Master Thesis

Tor Holm Slettebak

June 8, 2016

Summary

Write your summary here...

- Theory
- Method
- Analysis
- Conclusion

Preface

Write your preface here...

Contents

List of Figures

List of Tables

1. Introduction

Random citation [1] embeddeed in text.

1.1 Background

The Norwegian railway network covers large distances of Norway where sea, mountains and rivers causes the need of a large number of bridges. There is over 3000 railway bridges in Norway [6], many of which were built in the period 1900 - 1950, meaning many bridges are around 100 years old, meaning they are closing in on their designed lifespan, and is built with old methods and steel. The railway is in constant evolution, and over time the train velocities has increased as well as traffic density. With lifespans of around 100 years a steel railway bridge needs properties to withstand weather and continous loading. This means continous inspections of bridges are required. Every sixth year Norwegian bridges are subject of a major inspection, for uncovering corrotion, and other damages of fatigue. This is a process demanding time, resources and manpower. Therefore good estimates of traffic impact on older and newer bridges are a necessity.

Bridge weigh-in-motion technologies was first developed in the USA in 1978. The initial system consisted of strain sensors placed beneath the bridge and sensors beneath the road. But systems using only strain sensors have also been developed. The general principle of a BWIM system is that a vehicle's axles induce strain in the bridge proportional to the influence ordinate and the magnitude of axle load. Thus from knowing the influence line for a sensor location and the measured strain the axle weights can be calculated. In both road traffic and railway, static scales have been used to determine a vehicle or trains weight. The static nature of such a system requires that the vehicle stands still, which for traffic flow and general inconvenience for both the people performing the weighing and the people driving the vehicles. Likely because of this interest for developement of weight in motion technologies has increased over the years [SOURCE?].

Bridge weigh in motion, gives the abilty to determine traffic flow over a certain point, which may be more useful for the unregulated road traffic. BWIM gives the ability to monitor weight of trains, and thus to detect possible overloading of trains. The BWIM system should ideally be implemented so that it provides a continous data flow and automatic detection of trains and calculations of axle weights. This would provide information of bridge behaviour for different types of trains, different loads, and weather conditions. It also will provide data describing dynamic effects on the bridge. This data could be used to find the optimal crossing velocities for different train types. A permanent system providing continious data, could measure changes of bridge property over time, making it a bridge health monitor. Changes in the system could be detected without a major inspection. A BWIM system coult in theory detect internal changes of a bridge, which could go undetected by a visual inspection. B-WIM traffic data including vehicle loads and traffic density can be combined with degradation data to estimate how traffic density affects the aging of a bridge. A BWIM system could over time provide us with estimates of what future bridges spanning similar crossings will be subjected to. BWIM will provide research data for bridge construction.

Generally BWIM systems have been used for road bridges and many different systems have been developed and put to use in both Europa, USA and Australia. For railway bridges this is not the case, according to [2] only Liljencranz [4] and one other implementation of BWIM for railwaybridges has been made. Compared to road traffic, a railway bridge has constant properties making it suitable for BWIM systems. The trains of course always follows the same track on a single track bridge, thus a BWIM railway system don't need to make special considerations for transversal effects varying from train to train. Also single track railway bridges means that the BWIM system doesnt need to account for multiple vehicle events, but instead needing capabilities to cope with a large number of axles, and long strain signals.

1.2 Research objectives

The main goal of this master thesis is to develope and investigate methods of calculating influence lines for steel railway bridges. As well as to investigate how a BWIM system will work for steel railway bridge. To accomplish this a BWIM program has been developed by author using the script language Matlab, because of it's extensive math libraries, plotting abilities, toolboxes and simplicity which suits an early developement phase.

The goal of this master project.

- Implement a working BWIM system
- Implement methods for calculating the influence lines for a arbitrary sensor location.
- Identify good practices for building a BWIM system.
- Analyse how Bridge weigh in motion works for a typical Norwegian steel railway bridge, through measurement data from Leirelva bridge.

2. Theory

2.1 Bridge Weigh-in-Motion

A Bridge Weigh-in-Motion system is based on measurements of a bridge's deformation. The BWIM system uses these measurements to calculate passing vehicles axle loads. There are different approaches to assembling such a system, but they typically consists of a strain gauge measuring the strain induced by passing vehicles, a axle detector used to find the vehicle speed and spacing of axles and a computer or data storage device. An algorithm then is able to use the data gathered from the axle detector and strain gauge to calculate axle loads [8].

2.1.1 Moses' Algorithm

Moses' algorithm is based on the fact that a moving load along a bridge will set up stresses in proportion to the product of the value of the influence line and the axle load magnitude. The influence line being defined as the bending moment at the point of measurement due to a unit axle load crossing the bridge [8].

Moses' algorithm is built from the fact that a moving unit load on a bridge will induce stresses proportional to the product of the value of the influence line and the axle load magnitude.

Each individual girder's stress is related to moment:

$$\sigma_i = \frac{M_i}{W_i} \tag{2.1}$$

Expressing the moment in terms of strain gives

$$M_i = W_i \sigma_i = E W_i \varepsilon_i \tag{2.2}$$

Where:

σ_i = the stress in the i'th girder

M_i = the bendind moment in the i'th girder

W_i = the section modulus

E = The modulus of elasticity

ε_i = strain in the i'th girder

The sum of the individual girder moments is therefore:

$$M = \sum_{i=1}^{N} M_i = \sum_{i=1}^{N} EW_i\varepsilon_i = EW \sum_{i=1}^{N} \varepsilon_i \tag{2.3}$$

The sum of the girder strains is proportional to the gross bending moment. The total bending moment and the measured strain is thus directly related by EW. These constants can be calculated through the bridge's dimensions and material properties. However through measuring the effects of a known vehicle passing the bridge these constants can be derived.

Weigh in motion is an inverse type problem, the strain is measured and the cause of the strain is to be calculated. The theoretical bending moment corresponding to axle loads on the bridge at one strain sample, is given by:

$$M_k^T = \sum_{i=1}^{N} A_i I_{(k-C_i)} \tag{2.4}$$

$$C_i = (L_i \times f)/v \tag{2.5}$$

Where:

N = *the number of vehicle axles*

A_i = *the weight of axle i*

I_{k-C_i} = *the influence line ordinate for axle i at sample k*

L_i = the distance between axle i and the first axle in meters

C_i = *The number of strain samples corresponding to the axle distance* L_i

f = the strain gauge's sampling frequency, in Hz

2.2 Influence lines

A influence line can be defined as: "A graph of a response function of a structure as a function of the position of a downward unit load moving across the structure." For a BWIM system this response function typically is the bending moment at the sensor location. The influence line could be found through assembling a model of the bridge in any CAD or frame-program, this would however take a lot of time especially for more advanced bridge's. Depending on the support of the bridge the influence lines takes different theoretical forms, as seen in Figure 2.1. The true influence line for a bridge lie somewhere in between the simply supported and fixed version [7, p. 146].

Znidaric and Baumgärter [7], did a study on the effect of choice of influence line. This study shows errors up to 10% for a short 2 m bridge span and errors of several hundred percent for a 32 m bridge span, illustrated by figure 2.2 showing how a veihicles gross weight is affected when the influence line is varied from a simply supported version to a fixed support version. This underlines the importance of using correct influence lines for a B-WIM system.

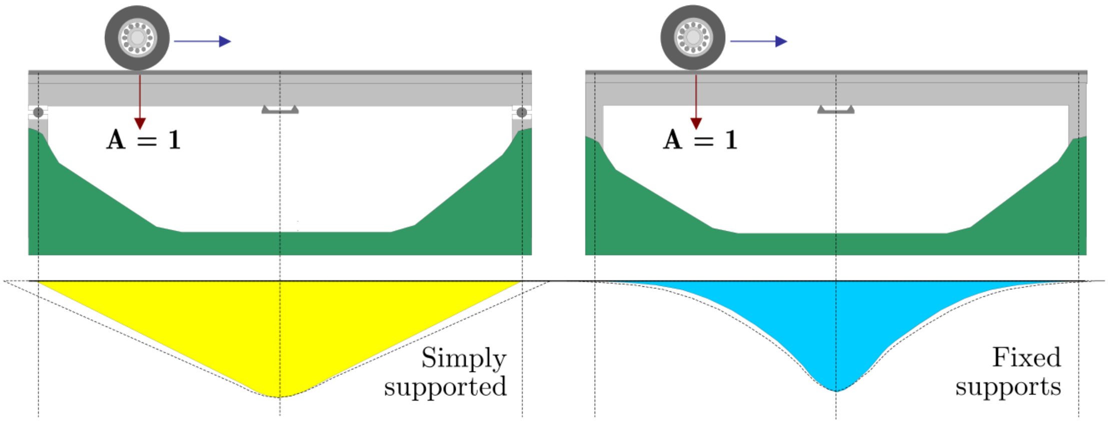


Figure 2.1: Influence lines for simply and fixed supported bridges, figure from [8]

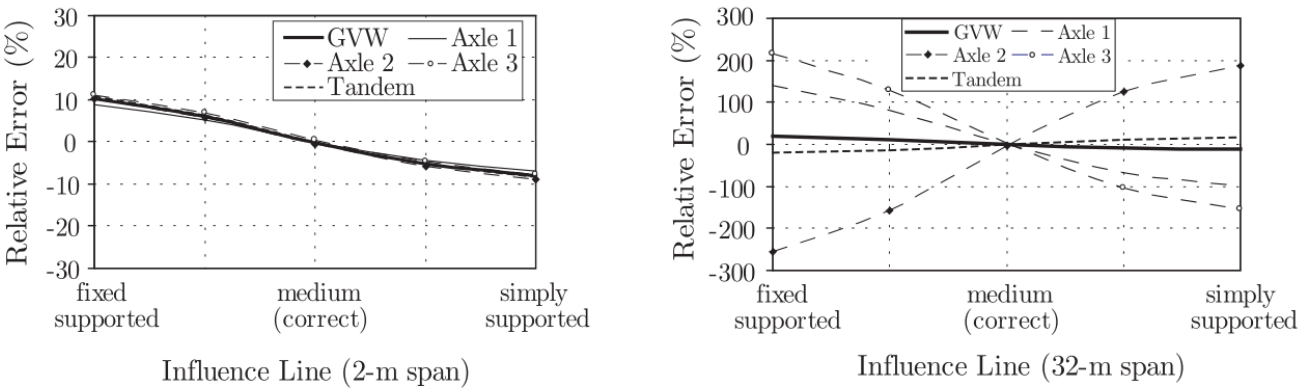


Figure 2.2: Errors of axle loads due to wrongly selected influence lines, figure from [8]

2.2.1 Using influence lines, in the BWIM system

Even if a correct influence line for a BWIM setup is found, wrong placement of the influence line with respect to the strain signal is a major source of error. In theory it should be possible to detect the excact point of a axle passing over the sensor, as it results in a peak in the strain signal. This peak corresponds to the major peak in the influence line. A good example of this is seen in figure 2.3, which shows the influence line aligned with the strain signal from a 3 axle vehicle. The first peak of the strain signal corresponding to the the first axle of the vehicle should occur at the same location as the the peak of the influence line, which should be precisely at the sensor location. For closely spaced axles it may be difficult to detect the individual peaks, because the both influence the sensor at the same time and because of system noise and dynamics.

2.2.2 Influence line through the Matrix Method

Quilligan [8] developed a 'matrix method' to calculate the influence line of a bridge through the measured strain induced by a vehicle. This method is derived from Moses', equation 2.6. The

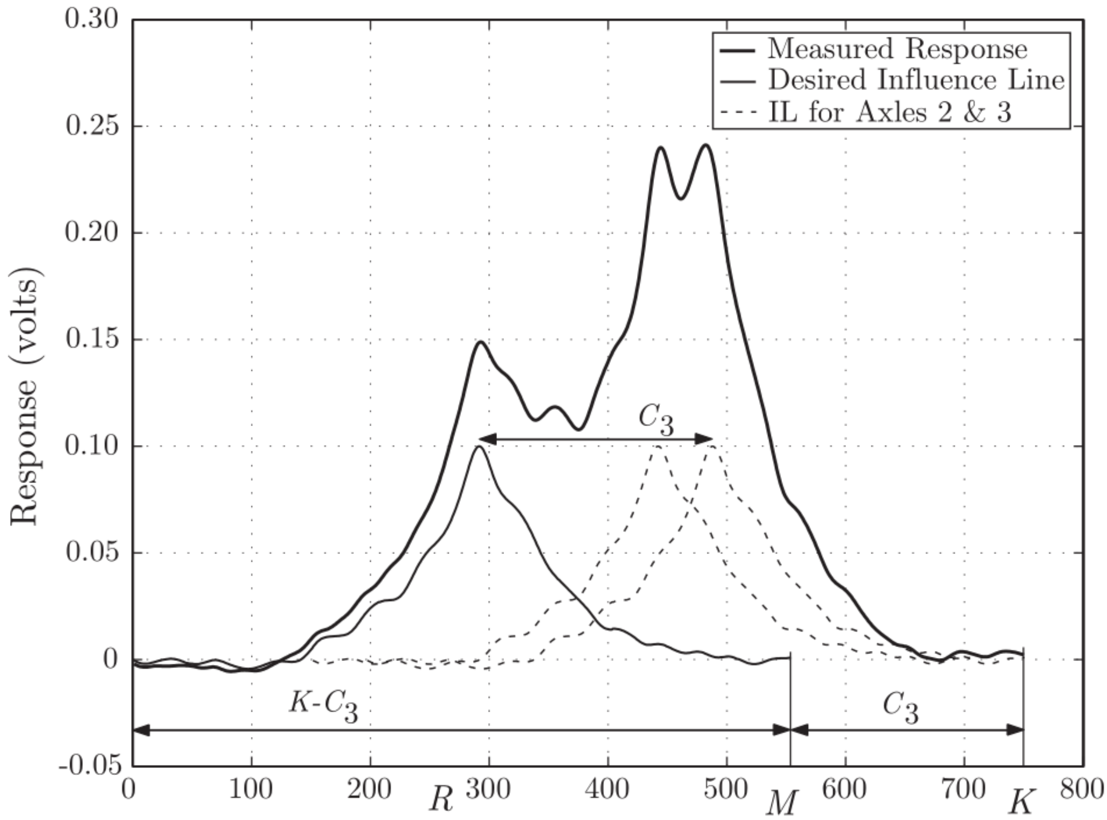


Figure 2.3: Placement of influence lines, influence line has been scaled.

matrix method calculates a influence line for a specific strain signal given a known train with known axle weights and velocity. The found influence line is therefore subject to system noise and dynamics which are likely to vary from vehicle to vehicle. An averaging of a sufficient number of calculated influence lines should reduce the dynamic effects. The following description of the matrix method is an extension of Quilligans thesis "Bridge Weigh-in Motion : Development of a 2-D multi-vehicle algorithm [8]", and shows the math for a general case with unlimited number of vehicle axles.

$$Error = \sum_{k=1}^{K}[\varepsilon_k^{measured} - \varepsilon_k^{theoretical}]^2 \tag{2.6}$$

Equation 2.6 were originally used to filter out the dynamic response of the bridge. The theoretical strain in this equation can be expressed as a product of axle loads and influence ordinates at sampling points, see equation 2.4, thus we can expand equation 2.6:

$$Error = \sum_{k=1}^{K}\left[\varepsilon_k^{measured} - \left(\sum_{i=1}^{N} A_i I_{(k-C_i)}\right)\right]^2 \tag{2.7}$$

The set of influence ordinates I that minimizes $Error$, forms the wanted influence line.

$$\frac{\partial Error}{\partial I_R} = \frac{\partial \sum_{k=1}^{K}\left[\varepsilon_k^{measured} - \left(\sum_{i=1}^{N} A_i I_{(k-C_i)}\right)\right]^2}{\partial I_R} \tag{2.8}$$

For a given number of known axle loads this equation comes down to a set of $(K - C_n)$ number of linear equations. Rearranging the equations and writing them in matrix form leads to:

$$\Big[Am\Big]_{K-C_N,K-C_N} \Big\{I\Big\}_{K-C_N,1} = \Big\{M\Big\}_{K-C_N,1} \tag{2.9}$$

Where:

$\Big\{M\Big\}$ = a vector depending on axle weights and measured strain, $M_{i,1} = \Big(\sum_{j=1}^{N} A_j \varepsilon_{(i+C_j)}\Big)$

$\Big[Am\Big]$ is a matrix depending only on the axle loads, defined by equation 2.10.

$$\Big[Am\Big] = \sum_{i=1}^{N}\sum_{j=i}^{N}\Big[Am\Big] + (A_i A_j \Big[D\Big]_{C_j - C_i}) \tag{2.10}$$

Which produces the upper triangle of the symmetric $\Big[Am\Big]$ which through the transpose operation can be used to build the full matrix. Where:

$\Big[D\Big]_{C_j-C_i}$ = a matrix containing only one diagonal of ones, where the diagonal is placed with an offset, $C_j - C_i$, from the center matrix diagonal.

Solving equation 2.9 for the influence ordinate vector gives the influence line for the strain history. This can be done through inversion of the $\Big\{M\Big\}$ (equation 2.10) or other numerical solutions like a Cholesky factorization. In this project this was done through Matlab's $\backslash$operator. When the influence line and the axle spacings are known, the axle weights can be calculated by solving

$$A = \Big\{I\Big\} \backslash \epsilon \tag{2.11}$$

2.2.3 Influence line through Optimization

testing [4]. For this thesis one of the goals where to assess the accuracy of and optimization algorith for finding a bridges influence lines. To develope such an algorithm test strain signal where produced by a matlab script, and the algorithm developed was to find the influence line used to produce the strain signal. In theory using optimization to identify influence lines should work well, and indeed it did for these produced theoretical strain signals.

The method of optimization developed for this project, utilized matlabs "fmincon" function which finds the minimum value of a constrained function. The function chosen to minimize was Moses' equation 2.7.

1. create initial guess of influence line
2. use initial guess values as input to fmincon
3. for each iteration of fmincon, a new influence line is built from the optimized values
4. The found influence line forms a influence ordinate matrix, based on known axle spacings
5. equation 2.7, is calculated resulting in an error value
6. steps 3 - 5 is repeated until either function tolerance is reached, or other tolerance values is exceeded.

2.3 Finding the train's speed

- By identifying a peak representing the same axle in the strain signals for two different sensors. The time difference between two such peaks is the time the train uses to travel the distance between the two sensors. Given the known distance between the sensors, s, the velocity is given by $v = s/t$.
- Through doing cross correlation between two sensors strain signals. Cross correlation measures the similarity between two signals as a function of the lag. This can be used to identify the lag between two similar signals. The cross correlation of two signals has maximum value at the lag equal to the delay. The time delay is then a product of the sampling frequency and the lag in samples.

2.4 The axle distances

Identifying axles in the strain signal is done through identifying standalone peaks, however due to system noise and dynamic effect finding excact positions of axles in the signal

2.5 Filtering and noise

All signals are subjected to noise, which can be defined as

> unwanted disturbances superposed upon a useful signal that tend to obscure its information content [9]

Noise in a BWIM system can be intrinsic noise, that is noise generated inside a system, and extrinsic noise which is noise generated outside the system. A train approaching the BWIM sensors may be a source of extrinsic noise. Performing bridge weigh-in motion relies upon the information provided by the sensor signals. When the distances between axles is to be found, noise is a source of distortion which may increase error of found distance, it may also make it difficult for the program to detect the desired peaks in the signal which corresponds to the trains axles. Smoothing the signal may therefore completely necessary for a BWIM system. During the developement of the software for this thesis, several attempts on finding and using appropriate filters have been made. Matlab contains many such filter functions which can be used, such as a Butterworth and SGOLAY filters.

2.5.1 Noise smoothing through fourier transformation

The following quotation from Matlabs: Practical Introduction to Frequency-Domain Analysis, see [5], describes how frequency analysis can be done with Matlab.

> Frequency-domain analysis shows how a signal's energy is distributed over a range of frequencies. A signal can be converted between the time and frequency domains with a pair of mathematical operators called a transform. An example of this is the Fourier transorm which decoposes a function into the sim of a number of sine wave frequency components. The 'spectrum' of frequency components is the frequency domain representation of the signal. The inverse Fourier transform converts the frequency domain function back to a time function.

Performing a fast fourier transformation in matlab on a vector signal, gives the oportunity to remove unwanted frequencies from the signal. When the signal is transformed into the frequency domain, setting all the frequencies above 30 Hz to zero and then transforming the signal back into the time domain would smooth a typical BWIM signal greatly. Figure **??** shows filtering of a strain signal where frequencies above 20 Hz have been eliminated.

2.5.2 Noise smoothing using a Butterworth filter

A low pass Butterworth filter, only allows low frequencies of a signal meaning the high frequencies being treated as noise.

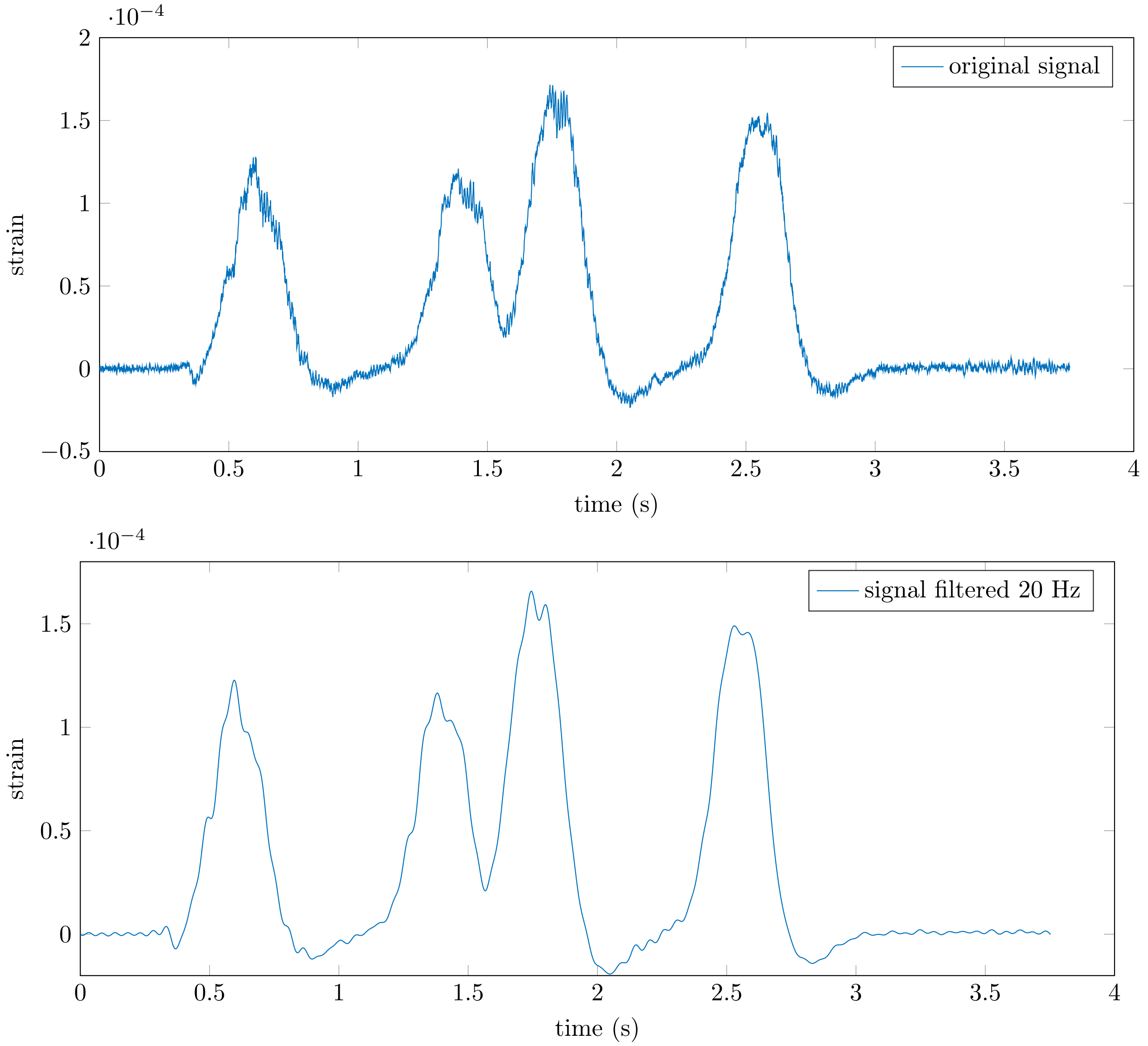


Figure 2.4: Figure showing filtering of a signal, where frequencies above 20 Hz in the signal have been eliminated

3. Method

3.1 Programming a BWIM system

This master project began by learning how a BWIM-system works, and to then create a working model performing BWIM. To make this as simple as possible building a simple beam model of a bridge in Matlab, and simulate moving loads crossing it. Matlab was chosen as the developement languague for the following reasons:

- Matlabs excellent plotting properties
- Simplicity
- Good tools for analysing and debugging the code
- it's large library of toolboxes and functions.

