Structural health monitoring meets data mining

Miao, S.

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

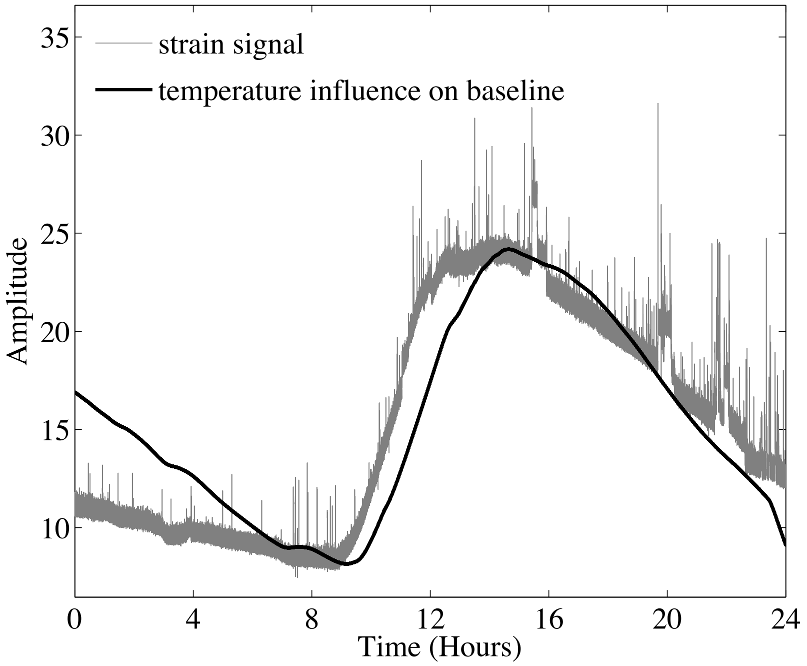

Figure 5.1: The influence of temperature on the strain signal - A linear model between the strain and temperature signals with a length of one day.

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.

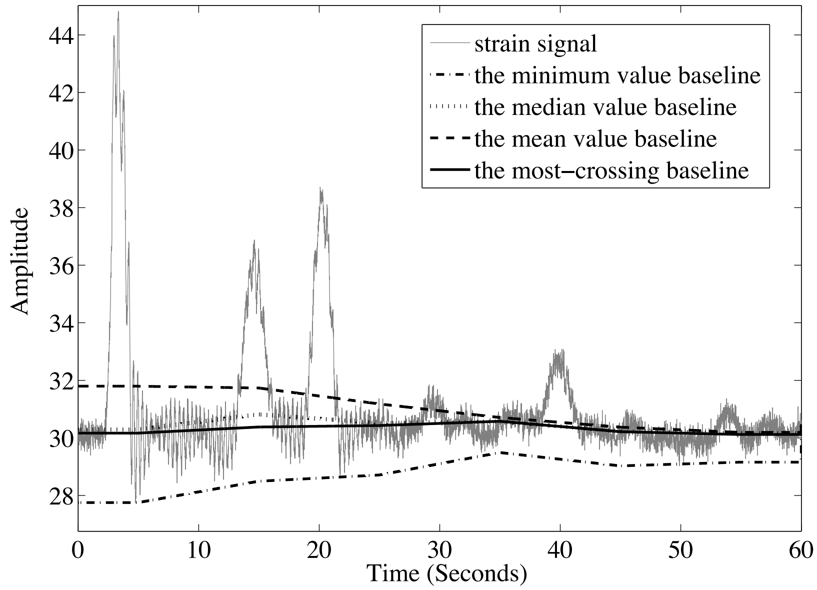

Figure 5.2: Comparison of several piece-wise baseline correction methods - The length of each window is 1000 data points (10 seconds). The baseline of each window is assumed to be a constant value, which is either obtained by calculating the mean value (the dashed line), the minimum (the dash-dotted 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].

