.3 Free Vibration Periods Extraction

In this section, we focus on extracting the free vibration periods of traffic events from our structural health monitoring system, which is a critical step to analyse the modal parameters of the bridge. The free vibration period means the period after a vehicle has passed, and before a next vehicle appears on the bridge. The reason for choosing this period is that the bridge is put in motion by the vehicle, but the actual weight does not actually influence the frequency of vibration after the vehicle has disappeared, nor do any other vehicles.

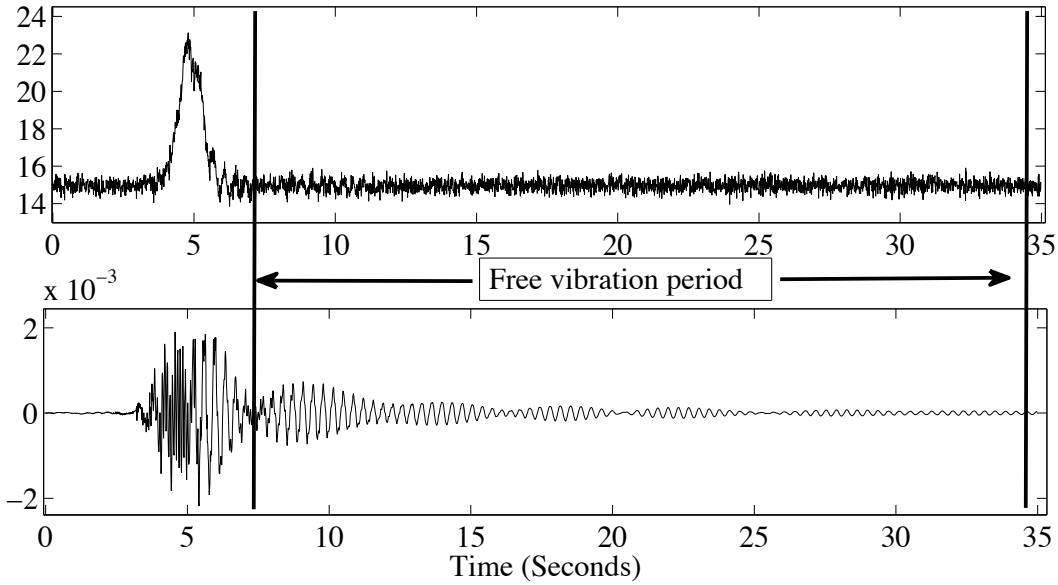

Figure 7.3: Free vibration period - The period between two vertical lines is referred to as the free vibration period, which starts after one truck just passes the bridge and ends before another vehicle appears on the bridge.

Shown as Fig. 7.3, free vibration fragments are extracted from the vibration signal, which corresponds to the traffic-free period directly after a truck-related peak in the strain signal. Details of how such periods can be identified in the data can be found in [53]. Following this procedure, we obtain a number of free vibration periods, which will be used for modal analysis in the next section.

7.3 Modal Parameter Extraction

In this section, we employ two methods to extract modal parameters: the PP method and the SSI method. The dataset employed for modal analysis is obtained by extracting truck events, with at least 20 seconds of free vibration period, from the testing dataset in the previous section, which is composed of 72 truck events on lane 1 and 77 truck events on lane 4.

7.3.1 The Peak-Picking Method

The PP method is a widely-used method to estimate modal parameters from output-only measurements [124, 125], in which the natural frequencies are simply obtained by choosing the peaks on the graphs of the power spectral densities (PSDs) [126, 127, 128]. The PSDs are basically obtained by converting the measurements to the frequency domain with the DFT [114, 127].

To obtain all the possible modes of the bridge, we apply the PP method to the free vibration periods of the selected dataset. After normalising the 149 spectra, we obtain a graph, shown as Fig. 7.4. From this figure, we can easily detect several interesting modes. Table 7.2 provides statistics of these modes. The approximate location of each mode is defined according to Fig. 7.4, and the occurrence of a mode is counted if there is at least one peak, whose amplitude is bigger than the average amplitude. The third column in Table 7.2 is calculated by counting what fraction of the 149 spectra actually show a peak at the specified location in the spectrum.

Table 7.2: Statistics of modes.

Mode	Frequency (Hz)	Occurrence
1	0.73 – 0.93	71.8%
2	2.59 – 2.78	100%
3	2.79 – 2.98	100%
4	5.50 – 5.77	97.3%
5	11.00 – 11.50	98.7%
6	14.82 – 15.55	12.1%
7	16.30 – 16.90	10.7%
8	18.30 – 18.80	48.3%

As illustrated below, the amplitude indicates the strength of each mode. Mode 2 and mode 3 are the principal modes of the bridge, which occur in every event. Mode 4 and mode 5 are also important modes, which have a strong amplitude and happen in most events. Mode 1 and mode 8 have moderate occurrence, but

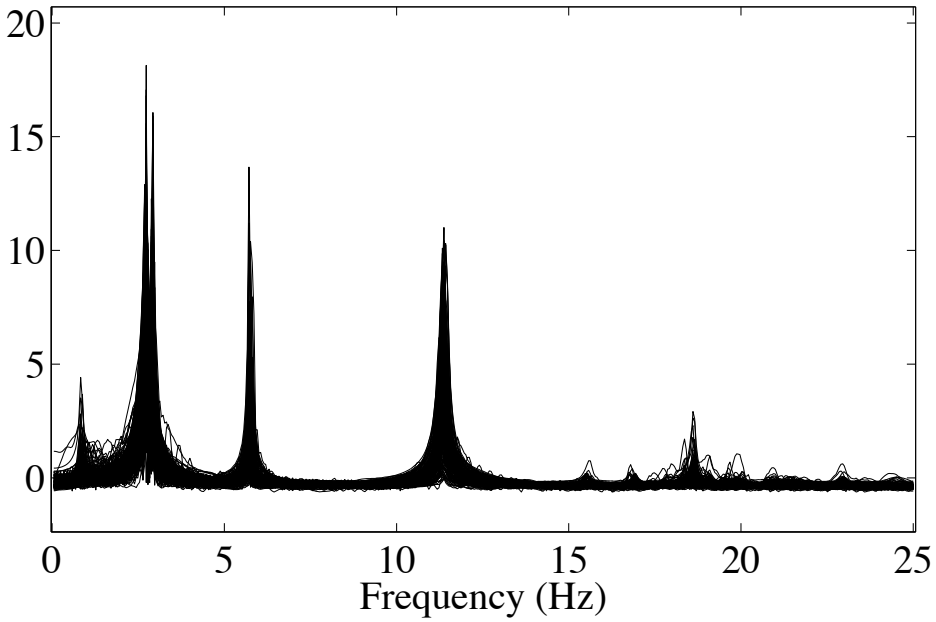

Figure 7.4: The vibration modes of the bridge - The spectra are derived from free vibration periods of 149 selected truck events.

their amplitudes are relatively weak. Mode 6 and mode 7 are so weak that they can be ignored in most cases.

The PP method is simple, and needs no model to fit to the measurements [113], so its identification is fast [128], and can be used on-site to verify the quality of the measurements [8, 114]. However, the PP method relies heavily on the frequency resolution [129]. When the assumptions of well separated modes and low damping are violated, the PP method often results in inaccurate and erroneous modes [8, 127]. To estimate modal parameters more accurately, we need employ more advanced methods.

7.3.2 The SSI Method

Compared with the PP method, the SSI method is a more advanced method for modal analysis. which is based on the stochastic state space model. To improve the result of the SSI method, the stabilization diagram is introduced to

distinguish physical modes from spurious modes. In this section, we will illustrate this method with some selected datasets, and present modal parameters derived from these datasets.

7.3.2.1 Stochastic State Space Model

The SSI method is especially suited for operational modal parameter identification when only output measurements are available. In the text below, the core steps of the SSI method are discussed. A detailed explanation is beyond the scope of this chapter and can be found in the references [11, 118]. The dynamic system of a vibration structure can be modelled by the following discrete-time state space model:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \tag{7.1}$$

where y_k is the measurement at discrete time instance k, x_k is the state vector, u_k is the input vector, A is the discrete state matrix, B is the discrete input (system control influence coefficient) matrix, C is a real output influence coefficient matrix and D is the out control influence coefficient matrix; w_k is the process noise due to disturbances and modelling inaccuracies, and v_k is the measurement noise due to sensor inaccuracy. Here, the process noise w_k and measurement noise v_k are assumed to be white noise vectors, with the following covariance matrices:

$$E\left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix}\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta_{pq} \tag{7.2}$$

where $E[\cdots]$ is the mathematical expectation operator, δ_{pq} is the Kronecker delta, and Q, R, S are process and measurement noise auto/cross-covariance matrices. The sequences w_k and v_k are assumed statistically independent of each other. In practice, the input vector u_k is not measured, and only the response of a structure is measured, so it is impossible to distinguish u_k from the process noise w_k and the measurement noise v_k. By implicitly modelling u_k with the noise terms w_k, v_k, the discrete-time stochastic state space model can be represented as:

$$\begin{aligned} x_{k+1} &= Ax_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{7.3}$$

Here the noise terms w_k, v_k still follow the white noise assumption. One drawback of the stochastic state space model is that if the input contains some dominant frequency components except for the white noise, these frequency components will appear as poles of the state matrix A.

Estimation of state matrices Based on Eq. 7.3, there are several techniques that can be used for system identification through ambient measurements. The technique employed in this chapter is called data-driven stochastic subspace identification. All the output measurements are organized in a block Hankel matrix $H \in R^{2li \times j}$ with $2i$ block rows and j columns (each data point in the measurement is viewed as one column). Every block consists of l rows. For statistical reasons, it is assumed that $j \to \infty$. The block Hankel matrix H can be represented as:

$$H = \frac{1}{\sqrt{j}} \left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \vdots & \vdots \\ y_{i-1} & y_i & \cdots & y_{i+j-2} \\ \hline y_i & y_{i+1} & \cdots & y_{i+j-1} \\ y_{i+1} & y_{i+2} & \cdots & y_{i+j} \\ \vdots & \vdots & \vdots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{2i+j-2} \end{array}\right] = \left[\frac{Y_{0|i-1}}{Y_{i|2i-1}}\right] = \left[\frac{Y_p}{Y_f}\right] \tag{7.4}$$

where $\frac{1}{\sqrt{j}}$ is the scaled factor, Y_p stands for the past output matrix, Y_f represents the future output matrix. The key element of the data-driven SSI is the projection of the row space of the future outputs into the row space of the past outputs. This projection can be defined as:

$$P_i = \frac{Y_f}{Y_p} = Y_f Y_p^T (Y_p Y_p^T)^\dagger Y_p \tag{7.5}$$

where $(\cdot)^\dagger$ represents the pseudo-inverse of a matrix.
The projection P_i can be factorised as:

$$P_i = \Gamma_i X_0 = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{i-1} \end{bmatrix} \begin{bmatrix} x_0 & x_1 & x_2 & \dots & x_{i-1} \end{bmatrix} \tag{7.6}$$

where Γ_i is the observability matrix, and X_0 represents the Kalman filter state sequence at time lag zero. With the help of the singular value decomposition (SVD), the projection P_i can be further decomposed as:

$$Y_i = USV^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 S_1 V_1^T \tag{7.7}$$

The order n of the system can be determined by neglecting the smaller singular values in S_2, and the observability matrix Γ_i and Kalman filter state sequence X_0 can be estimated by:

$$\begin{aligned} \hat{\Gamma}_i &= U_1 S_1^{1/2} \\ \hat{X}_0 &= S_1^{1/2} V_1^T \end{aligned} \tag{7.8}$$

The system parameter matrices A and C can be obtained based on the estimated observability matrix $\hat{\Gamma}_i$:

$$\begin{aligned} A &= \hat{\Gamma}_{i1}^{\dagger} \hat{\Gamma}_{i2} \\ C &= \hat{\Gamma}_{li} \end{aligned} \tag{7.9}$$

where $\hat{\Gamma}_{i1}$ denotes $\hat{\Gamma}_i$ without the last l rows, $\hat{\Gamma}_{i2}$ represents $\hat{\Gamma}_i$ without the first l rows, and $\hat{\Gamma}_{li}$ stands for the first l rows of $\hat{\Gamma}_i$.