This master project bagan with developing a understanding of how a BWIM system works, the math behind it and looking into how others have devloped such systems. When a sufficient understanding of this was reached work started on building a Matlab program capable of performing BWIM. The model of developement was based on a simple beam model as the bridge with moving loads crossing the longitudinal direction of the beam simulating a passing train like shown in figure 3.1. This simple beam model was used to develope and validate the BWIM algorithm.

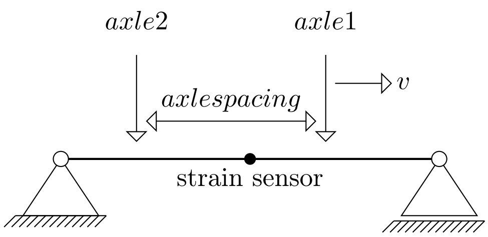


Figure 3.1: Beam model for developement

A simple flow diagram describing the intial BWIM program:

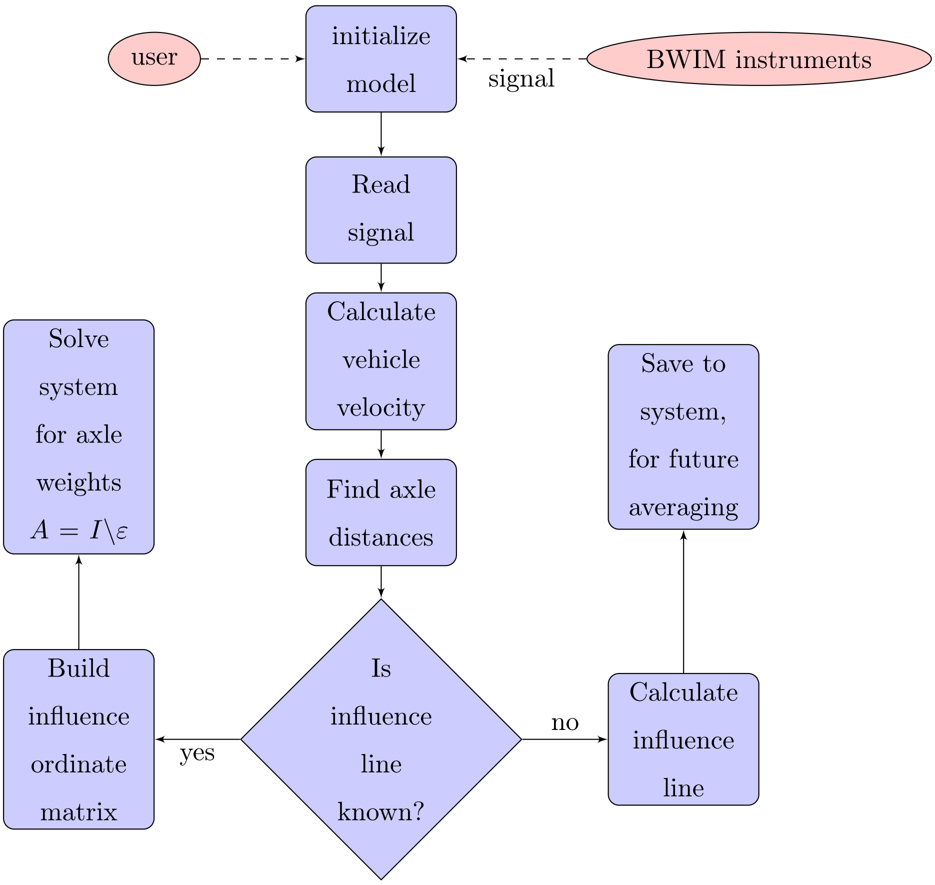


Figure 3.2: Flow chart describing a BWIM system

3.1.1 Producing a strain signal

Through the theoretical moment influence lines of the beam, a strain signal can be built through the moment-strain relationship, found in equation 2.3, for a given set of axle weights. A simple beam bridge model, as seen in figure 3.1, will not recreate a actual bridge strain signal but will be used to create a working BWIM system. The produced strain signal will differ from an actual strain signal mostly because of dynamics, from the train and bridge, and because actual boundary conditions of a bridge will differ from the boundary conditions of a simple beam model. The strain sensors will also produce noise distorting the signal. To make as good a signal as possible, some effort were placed into recreating the effect mentioned above. To add noise to the signal, white gaussian noise was included in the signal through Matlabs wgn function "http://se.mathworks.com/help/comm/ref/wgn.html". Such a produced signal can be seen in figure 3.3, which is produced by 8 axles moving across the bridge at 20 m/s.

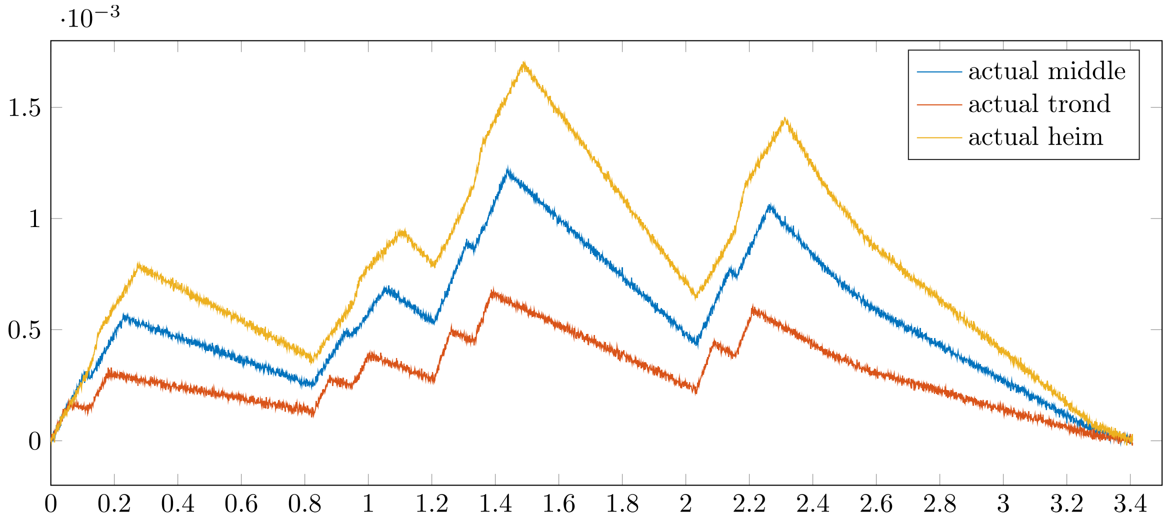


Figure 3.3: Strain signal created through beam model

This theoretical strain signal will vary greatly from an actual strain signal measured on a bridge. The beam model used to develope this strain signal, may be comparable with simple bridge types like a single span integral bridge, but will not be comparable with more complex bridges. However for developing a BWIM program the simple beam model will suffice as the different modules of BWIm will be roughly the same no matter the bridge type.

3.2 Leirelva bridge

The department of "Konstruksjonsteknikk" had access to Lerelva bridge for test purposes, which allowed installing sensors and using equipment on the underside of the bridge. Leirelva bridge is a typical Norwegian railway bridge built in 1921. It is a simple 25 meters steel truss bridge consisting of 5 verticals dividing the stringers into 6 sections. These stringers consist of angle profiles and plates built with the riveting teqnique. This bridge is of of particular interest for a BWIM system because few or none have installed and tested such a system on a bridge of this type. This type of bridge is typical for the norwegian railway and thus if a BWIM system could be proved to work on one such bridge it would easily be adapted to similar bridges.

3.3 System setup

To test the BWIM-program on actual data, a BWIM-system to gather strain data from actual train passings. The subject bridge were Lerelva-Bridge in Trondheim, figure 3.5, a typical Norwegian steel railway bridge. Three strain gauges, 3 mm 120 ohms from HBM, were placed by the support towards Trondheim on the first section of the longitudinal stringer, like shown in figure 3.4b and 3.6. The sensors were placed with 1 m spacing around the middle of the stringer section. These strain gauges were connected to a National Instruments compactDAQ with module

NI 9235 which produced an continuous data flow to a standard laptop, see figure 3.4a. A Kipor generator was brought for power.

(a) System setup from data gathering at Lerelva (b) Placement of strain gauges on stringer section

Figure 3.4: Instruments for aquiring strain data

Figure 3.5: Lerelva bridge with a train passing over

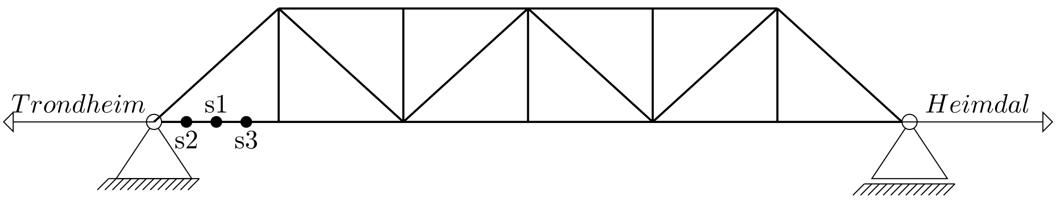


Figure 3.6: Sketch of bridge showing sensor locations for system setup at Leirelva bridge

3.4 Data gathered

The instruments discussed in 3.3, provides a measurement frequency of 1024 Hz. All data was gathered the same system setup during a single day. In all six trains were recorded passing the bridge. The system setup stored the signals for the three different sensors along with the time for the elapsed signals in a matrix for each train. The recordings of train passings were started and ended manually as no trigger was in place to start and end the signal. Some of the gathered signals therefore ended up being very long, which means it requires to have essential data extracted. One of the reccordings was also very short, but still usable. Three of the trains travelled towards Heimdal, and three towards Trondheim.

3.5 Trains

The trains recorded passing the bridge where of two types, a short two vagon commuter trains of type NSB92 as seen in figure3.7 and a freight train with a EL14 locomotive as seen in 3.8. The weight of the trains with passengers is unknown, resulting in axle weights being set equal the distibuted weight of the vagon. For the freight train the properties of the locomotive was found through [3].

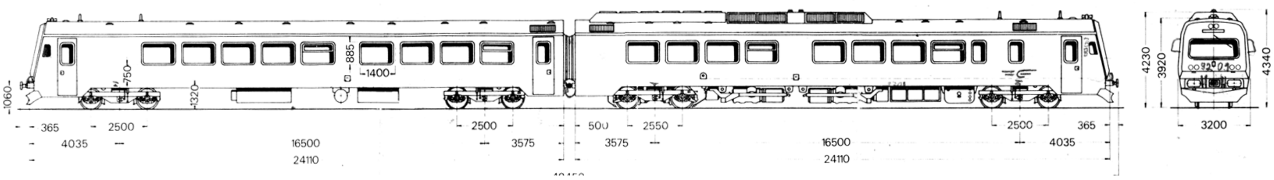


Figure 3.7: Axle distances of a NSB92 train

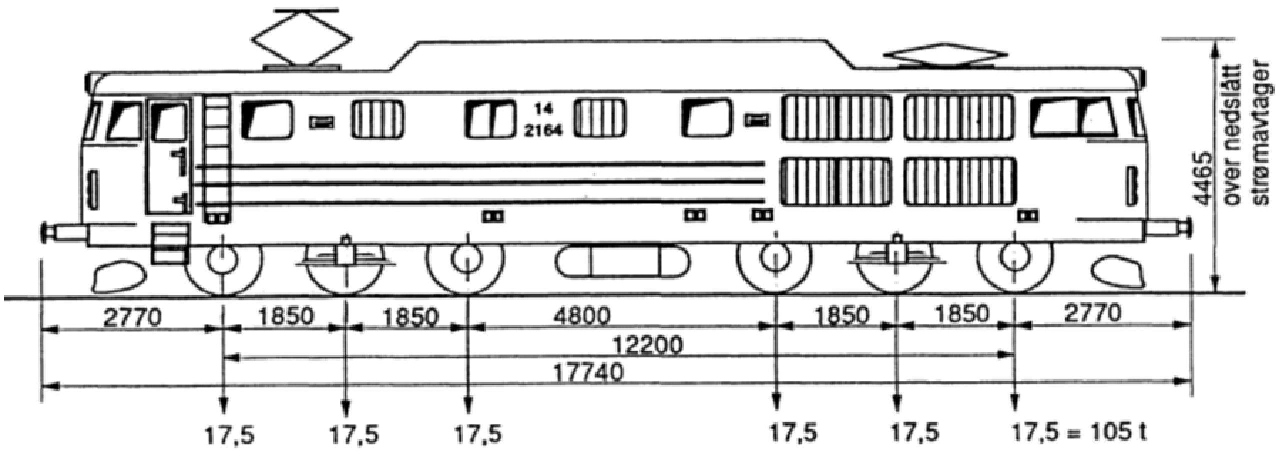


Figure 3.8: Axle distances and weights for a EL14 locomotive

4. Analysis

This chapter will analyse the how the developed BWIM system performs, with special emphasis on the influence lines produced by the Matrix method, and how they represent the bridge and trains. Sensor locations are as shown in figure 3.6, and is the source of the naming conventions in the plots in this chapter. Should include:

- Compare theoretical and calculated influence lines. Also include influence lines found through Abaqus.
- Check how influence lines found through matrix method and optimization reproduces the strain history
- Test obtained influence line by running the bwim routine on the hitherto unused freight train. (Depends on getting info about the train). Also Do this test on the other trains.

4.1 Strain data

The following figure 4.2, contains raw strain data for 6 different trains passing the Lerelva bridge. Each subfigure contains data from three different strain sensors placed as described in System setup, see section:3.3. Three of the trains comes from the north side; train 3, train 5 and train 7, and three from the south side; train 4, 6 and 8. The strain signals all appear similar in form, except for train 7, figure4.2e, which is a freight train. The other 5 trains are all of the same type, a NSB 92 type passenger train 3.7.

The strain signals have different levels of peak height suggesting that the trains actual axle weights differ from what is found in table 4.2. This will throw off the magnitude of the resulting average influence line found through the matrix method. And this error in calculated influence line will inevitably be found again in the calculated axle weights . To account for the different directions of the trains, the strain data for the trains going towards Trondheim has been reversed (correct word ?). This is not necessary for finding influence lines, but makes it easier placing the found influence lines in the same coordinate system. Some of the signals were originally very long, due to not knowing exactly when the train would pass. This means cutting the signal into a vector containing the essential data. Initially the goal was to identify exactly, or as closely as possible, the time the train entered the bridge. Due to noise and dynamic effects identifying this, proved a difficult process involving detection of peaks which lies close to peaks of noise. This proved possible to do for each individual signal, but a general method performing this for every signal was not within the authors capabilites. Therefore, to cut the signals as equally as possible the first and last major peaks of the signals were used as reference points for appending of samples before and after these peaks, as seen in figure 4.1. For this method to prove exact, the speed of the train should be taken into consideration when appending sample points so that the influence lines of the signals gets an as equal length as possible.

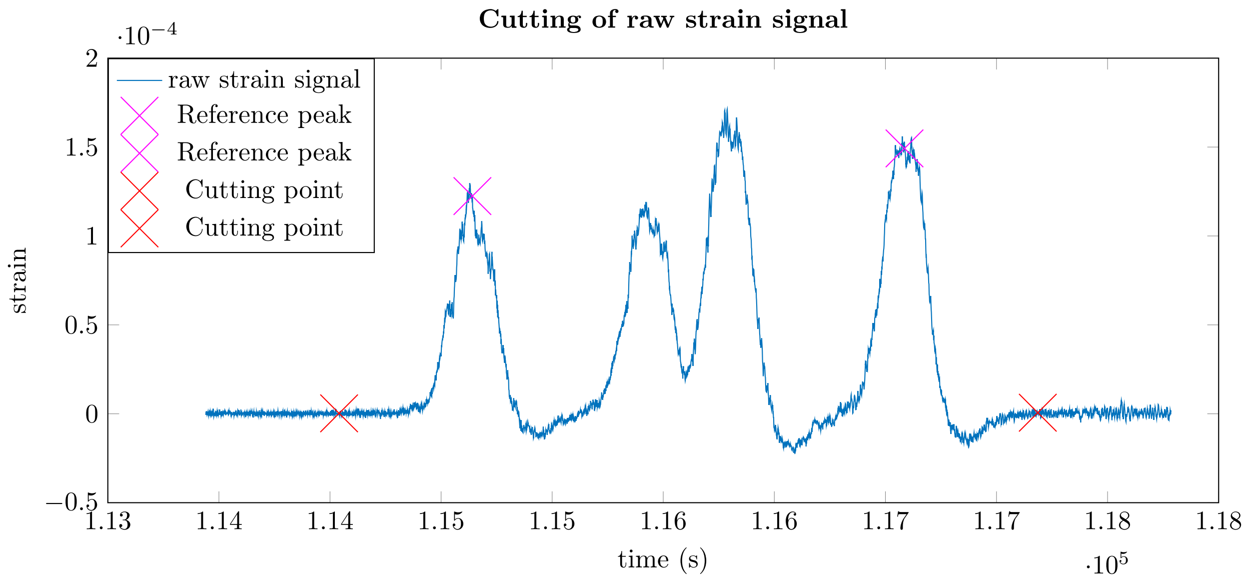
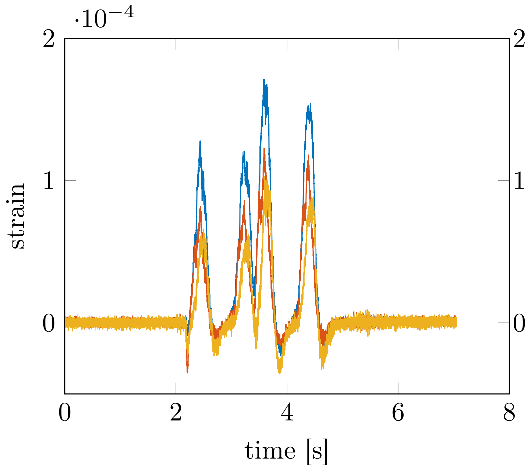
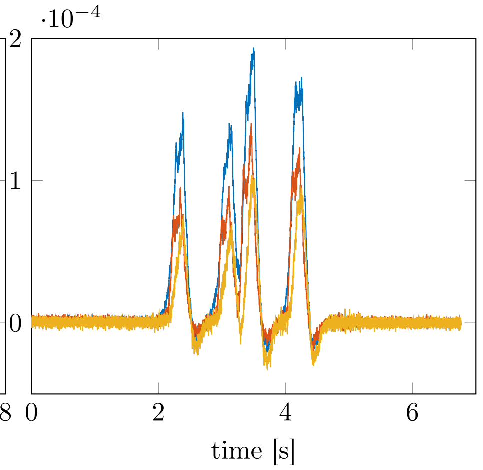


Figure 4.1: Cutting points of strain signals

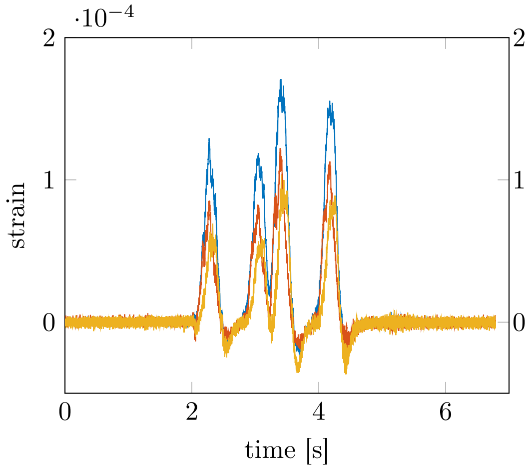
The strain data from the freight train, figure 4.2e, is not used for finding the bridge's influence line because the train data is unknown. Axle weights for this train was not found, and guesswork of this data would be difficult. The properties of the freight trains locomotive is known, and as discussed in 4.9 this could in theory be used to calibrate thes sensors, and to identify errors in the BWIM system.



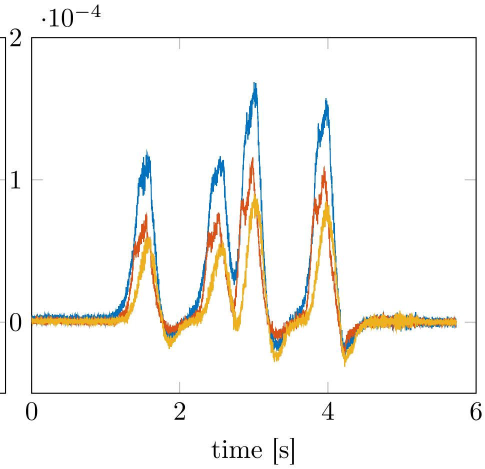
(a) Train 3, towards Heimdal



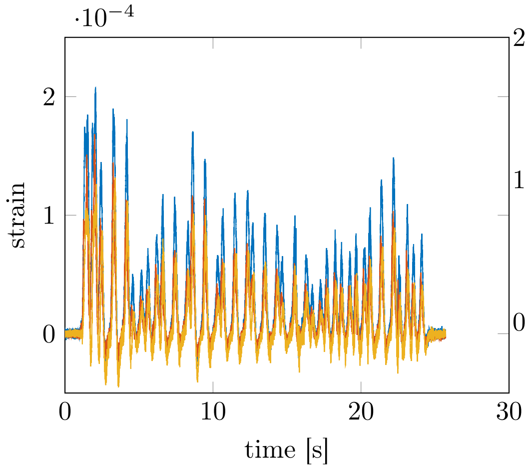
(b) Train 4, towards Trondheim



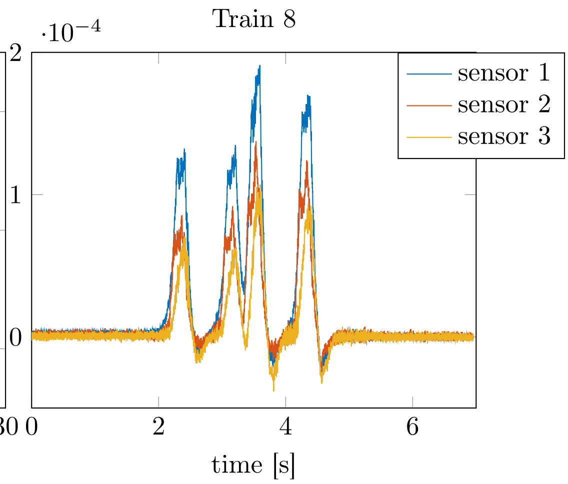
(c) Train 5, towards Heimdal



(d) Train 6, towards Trondheim



(e) Freight train, towards Heimdal



(f) Train 8, towards Trondheim

Figure 4.2: Strain data from Leirelva bridge

4.2 Finding the speed of the train

The importance of using the correct speed in a BWIM system becomes apparent when calculating influence line for a sensor. A wrongly determined speed will result in what looks like dynamic effects or an oscillating influence line, none of which should appear in a static influence line. If the influence line is known incorrect train velocities will still cause wrongly calculated axle weights. A correctly calculated speed is therefore of utmost importance for Bridge Weigh-in-Motion. As discussed in theory there are two ways used by existing BWIM systems to find the train's velocity. Both these methods have been implemented and tested, however they contained flaws making them unreliable, or unsuitable for this project.

- The method of peak identification 2.3, is very subjected to noise corrupting location of identified peaks. A train bogie typically consist of axles in pairs or threes, which will all influence the sensors simultaneously creating a major peak containing smaller peaks. In such a case the identification of a single peak can be difficult, and will likely provide faulty calculated velocity.

- The method of phase difference using cross correlation depends on strain signals where the trains velocities are known and the distance between two or more sensors. This method seems to work independently of noise which likely makes it superior to the peak method. This method will however require calibration for each setup of a BWIM system, due to the method needing a system constant depending on the bridge and the sensor placement. The velocity of the trains producing the strain signals in this thesis, was not known or attainable through NSB or Jernbaneverket and therefore this method were not applicable for finding velocities. Calibrating this method has neither been the focus of this thesis.

These two methods both work very well for a theoretical signal, however when noise and dynamics are introduced as well as more complicated bridge boundary conditions identifying the peaks representing the same axles becomes complex. A method indentifying peaks, will have to adapt to each signal because the magnitude of noise and dynamics vary for the different sensors and train passings. Due to this thesis' focus on the matrix method and influence lines, these methods have not been a priority and since correct train velocities are of utmost importance for calculating influence lines.