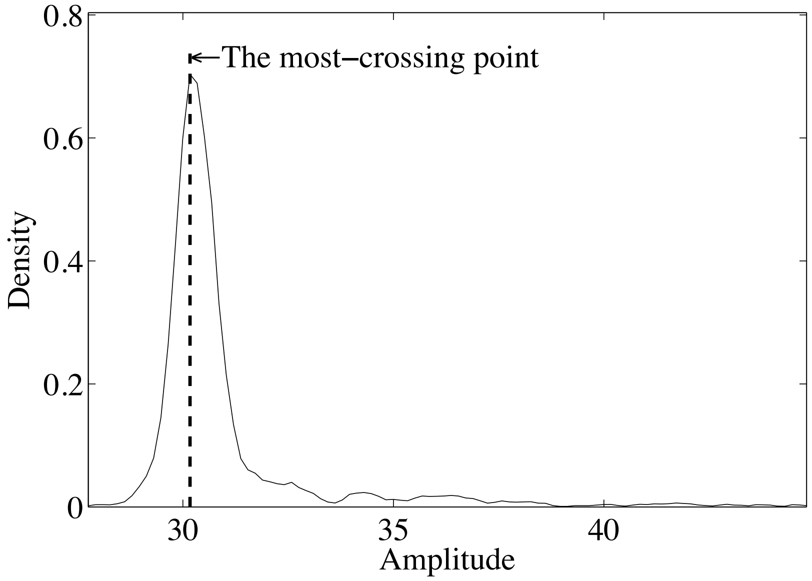

Figure 5.3: The kernel smoothed probability density function - The PDF is derived from the same dataset as Fig. 5.2; the most-crossing point is the first peak of the kernel smoothed PDF.

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]:

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

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.

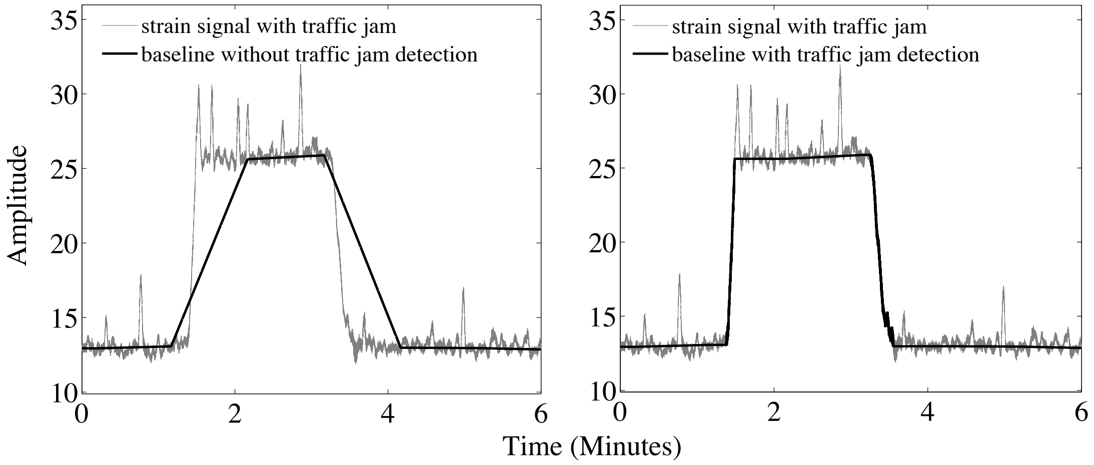

Figure 5.4: The reason for traffic jam detection - The baseline without traffic jam detection (left) and the baseline with traffic jam detection (right).

