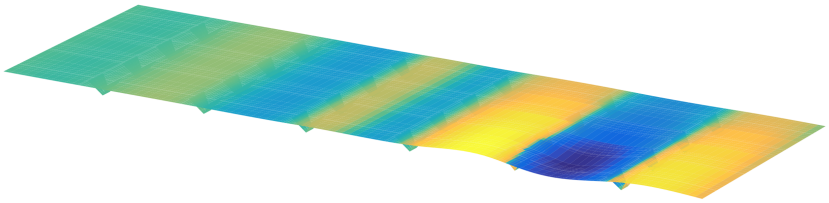
Damage Detection of Concrete Slab Bridges

Jeremy Barisch-Rooney

December 27, 2019



Acknowledgements

I would like to thank...

TODO: Acknowledgements

Contents

List of Figures

List of Listings

List of Tables

1 Introduction

The probability of a bridge to fail increases over time until it is no longer considered safe for use. Maintenance of a bridge is typically carried out when something goes wrong or according to a preventative maintenance schedule based on expert knowledge, neither approach making the best use of limited maintenance resources. The Nanfang'ao bridge collapse in Taiwan on 1 October 2019 in which six people were killed and 12 were injured is a recent example of the importance of maintainence of civil infrastructure. The bridge was only twenty years old.

Sensors can provide real-time information without the delay or cost of a manual maintenance check. What are the costs and benefits of installing a decision support system (DSS) based on real-time sensor data for the purpose of maintenance of a no-prestress no-postension concrete slab bridge? This question is the topic of this thesis, to answer it we need to answer a few sub questions. What analyses of sensor data do and do not provide valuable information to the user of a DSS? How can we collect sensor data for the purpose of analysis? And what are the costs and benefits of installing different types and quantites of sensors on a bridge?

A **decision support system** for bridge maintenance** is a software system that provides the user of the system with information on the current state of a bridge. The provided information should enable the user of the system to make a more informed decision about when and/or where maintenance should be carried out. The provided information can include real-time sensor data and an analysis thereof. The primary aim of the analysis techniques applied to sensor data in this thesis are to classify data in two states, a normal or healthy state, and an abnormal or damaged state.

Structural health monitoring (SHM) is a term used to describe a range of system implemented on civil infrastructure for the purpose of assisting and informing about the structure's "fitness of purpose".

In the development of a DSS for bridge maintenance it is **necessary to have sensor data** corresponding to both the damaged and undamaged states of the bridge in question. Due to the expensive nature of a bridge it is usually and unsurprisingly not permitted to impose a damaged state on a bridge. The use of finite element (FE) software allows us to create simulated sensor data via a finite element model (FEM) of a bridge on which different damage states can be imposed.

Because nearly every structure in civil infrastructure is unique a major part of any SHM system has to be geared towards long term evaluation of what is "normal" of "healthy" performance of the system [1].

A FEM is a model of a structure in software. All models are wrong TODO:REF Mike Lees' referenced something in his lecture and thus the simulated sensor data generated by the FEM will **differ from sensor data collected from sensors installed in real life**. The simulated sensor data should be close enough to reality to provide confidence that the anaysis techniques in this thesis can also work with real sensor data, while acknowleding that additional work would likely be needed to tune the anaysis techniques once real sensor data is available. After all, "in theory there is no difference between theory and practice, while in practice there is", TODO:REF ambiquity of who said this.

A finite element simulation of a bridge can produce sensor data that is necessarily different from data collected from real installed sensors. It is necessary to **verify the simulated data against some ground truth** in order to have confidence in the accuracy of the simulated data. And it is necessary to **test the developed analysis techniques on real data** to have confidence in the techniques, for when a DSS for bridge maintenance is installed in real life.

OpenSees (The Open System for Earthquake Engineering Simulation) is an open-source FE software package that anyone with a Windows, macOS or Linux machine, and an internet connection, can download and install. Depending on open-source rather than proprietary FE software enables users to explore or extend this research, thus allowing for reproducible research.

A FEM of bridge 705 in Amsterdam was available at TNO (Netherlands Organisation for Applied Scientific Research) for the FE software package Diana. This FEM was verified against measurements from an experimental campaign. The existence of a verified FEM for bridge 705 allows the generated FEM for OpenSees to be verified by comparing responses under explicit loading conditions from the OpenSees model of bridge 705 to responses from the verified Diana model under the same load.

This thesis could have gone one of two ways. The verified FEM of bridge 705 for Diana could have been used to simulate sensor responses for analysis. However Diana requires a relatively expensive proprietary licence for use (you must ask for a quote) and the file format of FEMs in Diana is rather awkward to modify. support this claim. By using OpenSees it was easier to target a greater number of bridges, by generating FEMs based on a high-level bridge specification. More importantly however OpenSees does not require a licence for use and is additionaly available for macOS users thus allowing for the research to be reproduced or extended.