Since neither of these methods were usable without calibration, an alternative way was developed. This method determined the velocity by recreating the strain signal, like shown in equation 4.6, for various train velocities and minimizing the difference between measured and recreated signal. It utilizes equation 2.7 and requires constant values of axle weights as well as known axle spacings. The only varying factor is the speed used in each iteration to calculate an influence line. A well suited Matlab function "fminsearch", was used to search for the optimal value of train velocity. "fminsearch finds the minimum of a scalar function of several variables, starting at an initial estimate. This is generally referred to as unconstrained nonlinear optimization." This method uses brute force, and its time consumption proved high. The accuracy of this methods seem reliable due to

The velocities of the trains found through this brute force method is shown in table 4.1, and all plots and results produced have been made using these velocities, except for specificly mentioned cases.

train	3	4	5	6	8
velocity (m/s)	20.99	21.7276	21.4857	16.83	20.591465

Table 4.1: Table of determined train velocities

4.3 Analysis of the influence lines calculated by the matrix method

For the theoretical strain signal for the simple beam model, shown in 3.3, the matrix method calulates a almost perfect influence line. Where the only source of error is likely due to noise, or round off errors. The influence line incorporates the properties of a bridge,

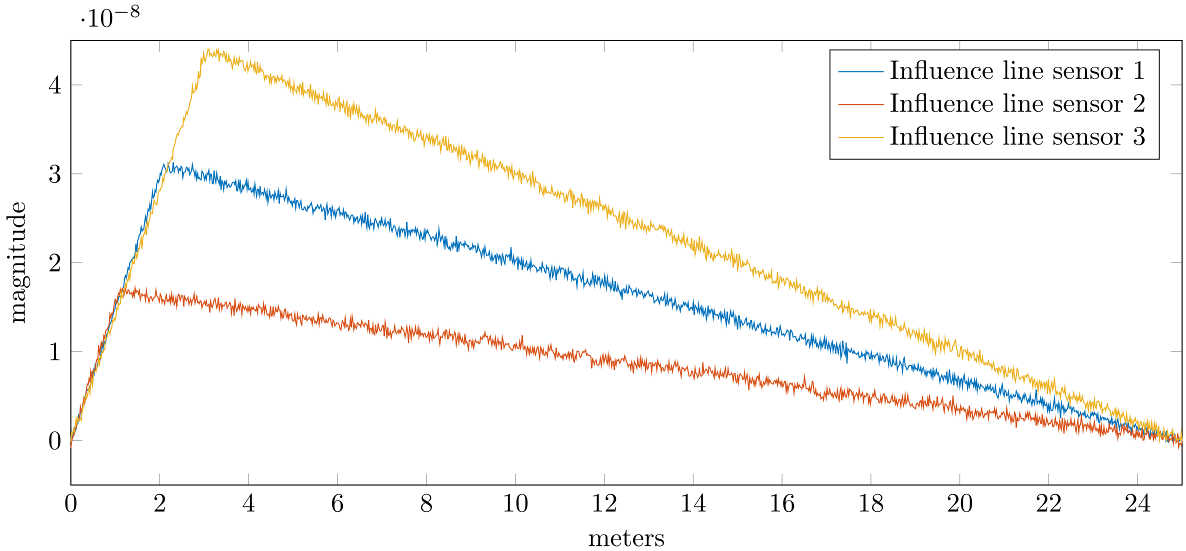


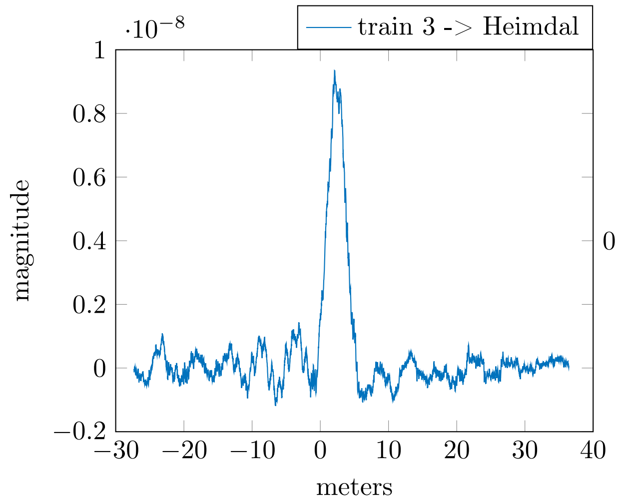
Figure 4.3: Influence line calculated from created strain

The analysis of the matrix method is based on 5 different train passings, and 3 sensor readings on each passing. The trains in these in this analysis is all of the type NSB 92 3.7. The weight of each train axle is not known, and therefore the axle weights have been calculated from the gross weight of the wagon and locomotive like shown in table 4.2. Passenger weight, or number of passengers, was not known and has therefore been neglected. Figure 4.4 show influence lines for 5 different trains passing the same sensor. These influence lines are based on roughly the same

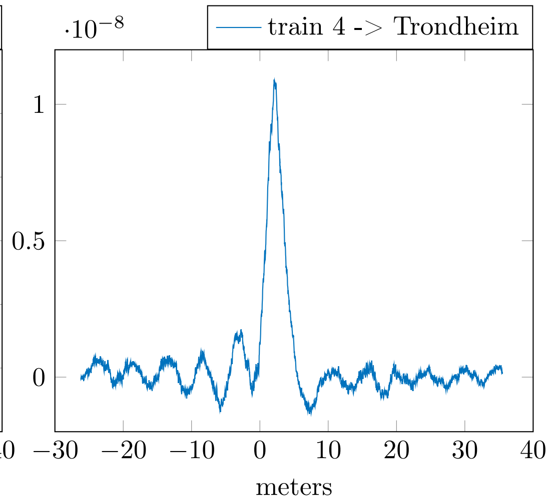
Axle	1	2	3	4	5	6	7	8
Axle weight [kg]	9500	9500	9500	9500	14575	14575	14575	14575
sum	38000				58300			
sum total	96300							

Table 4.2: Table of axle weights used to calculate Influence lines

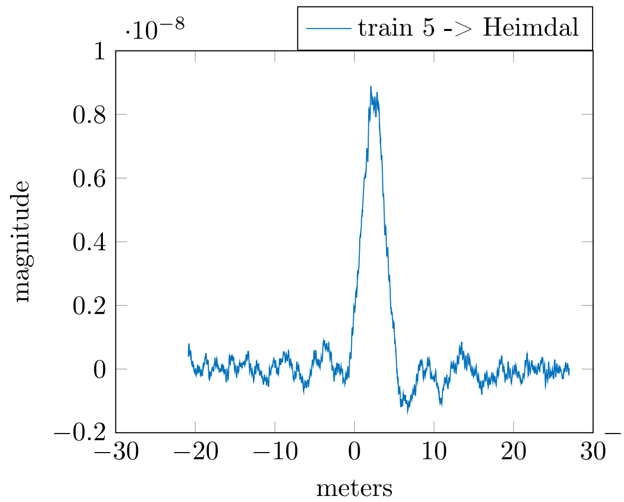
number of sampling points, however due to differing train velocities and that the strain signals does not have equal availability of data, since they sampling was started and ended manually, they may differ a little in length. The influence lines have been placed in a reference coordinate system based on the sensor location. The maximum peak location of the influence lines have been placed at the sensors location.



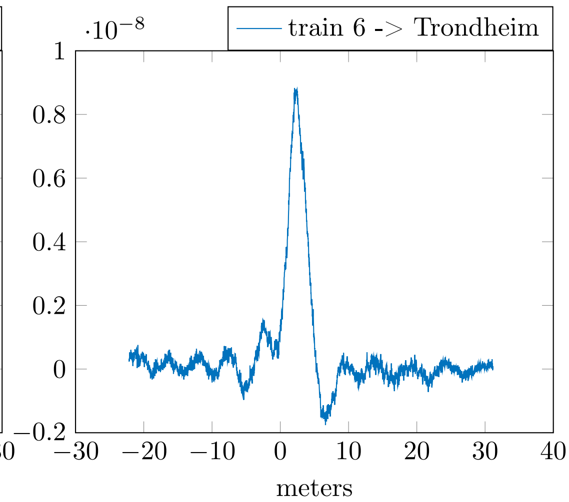
(a) Influence line train 3



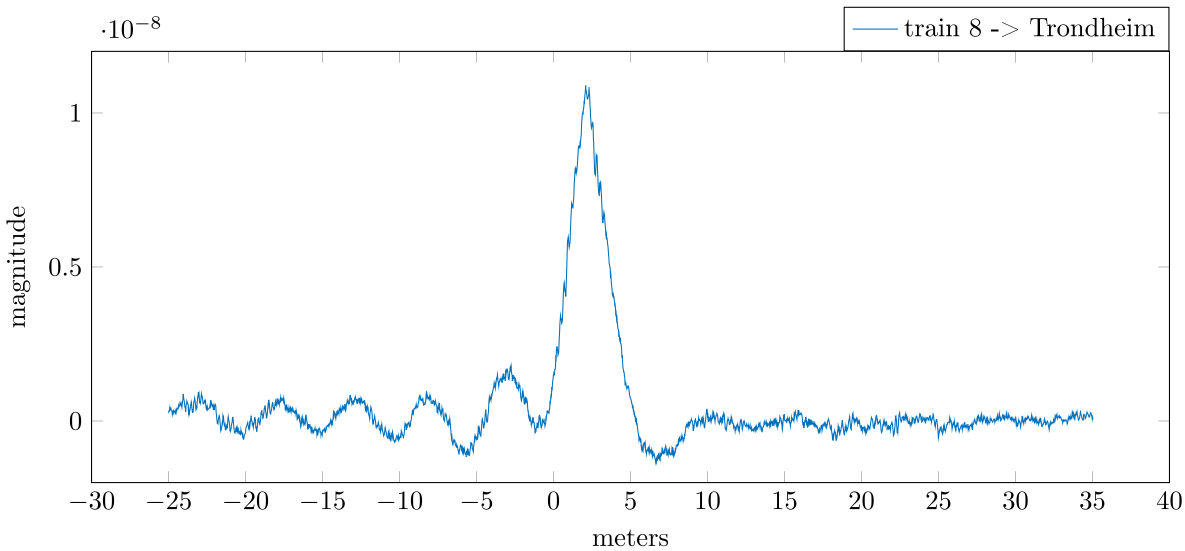
(b) Influence line train 4



(c) Influence line train 5



(d) Influence line train 6



(e) Influence line train 8

Figure 4.4: Influence lines found through the matrix method, for the middle sensor

A qualitative assessment of the influence lines in figure 4.4

- Train 3 and 5 travels in the same direction, and have a no distinct single peak, while train 4, 6 and 8 have more of a singular peak. This may be due to the different directions of the trains.
  - However other possibilities excist such as train velocity inducing different dynamic effects or that the sensor readings are subjected to noisy creating additional peaks.
- The different influence lines displays different values of magnitude, the influence line for train 4 and 8 have a magnitude higher than 1×10^{-8}. This is shown more clearly in plot 4.11, showing all the influence lying over each other.

As plainly seen in figure 4.4 there is big differences between the found influence lines. The trains are all of the same type meaning that the magnitudes of the influence lines, which should be the mostly dependent on axle weights, ought to be similar for all train passings. However as discussed in 4.1, the different magnitudes could be explained with the unknown values of axle weights. When the plots are laid on top of each other, as in figure 4.11, it is clearly visible that there is some variation in peak magnitude. Especially train 4 and 8 have a higher maximum peak magnitude than the others.

4.3.1 Accuracy of the matrix method through recreating the strain signal

One way of examining the accuracy of the matrix method is to recreate the strain signals by assembling the calculated influence lines in the influence ordinate matrix depending on axle spacings, and multiply this matrix with the axle weights vector. Figure **??**, shows how the signal shown in the method chapter 3.3 created for the beam model, have been recreated using the the influence line calculated through the matrix method.

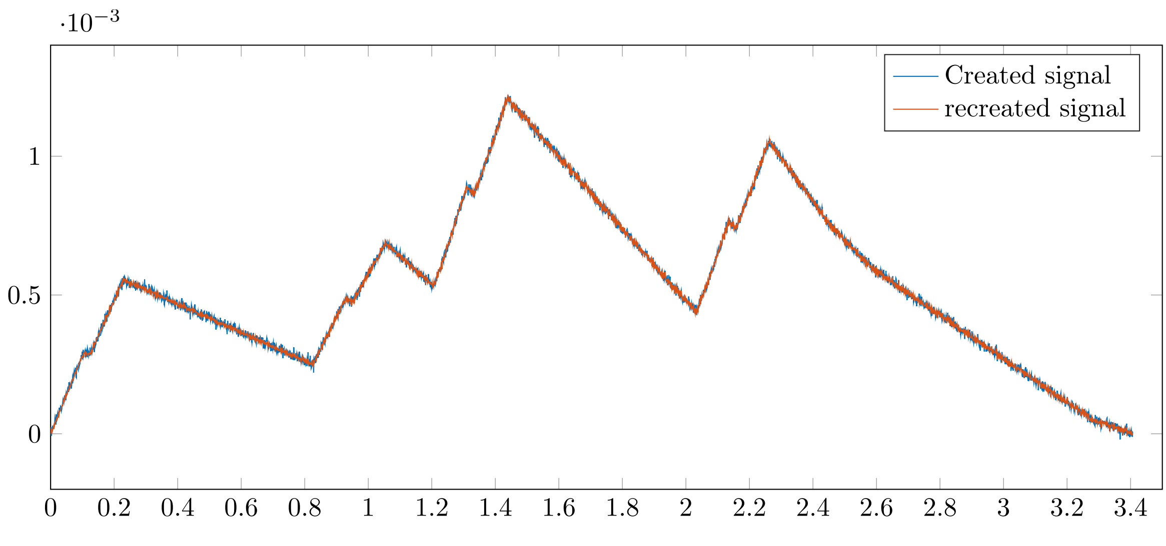
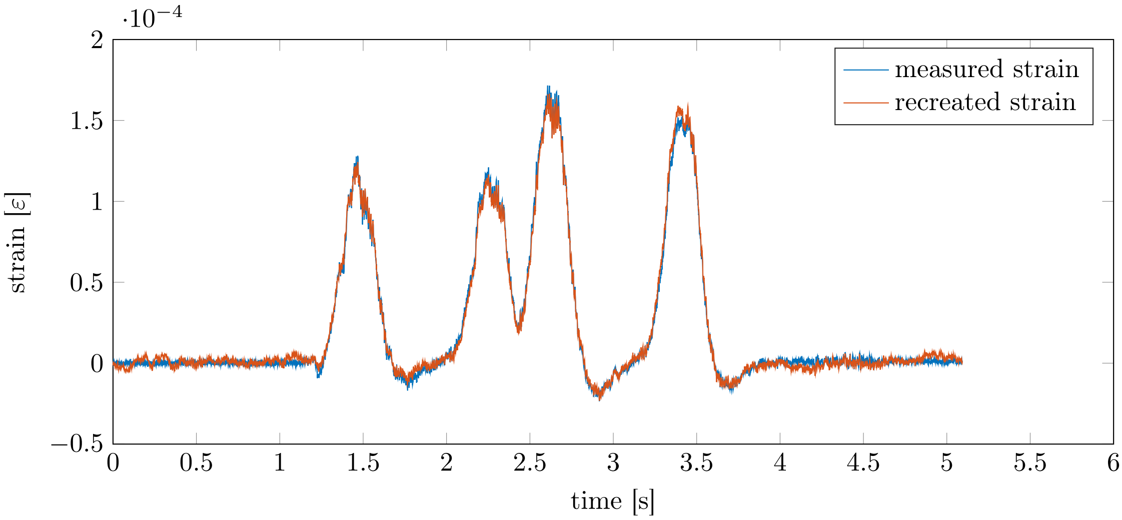
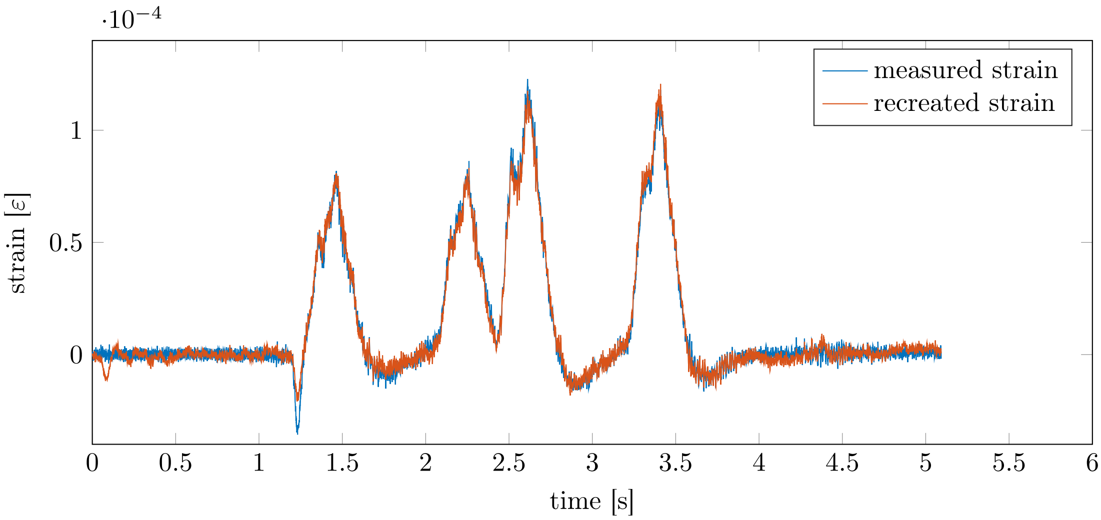


Figure 4.5: Recreated signal shown on top of created strain signal from 3.3

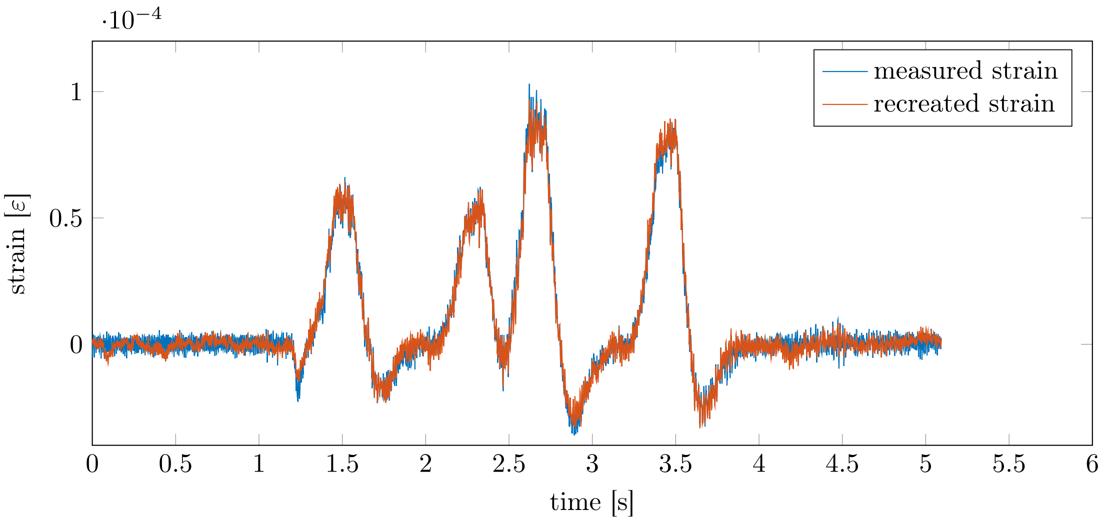
This figure show that the influence line created from a theoretical model bridge, where every property of the train is known, is not able to exactly recreate the strain signal. This is believed to be because of white noise added to the signal.



(a) Recreated strain atop measured strain, train 3 for sensor 1



(b) Recreated strain atop measured strain, or train 3 registered by sensor 2



(c) Recreated strain atop measured strain, train 3 for sensor 3

Figure 4.6: Recreated strain signals for train 3, overlayed measured signal to demonstrate accuracy of the matrix method

Figure 4.6 shows the strain signals from the three different sensors along with a recreated signal using the found influence lines of the sensors, signals for other trains can be found in appendix C.1. The signals being recreated are long and include sections where little or nothing happen during the first and last second of the signal. It is qualitatively difficult to compare the different figures due to the different magnitudes of the strain signals. To identify and compare errors the following equation 4.1, performing least square error will be used.

$$Error = \sum (\varepsilon_{meas} - \varepsilon_{calc})^2 \tag{4.1}$$

The recreated strain signals, see figure 4.6, illustrates the accuracy of the matrix method.

Error table			
	sensor 1	sensor 2	sensor 3
sum squared signal	$1.3207 \cdot 10^{-5}$	$5.2029 \cdot 10^{-6}$	$3.5630 \cdot 10^{-6}$
train 3	$Error = 6.6307 \cdot 10^{-8}$	$6.3778 \cdot 10^{-8}$	$4.6781 \cdot 10^{-8}$
error %	0.50205	1.22582	1.31297
sum squared signal	$1.6646 \cdot 10^{-5}$	$6.8390 \cdot 10^{-6}$	$3.6794 \cdot 10^{-6}$
train 4	$Error = 7.6854 \cdot 10^{-8}$	$3.9514 \cdot 10^{-8}$	$3.4617 \cdot 10^{-8}$
error %	0.46169	0.57779	0.94084
sum squared signal	$1.2888 \cdot 10^{-5}$	$5.0447 \cdot 10^{-6}$	$3.4902 \cdot 10^{-6}$
train 5	$Error = 5.5810 \cdot 10^{-8}$	$3.4720 \cdot 10^{-8}$	$4.0623 \cdot 10^{-8}$
error %	0.43303	0.68825	1.16391
sum squared signal	$1.5975 \cdot 10^{-5}$	$6.1166 \cdot 10^{-6}$	$3.5417 \cdot 10^{-6}$
train 6	$Error = 8.4405 \cdot 10^{-8}$	$4.2931 \cdot 10^{-8}$	$3.6182 \cdot 10^{-8}$
error %	0.52837	0.70188	1.02158
sum squared signal	$1.6782 \cdot 10^{-5}$	$6.7436 \cdot 10^{-6}$	$3.7381 \cdot 10^{-6}$
train 8	$Error = 6.7858 \cdot 10^{-8}$	$3.0772 \cdot 10^{-8}$	$3.4069 \cdot 10^{-8}$
error %	0.40435	0.45632	0.91138
averaged %	**0.46590**	**0.73001**	**1.0701**

Table 4.3: Errors of the recreated strain signals found in 4.6, rounded to four decimals, strain signal cut to include an extra 600 points of the bridge length

As table 4.3 and 4.6, shows the matrix method produces an influence line which recreates the strain signal with very little error. The squared sum of the signals compared to error is very small The error of this recreated strain mostly depends on the accuracy of speed, which decides the sample distance between axles. The averaged errors in the table shows that sensor 3, closest to the middle of the bridge, have the smallest average error. This could indicate that a sensor placement closer to the middle of the bridge resulting less error of calculated influence lines. However many other possibilities may also contribute to this

Error table, filtered signals			
	sensor 1	sensor 2	sensor 3
sum squared signal	$1.3207 \cdot 10^{-5}$	$5.2029 \cdot 10^{-6}$	$3.5630 \cdot 10^{-6}$
train 3	$Error = 8.3869{\cdot}10^{-8}$	$Error = 8.1484{\cdot}10^{-8}$	$Error = 7.8551{\cdot}10^{-8}$
error %	0.63503	1.56614	2.20463
sum squared signal	$1.6646 \cdot 10^{-5}$	$6.8390 \cdot 10^{-6}$	$3.6794 \cdot 10^{-6}$
train 4	$Error = 9.3855{\cdot}10^{-8}$	$Error = 5.6308{\cdot}10^{-8}$	$Error = 5.8075{\cdot}10^{-8}$
error %	0.56382	0.82334	1.57837
sum squared signal	$1.2888 \cdot 10^{-5}$	$5.0447 \cdot 10^{-6}$	$3.4902 \cdot 10^{-6}$
train 5	$6.8249 \cdot 10^{-8}$	$5.2464 \cdot 10^{-8}$	$6.9209 \cdot 10^{-8}$
error %	0.52955	1.04000	1.98296
sum squared signal	$1.5975 \cdot 10^{-5}$	$6.1166 \cdot 10^{-6}$	$3.5417 \cdot 10^{-6}$
train 6	$Error = 9.8692{\cdot}10^{-8}$	$Error = 5.2011{\cdot}10^{-8}$	$Error = 4.7314{\cdot}10^{-8}$
error %	0.61781	0.85033	1.33589
sum squared signal	$1.6782 \cdot 10^{-5}$	$6.7436 \cdot 10^{-6}$	$3.7381 \cdot 10^{-6}$
train 8	$Error = 8.7170{\cdot}10^{-8}$	$Error = 5.0457{\cdot}10^{-8}$	$Error = 6.2777{\cdot}10^{-8}$
error %	0.51943	0.74823	1.67938
average %	**0.57313**	**1.0056**	**1.7562**

Table 4.4: Errors of the recreated strain signals with original signal filtered for noise above 20 Hz, rounded to four decimals and using the same setup as for the previous model 4.3

The differences between the unfiltered and filtered errors, tables 4.4 and 4.5 respectively, are clear but not unexpected. They show that the filtering does not distort the error to an amount which destroys the accuracy of the influence line. The averaged error percentages for the different sensors in the error tables 4.5, 4.4 and 4.3 all show sensor 1 as being better able to recreate the strain signal values. This may be because only the signals from sensor 1 has been used to estimate the train velocities, and that one signal alone may produce a value better suited to that particular signal. Due to the main focus of this thesis being on other areas of BWIM systems, this has not been investigated further than this. Other possible reasons for this observed difference between error percentages are: that some sensor locations are better suited for BWIM, or that the signals with lower values of strain are more susceptible to noise.