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

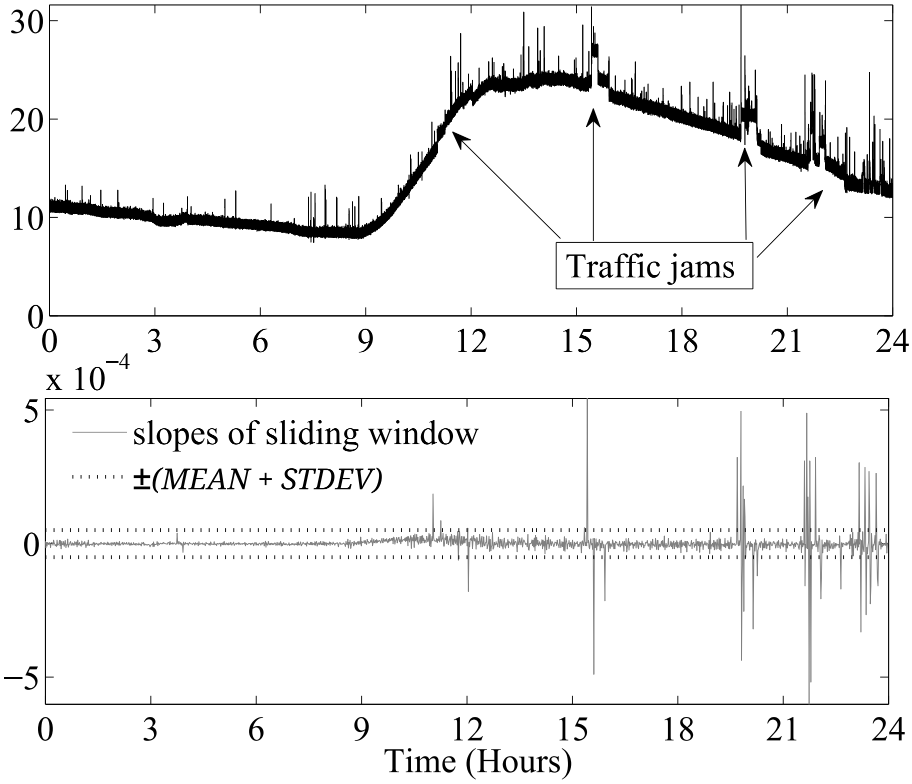

Figure 5.5: The traffic jam detection - The threshold for triggering the traffic jam detection procedure, which is set as the sum of the mean value ($MEAN$) and standard deviation ($STDEV$) of sliding window slopes.

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

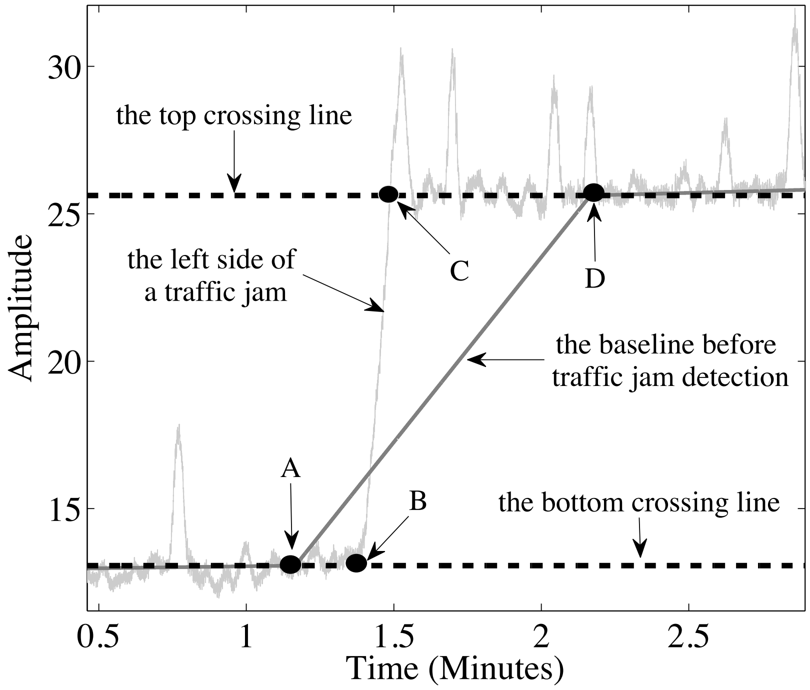

Figure 5.6: The traffic jam boundary detection - A and D are middle points of two successive sliding windows. The bottom-crossing line is a horizontal line across the middle point A. The top-crossing line is a horizontal line across the middle point D. The turning point B is the last intersection between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