Modal parameters The modal parameters are derived from the system parameter matrices A and C:

$$\begin{aligned} A &= \Psi \left[\mu_i\right] \Psi^{-1} \\ f_i &= \frac{|\lambda_i|}{2\pi} \\ \xi_i &= \frac{Re(\lambda_i)}{|\lambda_i|} \\ \Phi &= C\Psi \end{aligned} \tag{7.10}$$

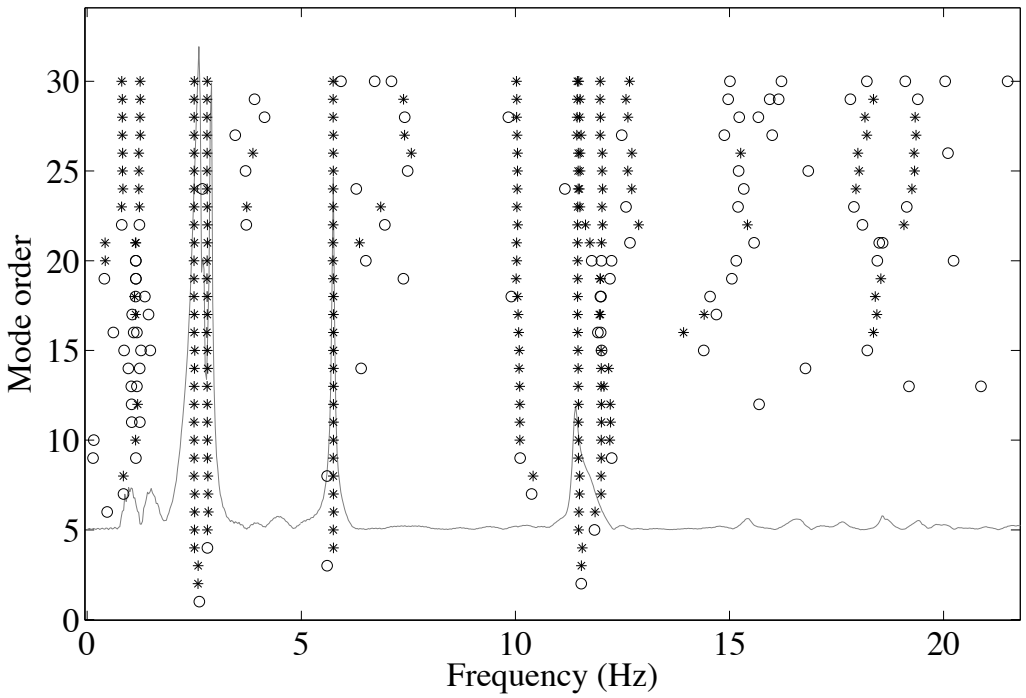

Figure 7.5: The stabilization diagram - The stabilization diagram is obtained by setting the stable criteria as 5% for natural frequencies and MAC, in which the stars represent stable physical poles, and the circles represent the spurious poles; the background spectrum is derived from the PP method.

where Ψ is the matrix of eigenvectors, μ_i are the discrete time poles, $\lambda_i = \frac{\ln(\mu_i)}{\Delta T}$ are the continuous poles, f_i are the natural frequencies, ξ_i are the damping ratios, and Φ is the mode shape matrix.

7.3.2.2 The Stabilization Diagram

It is assumed that all the input forces of the SSI procedure are white noise and the length of the recording is infinite. In practice, the measurements used for SSI are limited, and usually contain some other dominant frequency components. As shown in Eq. 7.5, the order of the system is obtained by ignoring the smaller singular values, which is usually higher than the actual system order. All of these factors may introduce spurious, numerical poles to the system. To address

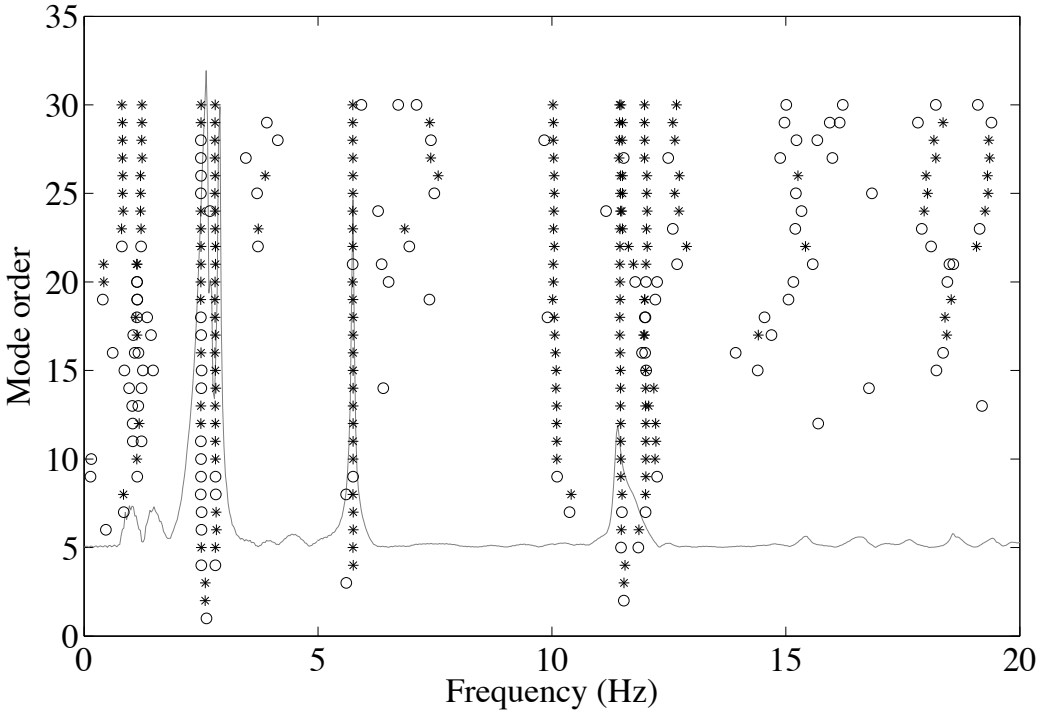

Figure 7.6: The stabilization diagram - The stabilization diagram is obtained by setting the stable criteria as 5% for natural frequencies and MAC, and 50% for damping ratios, in which the stars represent stable physical poles, and the circles represent the spurious poles; the background spectrum is derived from the PP method.

the physical and the spurious, numerical poles, the stability diagram [130] is introduced. The basic idea of the stabilization diagram is to iterate the system order n from a lower value to the maximum order. It is assumed that the lowest order is unstable, so the modal parameters of the current order are compared with those of one order lower. If the differences are under user-defined limits, then this order is considered to be a stable order. The limits are defined as:

$$\begin{aligned} \left| \frac{f_k - f_{k-1}}{f_k} \right| &< lim_f \\ \left| \frac{\xi_k - \xi_{k-1}}{\xi_k} \right| &< lim_\xi \\ (1 - MAC(k, k-1)) &< lim_{MAC} \end{aligned} \tag{7.11}$$

where $k > 1$ denotes the modal order, f is the frequency, ξ is the damping ratio, lim_f is the frequency limit, lim_ξ is the limit for the damping, lim_{MAC} is the limit for the modal assurance criterion (MAC). The MAC value ranges from 0 to 1, where 0 means that there is no similarity between the compared mode shapes, and 1 means these two mode shapes are consistent. The MAC can be defined as:

$$MAC(k, k-1) = \frac{|\Phi_k^H \Phi_{k-1}|^2}{(\Phi_k^H \Phi_k)(\Phi_{k-1}^H \Phi_{k-1})} \tag{7.12}$$

7.3.2.3 Experimental Settings on InfraWatch Dataset

To employ the SSI method for modal analysis, we select a dataset derived from 12 vibration sensors in the sensor network. The sensors are located at three cross-section of four different girders, which are equally spaced in both longitudinal and transversal directions.

The first activity to extract modal parameters from measurements with the SSI method, is creating a Hankel matrix with 24 block rows (30 rows per block), and 3,377 columns. One key parameter for SSI is the order of the system. Because of operational noise, it is impossible to obtain the system order precisely from the singular value of the Hankel matrix projection. If the system order is estimated with a lower value, some physical poles will be missed. Otherwise, spurious numerical poles may appear. The stabilization diagram is useful to separate physical poles from spurious numerical poles. In the stabilization diagram, the system order is tested from a minimal order 2 to a maximum order of 30. The physical poles are represented as stars and spurious poles are represented as circles. We assume the initial status of each pole is unstable, e.g, the two poles of mode order 2 are represented as circles.

The stable criteria are set as 5% for natural frequencies, and 5% for MAC. In practice, the damping ratios are difficult to be estimated accurately [131]. Shown in Fig. 7.5, the stabilization diagram is obtained by just employing natural frequencies and MAC stable criteria. The stabilization diagram in Fig. 7.6 is obtained

Table 7.3: Modal Parameters.

Mode	Mode shape	Frequency (SSI)	Frequency (PP)	Relative error
1	Bending	2.51 Hz	2.61 Hz	4.0%
2	Torsional	2.81 Hz	2.90 Hz	3.2%
3	Bending & Torsional	5.74 Hz	5.75 Hz	0.2%
4	Bending	10.09 Hz	–	–
5	Torsional	11.47 Hz	11.41 Hz	−0.5%
6	Bending & Torsional	11.99 Hz	–	–

by setting damping ratio criterion to a higher value (50%). Even when the stable criterion for damping ratios is much higher than that of natural frequencies and MAC, there is still one mode (around 2.51 Hz), that can be clearly observed from the background spectrum (derived from the PP method to one of the selected 12 vibration signals), is mistaken as a spurious mode. In this experiment, we exclude the damping ratio criterion from the stable criteria.

7.3.2.4 The Results of the SSI Method

The results of SSI method includes: natural frequencies, mode shapes and damping shapes. As illustrated in the stabilization diagram Fig. 7.6, damping ratios obtained with the SSI method are unreliable, so we won't further discuss them in this chapter. The results of the first two parameters are listed as follows:

Natural frequency We make a comparison between the modes obtained with the SSI method and the PP method, shown as Table 7.3. From both the table and the stabilization diagrams, we notice that there is a high coherence between the peaks in the spectrum and physical poles obtained with the SSI method. However, with the SSI method, we can obtain more poles, e.g, the modes around 10 Hz and 12 Hz, which are absent in the PP method, and there are some small peaks in the spectrum that are identified as spurious modes, e.g, the modes between 0.7 Hz and 1 Hz.