To really compare the methods of filtering however the found influence lines should be used to calculate axle weights. Averaging of the influence lines gives the following plots. An interesting discovery by studying these table, is that the longer the produced influence line becomes the more accurately it reproduces the strain.

Error table, minimal influence lines			
	Trondheim sensor	middle sensor	Heimdal sensor
sum squared signal	$1.3205 \cdot 10^{-5}$	$5.1993 \cdot 10^{-6}$	$3.5575 \cdot 10^{-6}$
train 3	$Error = 8.3046 \cdot 10^{-8}$	$Error = 6.9250 \cdot 10^{-8}$	$Error = 5.6861 \cdot 10^{-8}$
error %	0.62891	1.33192	1.59832
sum squared signal	$1.6634 \cdot 10^{-5}$	$6.7644 \cdot 10^{-6}$	$3.6746 \cdot 10^{-6}$
train 4	$Error = 1.0317 \cdot 10^{-7}$	$Error = 5.0548 \cdot 10^{-8}$	$Error = 4.0564 \cdot 10^{-8}$
error %	0.62024	0.74726	1.10391
sum squared signal	$1.2886 \cdot 10^{-5}$	$5.0407 \cdot 10^{-6}$	$3.4850 \cdot 10^{-6}$
train 5	$7.5816 \cdot 10^{-8}$	$4.4896 \cdot 10^{-8}$	$5.1032 \cdot 10^{-8}$
error %	0.58835	0.89067	1.46433
sum squared signal	$1.6308 \cdot 10^{-5}$	$6.3414 \cdot 10^{-6}$	$3.7159 \cdot 10^{-6}$
train 6	$Error = 1.1471 \cdot 10^{-7}$	$Error = 5.0396 \cdot 10^{-8}$	$Error = 4.1867 \cdot 10^{-8}$
error %	0.70340	0.79471	1.12670
sum squared signal	$1.6795 \cdot 10^{-5}$	$6.6767 \cdot 10^{-6}$	$3.7751 \cdot 10^{-6}$
train 8	$Error = 9.2468 \cdot 10^{-8}$	$Error = 3.8699 \cdot 10^{-8}$	$Error = 4.0678 \cdot 10^{-8}$
error %	0.55057	0.57961	1.07752
average %	**0.61829**	**0.86883**	**1.2742**

Table 4.5: Error table for minimal influence lines

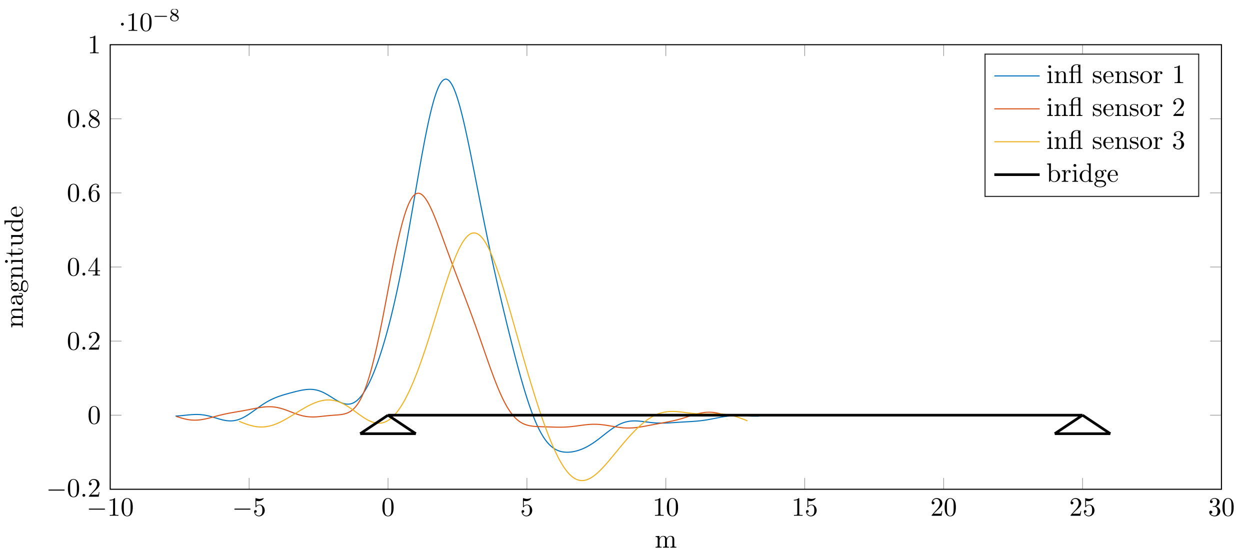


Figure 4.7: Influence lines for the sensors, from minimal strain signal

As figure 4.1 shows, the trains affects the sensor over a 2-3 second period. And the influence of a bogie stops shortly after it has passed the sensor, as the flatness after the last peak indicates. This shows that a bridge of this type will have a very local deformation due to loading. This means that a influence line for a sensor location on a bridge type like this will be short compared with bridge length. Influence lines made with the minimal cutting points can be seen in figure 4.7.

4.3.2 Influence lines from filtered strain

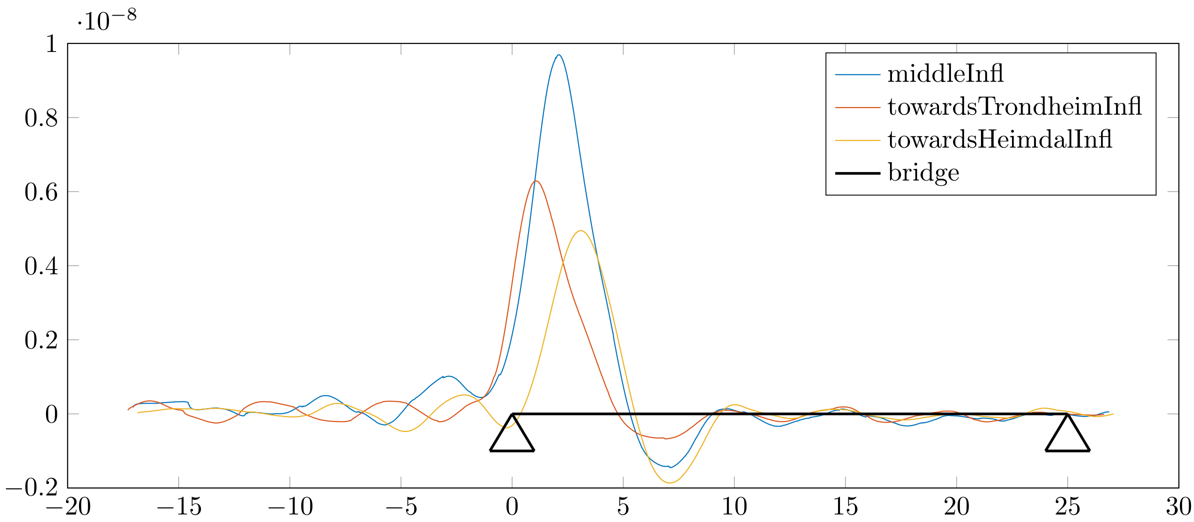


Figure 4.8: Influence line, filtered strain signals

4.3.3 Influence lines from unfiltered strain

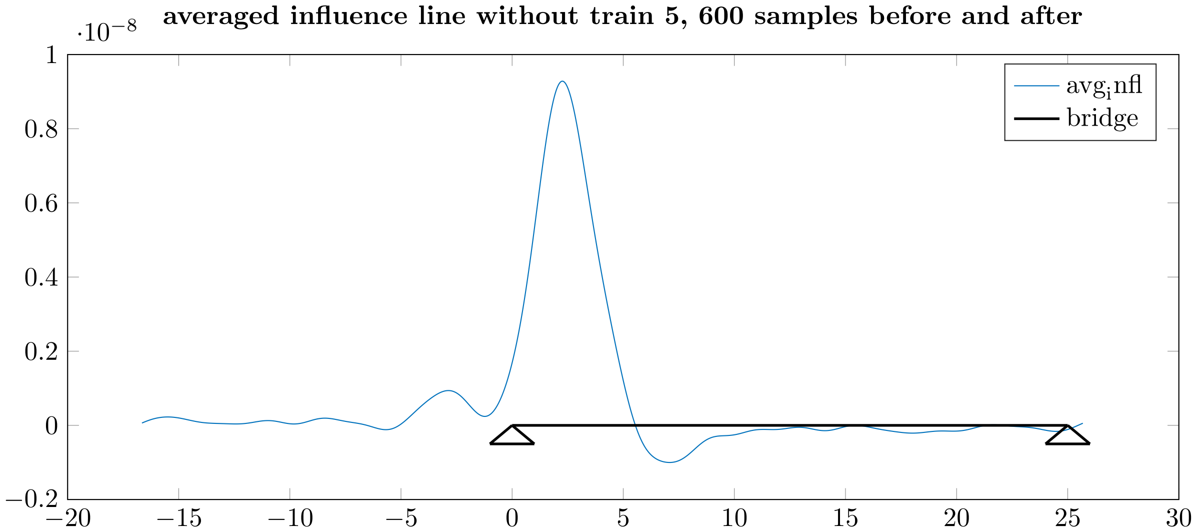


Figure 4.9: Influence line, filtered after averaging

4.4 Dynamic effects

The dynamic effects can clearly be seen in the plots of the influence lines for the various train passings. They appear as oscillations in the plots, and are more visible in the low magnitude areas of the influence line. These oscillations vary from train to train making it clear that the dynamic effects depends on the train. The varying influencing factors may be train speed and weight. In the source code producing these influence lines an assumption of train weight has been made, which makes all train axles equal in weight. What is interesting is the effects of an approaching train, which clearly induces oscillations in the bridge even though the train is as far as 40 meters away from the beginning of the bridge. The differences between the dynamic effects for the train passings may relate to velocity, axle weights and train acceleration (there may be more causes).

These dynamic effects are unwanted in the static influence line. In theory, averaging enough influence lines should reduce these effects enough to get usable data. This thesis does not contain enough train passings to achieve this. Also as mentioned before, one of the trains have amplified dynamic effects which throw off the results somewhat when performing averaging. Therefore exluding the results from train 5 as in would be reasonable. Figure **??** shows the average of the different influence lines where train 5 has been excluded. This plot still contain dynamic effects, which will need to be removed, but the amplitude of the oscillations have been visibly reduced.

Wrongly determined train velocity is a cause of oscillating influence lines, and can easily be mistaken for dynamic effects. Figure 4.10 is an example of a influence line determined from a wrongly set speed.

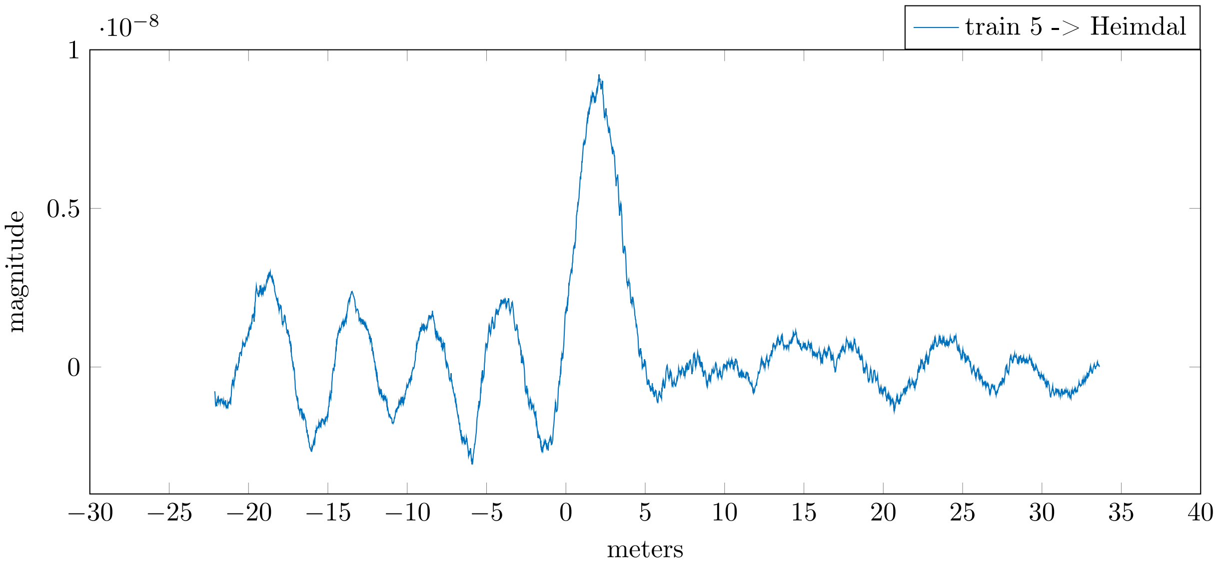


Figure 4.10: Influence line train 5

Shortening the number of samples in the case without train 5, reduces the dynamic effects furter, and a filtering of higher frequencies gives amore usable influence line as seen in ??. A general formula for identifying influence lines with too much oscillation should be developed. One way could be to use table 4.3 and exlude the trains which dominates error, or that differs most from the other trains.

The support towards Trondheim is of a special nature, it is connected to a very little bridge spanning perhaps 2 meters which cars may pass under. This may affect the train's entry and cause dynamic effects. It also provides a problem when deciding what should be part of the final influence line, what influences the sensor? One way to do it would be to simply cut the influence line at the samples corresponding to the bridge, however that does not seem likely to be a very good solution. Another way would be to smooth the influence line to the point where the entry part becomes itegrated with the the major influence line peak, which would result in a greatly distorted peak and is therefore not a good solution.

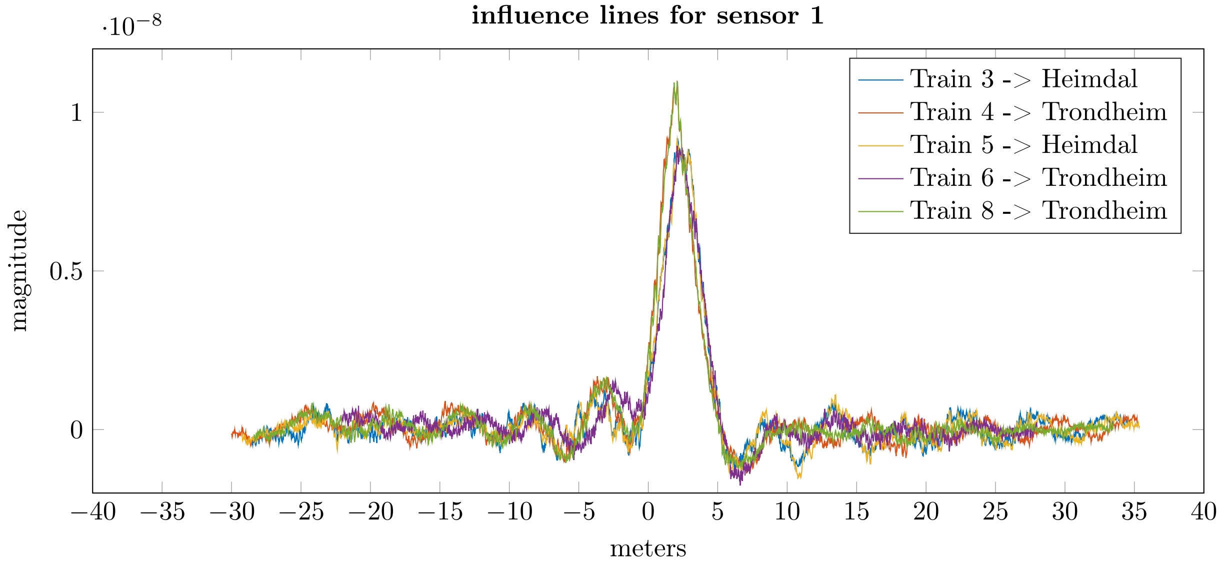


Figure 4.11: Influence lines from figure:4.4 on top of each other for sensor 1

Figure 4.11 shows all influence lines in one figure, which highlights the differences and similarities between the figure.

4.5 Averaging calculated influence lines

Clearly two of the influence lines, train 4 and train 8 has a maximum peak magnitude which differs from the others. These two trains both travels the bridge in the same direction, which could be a cause for differing magnitudes, however train 6, which also travels the same direction, does not follow this trend and in fact aligns with the other peaks of train 3 and 5. Based on this it can be assumed that direction of the train should not affect the magnitude of the maximum peak.

Another hypothesis for the differing peak heights, could be that the trains differ greatly in actual axle weights. A heavy train would cause higher values of measured strain and the measured strain is what the matrix method uses to find the influence lines. The values of axle weights used to produce the influence lines of the bridge are fixed at the values of a empty train. A quick study of equation 2.9, shows that increasing the values measured strain also would increase the values of the influence line. This is therefore a likely cause of differing magnitudes. (Look into possibility of identifying this effect in calculated axle weights !!!)

The average of these influence lines will likely have a maximal peak magnitude somewhere between the peaks of train 3,5 and 6 and train 4 and 8. This would cause problems when calculating the axle weights, the axle weights of train 3, 5 and 6 would be underestimated, and the axle weights of train 4 and 8 would be overestimated. This effect can be seen in the tables 4.6 and 4.7, showing calculated axle weights using this averaged influence line. The equivalent of figure 4.11 for sensors 2 and 3 can be found in appendix C.10 and C.11. These collection of influence lines also display a differing in magnitudes of the influence lines, with some differences. For sensor 2 train 4 and 8 still has higher maximum values, but train 8 produces a lower value of maximum compared with train 4. For sensor 3 the efect of differing magnitudes are almost invisible, for this sensor all trains seems to produce similar values except for train 6. This is also visible in table 4.6, where the axle weights for sensor 3 shows train 6 having the highest total value.

Another factor which could be the source of these effects are the velocity of the trains. A wrongly determined velocity causes oscillations in the influence lines as discussed previously, and maybe this also could cause differing maximum peak values. It may also be that different velocities could cause differing entry effects, which would provide

In the execution of the matrix method, the train velocity affects the distance between the

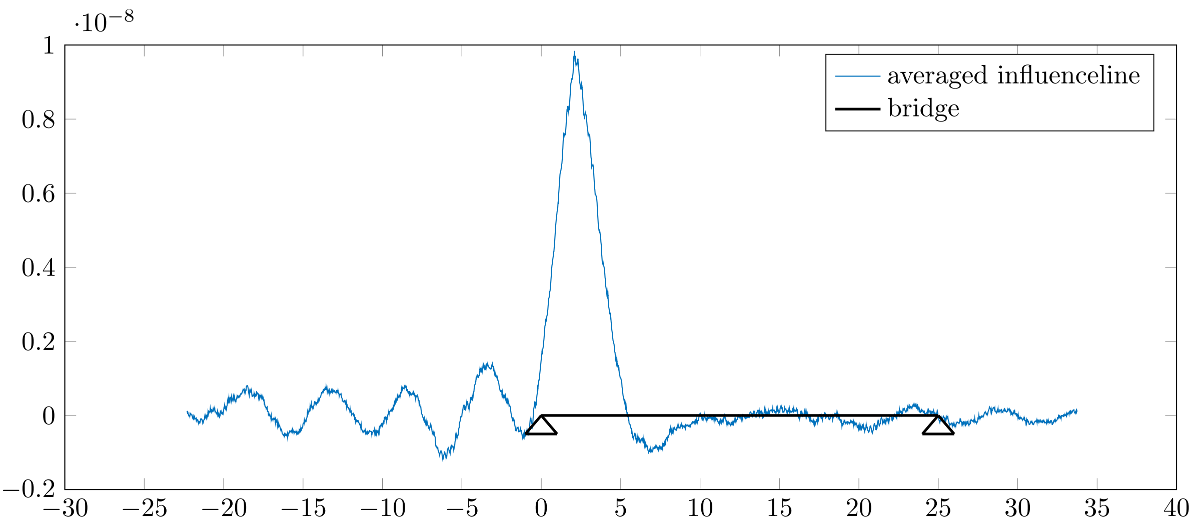


Figure 4.12: Averaged of the 5 trains

A possible way to place the found influence line is shown in figure 4.12, which places the influence line in the assumed position on the bridge. The maximum magnitude of the influence line should be found at the sensor location, thusly the averege influence line has been placed in the corrdinate system of the bridge accordingly. There is however the problem of noise, which makes identifying the actual max peak difficult.

Filtering the signals so that a singular smooth maximum peak can be identified. This could distort the actual signal, but will be an interesting approach.

This placement of the first peak and it's influence line also decides the follwoing placement of the influence line which are decided by the axle spacings found by the BWIM system. Therefore a wrongly placed first axle would provide increasing error for each axle found. Alternatively each influence line could be placed according to the identified axles or bogies. Since the axle distances is known for the signals in this thesis identifying the first axle or bogie and placing the rest

4.6 Optimized influence lines

During this project

4.7 Using calculated influence lines

The matrix method calculates a influence line vector, which is perfectly suited to that specific strain signal. The signals will however vary from train to train, both in length and magnitude. This means that the influence line for the sensor needs to be adapted to the strain signal. To perform a standard axle weight calculation, the influence line is required to be correctly aligned with the strain signal. The first peak of the strain signal, corresponding to the first axle of the train, should occur at the same location as the peak of the influence line which should be precisely at the sensor location. Identifying the first peak of the strain signal is subject to noise which corrupts any reading of peaks in the raw strain signa. Therefore filtering of noise is needed to correctly identify the signals peaks. A trains axle spacings, as seen in 3.7 which is the train type of the measurements, consists of short axle distances of about 2.5 meters. If the axle spacing between two axles are short compared to the bridge, or more specifically short compared to the width of the influence line, they both influence the signal simoultaneously and the peaks corresponding to the two axles thus lies very close to each other. The filtering can therefore not be to hard or soft, which results in problems when trying to automate the procedure of identifying axles. To correctly align the strain signal and influence line, the matlab code used in this thesis first smooths the strain signal to a degree where the desired number of peaks are identifyable before using matlabs findpeaks (REFERENCE THIS) procedure to find the peak locations, like seen in figure 4.13. When trying to place the influence line, it was found that axle detection needs to be very accurate for the calculation of axle loads. When using the method

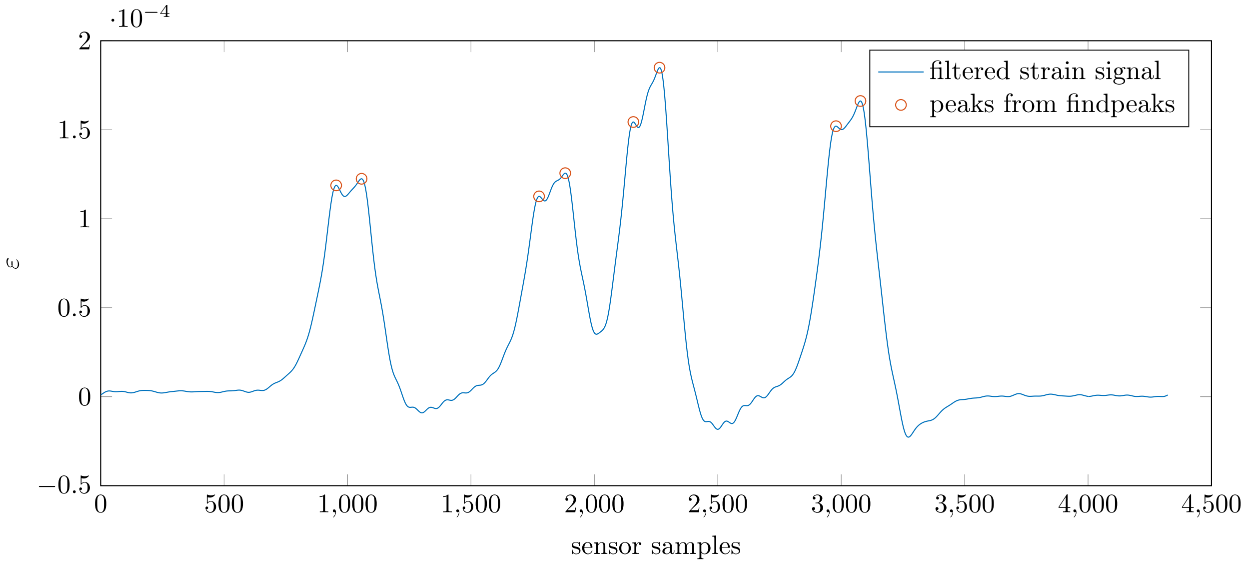


Figure 4.13: Axle peaks in strain signal

described above, with filtering to a degree where 8 peaks are found and to check how those peaks correspond with the known train's axle distances, proved to be accurate in some cases and very wrong in other cases. A wrongly found axle peak could for instance result in negative axle weights, and generally wrong axle weights. A more general method which seems to place the influence better is to filter the signal to a degree where only the major peaks are found. The location of the first such peak should roughly correspond to the centre between closely spaced axles, or a bogie centrum, on a train. Since the trains axles spacings are known, a successfully identified bogie location should place the influence line with a decent accuracy.