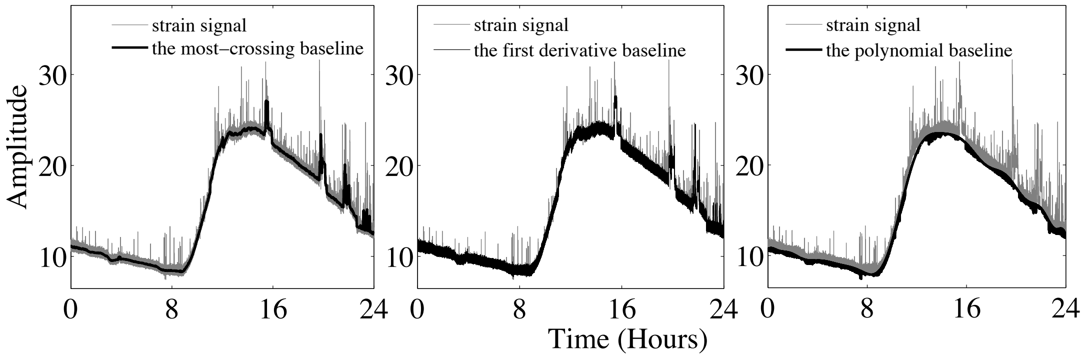

Figure 5.7: The comparison between three different baseline removal methods - The most-crossing baseline (left) is derived from a sliding window with length of 1 minute (6,000 data points). The first derivative baseline is obtained with a classification threshold of $MEAN+3\cdot STDEV$, and a false baseline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

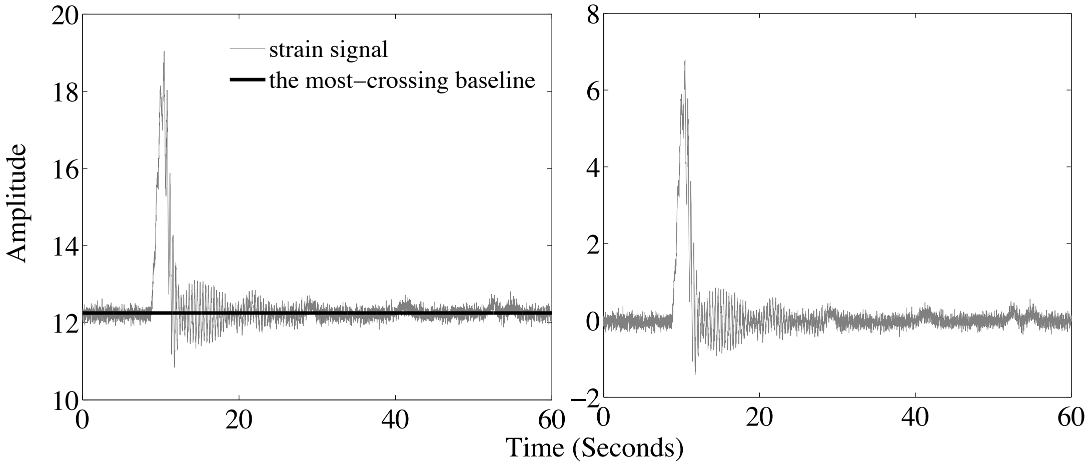

Figure 5.8: The most-crossing baseline over a short period signal - The baseline derived from the most-crossing method (left) and the baseline removed signal (right).