Extensibility is a measure of the ability to extend a software system. According to TODO:REF "An important kind of reuse is extensibility, i.e.,

the extension of software without accessing existing code to edit or copy it." The research in this thesis is not just reproduceable but also extensible. The benefit of this research being extensible is that an interested party should be able to download the software and perform research on another bridge or investigate how a new classification technique performs on sensor data, without having to start from scratch. By creating an extensible system it is my hope to facilitate further research in this area and prevent duplication of effort.

TODO:Paragraph on classification

Bridge data corresponding to states normal and abnormal was not available, however data was avilable from viaducts corresponding to two states, high and low temperature. In this thesis the **analysis techniques are tested** on this data to provide, an albeit limited, test that the techniques can perform a classification between states on real data.

This thesis first gives an overview of existing research into machine learning approaches for structural health monitoring (SHM), decision support systems and classification techniques. The methods section presents an in-depth description of how an extensible system is created for the collection of simulated sensor responses, how the inputs to this system should be structured, and what form the data-driven classification experiments will take. In the results section we take a look at the data generated by the data collection system, analyze the results of the classification experiments, and finally, present the costs and benefits of installing a decision support system for bridge maintenance.

2 Preliminaries

2.1 Abbreviations

ANN	Artificial neural network
CTE	Coefficient of thermal expansion
DSS	Decision support software
FEM	Finite element model
NDE	Non-disruptive evaluation
NDW	(Netherlands) National Data Warehouse for Traffic Information
OpenSees	Open System for Earthquake Engineering Simulation
RNN	Recurrent Neural Network
PyPI	Python Package Index

2.2 Definitions

`Bridge`	Model of a concrete slab bridge's geometry
Damage scenario	Healthy or specific damage state of a bridge
Simulation scenario	Combination of damage scenario and traffic scenario
Traffic scenario	Defines the traffic that flows over a bridge

2.3 Pseudocode

Pseudocode in this thesis is given in two formats. For code written in an imperative style the pseudocode will closely resemble the syntax of Python. For type declarations the pseudocode will resemble the syntax of Haskell. For most people the syntax of Python, or a similar imperative style language, will already be familiar. The syntax for data declarations used in this thesis is presented below in Listing **??**.

declarations used in this thesis. This pseudocode declares a type for describing a vehicle in terms of two pieces of data, a list of axle distance and a list of loads per wheel. Note that `foo :: bar` indicates a field of name `foo` and type `bar`. The square brackets represent the list type and the curly brackets represent a tuple type.

```
-- A vehicle as a list of axle distances and wheel loads.
data Vehicle {
    -- Distance betwen each pair of axles.
    axleDistances :: [Float],
    -- A tuple per axle, of wheel load intensity in kilo Newton.
    wheelLoads :: [(Float, Float)]
  }
```

Sometimes the reader will be presented with a type signature that looks like `foo :: A -> B -> C`. A type signature describes the input and output types of a function, the previous type signature can be read as "a function `foo` that takes an argument of type `A`, an argument of type `B`, and returns a value of type `C`". Another example is `bar :: [Float] -> Int` which can be read as "a function `bar` which takes a list of `Float` as input and returns a value of type `Int`".

3 Literature Review

The goal of this Section is to summarise the existing body of work related to the subject area of this thesis. This Section thus outlines the existing work on damage identification of civil infrastructure, traffic and bridge simulations, and real-life installations of sensors on bridges for the purpose of structural health monitoring.

3.1 Damage Identification

Methods of damage identification by monitoring changes of civil infrastructure include methods based on modal properties, methods based on a model-updating procedure, probabilistic approaches e.g. using Bayes theorem, and pattern recognition approaches such as artificial neural networks.

A significant amount of the research into damage identification of civil infrastructure is based on modal properties, in particular with focus on changes in natural frequencies and mode shapes.

Damage was applied to the I-40 bridge, a 130m girder bridge over the RIO Grande river, before it's demolition and data recorded form ambient vibration tests. In [2] it is noted that no change in dynamic properties were observed except in the fourth and most severe damage case. Furthermore, changes of similar magnitude were observed from ambient vibration tests on the undamaged structure at different times, indicating the methods are not robust to environmental noise.

In [3], four years later, the mode shapes from the same vibration tests on the I-40 were used to detect and localize the damage. The modal properties were shown to be statistically different from the state without additional damage imposed, for all damage cases. However the location of damage was only identifiable once the damage increased to a point that the bridge would have collapsed under a live load. TODO: which modal properties

In [4] changes in natural frequency and mode shapes from numerical simulations are used to determine the location and the extent of damage on a rigid frame and then to assess the safety of the structure. However this paper highlights two issues common in the literature. Modal parameters corresponding to a baseline or "healthy" state are required, and robustness to noise is not addressed in the work. The requirement of "baseline" data is not a fatal flaw and could be addressed in a number of ways: 1) the baseline state comes from sensor measurements taken for newly built structures, 2) existing structures could be monitored for *any* changes after sensor installation, not knowing whether the structure was already damaged or not, 3) a FEM is

used to generate an approximation of the baseline state. The robustness to noise is a more crucial problem because civil structure will be subjected to environmental factors such as temperature changes, the work [4] simply states "the existence of noise in the data processing should be addressed". TODO: More examples not addressing noise.