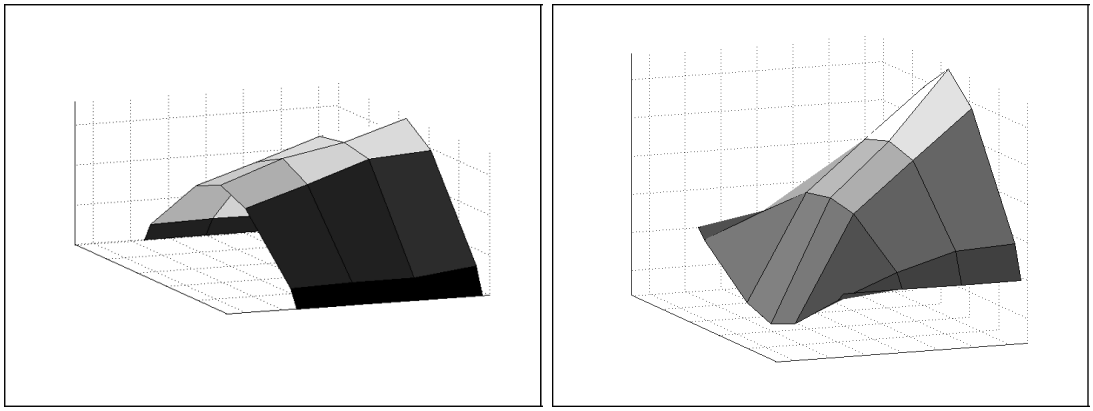

Figure 7.7: The first and the second mode shapes - The left picture shows the mode shape of the first mode, which is the first bending mode. The right picture shows the mode shape of the second mode, which is the first torsional mode.

Compared with the modes in Table 7.2, Table 7.3 has fewer modes. This is because not all the modes of a bridge can be excited by a single traffic event at the same time. The modes in the former table are derived from 149 truck events, and the results in the latter table are derived from a single truck event.

Mode Shapes Fig. 7.7 to Fig. 7.9 show the first six mode shapes derived from the SSI method. Because the sensor network just covers half of the bridge span, the mode shapes of the unmeasured half span are modelled using the existing measurements and structural knowledge.

7.4 The Influence of Environmental Factors

As mentioned in Section 7.1, modal parameters are not only sensitive to structural damage and degradation, but also to varying operational and environmental loadings, which include traffic, wind, humidity and temperature. In this section, we will look into the influence of temperature and vehicle mass on one of the most important modal parameters: natural frequencies.

If we simply take the bridge as a Euler-Bernoulli beam, the vehicle and bridge

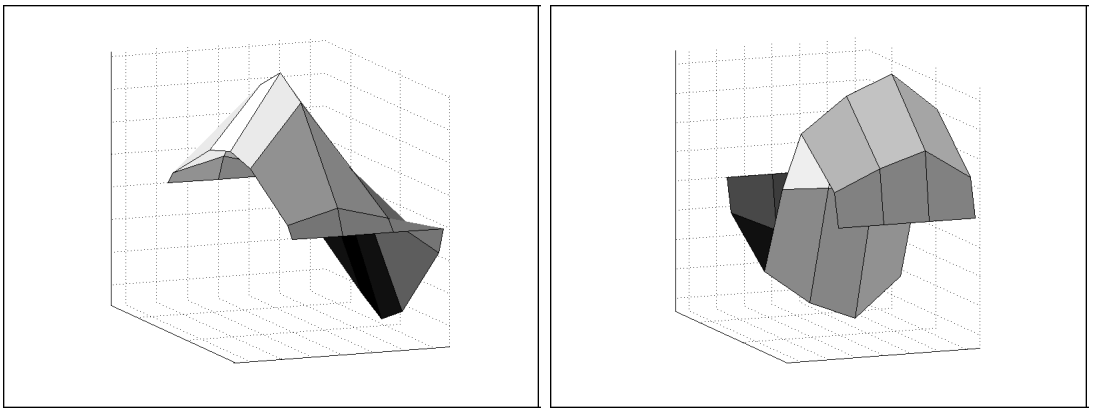

Figure 7.8: The third and the fourth mode shapes - The left picture shows the mode shape of the third mode, which is the first mixed mode, derived from the combination of bending and torsional behaviour. The right picture shows the mode shape of the fourth mode, which is the second bending mode.

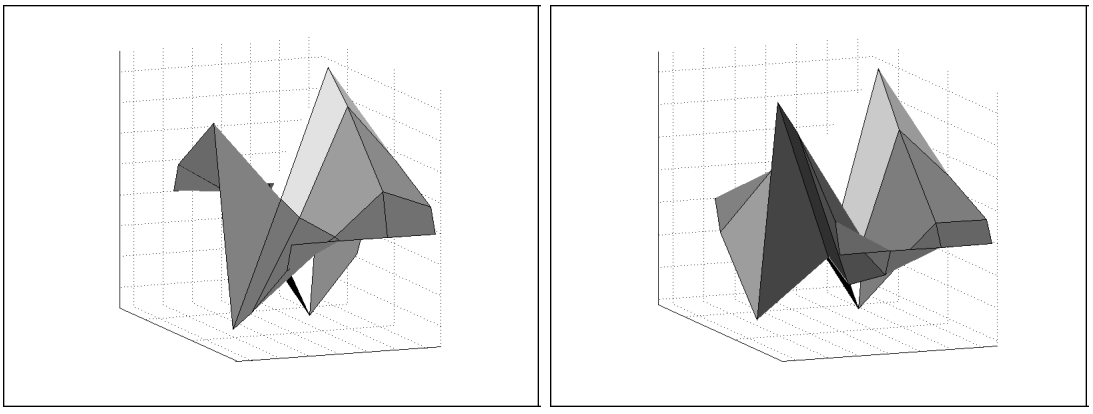

Figure 7.9: The fifth and the sixth mode shapes - The left picture shows the mode shape of the fifth mode, which is the second torsional mode. The right picture shows the mode shape of the sixth mode, which is the second mixed mode, composed of bending and torsional behaviour.

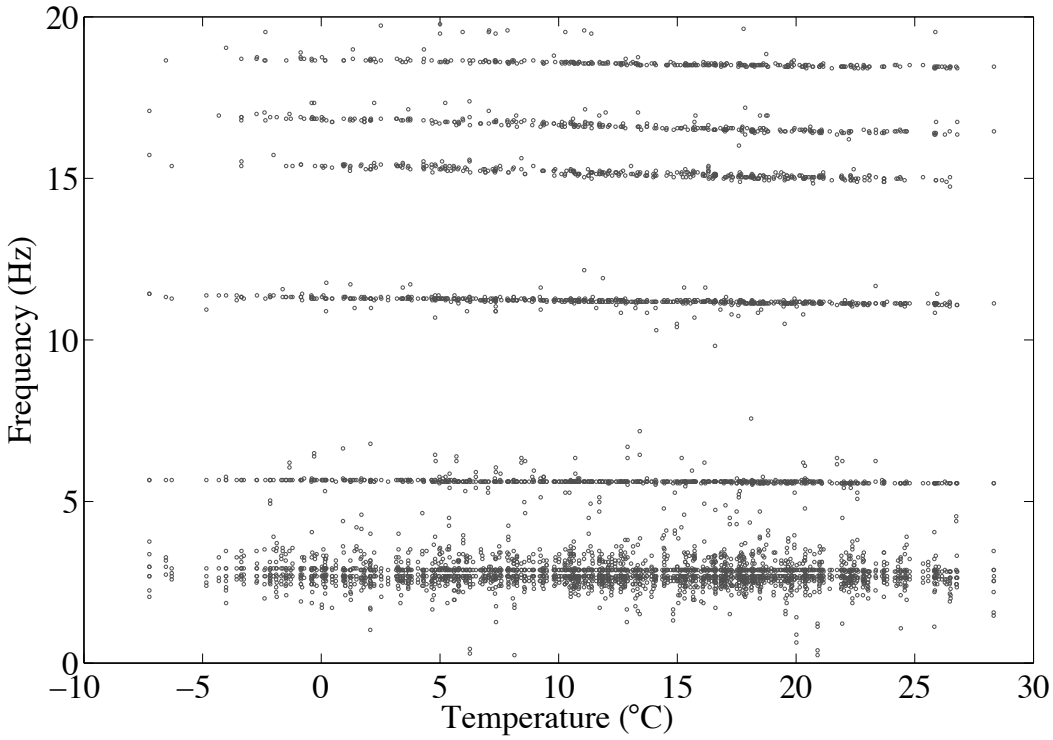

Figure 7.10: Natural frequencies and temperature - This picture illustrates a scatter plot between natural frequencies and temperature, in which natural frequencies derived from free vibration periods of 983 truck events; the data covers a period of more than two years, with a temperature range of 40 °C.

interaction system [132] can be modelled as a damped parallel spring mass system, and each natural frequency f_n of the system can be represented as follows:

$$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{7.13}$$

where k represents for the stiffness of the bridge, m represents the total mass on the bridge. According to Peeter et al. [50], the temperature may have an impact on the boundary conditions and the Young's modulus of the material of which the structure consists. The variation of temperature will cause changes in the stiffness k of the bridge, and vehicles on the bridge will add to the mass m. All these changes can be detected from the measurements collected from the bridge.

The dataset employed in this section covers a big time scale, ranging from January, 2009 to September, 2011. We select 5 minutes (100 Hz) at 1:00 AM from each day, and extract truck events with long free vibration periods. Following the procedure mentioned in Section 7.2, we obtain 983 truck events with free vibration periods longer than 1,024 data points (10.24 seconds). The method employed for modal analysis is the PP method, because of its simplicity and high coherence with the advanced SSI method.

7.4.1 The Influence of Temperature

To look into the influence of temperature on natural frequencies, we employ the free vibration periods of the selected 983 truck events. During the free vibration period, the bridge has already been excited by the vehicle, but the weight of the truck no longer influences the total bridge mass, so it helps to separate the influence of temperature from the influences of other factors.

We choose a free vibration period of 2,048 data points (slightly over 20 seconds long) from each truck event collected with a vibration sensor, and apply the PP method for modal analysis. The temperature of the bridge is estimated by the average value of one of the temperature sensors during the free vibration period. Shown as Fig. 7.10, the temperature of our selected truck events ranges from -8 °C to 32 °C, and there are clearly several modes within the first 20 Hz. By zooming in the scatter plot, we find out that the mode between 2 and 5 Hz is actually composed of two modes, so there are 7 modes visible in the scatter plot. Generally speaking, the natural frequencies decrease with increasing temperature, but the influence of temperature on different modes is not equal. To look into the influence of temperature in detail, we fit each mode separately with a linear regression model, shown as Fig. 7.12 to Fig. 7.18[1]. The linear model can be represented as:

$$f = a \cdot t + b \tag{7.14}$$

[1]Note that the detailed plots of the modes show a discrete behavior along the Y-axis. This is caused by the resolution of the spectrum resulting from the FFT operation. With an input consisting of 2,048 measurements, the distance between frequency bins is 0.0488 Hz.

Table 7.4: The coefficients of linear regression models between temperature and natural frequencies.

Mode	a	b	norm	r
1	$-5.551 \cdot 10^{-4}$	2.678	$9.188 \cdot 10^{-4}$	0.188
2	$-1.731 \cdot 10^{-3}$	2.901	$8.550 \cdot 10^{-4}$	0.501
3	$-2.852 \cdot 10^{-3}$	5.651	$8.069 \cdot 10^{-4}$	0.736
4	$-7.587 \cdot 10^{-3}$	11.305	$1.420 \cdot 10^{-3}$	0.859
5	$-1.741 \cdot 10^{-2}$	15.387	$3.692 \cdot 10^{-3}$	0.868
6	$-1.732 \cdot 10^{-2}$	16.838	$3.722 \cdot 10^{-3}$	0.907
7	$-9.277 \cdot 10^{-3}$	18.676	$2.201 \cdot 10^{-3}$	0.868

in which a and b are coefficients, t stands for input (temperature), and f stands for the predicted frequency. The goodness of fit is measured by the *norm of residuals* as well as the *correlation coefficient* r. We give the definition of *norm of residuals*, which is:

$$norm(d, 2) = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{7.15}$$

in which d_i stands for the difference between the i^{th} predicted value and the i^{th} actual value. The coefficients of each linear regression model are listed in Table 7.4, and a scatter plot between coefficients a and b is illustrated in Fig. 7.11.