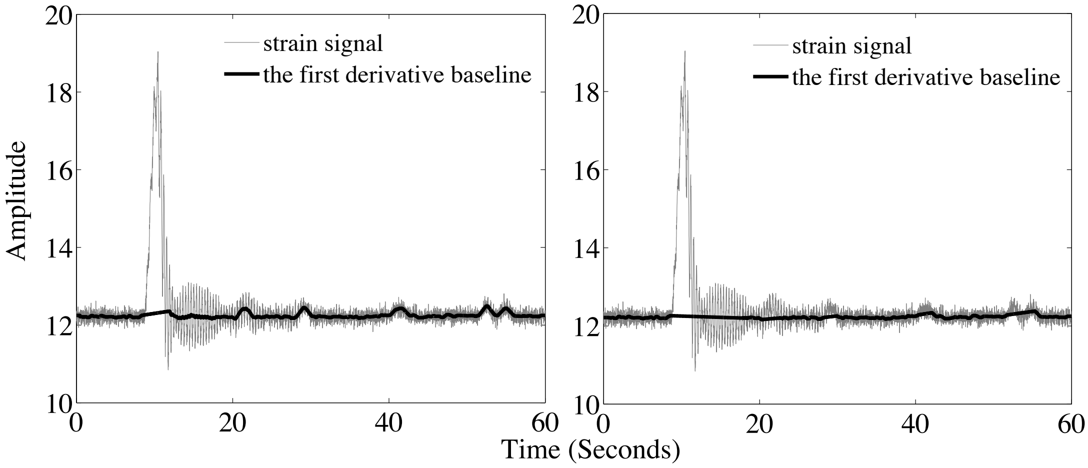

Figure 5.9: The first derivative baseline over a short period signal - The baseline obtained by just checking adjacent points (left) and the baseline obtained by correcting noise segments whose lengths are less than 150 data points.

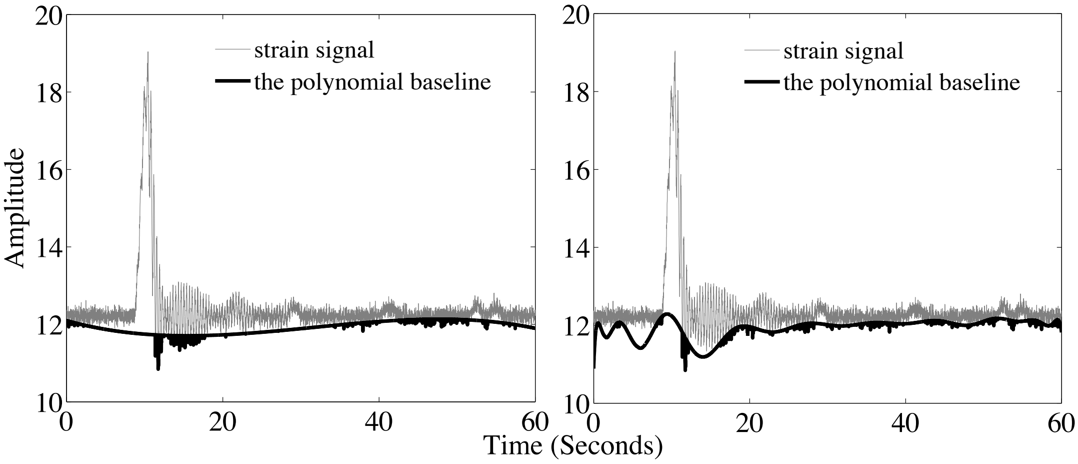

Figure 5.10: The polynomial baseline over a short period signal - The baseline derived from a 3-order polynomial fitting (left) and the baseline derived from a 20-order polynomial fitting (right).

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).

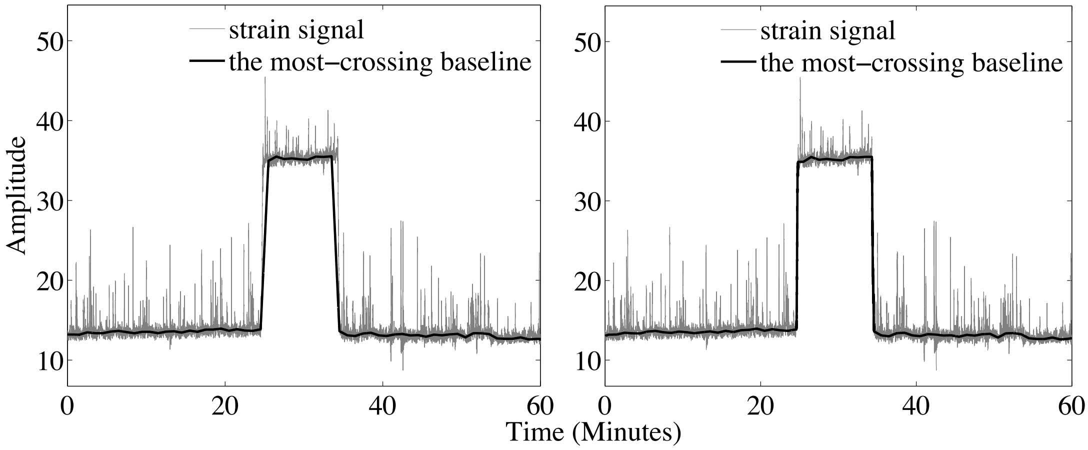

Figure 5.11: The most-crossing baseline for traffic jam signal - The traffic jam baseline before solving the boundary problem (left), and the baseline after traffic jam detection (right).