Compared to natural frequency, mode shapes have the advantage of capturing global information of the structure and thus allow for damage localization. In [5] introduced the use of the curvature of mode shapes which is obtained by differentiating the displacement mode shape twice. Changes in the curvature of the mode shape are localized to the damage and furthermore the absolute difference of the curvature mode shapes of the damaged and undamaged structures increase with increase in damage severity [6].

The 64m concrete Dogna bridge in Italy was built in 1978 and suffered from a strong flood in 2003. In 2008, prior to demolition, an experimental campaign was carried out where six damage configurations were applied to the bridge in the form of notches cut with a hydraulic saw. In [7] changes in modal curvature were succesfully used to identify the location of the damage. However the dynamic tests were all carried out under similar environmental conditions, thus the robustness to noise was not investigated.

A damping ratio is a dimensionless property that describes how the oscillations in a system decay after a disturbance.

Model updating approaches compare measurement data with responses from an analytical model. A model updating algorithm is then used to minimize the difference between analytical model's output and the measured response, which will be achieved by modifying some parameter of the model, such as the stiffness matrix in [8]. A problem with model updating approaches is that optimization algorithms may reach one of many local optimums. The local optimum that is found will depend on the initial parameters, which may be innacurate. In [9] a model-updating approach is applied which minimizes the difference in mode shapes. This approach was validated on the Z24 highway bridge in Switzerland, which is a 58m prestressed concrete bridge. The damage scenario considered is the lowering of one of the supporting piers (at 44m) by 95mm. {TODO: Were potential false positive scenarios considered?}

A Bayesian probabilistic approach was applied to a reinforced-concrete bridge column [10], this method compared the relative damage probabilites of different damage events based on data from vibration tests.

Methods based on ANNs [11] are based on labeled training data for different damage types, these methods depend largely on the quality of training data.

The most basic form of damage identification is determining whether damage has occured or not (detection). More sophisticated SHM methods attempt to determine the location (localization) or extent (assessment) of the damage. require either severe damage or make assumptions about the type of damage [12]. While an SHM system should be capable of a minimal amount of condition assessment (CA) the more likely scenario is an additional investigation for CA is triggered by the system [13].

3.2 Machine Learning

A number of damage identification experiments were applied on an aircraft wing showing damage localization and assessment to be possible with machine learning methods [14], however the experiments were in a controlled lab setting without environmental factors present. The same paper argues that damage prediction cannot be addressed by machine learning methods in general. why

A clustering approach was applied to detect damaged joints on the Sydney Harbour Bridge (SHB) [15]. This method attempts to group joints with similar behaviour and then detect damage when a sensor's responses are different from those in its group. This method succesfully detected a damaged joint and a joint with a damaged sensor.

3.3 Damage Types

Damaged sensors can be considered a special class of damage that is very important to identify because faulty data can interfere with damage detection of a structure. Damaged sensors can be detected via sensor data reconstruction. In this approach the sensor data is reconstructed based on spatial and temporal correlations among the sensor network. If there are discrepancies between the measurement data and the reconstructed data then the sensor may be faulty. Spatial correlations are used to reconstruct sensor data via principal component analysis [16], minimum mean square error estimation [17], support vector regression [18] and ANNs [19, 20]. [21] use a recurrent neural network (RNN) that takes into account spatial and past temporal data. In [22] a bidirectional RNN considers spatial and both past and future temporal correlations.

3.4 Simulation

Data for damage detection on bridges can be collected by simulating loads (vehicles) moving across the bridge deck. Damage detection on bridges using

neural networks was based on simulated vibration data in [11], in this case the "traffic" was simply a single truck simulated with two point loads, one per axle. The load traveled at a different constant speed in each simulation.

3.5 Environmental Factors

To avoid incorrect detection of anomalies due to environmental factors such as changes in temperature or a passing tram, the feature extraction of data is crucial in SHM. Feature extraction is arguably the most important and difficult step in ML-based health monitoring [14].

The temperature across a bridge deck was measured during a 24 hour cycle and correlated with the bridge's natural frequencies in [23]. A machine learning algorithm based on PCA is presented in [24] for separating the individual components of the deflection signal. An auto-associative neural network is employed for separating the effect of damage in extracted features from that caused by the environmental variations of the system [25].

When the noise level was under 10%, each individual component (temperature, live load, structural damage) was succesfully separated based on data from a computer model of a long-span bridge.

3.6 Health Monitoring Installations

3.7 Summary

Tables 3.7 and 3.7 summarize the literature which is most relevant to this thesis. The tables break down the literature into a number of criteria, in order to give a high-level view of the contributions of the literature. Table 3.7 breaks down the literature into bridge type, damage type, how traffic was simulated and if the FEM was validated. Table 3.7 gives an overview of the damage identification methods for the same literature as Table 3.7.