From these linear regression models, we can draw the following conclusions:

- All the natural frequencies decrease with increasing temperature.
- Different modes have different sensitivity to temperature.
- High-frequency modes are more sensitive to temperature than low-frequency mode, except for the last two modes.

7.4.2 The Influence of Traffic Events

For short periods, the stiffness of the bridge can be assumed constant, and the only factor influencing the natural frequencies is the mass of traffic. We assume that when a truck is on the bridge, the mass of the bridge increases, and the

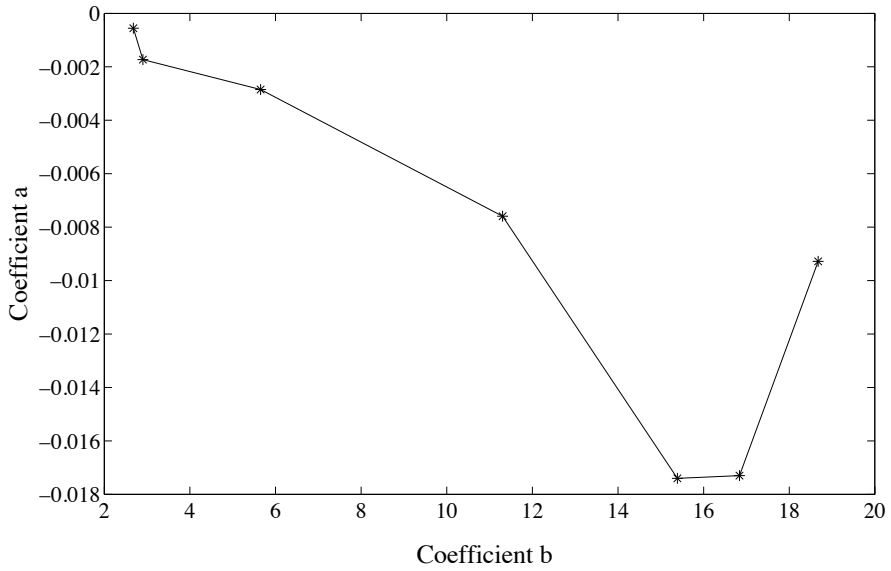

Figure 7.11: Coefficients of linear models - The Y-axis stands for the first coefficient (a) of the linear model; the X-axis stands for the second coefficient (b) of the linear model

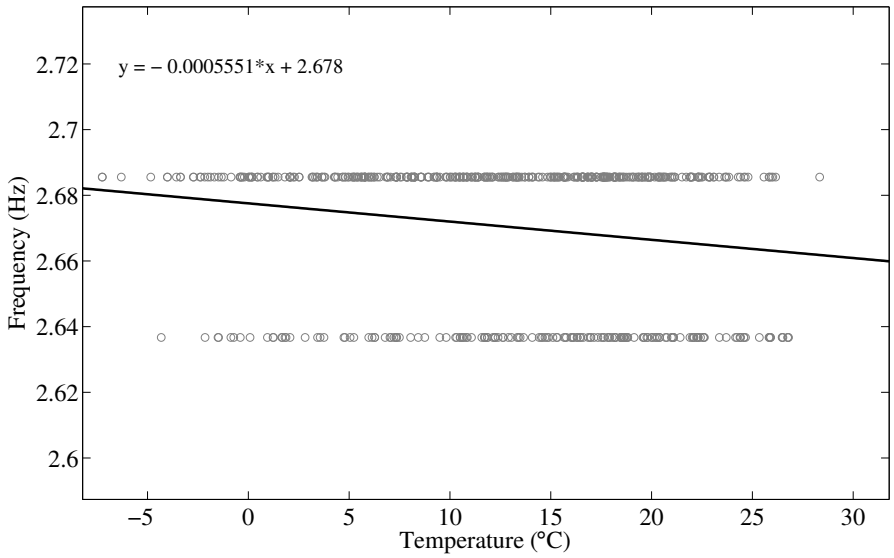

Figure 7.12: The linear modal between the first mode and temperature - The coefficients indicate that the first mode is practically insensitive to temperature.

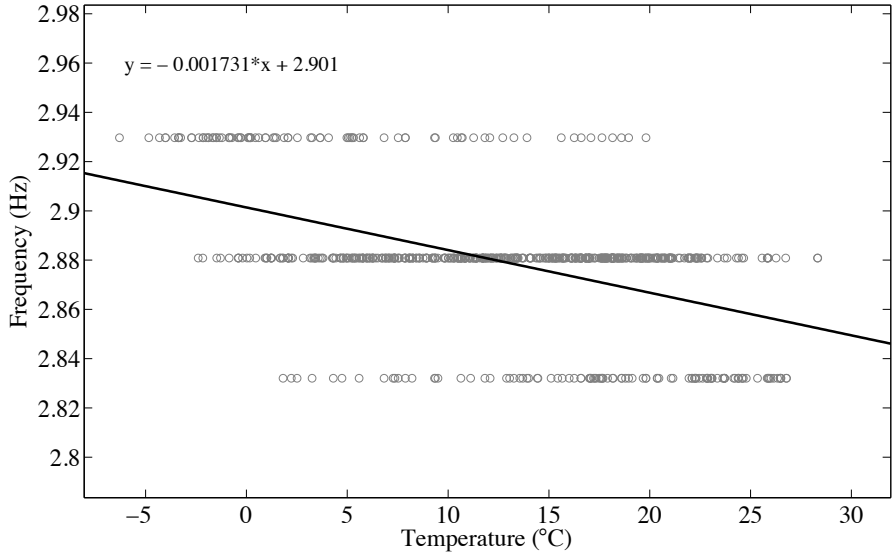

Figure 7.13: The linear modal between the second mode and temperature - The coefficients indicate that the second mode is more sensitive to the temperature than the first mode.

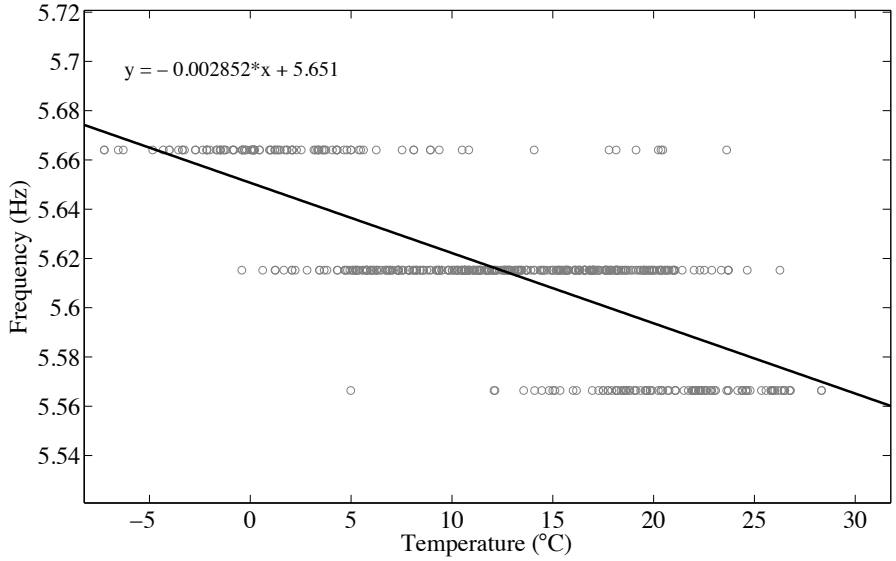

Figure 7.14: The linear modal between the third mode and temperature - The coefficients indicate that the third mode is more sensitive to the temperature than the first two modes.

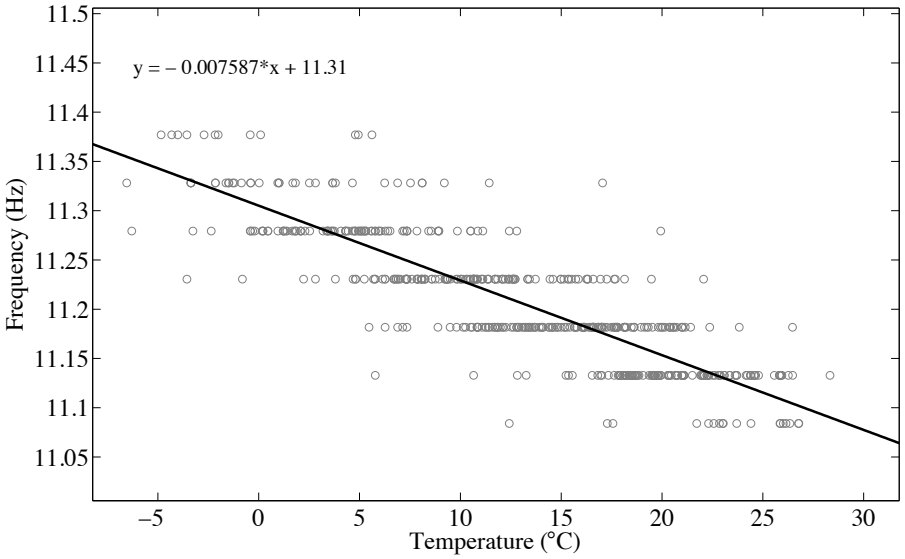

Figure 7.15: The linear modal between the fourth mode and temperature - The coefficients indicate that the fourth mode is more sensitive to the temperature than the first three modes.

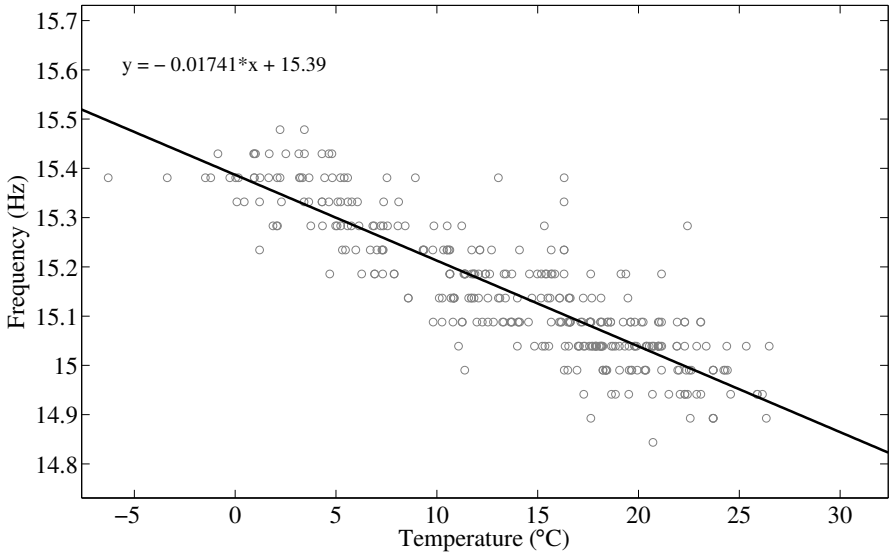

Figure 7.16: The linear modal between the fifth mode and temperature - The coefficients indicate that the fifth mode is more sensitive to the temperature than the first four modes.