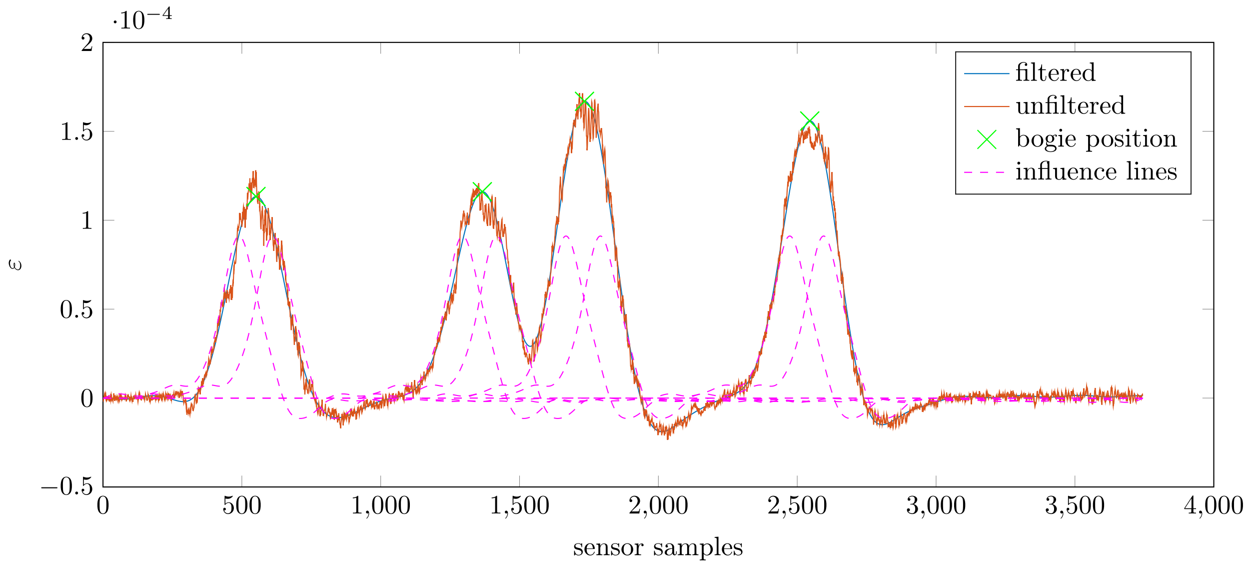


Figure 4.14: Placement of influence lines, based on bogie peaks in signal. The influence lines shown are scaled to fit strain signal magnitude.

4.8 Calculate the axle weights

The system setup described in section 3.3, gives us three different locaions for measuring strain and so we have three diffferent influence lines generated by the BWIM program. When calculating the axle weights corresponding to each train, we will then have three different estimates of the axle weights. However, since we do not know the exact weights of the trains, estimating correctness of the BWIM can not be done through comparing known and calculated axle weights.

Figures 4.16 and 4.15 are the influence lines used to calculate the axle weights in the tables 4.7 and 4.6. By studying the different influence lines, it is clear that the visible differences between the two variants are minimal. This can also be seen in their respected tables. The influence lines produced by filtered strain seems to produce a influence line of slightly lower magnitude, which when used to calculate axle weights results in slightly higher values. Sensor location clearly distinguishes the influence lines:

- The influence line from sensor 1, the middle section sensor, appear to be a mixture of the influence lines from the other sensors.
  - sensor 3 is closer to the end of the bridge's first section, and this can clearly be seen through the negative influence after the first 5 meters of the bridge.
  - sensor 2 has lesser negative magnitude after the first 5 meters. The entry effects of the averaged influence lines also appears to be least significant for this sensor location.
- sensor 3 is influenced by a larger section of the bridge.

The axle weights in table 4.7 is calculated using the influence lines from figure 4.16 . These The axle weights calculated for the minimal influence lines is similar to what is shown in the tables 4.7 and 4.6. A shorter influence line may still contain dynamic effects, but likely less than longer influence lines. Less because of .....

4.8.1 Accuracy of axle weights

As seen in tables 4.7 and 4.6, there are differences between the calculated axle weights for each sensor. The values for the different axles should be relatively similar for each calculation, but for some of the signals it is clear that values vary with up to 2000 kg which unlikely is explainable by passenger distribution in the train. A more reasonable explanation for these differences from axle to axle could be the placement of the influence line representing the axle, as has been discussed in 4.7. These are errors which may have one or more reasons.

- the sensors may not be correctly calibrated, resulting in differences in axle weights from sensor to sensor. This can be controlled by calculating the ration between the same axles for different sensors.
- The initially set axle weights remaing constant for each train resulting in influence lines of incorrect magnitude which when averaged will make the influence line have a magnitude

	sensor 1					sensor 2					sensor 3				
	trains and their axle weights for sensors														
axle	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8
1	8563	10689	8578	11156	10617	9006	11278	9570	11195	11233	8341	8763	8402	9111	8532
2	9343	10379	9170	10237	10284	7814	8628	7491	8176	8295	9581	9820	9440	10945	10043
3	8709	10294	8817	11353	10353	9521	11446	9983	11940	11668	8837	8563	9203	8752	8320
4	9057	9868	8451	10400	10285	8214	8586	7351	8336	8697	9073	9626	8547	10479	10262
sum car	35672	41230	35016	43146	41539	34555	39938	34395	39647	39893	35832	36772	35592	39287	37157
5	13392	15615	13546	15879	14865	14904	18402	15379	17434	17489	14064	14462	14660	13515	13475
6	14581	14893	13859	15985	16313	13059	13336	11674	13391	14079	16116	15509	15121	18038	17548
7	11303	15097	11479	15656	14380	13238	17679	13678	17332	17278	12374	13561	12595	13615	12636
8	14184	12962	13616	13549	14350	12496	10913	11196	10910	12026	14788	13792	13933	16003	15501
sum loc	53460	58567	52500	61069	59908	53697	60330	51927	59067	60872	57342	57324	56309	61171	59160
sum tot	89132	99797	87516	104215	101447	88252	100268	86322	98714	100765	93174	94096	91901	100458	96317

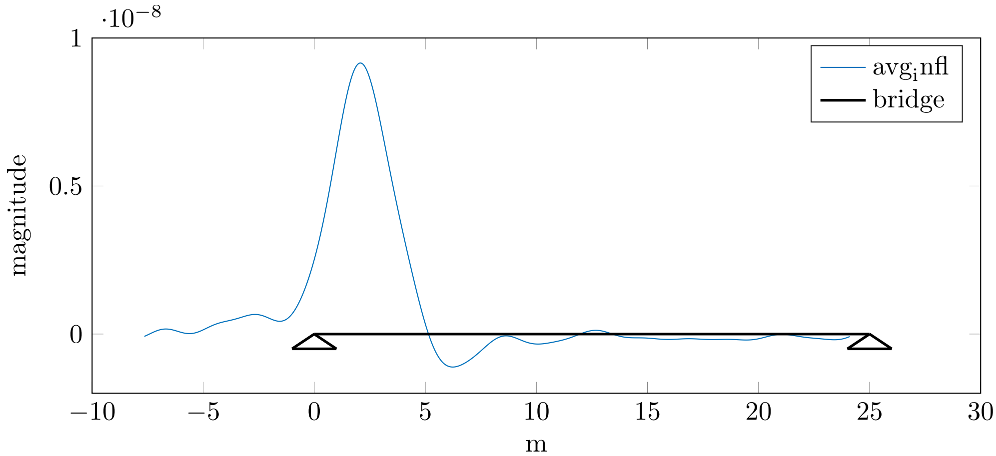
Table 4.6: Table of axle weights for averaged influence lines, all trains

	sensor 1					sensor 2					sensor 3				
	trains and their axle weights for sensors														
axle	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8
1	8819	10971	8837	11301	10858	9788	12060	10353	11531	11859	8042	8436	8093	8916	8241
2	9106	10086	8932	10052	10046	6847	7536	6423	7435	7367	9835	10100	9715	11083	10263
3	8940	10522	9055	11488	10561	10343	12185	10765	12282	12338	8625	8317	8994	8561	8081
4	8822	9620	8207	10252	10066	7231	7577	6313	7563	7740	9376	9928	8867	10687	10522
sum car	35687	41199	35031	43093	41531	34209	39358	33854	38811	39304	35878	36781	35669	39247	37107
5	13772	16046	13938	16095	15283	16131	19592	16561	17926	18582	13572	13946	14169	13138	12950
6	14255	14494	13517	15729	15949	11548	11597	10066	12196	12487	16510	15931	15561	18235	17885
7	11696	15513	11883	15874	14771	14504	18803	14944	17770	18271	11909	13092	12127	13310	12179
8	13866	12567	13280	13332	13990	11051	9210	9625	9748	10496	15178	14230	14359	16217	15850
Sum loc	53589	58620	52618	61030	59993	53234	59202	51196	57640	59836	57169	57199	56216	60900	58864
Sum tot	89276	99819	87649	104123	101524	87443	98560	85050	96451	99140	93047	93980	91885	100147	95971

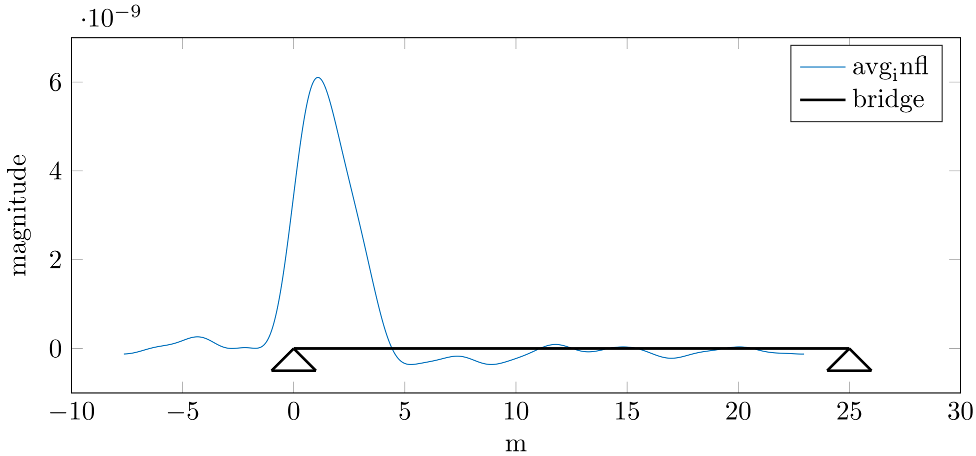
Table 4.7: Table of axle weights for averaged influence lines, where strains have been filtered, all trains

	sensor 1					sensor 2					sensor 3				
	trains and their axle weights for sensors														
axle	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8	train 3	train 4	train 5	train 6	train 8
1	8797	10984	8905	11393	10746	9139	11473	9802	11373	11272	8458	9126	8626	9632	8771
2	9431	10467	9162	10726	10627	7961	8553	7503	8949	8582	9677	9765	9427	11362	10232
3	8860	10476	9080	11255	10311	9479	11454	10045	11886	11452	8855	8918	9344	9046	8444
4	8956	9718	8217	10956	10475	8324	8475	7317	9240	9024	8845	9098	8134	10820	10192
sum car	36044	41645	35364	44330	42159	34903	39955	34667	41448	40330	35835	36907	35531	40860	37639
5	13598	15795	13884	16239	14856	15072	18646	15682	17677	17494	14189	14869	14916	14155	13712
6	14769	15069	13863	16527	16865	13304	13285	11703	14454	14578	16245	15406	15067	18496	17837
7	10980	14773	11322	15297	13793	12888	17424	13476	17120	16756	11988	13622	12372	13889	12354
8	13984	12654	13245	13941	14430	12477	10492	10988	11852	12197	14577	13214	13550	16187	15341
sum loc	53331	58291	52314	62004	59944	53741	59847	51849	61103	61025	56999	57111	55905	62727	59244
sum tot	89375	99936	87678	106334	102103	88644	99802	86516	102551	101355	92834	94018	91436	103587	96883

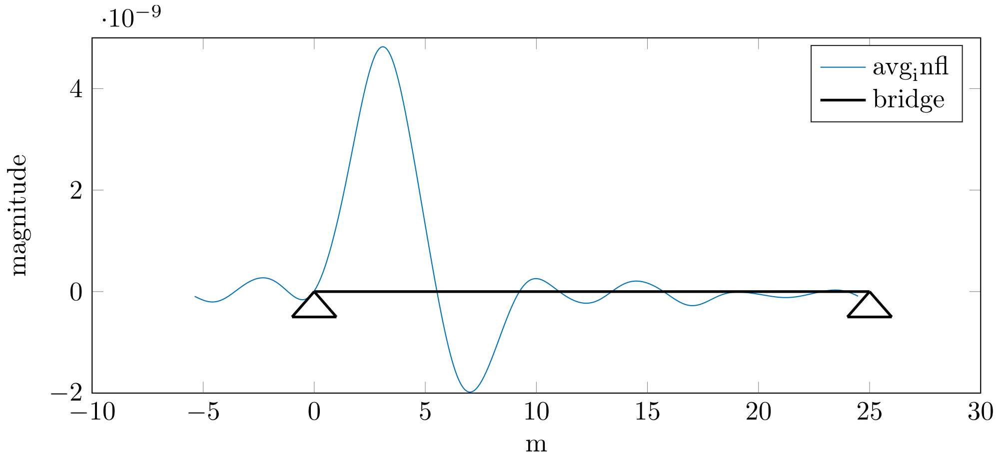
Table 4.8: Table of axle weights for minimal averaged influence lines



(a) Averaged influence line for sensor 1



(b) Averaged influence line for sensor 2

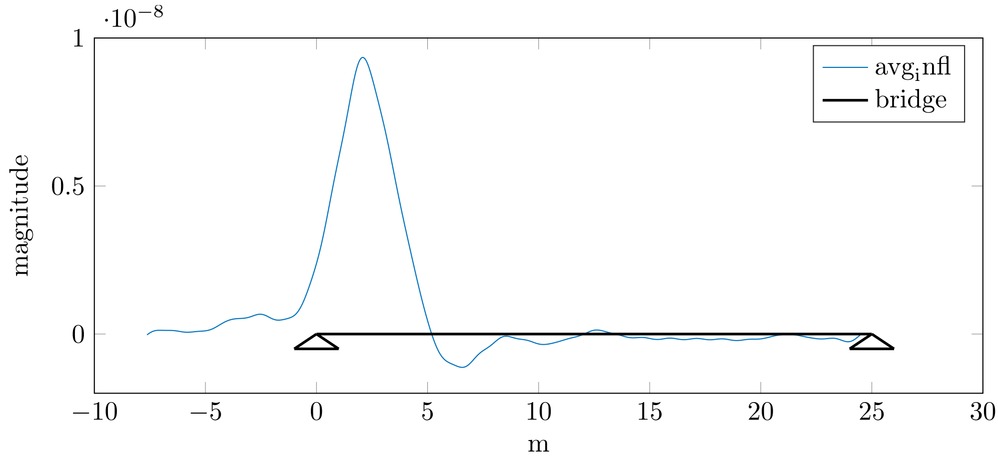


(c) Averaged influence line for sensor 3

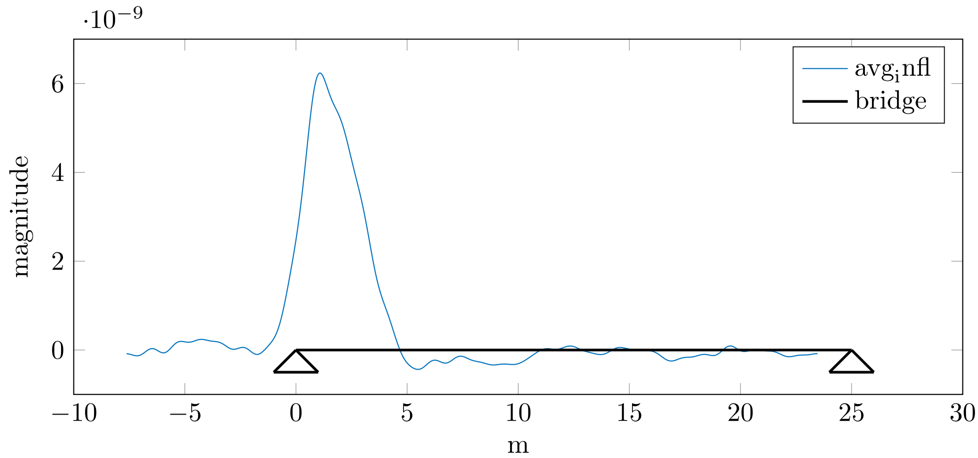
Figure 4.15: averaged influence lines used to calculate axle weights

somewhere between the found values.

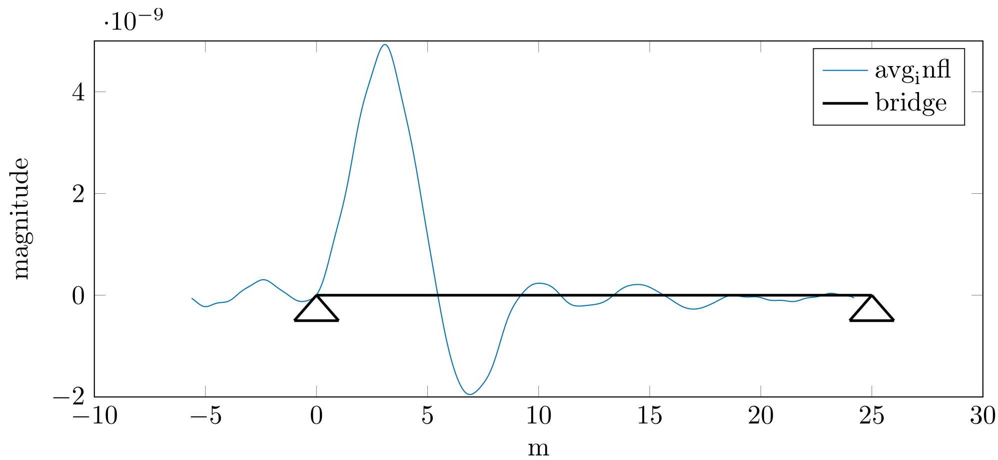
The ratio tables, 4.11, 4.10 and 4.9, highlight the differences and similarities between the influence lines. The ratio between axle weights for the different versions of the influence line



(a) sensor 1



(b) sensor 2



(c) sensor 3

Figure 4.16: averaged influence lines, based on filtered strains, used to calculate axle weights

differ little from each other, all ratios are within 10 % showing that the calculated gross train weights are reasonably constant from sensor to sensor. The tables show that the difference between minimal and standard length influence line are small, while the influence lines calculated

Gross weight train						avg ratio
	train 3	train 4	train 5	train 6	train 8	
sensor 1	89132	99797	87516	104215	101447	
sensor 2	88252	100268	86322	98714	100765	
ratio:	0.99013	1.00472	0.98636	0.94722	0.99328	0.98434
sensor 1	89132	99797	87516	104215	101447	
sensor 3	93174	94096	91901	100458	96317	
ratio:	1.04535	0.94287	1.05011	0.96395	0.94943	0.99034
sensor 2	88252	100268	86322	98714	100765	
sensor 3	93174	94096	91901	100458	96317	
ratio:	1.05577	0.93845	1.06463	1.01767	0.95586	1.00647

Table 4.9: Ratio table showing the ratio between gross train weight for the different sensors

from filtered signals have the highest values of difference. The tables also indicate that sensor location is of significance. Sensor 1 and sensor to produce the most consistent ratio values, where calculated gross vehicle weight is higher for every train compared with the same values from sensor 2. The same can not be said for the comparison of sensow 1 and 3 as well as sensor 2 and 3, where the ratio values vary from over 1 to under one for different trains. These differences can be seen by studying the influence lines for the different sensors. The influence lines for sensor 1 and 2 are visually similar while the influence line for sensor 3 have the lowest peak value but appears to have a wider zone of influence meaning that the sensor is affected more from the other sections of the bridge. Likley sources of error in calculated axle weights:

- Wrongly determined train velocity - resulting in incorrect influence lines, and error in alignment of influence lines with strain signal.
- Peak detected by placement algorithm is wrong.
- Averaged influence line does not represent the strain signal, that is the axle weights used to calculate the influence lines was not correct and resulted in too high or low magnitude of the influence lines peak.

4.9 Calibration of sensors and the program

THIS CHAPTER MAY NO LONGER BE RELEVANT DUE TO THE SENSORS BEING CALIBRATED.. PERHAPS INSTEAD DISCUSS HOW GENERAL CALIBRATION AND CONTROL OF INFLUENCE LINES WAS TRIED, BUT BECAUSE OF THE DIFFICULTY OF EXTRACTING THE LOCOMOTIVE OF THE FREIGHT TRAIN. a CALIBRATION AND CONTROL WAS NOT POSSIBLE TO DO. As seen in section 4.8, a calibration of the sensors may be necessary for correct calculations of a trains axle weights. In normal circumstances this

Gross weight train						avg ratio
	train 3	train 4	train 5	train 6	train 8	
sensor 1	89375	99936	87678	106334	102103	
sensor 2	88644	99802	86516	102551	101355	
ratio:	0.99182	0.99866	0.98675	0.96442	0.99267	0.98686
sensor 1	89375	99936	87678	106334	102103	
sensor 3	92834	94018	91436	103587	96883	
ratio:	1.03870	0.94078	1.04286	0.97417	0.94888	0.98908
sensor 2	88644	99802	86516	102551	101355	
sensor 3	92834	94018	91436	103587	96883	
ratio:	1.04727	0.94205	1.05687	1.01010	0.95588	1.00243

Table 4.10: Ratio table showing the ratio between gross train weight for the different sensors, from minimal influence lines

could be done in the following manner.

1. Have a train, of which the known properties are velocity, axle spacings and axle weights, perform one or more runs in both directions.
2. The obtained strain signals from these passings are used along with the Influence line to calculate the axle weights for at least one sensor.
3. The resulting axle weights should have a constant ratio between the same axles for different sensors.
4. The axle weights are scaled to equal the known values. These scalars is the calibrating constants for the sensors.
5. The scalars obtained could be used directly on the signal data, but the only part of the BWIM which directly requires this scaling for correct results are the calculated axle weights.

Due to limited field data, the only train which is usable for calibrating the sensors is the freight train. The freight train has one constant which the other trains do not have, a locomotive which will have axle weights approximately equal the given properties of the locomotive as listed in **??**. The calibration performed in this project was performed thusly:

1. Identify the first 2 major peaks of the signal corresponding to the 6 axles of the locomotive and cut the signal accordingly.
2. Find the speed of the train as well as possible using the methods descibed in section 2.3.

Gross weight train						avg ratio
	train 3	train 4	train 5	train 6	train 8	
sensor 1	89276	99819	87649	104123	101524	
sensor 2	87443	98560	85050	96451	99140	
ratio:	0.97947	0.98739	0.97035	0.92632	0.97652	0.96801
sensor 1	89276	99819	87649	104123	101524	
sensor 3	93047	93980	91885	100147	95971	
ratio:	1.0422	0.94150	1.0483	0.96181	0.94530	0.98782
sensor 2	87443	98560	85050	96451	99140	
sensor 3	93047	93980	91885	100147	95971	
ratio:	1.0641	0.95353	1.0804	1.0383	0.96804	1.0209

Table 4.11: Ratio table showing the ratio between gross train weight for the different sensors, using influence lines calculated from filtered strain signals

3. Use the influence line found through the other trains to calculate the axle weights for each sensor.

4. Find the scalar giving correct results for each sensor.

The strain signal however proved difficult or impossible to cut correctly, due to the length of the locomotive and the width of the influence line the next axle after the locomotive also influenced the signal, which would affect the results. The best suited sensor for this task proved to be the sensor closes to the support on the side towards Trondheim. Figure **??** shows the first peaks of the strain signal for this sensor, where the first two major peaks corresponds to the axles of the locomotive. The red circles named first cutting point and first boigie over, shows a possible cut of the signal could be made. The third and fourth major peak indicates axles of the vagons. The two first peaks should ideally have had the same level of magnitude, the fact that they do not shows that the first and second set of axles influence the sensor at the same time. The second point also is raised above the first point, which also is the case for the next peak corresponding to vagon axles. A safe signal could based on this not be found to perform calibration with, as it would provide a error. Therefore this a calibration of the sensors have not been achieved in this project.

This way of calibration is prone to error. The speed of the freight train is unknown, and the methods described in the thesis for obtaining the speed will not work as well as for the other trains, where all the axle spacings are known. So if a successful identification of the locomotive's axles could be found in the strain signal for at least one sensor, the system could be calibrated. However, due to the width of influence around the sensor, there may not be a set of peaks in the strain signal which influenced only by the locomotive's axles. Also, the method used so far to determine the speed has been based on the difference between the strain history produced by calculated influence lines, so determining the speed of the freight train will produce a probable error. The length of the locomotive from axle 1 to axle 6 is 12.2 meters, and the distance from the locomotive's last axle to the next axle of the train is unknown. Sources of error are many

- The program does not know the exact axle weights, which are required for an exact influence line.
- There might be errors in the calculated speed of the influence line. When the speed is wrong the influence line typically gains additional peaks and curves, which could be confused with dynamic effects.
- The noise levels vary for the different sensors and trains, this may be due to lower values of measurements which causes noise to become more significant compared to measured values. This difference from train passing to train passing will cause error when smoothing influence lines which may have to be adapted to different sensor locations.
- There is not enough data gathered to produce a general averaged influence line.
- Error in the placement of found influence lines, which is a likely cause of error due to filtering.