Comment on reproducability

Paper	Bridge type	Damage type	Traffic simulation	Validation of FEM
[11]	214m Suspension	Corrosion and loosening of joints	One vehicle represented as two points loads	"Reasonable agreement" with measurements

Paper	Measurement	Feature	Analysis	Noise	Accuracy
[11]	Vibration	Frequency spectra	ANN	“Moderate”	70%

4 Background and Motivation

The Literature Review provided information on *what* has already been achieved in the subject area of this thesis. This Section however focuses on the question of *why*. Why should sensors be installed on concrete slab bridges for the purpose of anomaly detection? This Section will provide background information to the reader, answering this questin and motivating this thesis. Doing so this Section will examine the different types of Bridges, types of damage that occur, how health monitoring takes place, and why an extensible system for researching damage detection on bridges is desirable.

4.1 Existing Bridges

The Dutch national main road network consists of 3,200km of road. Assets in the road network are divided into four categories: pavements, structures, traffic facilities and environmental assets. Each structure is categorized into a type that has its own maintenance characteristics. Table 1 outlines the categorization of the 3,283 structures in the network.

Table 1: Structures in the Dutch national main road network. Each type of structure has its own maintenance characteristics. The table lists for each structure type the total number in the Dutch national main road network and the total deck area.

Stucture type	Number	Deck Area (m2)
Concrete bridge	3,131	3,319,002
Steel bridge (fixed)	88	301,997
Movable bridge	43	347,876
Tunnel	14	475,228
Aqueduct	7	86,491
Total	3,283	4,530,593

4.2 Bridge Maintenance

In this subsection we briefly review the cost of bridge maintenance, with a focus on Dutch concrete slab bridges. [26] should be considered the de-facto reference for this subsection.

Bridge maintenance is a requirement in the life-cycle of a bridge in order to extend the life of a bridge and keep it within operational conditions. The aims of bridge maintenance are

- Effective management of operational programs
- Realistic budgeting at national level
- Tuning bridge mainagement with other maintenance programs

TODO: Paragraph on overview of operational programs

Bridges are a type of structure that require a large investment, though they also have a long service life of 50 to 100 years. Annual maintenance costs are relatively small compared to the initial investment cost (<1%), however over the lifetime of the bridge the maintenance costs are on the order of the initial investment. the annual maintenance cost and the cost of replacement are given for each type of structure in the Dutch national main road network in table 2.

Table 2: Annual maintenance cost and cost of replacement in millions of euros, for each type of structure in the Dutch national main road network.

Stucture type	Total Replacement Cost (€M)	Annual Maintenance Cost (€ M)
Concrete bridge	6,600	37
Steel bridge (fixed)	600	7
Movable bridge	1,100	10
Tunnel	1,700	13
Aqueduct	250	1
Total	10,250	68

The maintenance cost of a concrete bridge can be estimated by determining the maintenance cost of frequently used components such as concrete elements, extension joints and bearings. These costs estimates of the frequently used components first require a description of minimal acceptable condition of the components. Then, in combination with an estimation of maintanence intervals (which can come from subjective and conflicting sources) and prioritization of the available budgets, a maintenance plan of a bridge can be presented. An example of such a plan for a typical concrete highway bridge is shown in Figure 1.

The Dutch national road network contains over 3,000 highway bridges. Of these, most are 30 or more years old. A significant amount of bridges were constructed in the 1970s, which is typical for many Western European road networks. Fitting a Weibull distribution to the lifetime of demolished concrete bridges suggests an expected lifetime of 41 years. This in turn would mean that the many concrete bridges constructed in the 1970s and earlier

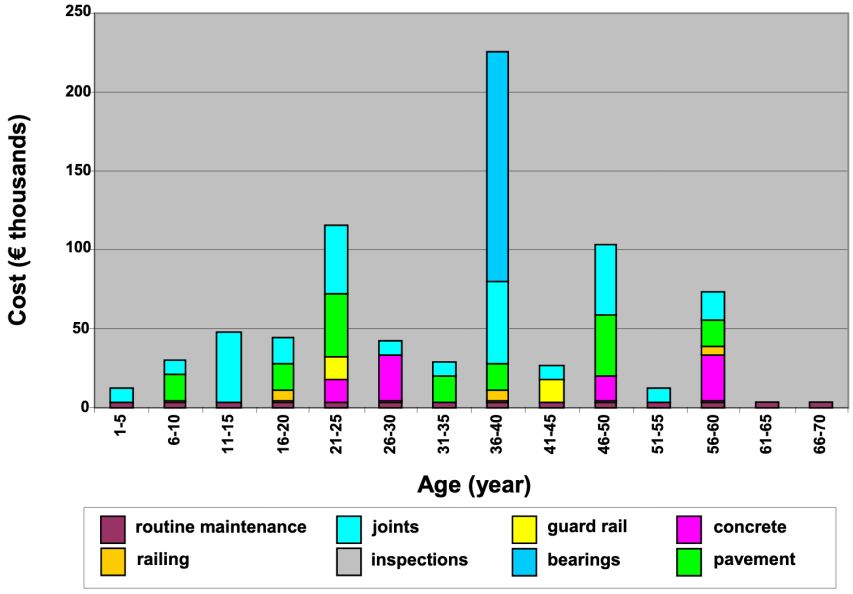


Figure 1: The maintenance cost of a typical concrete highway bridge. The y-axis shows the cost in thousands of euros. Each bar is for a period of five years and the cost is based on underlying components as indicated by the legend.