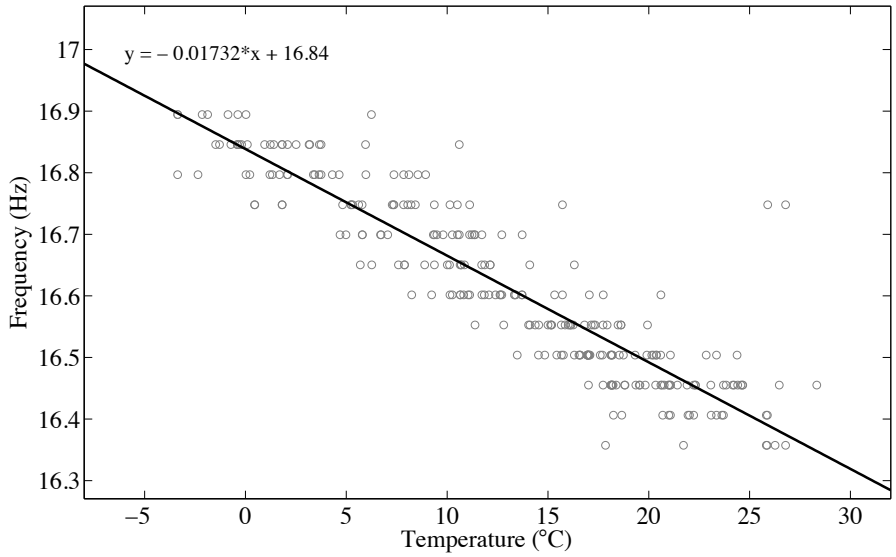

Figure 7.17: The linear modal between the sixth mode and temperature - The coefficients indicate that the sixth mode is less sensitive to the temperature than the fifth mode, but more sensitive than the first four modes.

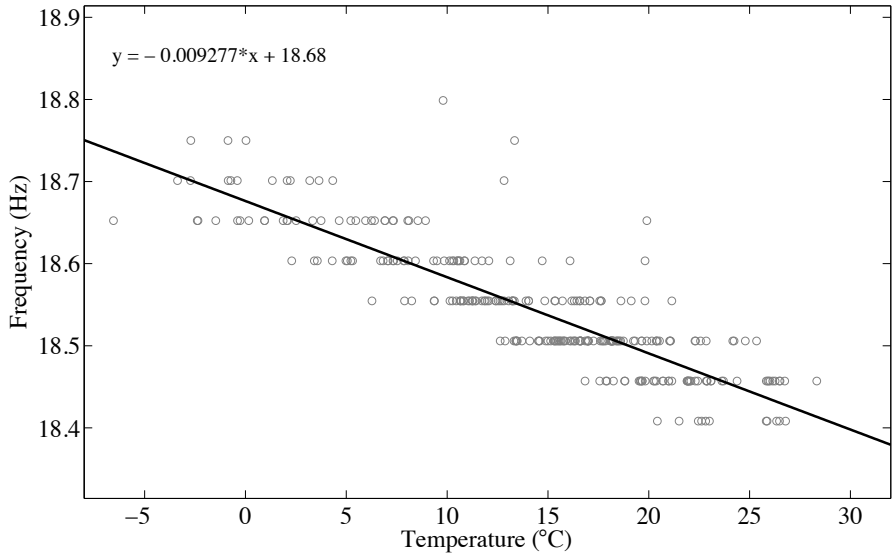

Figure 7.18: The linear modal between the seventh mode and temperature - The coefficients indicate that the seventh mode is more sensitive to the temperature than the first four modes, but less sensitive to the fifth and sixth modes.

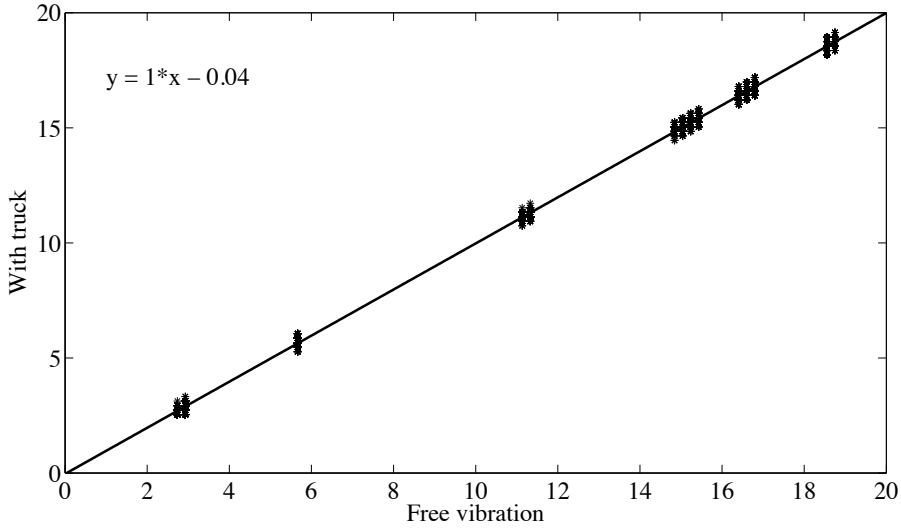

Figure 7.19: The influence of mass on all modes - This picture illustrates a scatter plot of natural frequencies obtained with trucks on the bridge and during free vibration periods.

natural frequencies should decrease. To verify the assumption, we apply DFT to the periods when trucks are on the bridge and the periods of free vibration respectively. As illustrated in Fig. 7.19, generally speaking, natural frequencies obtained with vehicles on the bridge are less than those obtained from free vibration periods. To look into the influence of traffic mass on each mode, we make a statistical analysis, shown as Table 7.5. In this table, $f_{free=mass}$ indicates that the frequency obtained during free vibration period is equal to the frequency obtained with a vehicle on the bridge; $f_{free>mass}$ indicates that the former frequency is bigger than the latter frequency; $f_{free<mass}$ indicates that the former frequency is smaller than the latter frequency.

The statistical results in Table 7.5 meet well with the linear function illustrated in Fig. 7.19, while the table reveals more details:

- In all the modes, most frequencies obtained with vehicles on the bridge are equal to frequencies obtained with free vibration periods.
- The numbers in columns 3 and 4 indicate that the mass of vehicles influences

Table 7.5: The statistical analysis of the influence of mass on each mode.

Modes	$f_{free=mass}$	$f_{free>mass}$	$f_{free<mass}$
1	106	58	34
2	407	327	30
3	467	110	102
4	571	261	40
5	433	190	155
6	300	196	120
7	306	128	104

the frequencies of all the modes, some of which are positive ($f_{free>mass}$), and some of which are negative ($f_{free<mass}$), and the positive numbers are bigger than the negative numbers.

- The positive numbers in mode 2 and 4 are bigger than the negative numbers, which indicates that these two modes are more sensitive to traffic mass. Referring to Table 7.3, we further find that both of these two modes are torsional modes.

Chapter 8

Conclusion

8.1 Conclusion

In this thesis, we have applied data mining techniques to the SHM domain, more precisely, to a highway bridge under normal in-service conditions. Long-term changes in the system can be analysed through a bridge's dynamic response, variables of which are known as modal parameters: natural frequencies, damping ratios and mode shapes. In reality, modal parameters are not only sensitive to structural damage and degradation, but also to varying operational and environmental conditions, such as traffic, sunshine, wind and most importantly, temperature. Understanding the performance of the bridge, and investigating the influences of operational and environmental variables, are two tasks we have discussed in this thesis.

We explore the nature of the bridge through measurements collected with a sensor network installed on the bridge, which consists of three types of sensor: strain, vibration and temperature sensors. The signals collected by strain sensors are sensitive to traffic loadings (including normal traffic events and traffic jams), as well as temperature changes and various types of noise. Individual vehicles are represented as peaks (with various durations and amplitudes), and traffic jams are represented as sharp baseline jumps, which last much longer than normal

traffic peaks. Temperature changes cause a gradual baseline drift, which is a low-frequency effect. Finally, noise is any high-frequency component, which appears as small fluctuations in strain signals. The vibration signals are sensitive to noise and normal traffic events, but not sensitive to low-frequency components, such as environmental changes and traffic jams. Temperature is a low-frequency signal, which is just sensitive to local temperature changes, with some level of delay to environmental temperature, in the order of several hours.

In the end, all sensors are attached to the same bridge, and respond to dynamics occurring on the bridge, so there must be some kinds of dependencies among the various sensors. To look into the dependency between each sensor type pair, we employ datasets of different scales, and analyse them in both the time and the frequency domains. In the time domain,

- The dependency between strain and temperature sensors is strong at a large scale, but is weak at a small scale; temperature is not affected by traffic loadings on the bridge.
- The dependency between strain and vibration sensors is weak at both large and small scales, because the former are not only sensitive to traffic events, but also to temperature, and the latter are just sensitive to traffic events; what's more, the responses of these two types of sensors to traffic events are different.
- The dependency between vibration and temperature sensor types is fairly weak at both big and small scales.

In the frequency domain,

- The dependency between the strain and temperature sensor types is weak at a small scale. At a large scale, due to the daily and seasonal fluctuations, the low components of strain spectrum correlates with that of temperature spectrum.
- When the bridge is excited by traffic events, some spectral components in the strain and vibration spectra are highly correlated.

- the dependency between the vibration and temperature sensor types is the same as that in the time domain.

Having a big picture about dependencies among sensor types in both the time and the frequency domains at various scales, we analysed the dependency between the strain sensor type and the temperature sensor type in the time frequency at a big scale, and employed an exponential decay model to overcome delays caused by concrete properties. We analysed the dependency between the strain sensor type and the vibration sensor type in the frequency domain at a small scale, and applied bandpass filters to the spectra. We observed that the dependency between vibration and temperature sensors is fairly weak in both the time and the frequency domains at any scales, so we first conducted modal analysis on vibration signals, and then associate natural frequencies with temperature.

In the sensor network, there are 145 sensors of different properties (sensor type, location, orientation), and the level of correlation between different sensors appears to depend on these properties. To further look into the dependencies in detail, we employed Subgroup Discovery techniques to analyse correlations of all sensor pairs of different types, and obtained a number of interesting patterns (rules).

As mentioned above, temperature has a strong influence on strain signals. To separate the influence of temperature from other effects, we proposed a baseline correction method (the *most-crossing* method), which is based on the probability density function. Within a sliding window, we assume the baseline is a constant value, and divide data points into two categories: noise and peaks. The PDF of the noise category is different from that of peaks. We take the peak value of the former PDF as baseline, and model adjacent sliding windows with linear interpolation. The most-crossing method is not only capable of catching baseline drift in our strain signals, but also works well on other kinds of datasets.

In the entire thesis, traffic events play a crucial rule. We propose two supervised methods to identify traffic events: one of which is a classification method based on video labels, as described in Section 7.2.2. The other one is a predefined pattern detection method based on template, landmarks and constraints, as proposed in

Chapter 6. The former method is precise, but requires a lot of effort to manually label traffic events from video; the latter method is based on prior knowledge and the MDL principle, which is fast, and of satisfactory precision. In Chapter 7, we employed the first method to illustrate the procedure of data selection and modal parameter extraction. To look into the influence of environmental factors, we needed to process a huge amount of datasets, so the second method is chosen for traffic event identification for this purpose.

There are a number of modal analysis methods to extract modal parameters, such as the PP method and the SSI method. We employed the SSI method to extract natural frequencies, mode shapes and damping ratios, and verified that the natural frequencies obtained with the SSI method are correlated well with those obtained with the PP method. Because the PP method is simple and of acceptable accuracy, we employed it to extract natural frequencies from datasets selected from more than two years' measurements, and associated them with temperature. Generally speaking, natural frequencies decrease when temperature increases. We looked into the relationship between temperature and natural frequencies mode by mode, and found that high-frequency modes are more sensitive to temperature than low-frequency modes. We also analysed the influence of vehicle mass on natural frequencies, and found that natural frequencies are less sensitive to traffic mass.

8.2 Discussion

As mentioned in Chapter 1, this thesis focuses on Part 2 and Part 3 of the SHM process. In this section, we will discuss some topics related to Part 1: Operational Evaluation. The first topic we want to discuss is whether the sensor network is the best solution for structural health monitoring. To answer this question, we should compare it with some other solutions. In the literature, there is some research utilising radar systems, which are flexible and capable of catching the (absolute) displacements accurately. However, the radar systems are more suitable for short-term measurements than long-term tasks. In long-term

SHM, we are usually interested in more factors related to the bridge than just displacements. The sensor network composed of multiple sensor types is more informative, which is usually preferred for long-term measurements.