During the time spent in this thesis, attempts were mad to validate the BWIM Matlab program as well as the resulting influence lines. Due to later discovered bugs in the program causing a systematic error in the averaged influence lines, it was for some time believed that the sensors was uncalibrated. With the correction of this bug it is no longer possible to identify signs of the sensors being uncalibrated. However some time went to investigate how the sensors could be calibrated through the BWIM program. One possible way was to calibrate the sensors based on the resulting axle weights for a known train. This also coincides with how the system best could be validated, by using the identified influence lines to calculate the axle weights of a train with known properties. This would give the actual errors of the calculated influence lines.

5. Conclusion and summary

This thesis is based on a developed BWIM program.

5.1 How does the matrix method perform

The plots showing recreated strains, figure 4.6 and appendix figures C.1 to C.4 show the good accuracy of the matrix method. Given accurate values of train velocity and axle distances the individual influence lines are able to almost exactly recreate the signals. This means that the matrix method by itself is a superior tool, but that it requires high levels of accuracy from the rest of a the BWIM system.

The matrix method runtime depends on the signal length and number of train axles. Relying on the symmetry of the matrices it is possible to only form half the matrix and use the transpose operation to form the full a matrix, which saves computational time.

The problem with the matrix method is that it is subjected to dynamic effects from the bridge. This can be solved by having several calibration runs in both directions and average the resulting influence lines. A train moving at very low velocity would likely also help minimize those dynamic effects and together with several calibration runs this would likely eliminate or make the the dynamic effects negligible.

5.2 The placement algorithm

Section 4.7 shows how the influence lines have been placed according to strain signal. This method has not been controlled properly, but seems to result in satisying accuracy. It does not require any special input from user, and is likely reliable. That is it will not increase the error of future calculation with much.

This method can easily be improved, instead of using only the first peak of the strain signal to align the influence lines all the peaks could be used to identify a even safer placement. Possibly an optimization routine, much like the one used in this thesis to find the trains velocities, could be used to find the best possible placement of the influence lines. One way of performing this optimization of alignment could be:

1. Use the placement found by the existing method as an initial guess of placement.

2. For each iteration of optimization, the axle weight is calculated and used to recreate the strain signal.
3. A mean square error or similar function, is calculated
4. The alignment of influence lines which minimizes this error will likely be a good or perfect placement.

This way of optimizing placement of influence lines could be used in a similar manner to find the axle distances of the train. That way detection of peaks corresponding to axles would not be necessary, also this method likely would be less subjected to, or even independent of, noise. However such an algorithm could identify solutions with minimal errors but which do not represent the actual train or signal.

5.3 Axle detection

Jernbaneverket controls the flow of train traffic in Norway, and given live locations of trains the BWIM algorithm might not need to find axle distances. Instead a query of existing systems could provide axle distances for the pasing train. This could in theory eliminate a source of error in BWIM algorithm. Detecting axle peaks correctly likely requires an optimization type algorithm for a bridge of this type. There might also exist other methods of getting good estimates of axle distances, but the method of doing this through peaks in a strain signal is too susceptible to noise and dynamics. The peak method appears to be a good way to identy train bogies, and maybe even the center position of the bogies, which may be acceptible for some BWIM systems.

5.4 General summary of the master thesis

In section 1.2 there was a list of the goals of this thesis and project. The developement of the a simple BWIM program can be called a success as result have been verified to work, the system is capable of reading and using any sort of strain signal and identifying peaks corresponding to specific axles or at least bogie. The speed of the trains have been identified using a brute force method, also other methods of finding vehicle speed have been developed and tested with the theoretical beam model. The BWIM system has been implemented with the matrix method enabling the calculation influence lines for a strain signal where the properties of the passing train is known. The influence lines have been the main focus of this master thesis, and efforts went into developing alternatives to the linear matrix method. This was done through optimization, and was successful for the theoretical strain signals produced through the beam model, but more complex bridge structures the optimization proved more subjected to the initial guess of the influence line. With optimization there might also exist more than one satisfying solution satisfying tolerance limits for the routhine. When strain signals from Leirelva was was used to test and further develope the Optimization routing it was found that it required special considerations compared to the matrix method which performed no matter the signal complexities.

When it came to aligning the identified influence lines with the equation system several solutions were tried and tested. Due to noise levels of the signal, this is a source of error and identifying individual peaks for alignment was found possible to do for each signal but no general method was successfully developed which was able to do this for every signal. Therefore a better solution proved to be identification of peaks produced by a bogie. This was done through smoothing the signal to the point where individual axles is not visible,

5.5 The main challenges of BWIM

Hva er hovedutfordringene med BWIM?

- Noise leading to problems with:
  - Axle detection
  - Alignment of peaks to build system for calculating axle weights.
  - something ?
- Having sufficient calibration vehicles to find a representative influence lines
- Dynamic effects in influence lines

5.6 Possible improvements, and suggestions of future work

Hvordan skal man sette opp et nytt forsøk slik at man kan verifisere og forbedre fremgangsmåten? Har du noen forslag til videre arbeid?

5.7

I konklusjonen (og tidligere i diskusjonen) kan du komme med erfaringene dine med implementering av BWIM algorithmen.

Bibliography

[1] John Doe. *The Book without Title*. Dummy Publisher, 2100.

[2] Ignacio González. "Application of monitoring to dynamic characterization and damage detection in bridges". PhD thesis. KTH, Architecture and the Built Environment, 2014.

[3] jernbane.net. *NSB - Cargo Net El14*. URL: `http://jernbane.net/bo/subpage.php?s=3&t=27`.

[4] A. Liljencrantz, R. Karoumi, and P. Olofsson. "Implementing bridge weigh-in-motion for railway traffic". English. In: *Computers and Structures* 85.1-2 (2007), pp. 80–88. URL: `www.scopus.com`.

[5] MATLAB. *Practical Introduction to Frequency-Domain Analysis*. URL: `http://se.mathworks.com/help/signal/examples/practical-introduction-to-frequency-domain-analysis.html` (visited on 06/06/2015).

[6] Jernbaneverket Njål Svingheim. *By train in Norway*. URL: `http://www.jernbaneverket.no/en/startpage1/News/By-train-in-Norway/` (visited on 06/06/2015).

[7] E. O'Brien, B. Jacob, and COST 323. "Second European conference on weigh-in-motion of road vehicles : Lisbon, 14th - 16th September, 1998". In: (1988), pp. 139, 152.

[8] Michael Quilligan. "Bridge Weigh-in Motion : Development of a 2-D multi-vehicle algorithm". NR 20140805. PhD thesis. KTH, Civil and Architectural Engineering, 2003, pp. viii, 144.

[9] J. Radatz and Institute of Electrical Electronics Engineers Standards Coordinating Committee 10. *The IEEE standard dictionary of electrical and electronics terms (6th ed., Vol. 100-1996, Institute of Electrical and Electronics Engineers). New York: Institute of Electrical and Electronics Engineers*. 1996.

Appendices

A. Dynamics

A train traversing a railway bridge creates actions in longitudinal, lateral, and vertical directions. Braking and traction from a passing train causes longitudinal forces Rocking, or rotations around an axis parallel to the longitudinal axis of the bridge, and vertical dynamic forces are created by structure-track-vehicle conditions and interactions.

A.1 Rocking and vertical dynamic forces

Lateral rocking of moving vehicles provide amplification of vertical wheel loads. This amplification increases the stresses in the members supporting the track. Superstructure-vehicle interaction creates a vertical dynamic amplification of moving loads, which will result in vibrations causing additional stresses in members supporting the track. The unloaded simply supported beam frequency $\omega_1 = \frac{\pi^2}{L^2}\sqrt{\frac{EI}{m}}$, provides a basic indicator of superstructure vertical dynamic response.

B. Trains

B.1 NSB92

B.2 Freight train - EL14

C. Figures

C.1 Recreated strain signals

C.2 Influence lines all sensors

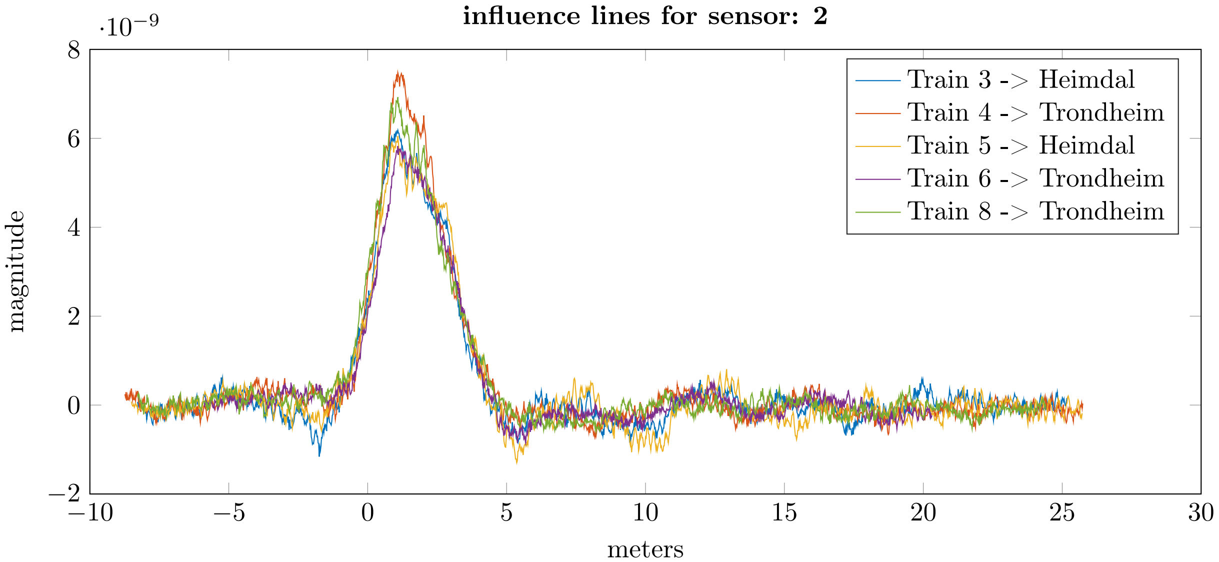


Figure C.10: Influence lines from figure:4.4 on top of each other for sensor 2

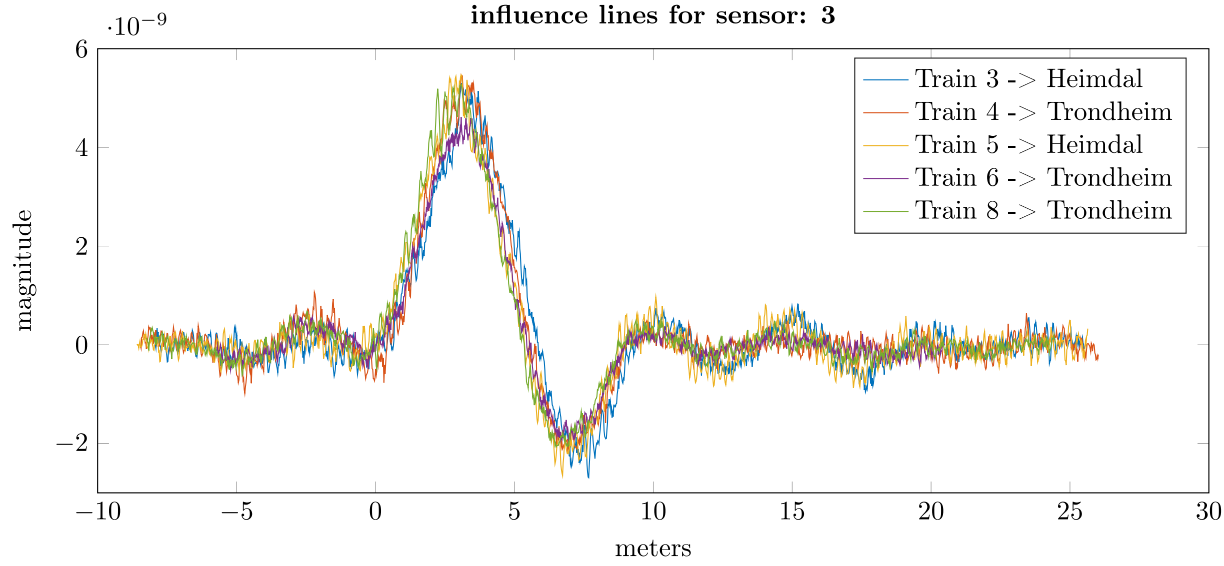


Figure C.11: Influence lines from figure:4.4 on top of each other for sensor 3

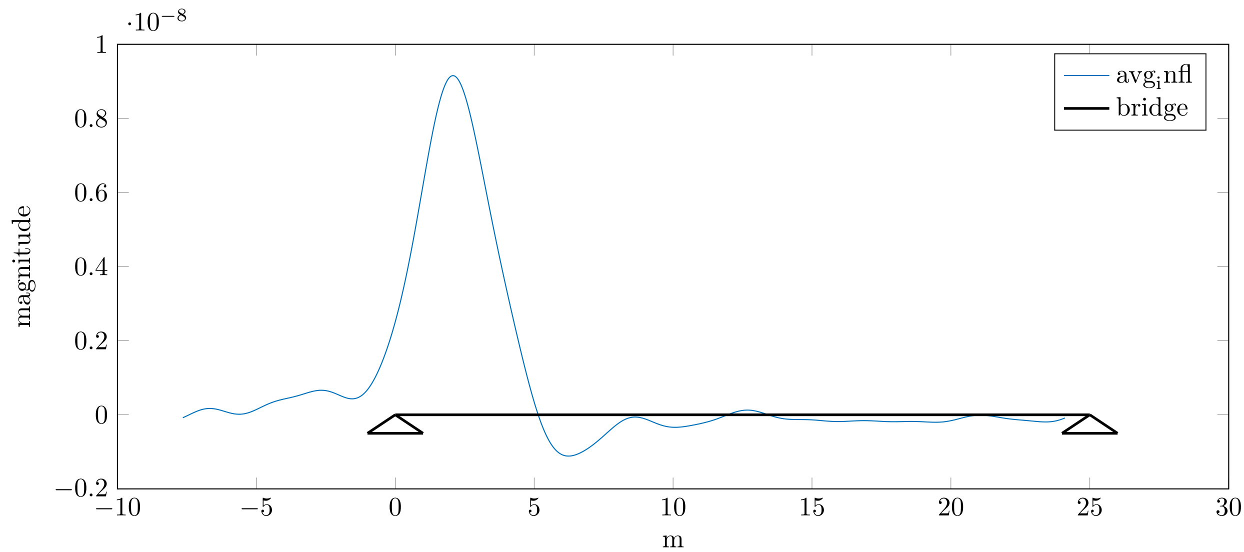


Figure C.12: Averaged influence line for sensor 2

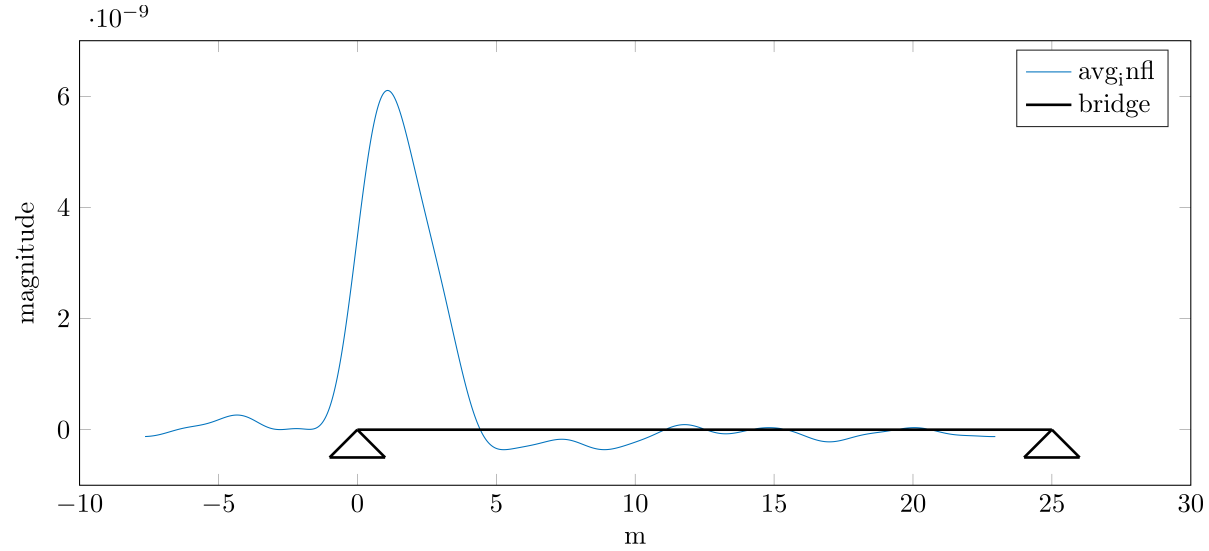


Figure C.13: Averaged influence line for sensor 2

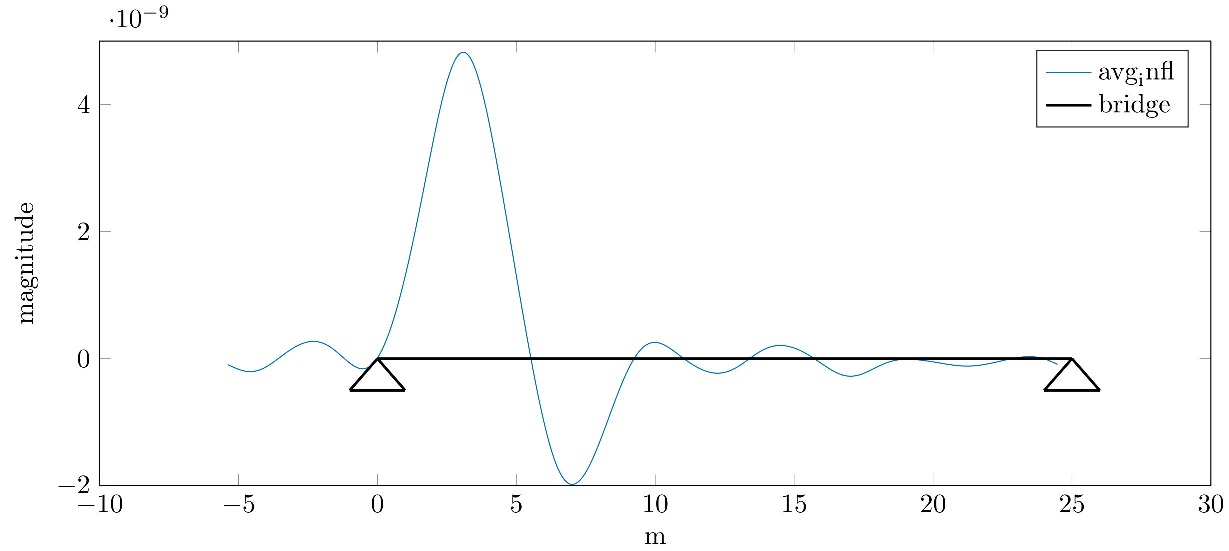
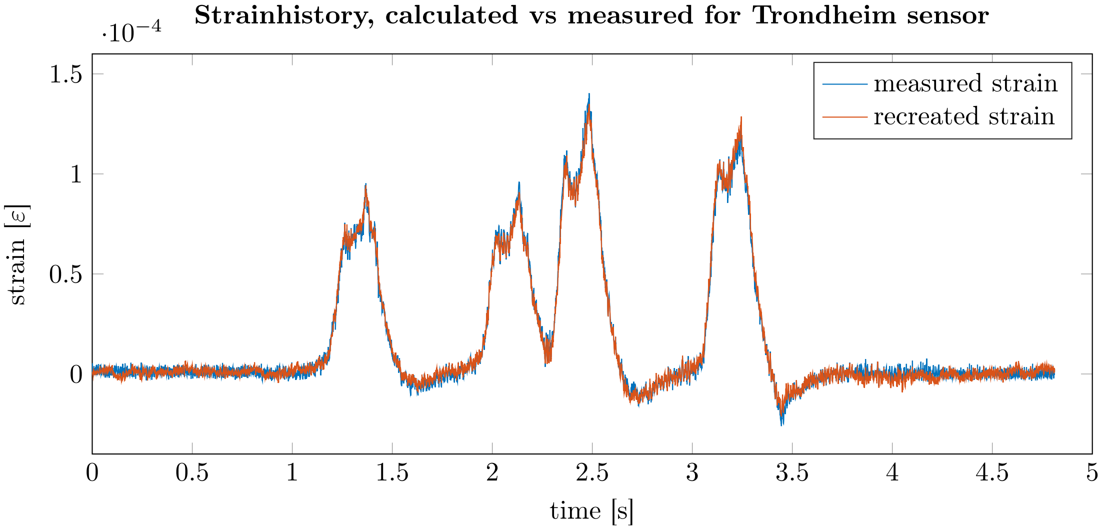
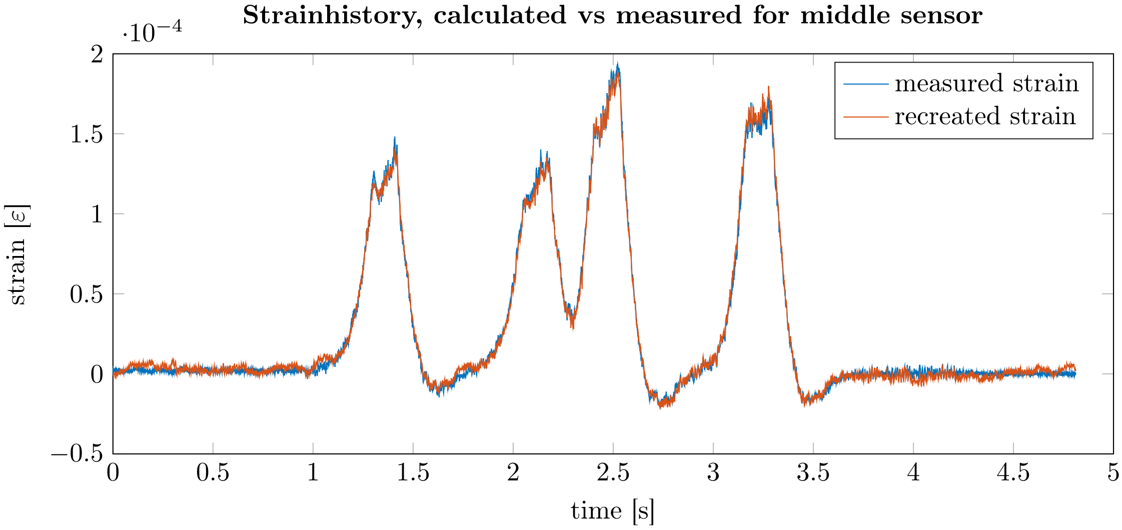


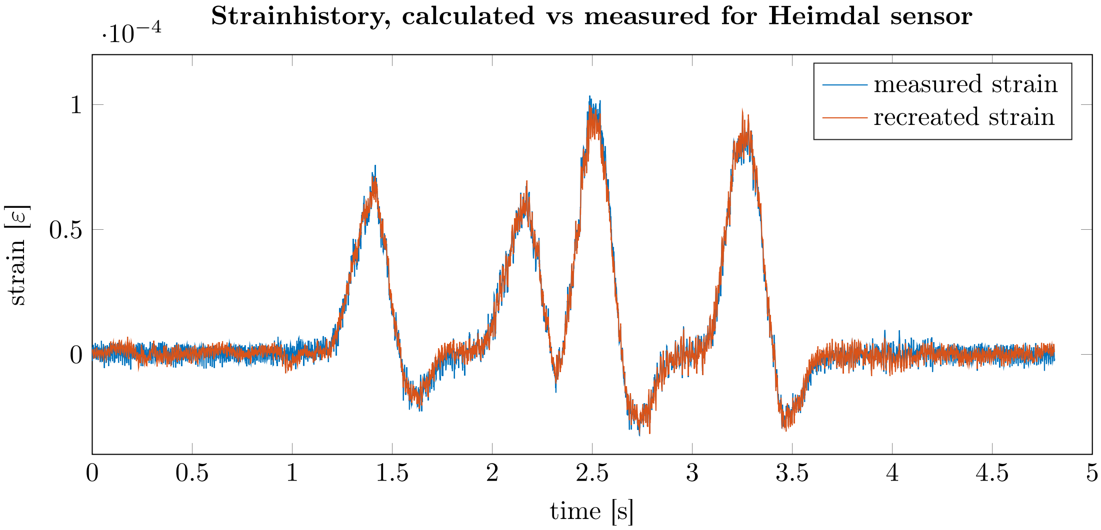
Figure C.14: Averaged influence line for sensor 3