would be due for replacement. However, of these demolished bridges, many were demolished due to a change in functional or economical requirements, rather than due to technical failure. Including the ages of current bridges in the fitted distribution increases the expected lifetime to 75 years, which is more in line with the design for 80 years of most Dutch highway bridges, design codes in the Netherlands require a design lifetime between 50 and 100 years.

Figure 2 shows an initial peak in the expected cost of replacement of Dutch bridges, this is due to a combination of the distribution of when the current bridges were originally built (largely in the 1970s), their expected lifetime and their replacement cost. In an aging bridge stock the cost of maintenance can be assumed constant, averaged over the large number of structures. After a long time the cost of replacement will be approximately 85€ million, approximately half the cost of annual maintanence of concrete bridges at 37€ million.

4.3 Damage Types

Damage scenarios can be classified as short-term or long-term. Short-term damage scenarios are defined as a change of the properties of structural materials and elements, and of the behaviour of the whole structure, due to effects that occur during a very short period of time. Long-term scenarios are time-dependent and may not only be related to external factors but also due to a change of state of materials with time. Tables 3 and 4 [27] outline some of the predominant types of damage due to short-term and long-term scenarios respectively.

TODO: Factor examples/consequences out of table

Table 3: Types of damage due to short-term events.

Event	Examples/Consequences
Collision	Impact by overweight vehicle or boat in the river
Blast	Impact by vehicle followed by explosion
Fire	Impact by vehicle followed by explosion and fire
Prestress loss	Sudden failure of a prestress tendon
Abnormal loading conditions	Loading concentration and/or overloading in a specific site along the l
Excessive vibration	Earthquake
Impact	Impact pressure by water and debris during floods

When damage is detected based on sensor measurements another possibility is of course that the sensor is itself faulty. Sensors can become faulty

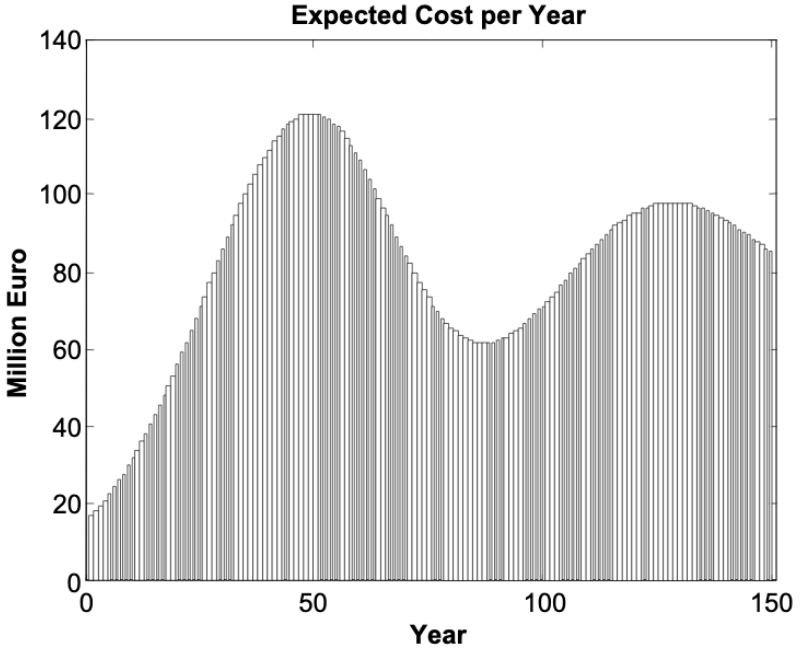


Figure 2: The expected cost of replacement of concrete bridges in the Dutch national main road network. The expected cost is calculated by summing over all concrete bridges, their ages and replacement costs. The initial peak is largely due to a surge in construction around the 1970s. The cost of replacement will tend to 85€ million in the long run.

Table 4: Types of damage due to long-term events.

Event	Examples/Consequences	Critical comp
Corrosion	Degradation of the bearings	Deck
	Loss of cross-section area in the prestressing tendons	Deck
Time-dependent properties of the structural materials	Excessive creep & shrinkage deformations	Deck
	Concrete deterioration	All
Low stress - high frequency fatigue	High frequency and magnitude of traffic loads	Deck
High stress - low frequency fatigue	Temperature induced cyclic loading	Abutment
Environmental effects	Freezing water leading to concrete expansion	All
Water infiltration/Leaking	Deterioration of the expansion joints; concrete degradation in the zone of the tendon anchorages	Deck
Pier settlement	Change in the soil properties	Deck

for a number of reasons, increased noise, bad installation, battery issues, harsh environment etc. [28]. Maintaining a healthy sensor network is important because faulty sensors can cause not only permanent loss of data but also inaccurate damage detection, if for example the detection system is being trained on the faulty sensor data.