The second topic relates to the optimal number and distribution of sensors within a sensor network. In a sensor network, if we employ too few sensors, the measurements are not enough to catch the performance of the bridge; however, if we employ too many sensors, there will be a lot of redundant measurements, which increases the burden of data-storage. Another factor related to the number of sensors is the distribution of sensors. Given a bridge, not every location on it is of the same importance. There are some points that are sensitive to loads, while some points are not. Understanding the structural properties of the bridge helps us determine the distribution of sensors, and then figure out the optimal number of sensors.

In our sensor network, there is considerable redundancy, and the sensor distribution can also be improved. As introduced in Chapter 3, the sensors in the sensor network are distributed along three cross-sections of the last half part of a single span. The number of sensors within each cross-section is also different, one of which covers 78.6% of all sensors. The improved sensor network should at least cover the whole span (to improve modal analysis), and the sensor number within each cross-section should be approximately equal.

8.3 Future work

In the future, we try to improve our work in the following directions:

Minimal sensor network We will build a minimal sensor network model. To monitor the health of a bridge, it is necessary to employ a number of sensors. However, it is not true that the more sensors there are, the better for an SHM system. To figure out the optimal sensor number (minimal sensor network), we have explored the dependencies among multiple sensor types in Chapter 4. In the future, we will also take the dependencies within the same sensor type into account.

Automatic baseline correction method In Chapter 5, we have proposed a baseline correction method, the most-crossing method. One of the important parameters in the method is the size of the sliding window. In the method, we empirically select a parameter as the size of the sliding window. In the future, we will introduce the MDL principle to the most-crossing method to select the optimal window size. We suppose that the optimal window size will lead to the minimal MDL score.

Multiple-scale pattern detection In Chapter 5, we focus on catching the trend (baseline) hidden in the time series. The baseline can be viewed as a large-scale pattern. In Chapter 6, we pay more attention to extract the predefined patterns, which are usually of small scale. In the future, we will develop a method to identify patterns of multiple scales, by combining the most-crossing method with the predefined pattern detection method. We suppose that the MDL scores of patterns of the same scale follow similar density distributions, and the critical points between adjacent MDL-score distributions can be taken as thresholds of multi-scale patterns.

More accurate modal analysis results In Chapter 7, we employed a number of datasets for modal analysis. The experimental results indicate that the temperature has a clear influence on natural frequencies, and the influence of mass on natural frequencies is obscure. To look into the environmental influence on modal parameters in more detail, we will employ more datasets for modal analysis, and instead of just considering the environmental influence on natural frequencies, we will also take mode shapes and damping ratios into account.

References

[1] FHWA: Reliability of visual inspection for highway bridges. Technical Report FHWA-RD-01-020, Federal Highway Administration (2001)

[2] Farrar, C.R., Doebling, S.W., Nix, D.A.: Vibration-based structural damage identification. Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences **359** (2001) 131–149

[3] Sohn, H., Farrar, C.R., Hemez, F., Czarnecki, J.: A review of structural health monitoring literature 1996 – 2001. Technical Report LA-13976-MS, Los Alamos National Laboratory (2004)

[4] Schulze, G., Jirasek, A., Yu, M.M., Lim, A., Turner, R.F., Blades, M.W.: Investigation of selected baseline removal techniques as candidates for automated implementation. Applied Spectroscopy **59** (2005) 545–574

[5] James III, G.H., Carne, T.G., Lauffer, J.P.: The natural excitation technique (NexT) for modal parameter extraction from operating wind turbines. Sandia National Laboratories Report (1993)

[6] Brincker, R., Zhang, L., Andersen, P.: Modal identification of output-only systems using frequency domain decomposition. Smart Materials and Structures **10** (2001) 441–445

[7] Li, H., Li, S., Ou, J.: Modal identification of bridges under varying environmental conditions: Temperature and wind effects. Structural Control and Health Monitoring **17** (2010) 495–512

[8] Reynders, E.: System identification methods for (operational) modal analysis: Review and comparison. Archives of Computational Methods in Engineering **19** (2012) 51–124

[9] Bendat, J.S., Piersol, A.G.: Engineering Applications of Correlation and Spectral Analysis. Wiley (1993)

[10] Maia, N.M.M., Silva, J.M.M.: Theoretical and Experimental Modal Analysis. Research Studies Press (1997)

[11] Peeters, B., De Roeck, G.: Reference-based stochastic subspace identification for output-only modal analysis. Mechanical Systems and Signal Processing **13** (1999) 855–878

[12] Brincker, R., Andersen, P.: Understanding stochastic subspace identification. In: Proceedings of International Modal Analysis Conference. (2006)

[13] Moaveni, B., Asgarieh, E.: Deterministic-stochastic subspace identification method for identification of nonlinear structures as time-varying linear systems. Mechanical Systems and Signal Processing **31** (2012) 40–55

[14] Zhang, G., Tang, B., Tang, G.: An improved stochastic subspace identification for operational modal analysis. Measurement **45** (2012) 1246–1256

[15] Döhler, M., Andersen, P., Mevel, L.: Operational modal analysis using a fast stochastic subspace identification method. Topics in Modal Analysis I, **5** (2012) 19–24

[16] Gamberger, D., Lavrac, N.: Expert-guided subgroup discovery: Methodology and application. Artificial Intelligence Research **17** (2002) 501–527

[17] van Leeuwen, M., Knobbe, A.: Non-redundant subgroup discovery in large and complex data. In: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III. ECML PKDD'11, Berlin, Heidelberg, Springer-Verlag (2011) 459–474

[18] Herrera, F., Carmona, C.J., González, P., del Jesus, M.J.: An overview on subgroup discovery: foundations and applications. Knowledge and Information Systems **29** (2011) 495–525

[19] Smith, S.M.: The Scientist and Engineer's Guide to Digital Signal Processing. Second edn. California Technical Publishing (1999)

[20] Weisstein Eric W. "Integer." From Mathworld–A Wolfram Web Resource. `http://mathworld.wolfram.com/Integer.html`.

[21] Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Mathematics of Computation **19** (1965) 297–301

[22] Hespanha, J.P.: Linear System Theory. Princeton University Press (2009)

[23] Stranneby, D., Walker, W.: Digital Signal Processing and Applications. Elsevier (2004)

[24] Esling, P., Agon, C.: Time-series data mining. ACM Computing Surveys (CSUR) **45** (2012) 12:1–12:34

[25] Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proceedings of the International Conference on Management of Data. (1994) 419–429

[26] Yi, B., Faloutsos, C.: Fast time sequence indexing for arbitrary Lp norms. In: Proceedings of International Conference on Very Large Data Bases. (2000) 385–394

[27] Alcock, R.J., Manolopoulos, Y.: Time-series similarity queries employing a feature-based approach. In: Proceedings of Hellenic Conference on Informatics. (1999) 27–29

[28] Chen, Y., Nascimento, M., Ooi, B., Tung, A.: SpADe: On shape-based pattern detection in streaming time series. In: Proceedings of International Conference on Data Engineering. (2007) 786–795

[29] Pearson, K.: Notes on regression and inheritance in the case of two parents. In: Proceedings of the Royal Society of London. Volume 8. (1895) 240–242

[30] Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proceedings of KDD workshop. (1994)

[31] Srisai, D., Ratanamahatana, C.A.: Efficient time series classification under template matching using time warping alignment. In: Proceedings of International Conference on Computer Sciences and Convergence Information Technology. (2009) 685–690

[32] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of International Conference on Knowledge Discovery and Data Mining. (2012) 262–270

[33] Niennattrakul, V., Srisai, D., Ratanamahatana, C.A.: Shape-based template matching for time series data. Knowledge-Based System **26** (2012) 1–8

[34] Fu, A., Keogh, E., Lau, L., Ratanamahatana, C., Wong, R.: Scaling and time warping in time series querying. The International Journal on Very Large Data Bases **17** (2008) 899–921

[35] Ambrosio, A.: Vector p-norm (2013) available at `http://planetmath.org/VectorPnorm`.

[36] Keogh, E.: Exact indexing of dynamic time warping. In: Proceedings of International Conference on Very Large Data Bases. (2002) 406–417

[37] Ratanamahatana, C., Keogh, E.: Making time-series classification more accurate using learned constraints. In: Proceedings of SIAM International Conference on Data Mining. (2004) 11–22

[38] Yang, Y.B., Chang, K.C.: Extracting the bridge frequencies indirectly from a passing vehicle: Parametric study. Engineering Structures **31** (2009) 2448–2459

[39] Song, W., Dyke, S.J.: Ambient vibration based modal identification of the emerson bridge considering temperature effects. In: Proceedings of World Conference on Structural Control and Monitoring. (2006)

[40] Xia, Y., Hao, H., Zanardo, G., Deeks, A.: Long term vibration monitoring of an RC slab: temperature and humidity effect. Engineering Structures **28** (2006) 441–452

[41] Reynolds, P., Pavic, A.: Comparison of forced and ambient vibration measurements on a bridge. In: Proceedings of International Modal Analysis Conference. (2001) 846–851

[42] Meeng, M., Knobbe, A.: Flexible enrichment with cortana-software demo. In: Proceedings of Machine Learning Conference of Belgium and The Netherlands. (2011)

[43] Meruane, V., Heylen, W.: Structural damage assessment under varying temperature conditions. Structural Health Monitoring **11** (2011) 345–357

[44] Cunha, A., Caetano, E., Magalhães, F., Moutinho, C.: Recent perspectives in dynamic testing and monitoring of bridges. Structural Control and Health Monitoring **20** (2013) 853–877

[45] Cornwell, P., Farrar, C.R., Doebling, S.W., Sohn, H.: Environmental variability of modal properties. Experimental Techniques **23** (1999) 45–48

[46] Brownjohn, J.M.W., Moyo, P., Omenzetter, P., Chakraborty, S.: Lessons from monitoring the performance of highway bridges. Structural Control and Health Monitoring **12** (2005) 227–244

[47] Xia, Y., Chen, B., Zhou, X.Q., Xu, Y.L.: Field monitoring and numerical analysis of tsing ma suspension bridge temperature behavior. Structural Control and Health Monitoring **20** (2013) 560–575

[48] Xu, Y.L., Chen, B., Ng, C.L., Wong, K.Y., Chan, W.Y.: Monitoring temperature effect on a long suspension bridge. Structural Control and Health Monitoring **17** (2010) 632–653

[49] Ni, Y.Q., Hua, X.G., Fan, K.Q., Ko, J.M.: Correlating modal properties with temperature using long-term monitoring data and support vector machine technique. Engineering Structures **27** (2005) 1762–1773

[50] Peeters, B., De Roeck, G.: One-year monitoring of the Z24-bridge: environmental effects versus damage events. In: Proceedings of International Modal Analysis Conference. (2000) 1570–1576

[51] Liu, C.Y., Dewolf, J.T., Fasce, P.E.: Effect of temperature on modal variability of a curved concrete bridge under ambient loads. Journal of structural engineering **133** (2007) 1742–1751

[52] Farrar, C.R., Doebling, S.W., Cornwel, P.J., Straser, E.G.: Variability of modal parameters measured on the alamosa canyon bridge. In: Proceedings of International Modal Analysis Conference. (1997) 257–263

[53] Miao, S., Knobbe, A., Koenders, E., Bosma, C.: Analysis of traffic effects on a Dutch highway bridge. In: Proceedings of IABSE. (2013)

[54] Karoumi, R., Wiberg, J., Liljencrantz, A.: Monitoring traffic loads and dynamic effects using an instrumented railway bridge. Engineering Structures **27** (2005) 1813–1819.