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.

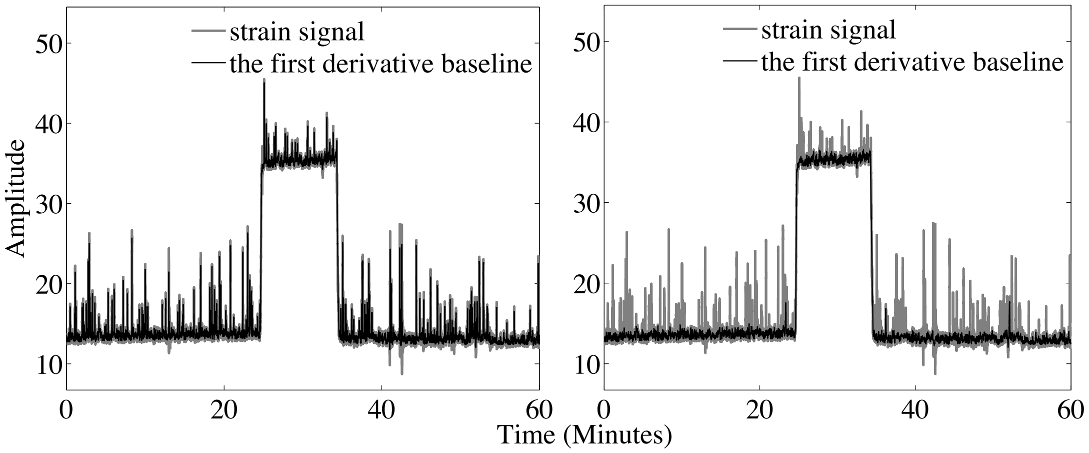

Figure 5.12: The first derivative baseline for a traffic jam signal - The first derivative-based baseline obtained by just checking adjacent points (left) and the baseline obtained by correcting noise segments whose lengths are less than 150 data points (right).

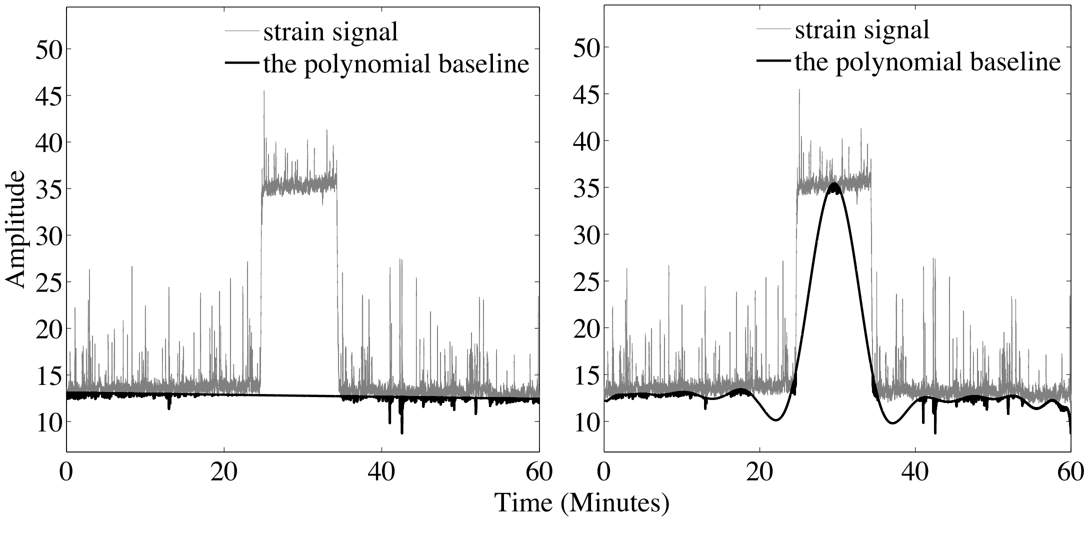

Figure 5.13: The polynomial baseline for traffic jam signal - The baseline derived from the 1 degree polynomial (left) and the 25 degree polynomial (right).