4.4 Health Monitoring

In this subsection we review some of the current methods of health monitoring of bridges. **(author?)** [12] should be considered the default reference for this subsection.

Current state of the art health monitoring methods only indicate whether damage has occured in a bridge, not determining the location or severity of the damage. verify by looking at more recent methods This class of methods are referred to as "global health monitoring" methods. Global health monitoring methods are considered sufficient since knowing that damage has occured will allow a more accurate inspection to take place.

Local health monitoring refers to methods that find the location and possibly the extent of damage. Non-destructive evaluation (NDE) refers to methods of determining the location of damage without damaging the structure, for example with guided ultrasonic waves. NDE can be time consuming and expensive, and access to a location on the bridge may be difficult or not even possible.

In the USA the Federal Highway Administration (FHA) requires that the condition of bridges be evaluated every two years. Such an inspection typically takes the form of a tap test. A tap test is a test where the surface of the bridge is tapped in order to find variations in the sound response. However the tap test is limited to finding damage near the surface and in cases, significant cracks. And consider that in the USA there are over 500,000 highway bridges with a span length of over 7m, inspection of all these bridges with a limited budget and staff not always achievable.

Most global health monitoring techniques are based on finding changes in resonant frequencies or mode shapes. However for concrete structures the deterioration of reinforced steel has little effect on natural frequency. Some methods attempt to find the location and length of cracks based on natural frequency, however these methods assume the only damage is cracking, an assumption that will certainly not always be true in a real-life setting.

Damage detection methods based on Bayes' theorem attempt to determine the most likely damage event by comparing the relative damage probabilities of different damage events. Techniques based on artificial neural

networks (ANN) require training data to map from a measurement to a damage scenario. The quality of the trained neural network depends on the quality of the data, "garbage in, garbage out".

Bridges will be subjected to environmental factors that affect the responses recorded by sensors. If the change in responses from environmental factors is considered as noise, then the change from damage must be notably greater than the change from noise in order for the health monitoring technique to detect that damage has occured. Methods which compare responses to baseline responses from the undamaged structure will suffer from this problem. Methods based on novelty detection do not require baseline data but typically do not provide a measure of the damage severity.

Health monitoring based on an analytical model can be challenging because the required data for building an analytical model is not always available. This is because civil infrastucture is not always built precisely to the original design due to changes in orders and due to on-site construction constraints. Moreover, concrete does not guarantee uniform material properties, which might be assumed in an analytical model.

4.5 Sensor Technology

4.6 Extensibility

In order for the developed DSS to be truly extensible it is not limited to depend on a single finite element program. The system has as a parameter a method of communication with a finite element program, such that data can be collected and analyzed from different finite element programs, in this case OpenSees and Diana.

Due to the expensive nature of installing sensors in real life and of damaging a bridge which is likely prohibited, the software system includes a component for simulating sensor responses from reinforced concrete bridges. In order for this simulation to be extensible and allow for further research on bridges other than bridge 705, the specification of the bridge is simply a parameter of the system.

The developed decision support system has a number of **parameters** such that users wishing to extend the software further are not limited to focus on bridge 705 or to use a specific finite element program. The specification of a bridge is a parameter of the system, as is the type and intensity of traffic on the bridge. Furthermore, as mentioned earlier, different finite element programs can be integrated with this system, which may be useful if a finite element model of a bridge for a different finite element program is already

available to the user.

For a software system to be extensible, the source code must be available to any user wishing to extend said software. The benefits of **open source software** are well known, in particular open source software allows *any individual with an interest* to develop or *extend* the software. Open source software can thus leverage the knowledge of the community and prevent duplication of efforts which can occur when software is developed behind closed doors. Open source software also provides transparency to anyone wishing to investigate the software and may produce more reliable software due to more people having eyes on it.

extending to other types of bridges

Existing Work

This section contains a review of the most relevant material studied during this thesis work. The section begins with an overview of related works followed by a more in-depth look at the most relevant material. The aim of this section is to place the thesis in context and to provide background information to the reader on employed techniques. The section concludes by relating the reviewed material back to this thesis.

An overview

TODO: overview of related works

The application of machine learning to structural health monitoring

[14] illustrates the utility of a data-driven approach to structural health monitoring (SHM) by a number of case studies attempting to identify damage on an aircraft wing. In particular the paper focuses on pattern recognition and machine learning (ML) algorithms that are applicable to damage identification problems.

The question of *damage detection* is to identify if a system has departed from normal (i.e. undamaged) condition, simple "is there damage or not?". The more sophisticated problem of *damage identification* seeks to determine a greater level of information on the damage status, even to predict the future of the situation. The problem of damage identification can be considered as a hierarchy of levels as described in [29].