[55] Zhu, X.Q., Law, S.S.: Moving load identification on multi-span continuous bridges with elastic bearings. Mechanical Systems and Signal Processing **20** (2006) 1759–1782

[56] Wikipedia: Probability density function (2013) available at `http://en.wikipedia.org/wiki/Probability_density_function` .

[57] Marron, J.S., Wand, M.P.: Exact mean integrated squared error. The Annals of Statistics **20** (1992) 712–736

[58] Silverman, B.W.: Density estimation for statistics and data analysis. Chapman and Hall, London (1986)

[59] Dietrich, W., Rudel, C.H., Neumann, M.: Fast and precise automatic baseline correction of one- and two-dimensional nmr spectra. Journal of Magnetic Resonance **91** (1991) 1–11

[60] Gan, F., Ruan, G.H., Mo, J.Y.: Baseline correction by improved iterative polynomial fitting with automatic threshold. Chemometrics and Intelligent Laboratory Systems **82** (2006) 59–65

[61] Hartigan, J.A.: Clustering Algorithms. John Wiley & Sons, Inc., New York, NY, USA (1975)

[62] Ralston, H.R., Wilcox, G.E.: A computer method of peak area determinations from Ge-Li gamma spectra. In: Proceedings of Modern Trends in Activation Analysis. (1968) 1238–1243

[63] Shao, X.G., Ma, C.X.: A general approach to derivative calculation using wavelet transform. Chemometrics and Intelligent Laboratory Systems **69** (2003) 157–165

[64] Mosier-Boss, P.A., Lieberman, S.H., Newbery, R.: Fluorescence rejection in raman spectroscopy by shifted-spectra, edge detection, and FFT filtering techniques. Applied Spectroscopy **49** (1995) 630–638

[65] Friedrichs, M.S.: A model-free algorithm for the removal of baseline artifacts. J Biomol NMR **5** (1995) 147–153

[66] Lieber, C.A., Mahadevan-Jansen, A.: Automated method for subtraction of fluorescence from biological raman spectra. Applied Spectroscopy **57** (2003) 1363–1367

[67] Rowlands, C., Elliott, S.: Automated algorithm for baseline subtraction in spectra. Journal of Raman Spectroscopy **42** (2011) 363–369

[68] Wei, L., Keogh, E., Van Herle, H., Mafra-Neto, A.: Atomic wedgie: Efficient query filtering for streaming times series. In: Proceedings of International Conference on Data Mining. (2005) 490–497

[69] Niennattrakul, V., Ratanamahatana, C.: Shape averaging under time warping. In: Proceedings of International Conference on Electrical Engineering/Electronics, Computer,Telecommunications, and Information Technology. (2009) 626–629

[70] Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of ACM SIGMOD Workshop on Research Numbers in Data Mining and Knowledge Discovery. (2003) 2–11

[71] Bagnall, A., Ratanamahatana, C., Keogh, E., Lonardi, S., Janacek, G.: A bit level representation for time series data mining with shape based similarity. Data Mining and Knowledge Discovery **13** (2006) 11–40

[72] Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems **3** (2001) 263–286

[73] Perng, C.S., Wang, H.X., Zhang, S.R., Parker, D.S.: Landmarks: a new model for similarity-based pattern querying in time series databases. In: Proceedings of International Conference on Data Engineering. (2000) 33–42

[74] Bandera, J.P., Marfil, R., Bandera, A., Rodríguez, J.A., Molina-Tanco, L., Sandoval, F.: Fast gesture recognition based on a two-level representation. Pattern Recognition Letters **30** (2009) 1181–1189

[75] Shatkay, H., Zdonik, S.: Approximate queries and representations for large data sequences. In: Proceedings of International Conference on Data Engineering. (1996) 536–545

[76] Mohammad, Y., Nishida, T.: Constrained motif discovery in time series. New Generation Computing **27** (2009) 319–346

[77] Grünwald, P.D.: The Minimum Description Length Principle. The MIT Press (2007)

[78] Vespier, U., Knobbe, A., Nijssen, S., Vanschoren, J.: MDL-Based analysis of time series at multiple time-scales. In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. (2012) 371–386

[79] Vespier, U., Nijssen, S., Knobbe, A.: Mining characteristic multi-scale motifs in sensor-based time series. In: Proceedings of International Conference on Information and Knowledge Management. (2013) 2393–2398

[80] Liu, T., Chen, H.: Real-time tracking using trust-region methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004) 397–402

[81] Rakthanmanon, T., Keogh, E., Lonardi, S., Evans, S.: Time series epenthesis: Clustering time series streams requires ignoring some data. In: Proceedings of International Conference on Data Mining. (2011) 547–556

[82] Hu, B., Rakthanmanon, T., Hao, Y., Evans, S., Lonardi, S., Keogh, E.: Discovering the intrinsic cardinality and dimensionality of time series using MDL. In: Proceedings of International Conference on Data Mining. (2011) 1086–1091

[83] Walraven, G.: Basic Arrhythmias. Prentice Hall (2010)

[84] Wikipedia: QRS complex (2014) available at `http://en.wikipedia.org/QRS_complexs`.

[85] Afonso, V.X.: ECG QRS detection. Biomedical Digital Signal Processing: C-language Examples and Laboratory Experiments for the IBM PC (1993) 236–264

[86] Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR time series classification/clustering page (2011) available at `http://www.cs.ucr.edu/~eamonn/time_series_data/`.

[87] Amit, Y., Grenander, U., Piccioni, M.: Structural image restoration through deformable templates. Journal of the American Statistical Association **86** (1991) 376–387

[88] Liu, R., Lu, Y., Gong, C., Liu, Y.: Infrared point target detection with improved template matching. Infrared Physics & Technology **55** (2012) 380–387

[89] Lüthi, M., Jud, C., Vetter, T.: Using landmarks as a deformation prior for hybrid image registration. In: Proceedings of International Conference on Pattern Recognition. (2011) 196–205

[90] Jin, Z., Lou, Z., Yang, J., Sun, Q.: Face detection using template matching and skin-color information. Neurocomputing **70** (2007) 794–800

[91] Caprari, R.S.: Duplicate document detection by template matching. Image and Vision Computing **18** (2000) 633–643

[92] Müller, M., Röder, T.: Motion templates for automatic classification and retrieval of motion capture data. In: Proceedings of ACM SIGGRAPH/Eurographics symposium on Computer animation. (2006) 137–146

[93] Ge, X., Smyth, P.: Deformable markov model templates for time-series pattern matching. In: Proceedings of International Conference on Knowledge Discovery and Data Mining. (2000) 81–90

[94] Frank, J., Mannor, S., Pineau, J., Precup, D.: Time series analysis using geometric template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 740–754

[95] Agrawal, R., Lin, K., Sawhney, H.S., Shim, K.: Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In: Proceedings of International Conference on Very Large Databases. (1995) 490–501

[96] Ye, L., Keogh, E.: Time series shapelets: A new primitive for data mining. In: Proceedings of International Conference on Knowledge Discovery and Data Mining. (2009) 947–956

[97] Palpanas, T., Vlachos, M., Keogh, E., Gunopulos, D., Truppel, W.: Online amnesic approximation of streaming time series. In: Proceedings of International Conference on Data Engineering. (2004) 338–349

[98] Korn, F., Jagadish, H.V., Faloutsos, C.: Efficiently supporting ad hoc queries in large datasets of time sequences. In: Proceedings of International Conference on Management of Data. (1997) 289–300

[99] Chan, K., Fu, A.: Efficient time series matching by wavelets. In: Proceedings of International Conference on Data Engineering. (1999) 126–133

[100] Wang, X., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. Data Mining and Knowledge Discovery **13** (2006) 335–364

[101] Gullo, F., Ponti, G., Tagarelli, A., Greco, S.: A time series representation model for accurate and fast similarity detection. Pattern Recognition **42** (2009) 2998–3014

[102] Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time-series data. International Journal of Computer Research **10** (2001) 49–61

[103] Aßfalg, J., Kriegel, H.P., Kröger, P., Kunath, P., Pryakhin, A., Renz, M.: Similarity search in multimedia time series data using amplitude-level features. In: Proceedings of International Conference on Advances in Multimedia Modeling. (2008) 123–133

[104] Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Transactions on Database Systems **27** (2002) 188–228

[105] Megalooikonomou, V., Li, G., Wang, Q.: A dimensionality reduction technique for efficient similarity analysis of time series databases. In: Proceedings of International Conference on Information and Knowledge Management. (2004) 160–161

[106] Wang, Q., Megalooikonomou, V.: A dimensionality reduction technique for efficient time series similarity analysis. Information Systems **33** (2008) 115–132

[107] Megalooikonomou, V., Wang, Q., Li, G., Faloutsos, C.: A multiresolution symbolic representation of time series. In: Proceedings of International Conference on Data Engineering. (2005) 668–679

[108] Lin, J., Vlachos, M., Keogh, E., Gunopulos, D., Liu, J., Yu, S., Le, J.: A MPAA-based iterative clustering algorithm augmented by nearest neighbors search for time-series data streams. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2005) 333–342

[109] Vlachos, M., Gunopoulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: Proceedings of International Conference on Data Engineering. (2002) 673–684

[110] Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proceedings of International Conference on Very Large Data Bases. (2004) 792–803

[111] Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of International Conference on Management of Data. (2005) 491–502

[112] Sohn, H., Dzwonczyk, M., Straser, E.G., Kiremidjian, A.S., Law, K.H., Meng, T.: An experimental study of temperature effect on modal parameters of the alamosa canyon bridge. Earthquake Engineering and Structural Dynamics **28** (1999) 879–897

[113] Peeters, B., De Roeck, G., Hermans, L., Wauters, T., Kramer, C., De Smet, C.A.M.: Comparison of system identification methods using operational data of a bridge test. In: Proceedings of International Conference on Noise and Vibration Engineering. (1998) 923–930

[114] Peeters, B., De Roeck, G.: Reference based stochastic subspace identification in civil engineering. Inverse Problems in Engineering **8** (2000) 47–74

[115] Bodeux, J.B., Golinval, J.C.: Application of ARMAV models to the identification and damage detection of mechanical and civil engineering structures. Smart Materials and Structures **10** (2001) 479–489

[116] James III, G.H., Carne, T.G., Lauffer, J.P.: The natural excitation technique (NexT) for modal parameter extraction from operating structures. Modal Analysis **10** (1995) 260–277

[117] Ibrahim, S.R.: Efficient random decrement computation for identification of ambient responses. In: Proceedings of International Modal Analysis Conference. (2001) 1–6

[118] Van Overschee, P., De Moor, B.: Subspace Identification for Linear Systems: Theory, Implementation and Applications. Kluwer Academic Publishers (1996)

[119] Thai, H., DeBrunner, V., DeBrunner, L.S., Havlicek, J.P., Mish, K., Ford, K., Medda, A.: Deterministic-stochastic subspace identification for bridges. In: Proceedings of Workshop on Statistical Signal Processing. (2007) 749–753

[120] Foti, D., Diaferio, M., Giannoccaro, N.I., Mongelli, M.: Ambient vibration testing, dynamic identification and model updating of a historic tower. NDT & E International **47** (2012) 88–95

[121] De Roeck, G., Peeters, B., Ren, W.X.: Benchmark study on system identification through ambient vibration measurements. In: Proceedings of International Modal Analysis Conference. (2000) 1106–1112

[122] Bakir, P.G.: Automation of the stabilization diagrams for subspace based system identification. Expert System with Applications **38** (2011) 14390–14397

[123] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations **11** (2009) 10–18

[124] Shen, F., Zheng, M., Feng Shi, D., Xu, F.: Using the cross-correlation technique to extract modal parameters on response-only data. Sound and Vibration **259** (2003) 1163–1179

[125] Kim, B.H., Lee, J., Lee, D.H.: Extracting modal parameters of high-speed railway bridge using the TDD technique. Mechanical Systems and Signal Processing **24** (2010) 707–720