(a) Recreated strain, Trondheim sensor, train4

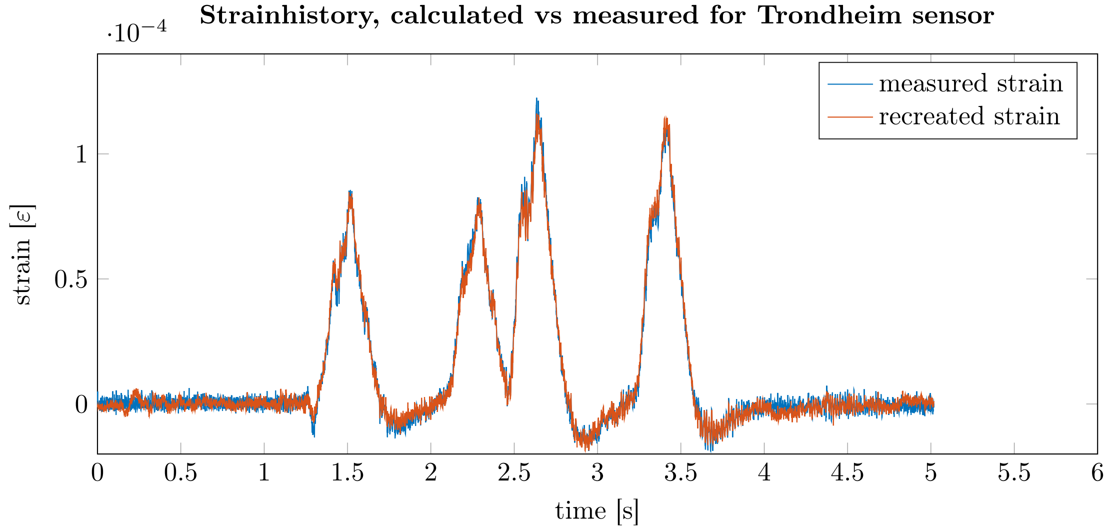


(b) Recreated strain, middle sensor, train4

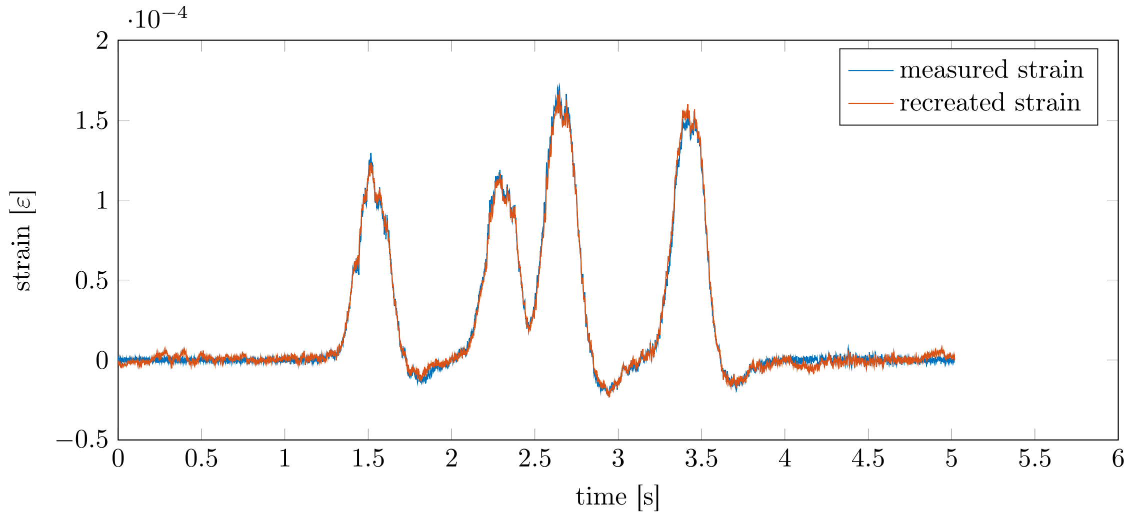


(c) Recreated strain, Heimdal sensor, train4

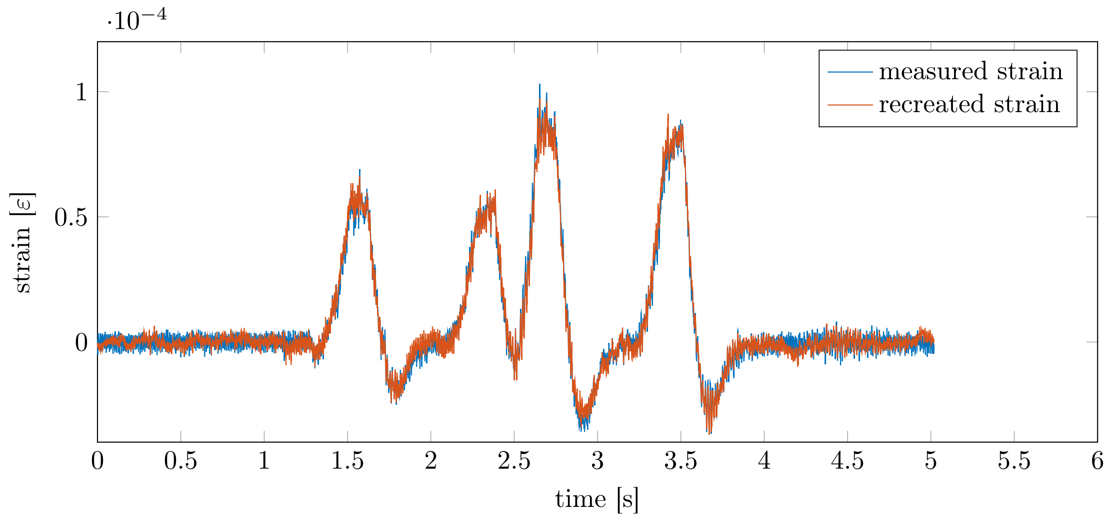
Figure C.1: Recreated strain signals for train 4



(a) Recreated strain, Trondheim sensor, train5

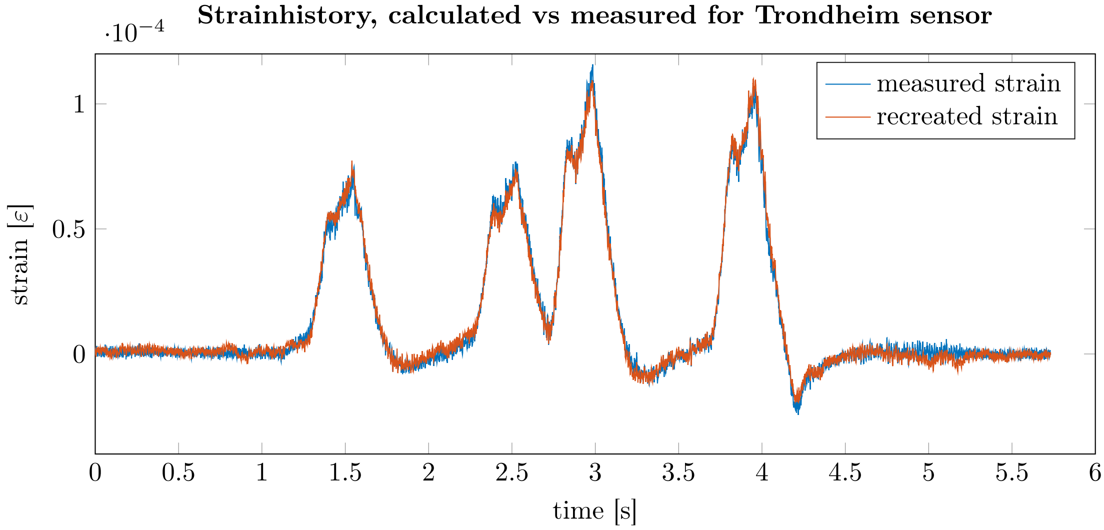


(b) Recreated strain, middle sensor, train5

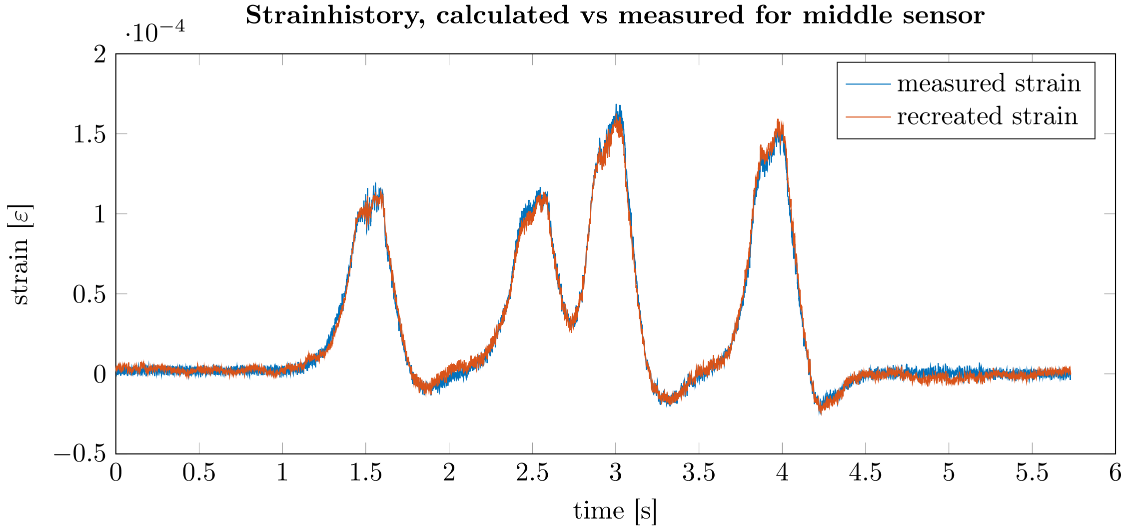


(c) Recreated strain, Heimdal sensor, train5

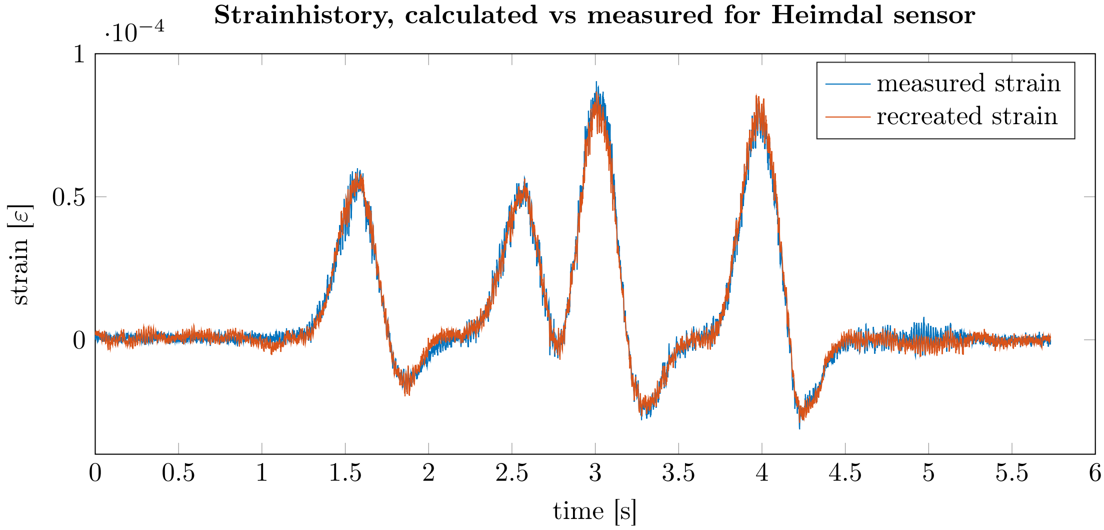
Figure C.2: Recreated strain signals for train 5



(a) Recreated strain, Trondheim sensor, train6

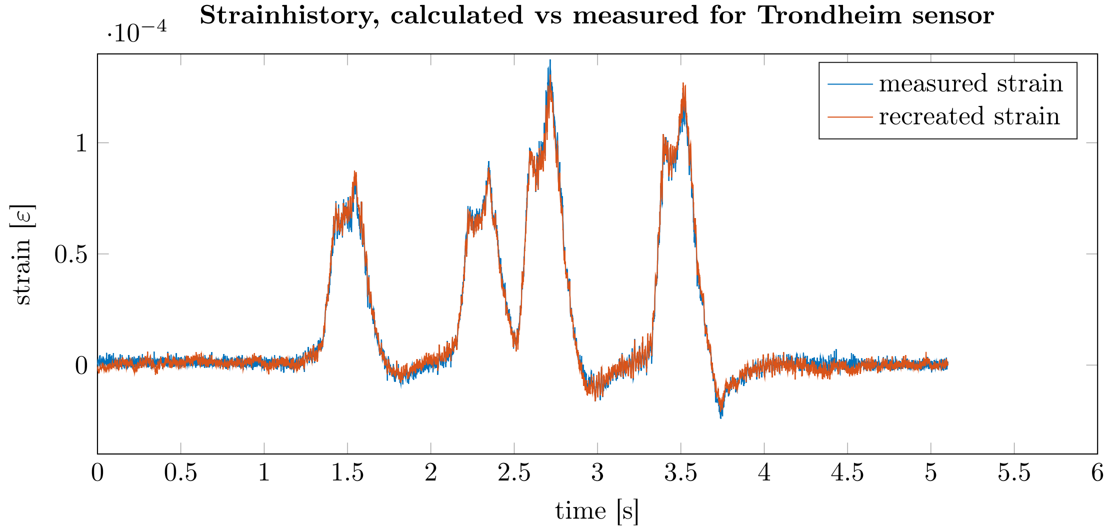


(b) Recreated strain, middle sensor, train6

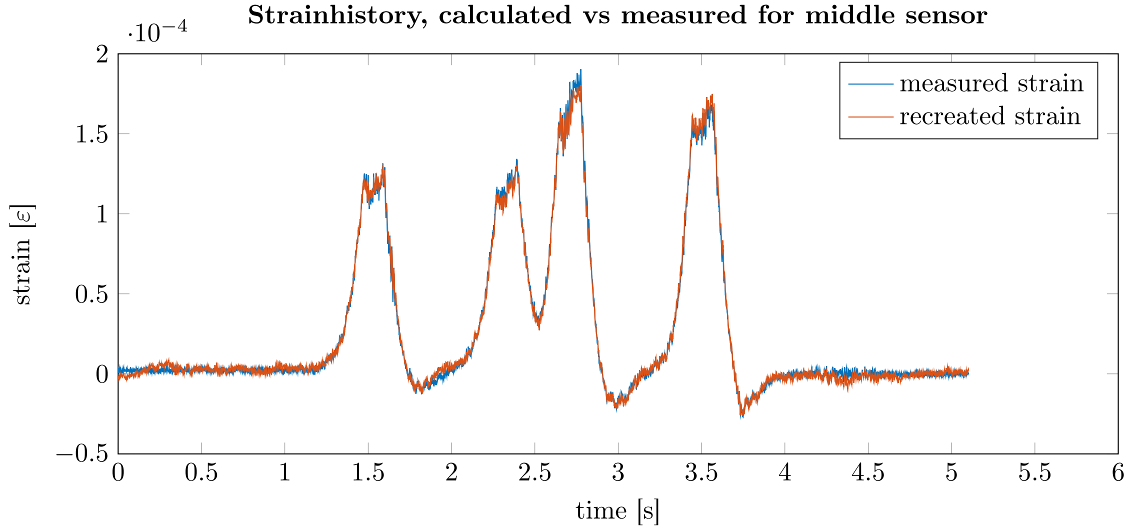


(c) Recreated strain, Heimdal sensor, train6

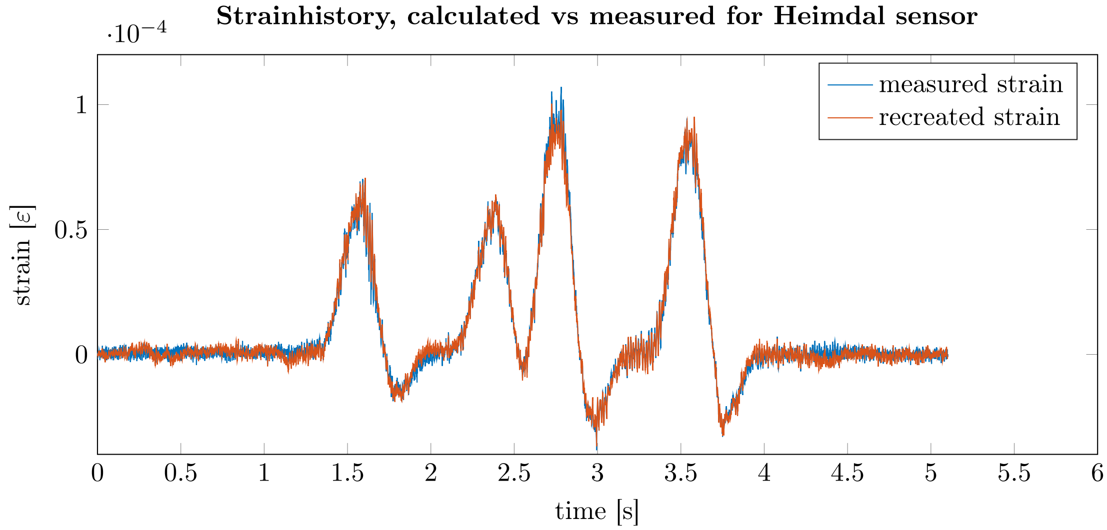
Figure C.3: Recreated strain signals for train 6



(a) Recreated strain, Trondheim sensor, train8

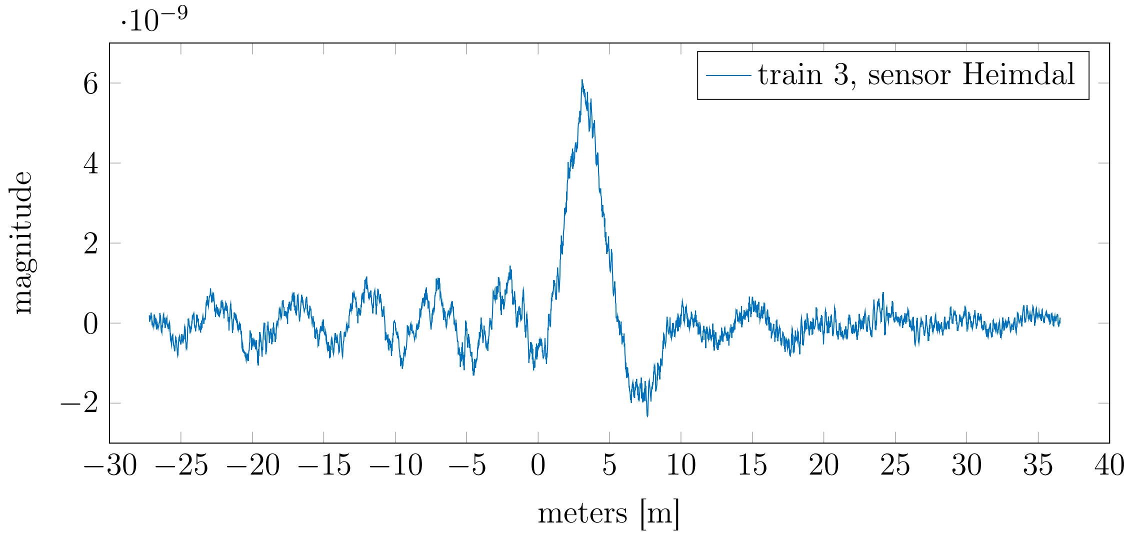


(b) Recreated strain, middle sensor, train8

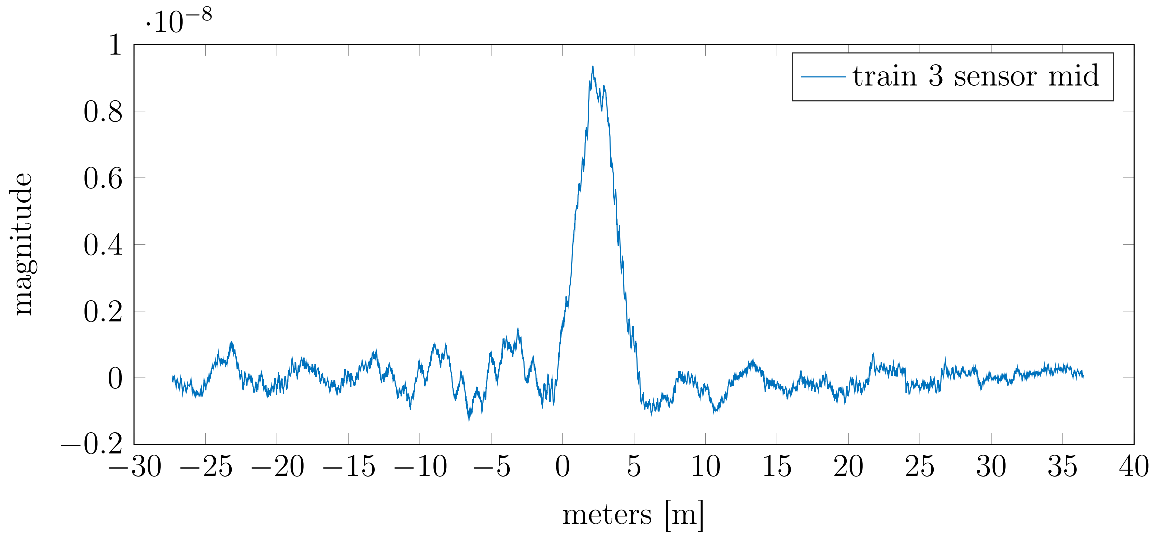


(c) Recreated strain, Heimdal sensor, train8

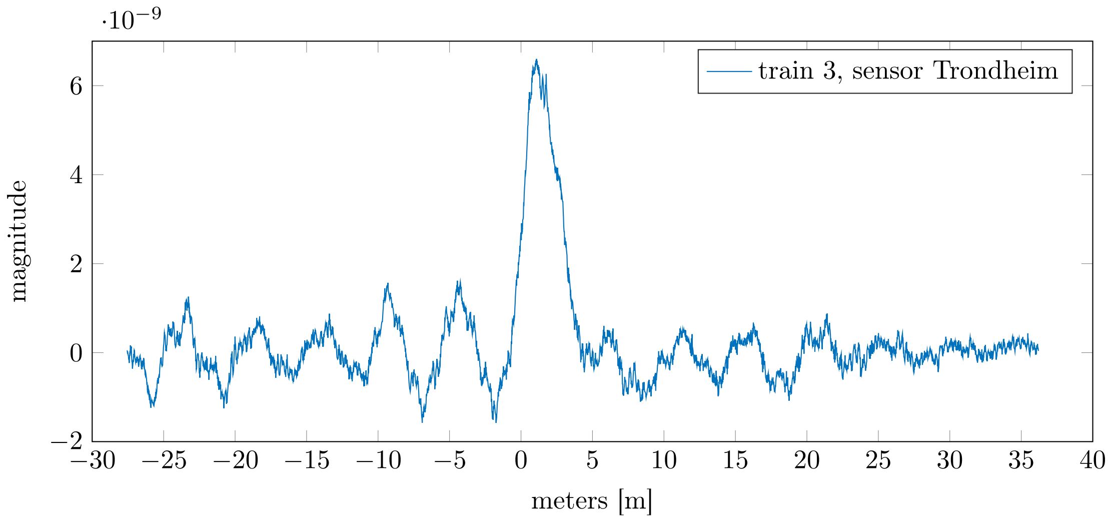
Figure C.4: Recreated strain signals for train 8