- Level 1. (Detection) indication that damage might be present in the structure.
- Level 2. (Localization) information about the probable position of the damage.
- Level 3. (Assessment) an estimate of the extend of the damage.
- Level 4. (Prediction) information about the safety of the structure.

[14] argues that ML can provide solutions to these problems upto level 3, but that in general level 4 cannot be addressed by ML methods.

Applying ML for the purpose of SHM is usually only a single step in a broader framework of analysis. Figure 3 shows the waterfall model ([30]) which begins with sensing (when to record responses) and ends with decision making. ML methods are only step four in this model. An important part of this entire process is feature extraction, step three, which can be regarded as a process of amplification, transforming the data to keep only information that is useful for the ML analysis. Another aim of feature extraction is to reduce the dimensionality of the data, to avoid the explosive growth of the data requirements for training with the data dimensions, known as the *curse of dimensionality* TODO:REF.

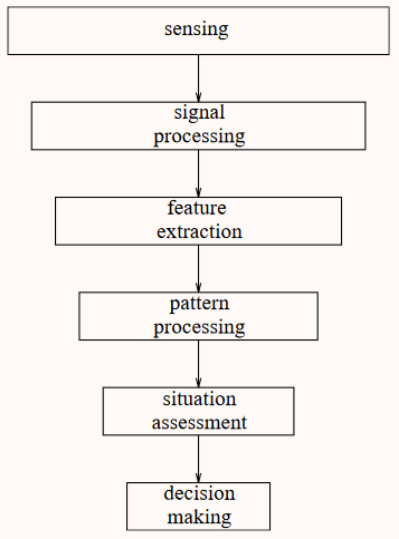


Figure 3: The *waterfall* model.

An experiment was setup to identify damage on the wing of a Gnat artefact. Damage scenarios for testing were created by making a number of cuts into copies of the wing panel. Transmissibility between two points was chosen as a measurement based on success in a previous study TODO:REF, it is the ratio of the acceleration spectra between two points $A_j(\omega)/A_i(\omega)$. This was measured for two pairs of perpendicular points on each wing; in the frequency range 1-2kHz, which was found to be sensitive to the type of damage investigated. The measurements were transformed into features for novelty detection by manual investigation of 128-average transmissibilities from the faulted and unfaulted panels, selecting for each feature a range of spectral lines as shown in TODO:FIG. 18 features were chosen.

To address the first level of Rytter's hierarchy, damage detection, an outlier analysis was applied. This outlier analysis calculates a distance measure (the squared Mahalanobis distance) for each testing observation from the training set. 4 of the 18 features could detect some of the damaged scenarios and could detect all of the unfaulted scenarios, other features produced false positives and were discarded. Two combined features managed to detect all damage types and raised no false positives.

The second level of Rytter's hierarchy is damage localization. This problem can be approached as a regression problem, however here it is based on the classification work done for damage detection where transmissibilities are used to determine damage classes for each panel. A vector of damage indices for each of the panels is given as input to a multi-layer perceptron (MLP) which is trained to select the damaged panel. The paper argues that "it may be sufficient to classify which skin panel is damged rather than give a more precise damage location. It is likely that, by lowering expectations, a more robust damage locator will be the result". This approach has an accuracy of 86.5%, the main errors were from two pairs of adjacent panels, whose damage detectors would fire when either of the panels were removed. The approach depends on the fact that damage is local to some degree, and the damage detectors don't fire in all cases, which was true in this case.

, the assessment was based on the previous detection technique.

Neural Clouds for monitoring of complex systems

In one-class classification, a classifier attempts to identify objects of a single class among all objects by learning from a training set that consists only of objects of that class. One-class classifiers are useful in the domain of system condition monitoring because often only data corresponding to the normal range of operating conditions is available. Data corresponding to the class of

abnormal conditions, when a failure or breakdown of a system has occurred, is often not available or is difficult or expensive to obtain.

The Neural Clouds (NC) method presented in [31] is a one-class classifier which provides a confidence measure of the condition of a complex system. In the NC algorithm we are dealing with measurements from a real object where each measurement is considered as a point in n-dimensional space.

First a normalization procedure is applied to the data to avoid clustering problems in the subsequent step. The data is then clustered and the centroids of the clusters extracted. The centroids are then encapsulated with "Gaussian bells", and these Gaussian bells are normalized to avoid outliers in the data.

The summation of the Gaussian bells results in a height `h` for each point `p` on the hyperplane of parameter values. The value of `h` at a point `p` can be interpreted as the probability of the parameter values at `p` falling within the normal conditions represented by the training data.

In comparison to other one-class classifiers, the NC method has an advantage in condition monitoring in that it creates this unique plateau where height can be interpreted as probability of the system condition. Figure 4 shows this plateau in comparison with other one-class classifiers, Gaussian mixture and Parzen-window.