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

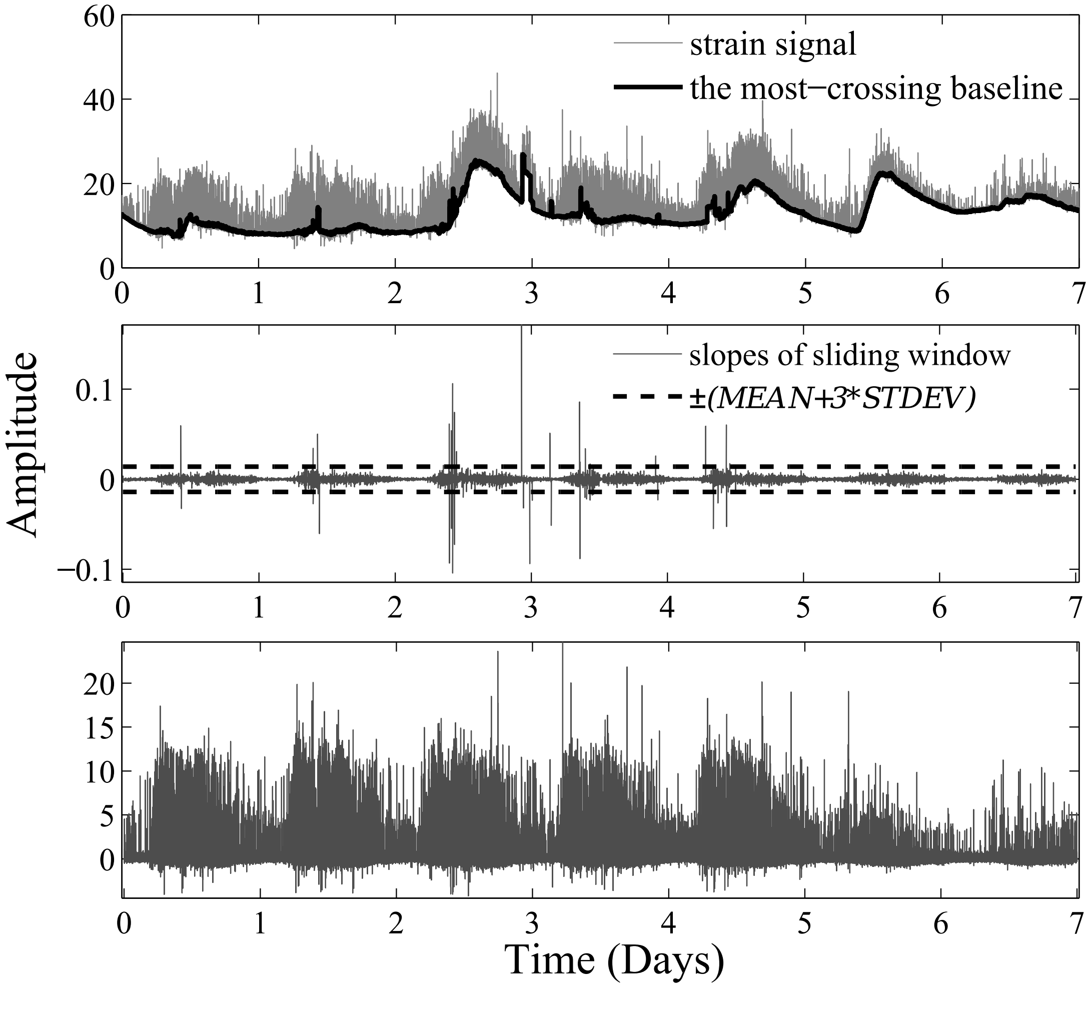

Figure 5.14: Traffic event statistics of 7 days (During the period between Dec 8, 2008 and Dec 14, 2008) - Top picture: the strain signal of 7 days at 1 Hz and its baseline obtained with the most crossing method. Middle picture: the slope values of adjacent sliding windows (with length 60 data points) 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.