[126] Felber, A.J.: Development of a hybrid bridge evaluation system. PhD thesis, University of British Columbia, Vancouver, Canada (1993)

[127] Ren, W.X., Zong, Z.H.: Output-only modal parameter identification of civil engineering structures. Structural Engineering and Mechanics **17** (2004) 429–444

[128] Masjedian, M.H., Keshmiri, M.: A review on operational modal analysis researches: Classification of methods and applications. In: Proceedings of International Operational Modal Analysis Conference. (2009)

[129] Kim, B.H., Stubbs, N., Park, T.: A new method to extract modal parameters using output-only responses. Sound and Vibration **282** (2005) 215–230

[130] Cauberghe, B., Guillaume, P., Verboven, P., Parloo, E., Vanlanduit, S.: The secret behind clear stabilization diagrams: The influence of the parameter constraint on the stability of the poles. In: International Congress and Exposition on Experimental and Applied Mechanics. (2004)

[131] Gomez, H.C., Ulusoy, H.S., Feng, M.Q.: Variation of modal parameters of a highway bridge extracted from six earthquake records. Earthquake Engineering and Structural Dynamics **42** (2013) 565–579

[132] Pakrakshi, V., Connor, A., Basu, B.: Reliability of flexible guideways with tuned mass dampers. In: Proceedings of Symposium on Bridge Engineering. (2004)

Nederlandse Samenvatting

Door ontwikkelingen in meet- en dataverwerkingstechnieken wordt in verschillende domeinen het monitoren van een fysiek systeem door middel van een sensornetwerk een realistische optie. Deze monitoringsystemen worden Structural Health Monitoring (SHM) systemen genoemd (Constructieve Gezondheid Meetsystemen). De definitie van SHM is het implementeren van een schade-detectiesysteem en het kwalificeren van technische constructies, waarbij het verzamelen van data, het extraheren van schadegevoelige kenmerken en statistische analyse zijn inbegrepen. Omdat de meeste SHM processen kunnen worden verwerkt met technieken uit het Data-Mining-domein, heb ik in dit onderzoek deze twee onderzoeksgebieden gecombineerd.

Het monitoringssysteem dat is gebruikt in dit onderzoek is een sensornetwerk dat is geïnstalleerd op een Nederlandse snelwegbrug, die als doelstelling heeft om de dynamische gezondheidsaspecten van de brug en de degradatie over lange duur te monitoren. Deze doelstelling kan niet eenvoudig worden afgeleid van de meetresultaten omdat naast de verkeersimpact het sensornetwerk ook gevoelig is voor variabele omgevingsfactoren zoals vocht, wind en in belangrijke mate temperatuur.

Ik heb op verschillende schalen de specifieke eigenschappen van elke sensor en de afhankelijkheden tussen de sensoren onderzocht. De hieruit verworven resultaten leveren een goed inzicht in het sensornetwerk en helpen de sensoren te selecteren die het meest gevoelig zijn voor de belasting voor modale analyses.

De verzamelde meetresultaten van een gegeven sensor zijn niet altijd direct bruikbaar. Gedurende een verkeersvrije periode zullen de sensoren voornamelijk ruis

waarnemen, terwijl de grote hoeveelheid voertuigen tijdens de spits resulteren in meetgegevens die te gecompliceerd zijn voor modale analyse. Om een goede dataset te genereren, heb ik *free vibration* (vrije trilling) periodes gebruikt waarbij de trilling is veroorzaakt door afzonderlijke vrachtwagens.

In de meetresultaten kan een verkeersgebeurtenis worden gezien als een patroon, waardoor een gebeurtenis kan worden behandeld als een patroon-detectieprobleem. Gebaseerd op *landmarks* (kenmerken) en beperkingen hiertussen heb ik een nieuwe detectiemethode voor voorgedefinieerde patronen ontwikkeld.

Om de reactie op de temperatuur van de rest te onderscheiden, heb ik een basis correctie methode, de *most-crossing* methode ontwikkeld. Deze methode is gebaseerd op de kansdichtheidfunctie (Probability Density Function, PDF). In combinatie met het *Minimum Description Length* (MDL) principe kan deze methode worden gebruikt om nuttige patronen te detecteren, in potentie op meerdere tijdsschalen.

Op basis van de verworven datasets van hoge kwaliteit heb ik modale analyses uitgevoerd met de eenvoudige piekselectie (Peak-Picking) methode en de uitgebreide *Stochastic Subspace Identification* (SSI) methode. De resultaten van beide methoden komen goed overeen. De invloed van temperatuur en gewicht van verkeer op de eigenfrequenties is geanalyseerd, waaruit blijkt dat de eigenfrequenties dalen bij stijgende temperaturen, maar dat de invloed van gewicht minder duidelijk is.

English Summary

With the development of sensing and data processing techniques, monitoring physical systems in the field with a sensor network is becoming a feasible option for many domains. Such monitoring systems are referred to as Structural Health Monitoring (SHM) systems. By definition, SHM is the process of implementing a damage detection and characterisation strategy for engineering structures, which involves data collection, damage-sensitive feature extraction and statistical analysis. Most of the SHM process can be addressed by techniques from the Data Mining domain, so I conduct this research by combining these two fields.

The monitoring system employed in this research is a sensor network installed on a Dutch highway bridge, which aims to monitor dynamic health aspects of the bridge and its long-term degradation. Meeting these requirements is non-trivial, because the measurements collected with the sensor network are not only sensitive to traffic events, but also to varying environmental loadings, such as humidity, wind and most importantly, temperature.

I have explored the specific focus of each sensor type under multiple scales, and analysed the dependencies between sensor types. The obtained results have provided us with a thorough understanding of the sensor network, and helped us select the sensors that are sensitive to positive loads for modal analysis.

The measurements collected with a selected sensor are not always directly useful. During traffic-free time, the bridge is not excited, and the sensor just collects random noise, while during rush hour, the bridge is excited by multiple traffic events, which makes the measurements too complicated for modal analysis. To

obtain high-quality datasets, I have proposed to employ free-vibration periods, which are generated by single truck events.

In the measurements, a traffic event can be viewed as a pattern, so that the traffic event identification task can be addressed as a pattern detection problem. Based on landmarks and constraints, I have proposed a novel predefined pattern detection method.

To separate the temperature response from others, I have proposed a baseline correction method, the *most-crossing* method, which is based on the Probability Density Function (PDF). Combined with the principle of Minimum Description Length (MDL), the method can be used to detect useful patterns, potentially at multiple scales.

Based on the obtained high-quality datasets, I have conducted modal analysis with both the simple Peak-Picking method and the advanced Stochastic Subspace Identification method. The results of these two methods meet well. I have analysed the influence of temperature and traffic mass on natural frequencies, and verified that natural frequencies decrease with temperature increases, but the influence of traffic mass is not as obvious as that of temperature.

Acknowledgements

Many thanks to the following people:

My coworkers at LIACS (Leiden University): Joaquin Vanschoren, Siegfried Nijssen, Ugo Vespier, Ricardo Cachucho, Marvin Meeng, Wouter Duivesteijn, Jan van Rijn, Rob Konijn, Claudio Sá, Benjamin van der Burgh, Kleanthi Georgala, Harm de Vries, Michael Mampaey, Alberto Baggio, Ana Loureiro, and Geraldine Ribeiro.

My coworkers at LIAS (Leiden University): Hilde De Weerdt, HouLeong Ho, Mingkin Chu, and Julius Morche.

Our partners in the Faculty of Civil Engineering and Geosciences, Delft University of Technology: Eddy Koenders, René Veerman, and Fred Schilperoort.

My cowokers at IDM, Lanzhou University (China): Xiaoyun Chen, Longjie Li, Junjun Cheng, Mingwei Leng, Yi Chen, Pengfei Chen, Xin Zhang, Min Yue, Yanshan He, Weiguo Song, Liangzhai Ma, Yukai Yao, Guohua Liu, Youli Su, Baojun Gao, Ping Wen, Sha Liu, Juan Zhao, Yangyang Liu, Qiaojin Xing, and Tao Chen.

My friends in the Netherlands: Di Liu and Yan Liu, Zhao Zhou and Yang Yang, Tao Ma, Fujun Gou, Xiaoli Lu, Wen Pan, Irene Martorelli, Narjis Sellam, Rui Li, Xiaohu Li and Changchun Song, Yuanyuan Zhao and Sipeng Zheng, Jiongwei Wang and Xiaoqun Yang, Shengnan, Zhao, Guiling Chen and Jiabao Yan, Hao Qiu, Zhongxiao Wang, Wen Liang, Rongfang Liu, Jun Wang and Jinfeng Shen, Yongyi Wang and Yanan Wang, Chunli Song and Min Cheng, Tao Peng and Yuanyuan Li, Gesterkamp Lennert and Yuan Li, Yinxuan Huang, Chengcheng

8. ACKNOWLEDGEMENTS

Li, Jianbin Jiang, Bo Chen, Zheng Guan, Jin Wang, Xiaolei Niu, Xingrong Ma, Jiayuan Li, Jiaqi Zhao, Zhiguo Zhou, Fuyu Cai, Song Wu, Lu Cao, Yuanhao, Guo, Yanming Guo, Zhiwei Yang, Kaifeng Yang, Zhan Xiong, Weidong Zhuang, Hao Wang, Yu Liu, Mohamed Tleis, Muhammad Fawad Khan, Mohd Hafeez Osman, Wei Wang, Bing Lu, Chengjie Xing, Jifeng Liu, Jianqiang Sun, Fei Liu, Kai Fang and Xiaofei Zhang, Juan Zhou and Botao Xin, Peng Ye, Hua Wang and Hua Zhu.

My friends in China: Xingchen Liu and Shuzhen Liu, Xin Liu and Cuihua Mi, Lili Jiang and Cong Hu, Shilin Huang, Buyu Wang, Gang Chen and Lijuan Su, Xiaowei Li, Hong Peng, Shouliang Li, Guidong Zhang, Xin Jin, Lin Wang, Jingzhi Zhang, Jun Li, Jing Su, Rong Ma, and Zhili Zhao.

My family: Dezhong Miao and Cuirong Yuan, Minghe Wang and Airong Xu. My wife Yuli Wang and my son Hangcheng Miao.

Thank you for being part of my PhD journey.

Curriculum Vitae

Shengfa Miao was born in Shanxian, Heze, Shandong, China on September 8, 1981. In 2001, he received his secondary school degree from the fifth Shanxian Middle School and started his bachelor study in Computer Science, at Lanzhou University, Lanzhou, Gansu, China. He received the bachelor degree in July 2005, and was awarded the title "Outstanding Graduate Student". In the same year, he was employed by Lanzhou university as a teaching assistant. In September 2007, he was admitted by Lanzhou University as a master student in the major of Data Mining, under the supervision of Prof. Xiaoyun Chen. He graduated in June 2009 with a thesis entitled "The Application and Research of Data Warehouse and Search Engine in the Management of Alumni Resources". Right after his graduation, he started his PhD research following Prof. Xiaoyun Chen.

In November 2010, he obtained a PhD position at Leiden Institute of Advanced Computer Science, Leiden University, the Netherlands, under the supervision of Prof. dr. J.N. Kok and dr. A.J. Knobbe. His research mainly focused on the InfraWatch project. The project aims to monitor structural health by installing a sensor network on a structure, specifically, a highway bridge, and extracting modal parameters from datasets collected with the sensor network. As being presented in this thesis, Data Mining plays a key role in the procedure of Structural Health Monitoring.

As of September 2014, Shengfa works as a research associate in Prof. dr. H.G.D.G. De Weerdt's group, at Leiden University Institute for Area Studies, Leiden University, on a project entitled "Digging into Data: Automating Chinese Text Extraction".