(a) Influence line for sensor towards Heimdal, train 3

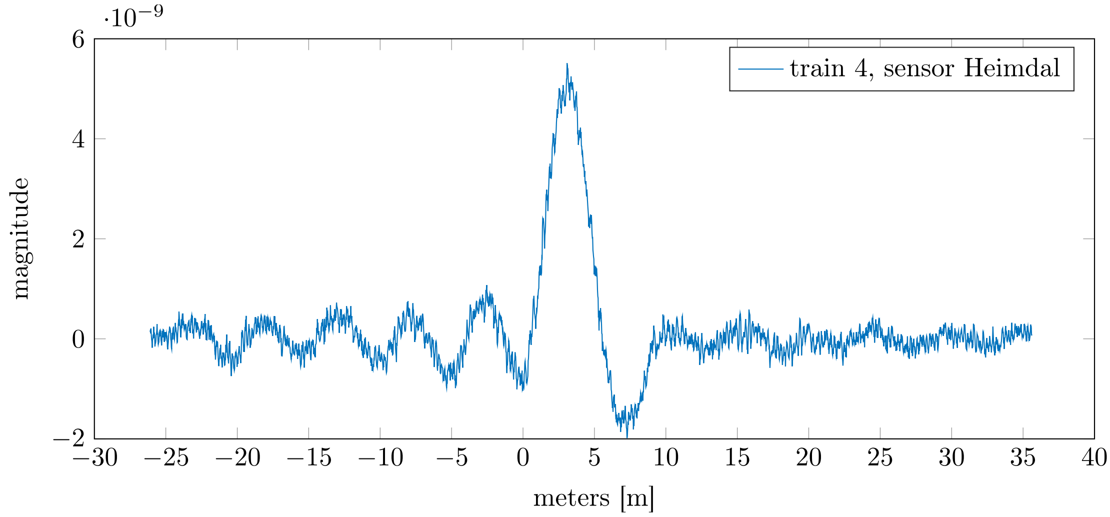


(b) Influence line for middle sensor, train 3

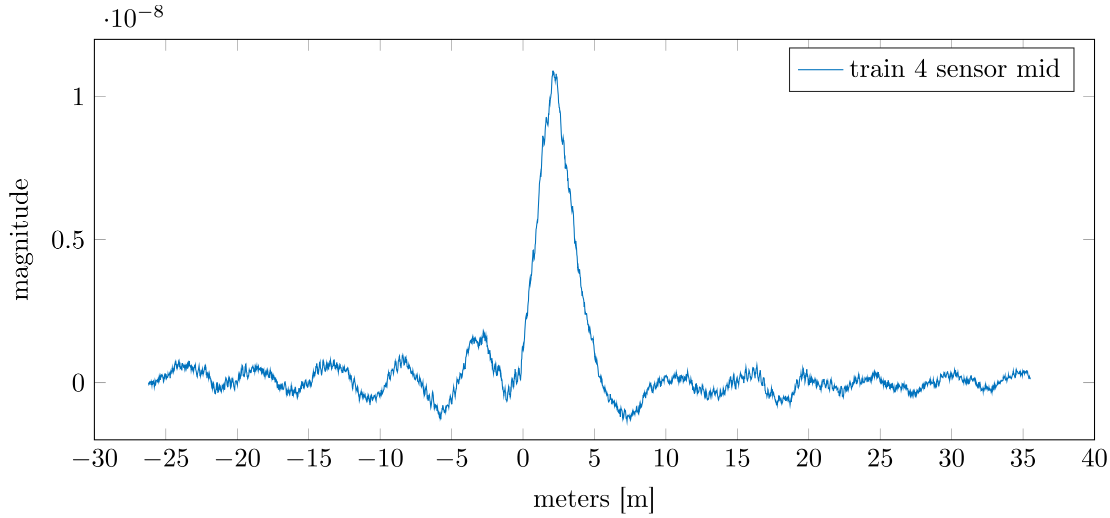


(c) Influence line for sensor closest Trondheim, train 3

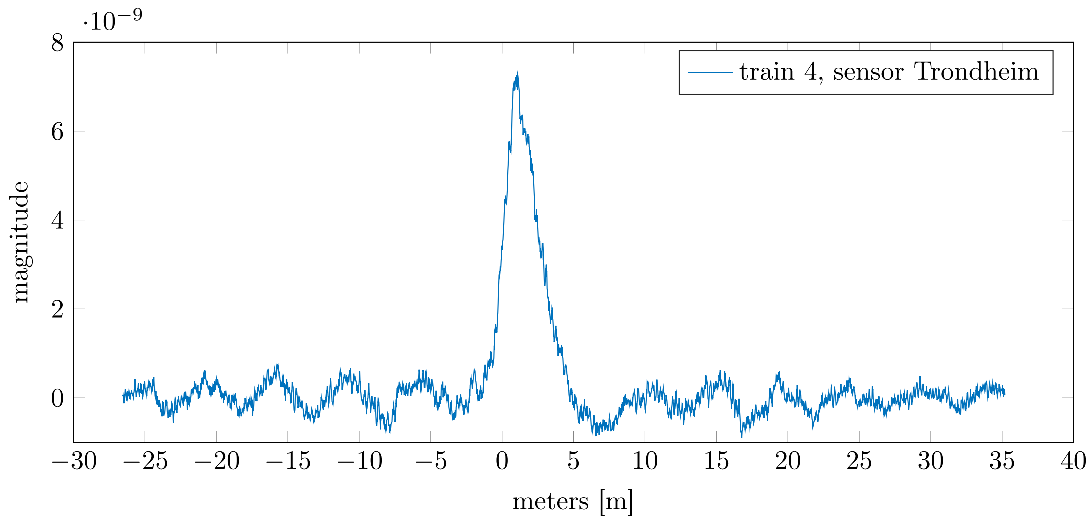
Figure C.5: Influence lines train 3



(a) Influence line for sensor towards Heimdal, train 4

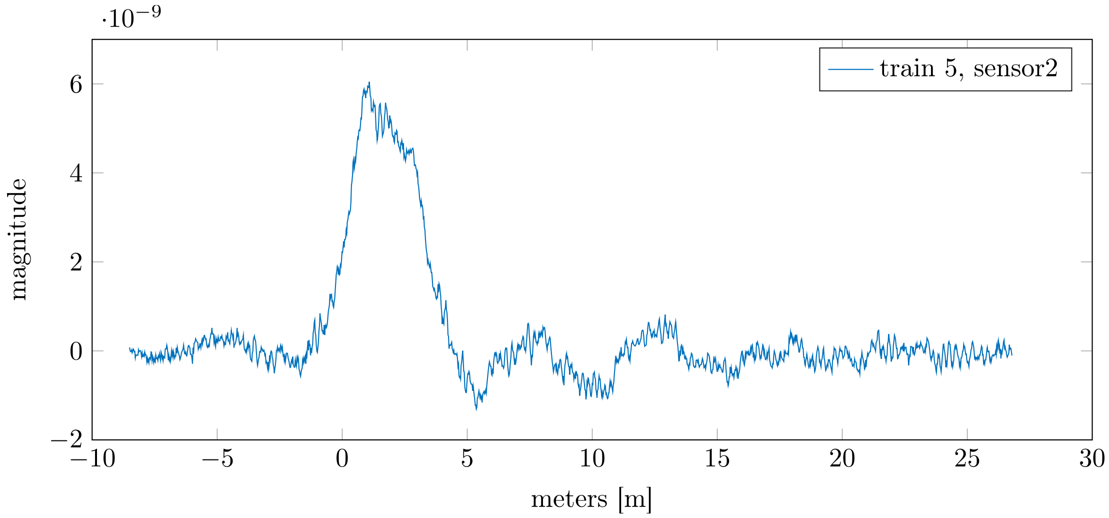


(b) Influence line for middle sensor, train 4

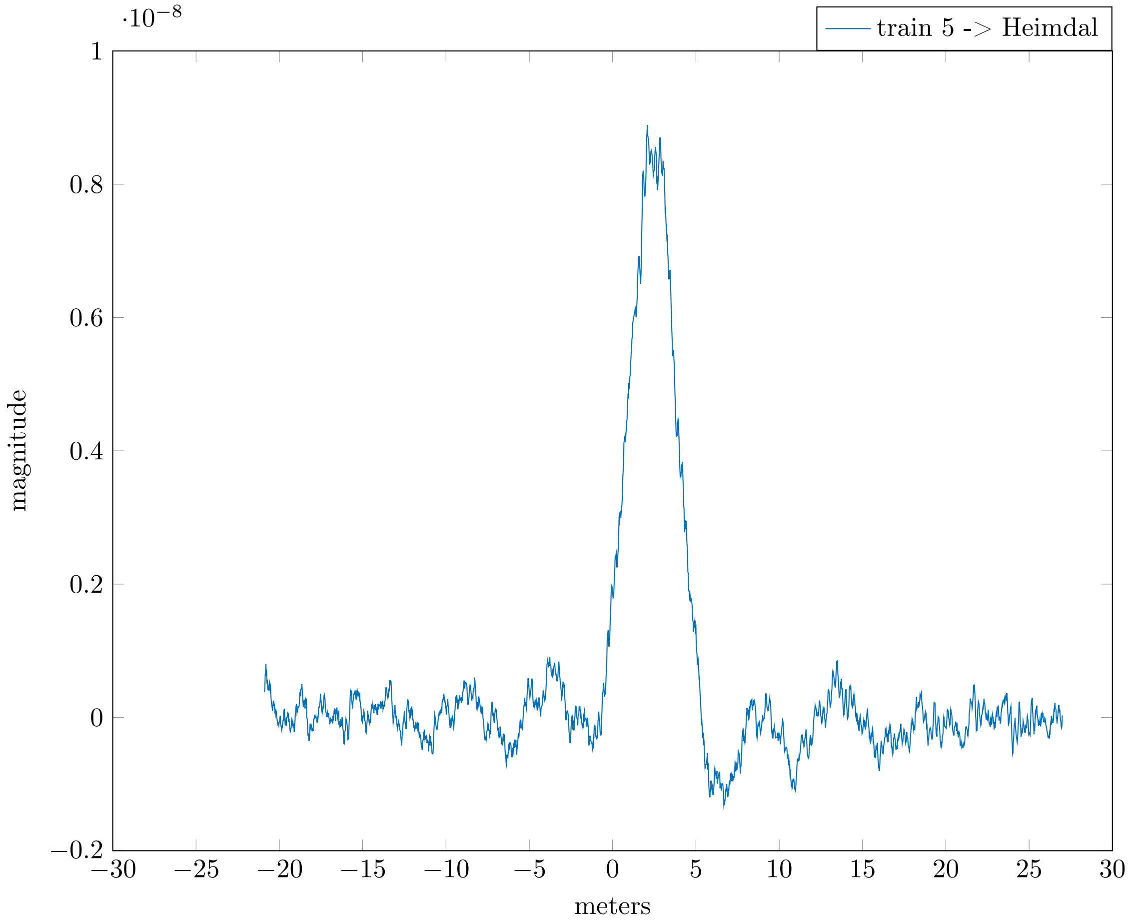


(c) Influence line for sensor closest Trondheim, train 4

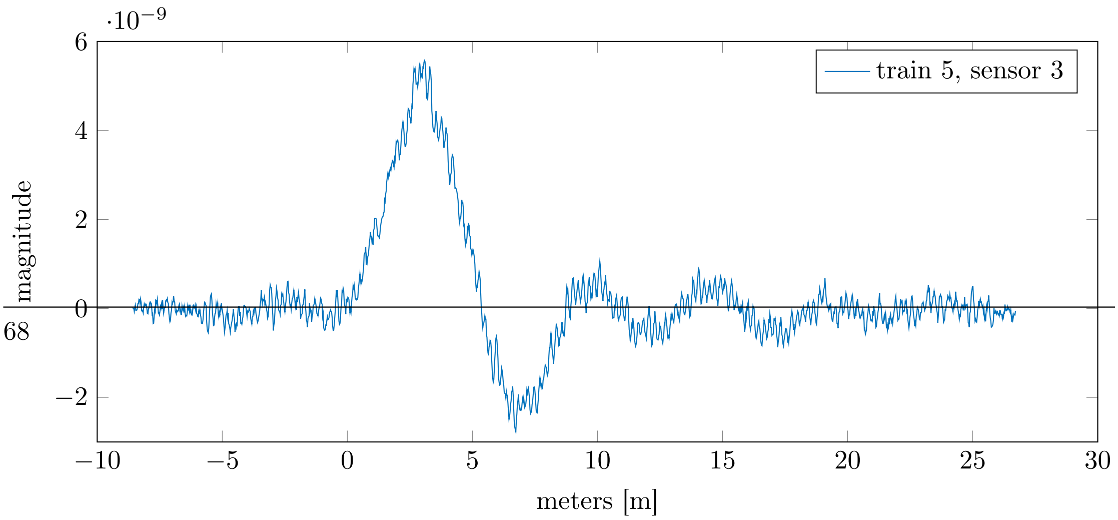
Figure C.6: Influence lines train 4

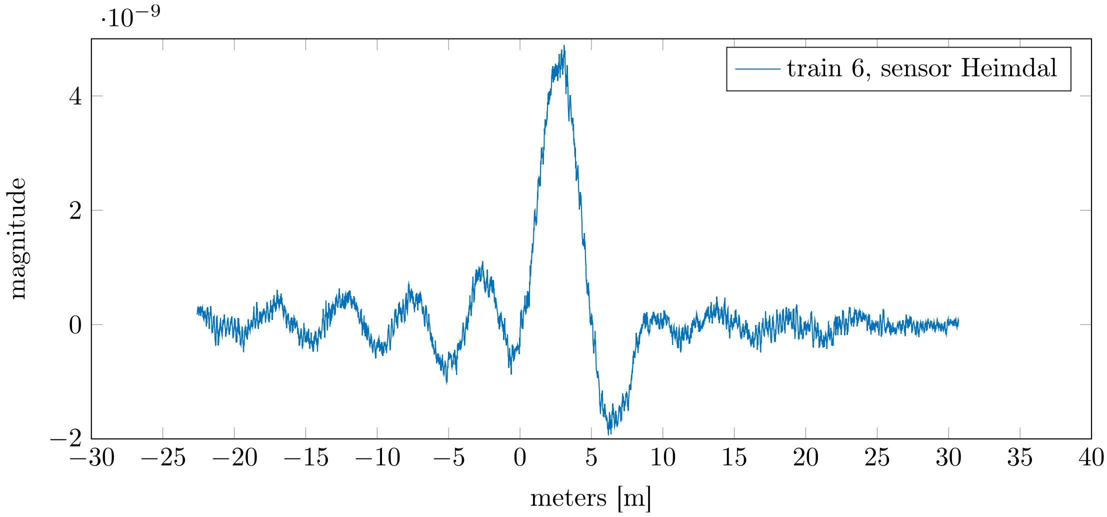


(a) Influence line for sensor towards Heimdal, train 5

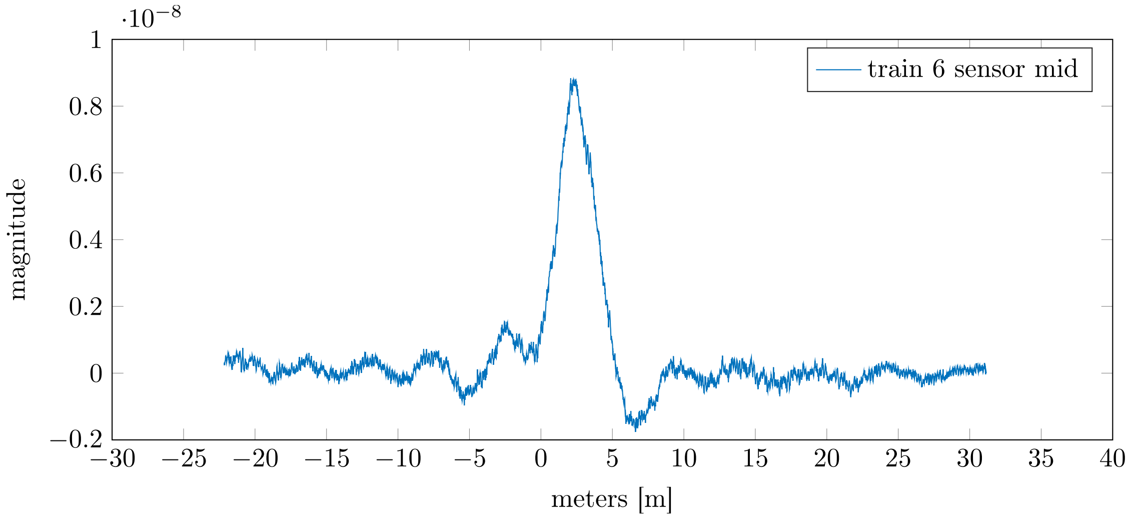


(b) Influence line for middle sensor, train 5

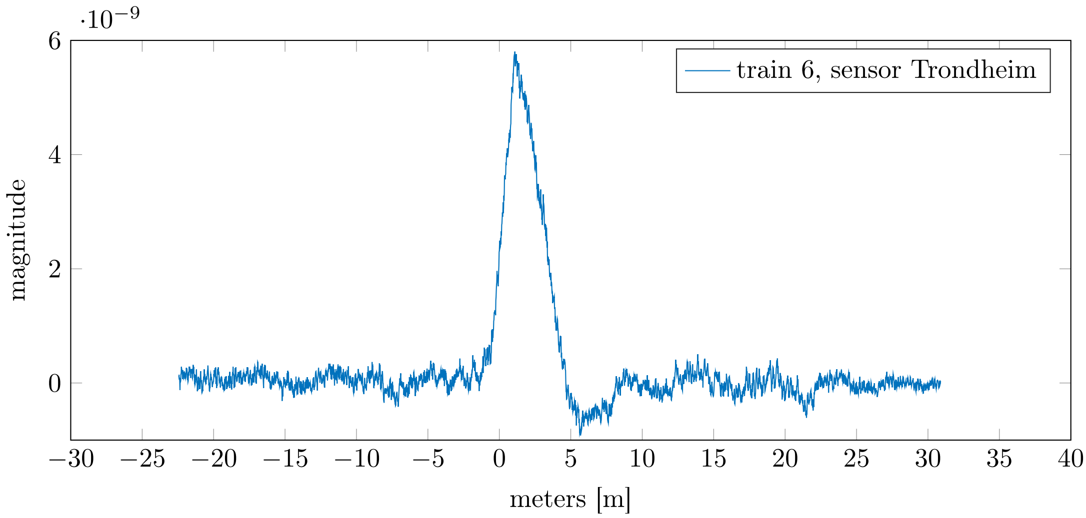




(a) Influence line for sensor towards Heimdal, train 6

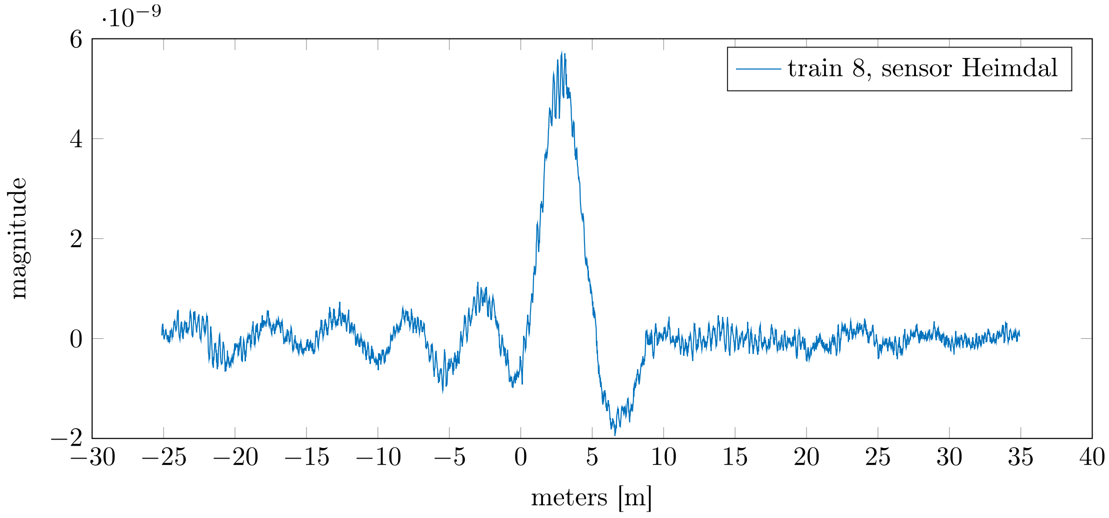


(b) Influence line for middle sensor, train 6

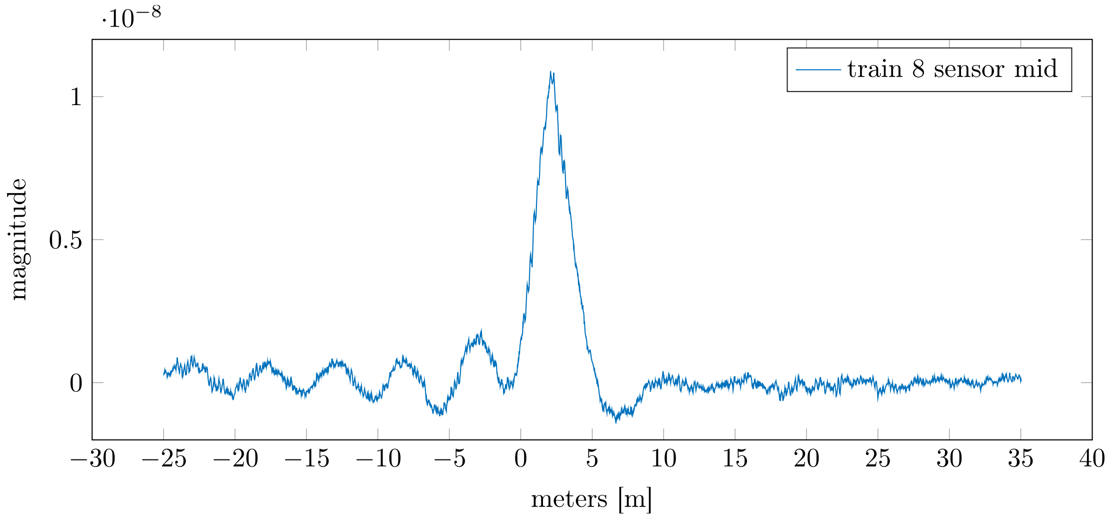


(c) Influence line for sensor closest Trondheim, train 6

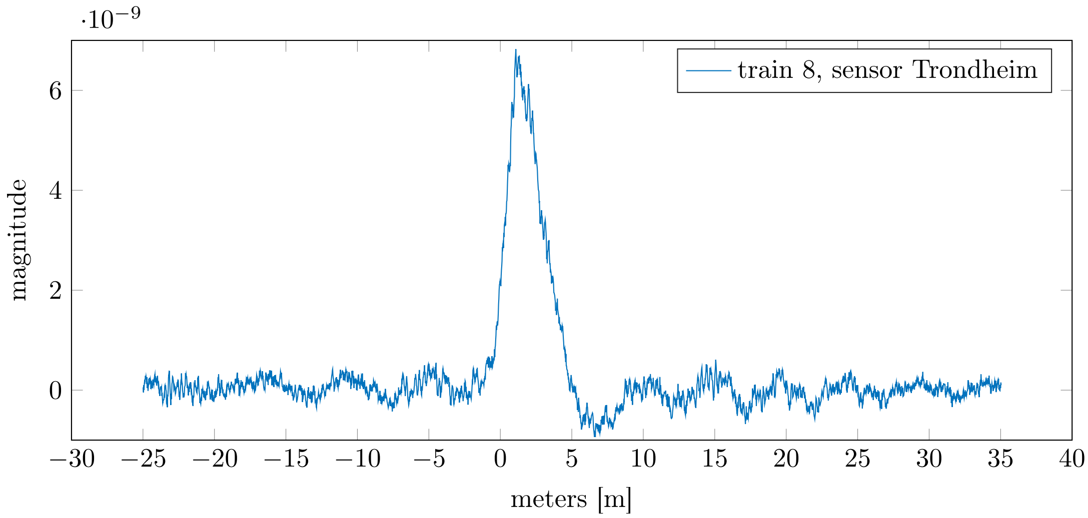
Figure C.8: Influence lines train 6



(a) Influence line for sensor towards Heimdal, train 8



(b) Influence line for middle sensor, train 8



(c) Influence line for sensor closest Trondheim, train 8

Figure C.9: Influence lines train 8

D. Algorithms and code

```
function [  M_c, A, C ] = findInfluenceLines( strainHistory, TrainData)
% This method will compute the necessary matrices and vectors for solving
% the system giving influence lines for a BWIM system
% This will be acomplished through Moses' equation
% The axle loads [A] are known, so the influence ordinates which minimises
% the following equation give best the best solution
% E = sum_from_k=1_to_k=numOfScans((M_k^M - M_k^T)^2)
% M_k^M = measured strain at scan k - measured response
% M_k^T = theoretical response
% TrainData is a matlab struct containg info of train speed, axle spacings
% and so on
% The returned variables: M_c =  a vector depending on axle weights and measured strain
% A = matrix depending only on axle spacing and axle weights
% C = axle distances in signal samples
format long;
speed = TrainData.speed;
frequency = 1/TrainData.delta;
k = length(strainHistory);
n = length(TrainData.axleWeights); % number of axles
C = zeros(1,n);
% The matrix size

C(1) = 0;
% calculates axle distances in signal samples, depending on sampling frequency and train speed
for i = 1:n-1
        C(i+1) = round((sum(TrainData.axleDistances(1:i)))*frequency/speed);
end
% % Defining the matrix size
if C(length(C)) > k
%    Extract the necessary parts of the C vector
    Cnew = C(C<k);
end
m = k-C(length(C));
M_c = zeros(m, 1);
for i = 1:m
```

```
    for j = 1:n
        M_c(i,1) = M_c(i,1) + TrainData.axleWeights(j)*strainHistory(i+C(j));
    end
end

% Now creating the A matrix, which depends on the axle weights
% The diagonal <-> sum of the squares of the axle weights
% The loop only calculates the upper triange of the matrix
A = zeros(m, m);

for i = 1:n
    for j = i:n
        offset = C(j)-C(i);
                if((m) - abs(offset))>0 % axle n does influence strain
            oneVec = ones(1,m - abs(offset));
            diagonal = diag(oneVec,offset);
            A = A + TrainData.axleWeights(i)*TrainData.axleWeights(j)*diagonal;
        end
    end
end
% Form the full matrix through the transpose of the upper triangle
A = A + tril(transpose(A),-1);
end
```