It is important to note that when significant changes occur in the normal state of the system, perhaps due to environmental changes, then the NC classifier should be retrained in order to avoid a false alarm. However, if a NC classifier is continually being retrained with real-time data then it may not detect a gradual long-term change to the system.

Combining data-driven methods with finite element analysis for flood early warning systems

In [32] a system for real-time levee condition monitoring is presented based on a combination of data-driven methods and finite-element analysis. Levee monitoring allows for earlier warning signals incase of levee failure, compared to the current method of visual inspection. The problem with visual inspection is that when deformations are visiable at the surface it means that levee collapse is already in progress.

Data-driven methods are model-free and include machine learning and statistical techniques, whereas finite-element analysis is a model-based method. One advantage of data-driven methods are that they do not require information about physical parameters of the monitored system. As opposed to finite-element analysis which in the case of levee condition monitoring requires parameters such as slope geometry and soil properties. The model-

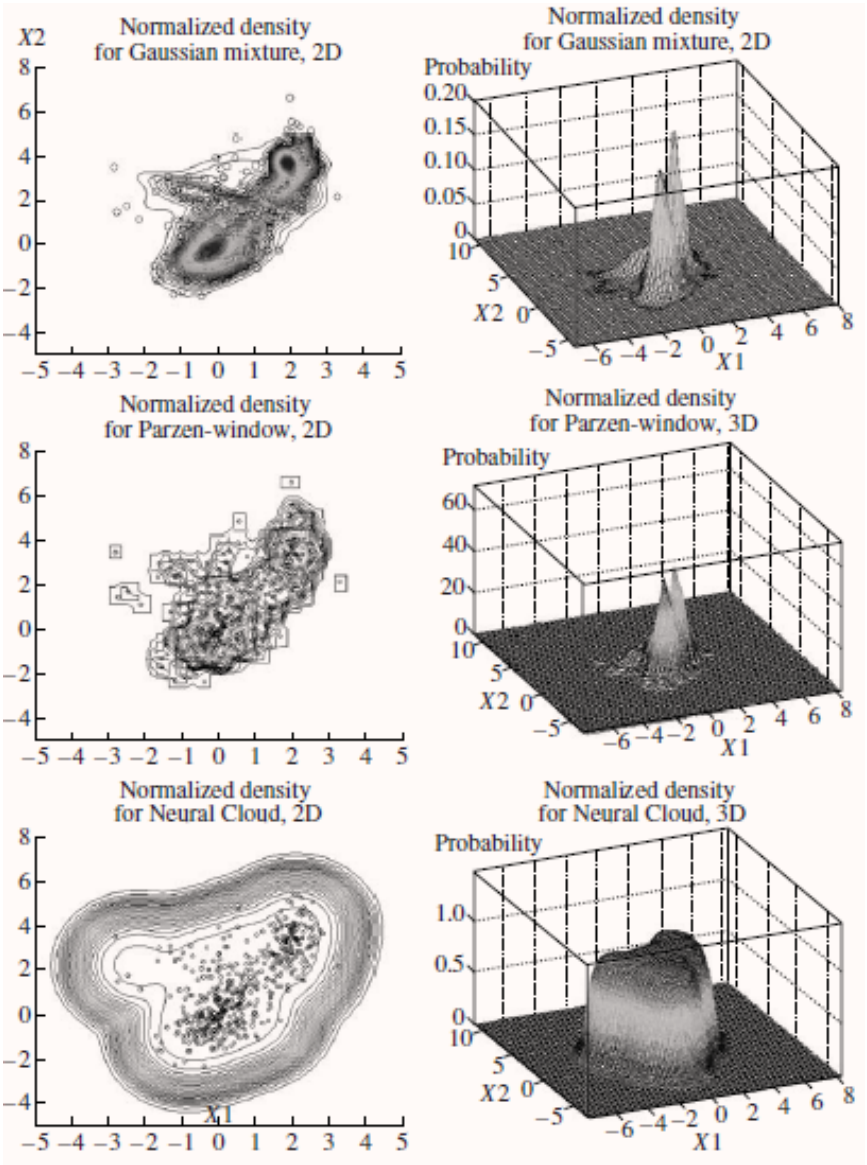


Figure 4: Comparison of Neural Clouds with other approaches, namely Gaussian mixture and Parzen-window. At the left side 2D contour line plots are pictures and at the right normalized density 3D plots.

based methods provide more information about the monitored object, but are more expensive to evaluate and thus difficult to use for real-time condition assessment.

In this paper the data-driven and finite-element components of the system which were developed are referred to as the Artificial Intelligence (AI) and Computer Model (CM) respectively. The AI and CM can be combined in two ways. In the first case the CM is used for data generation. Data is generated by the CM corresponding to normal and abnormal conditions. The normal behaviour data is used to train the AI and both the normal and abnormal behaviour data can be used for testing the AI. In the second case shown in Figure 5 the CM is used for validation of the alarms generated by the AI. If the AI detects abnormal behaviour then the CM is run to confirm the result. If the AI was correct a warning is raised, else the new data point is used to retrain the AI.

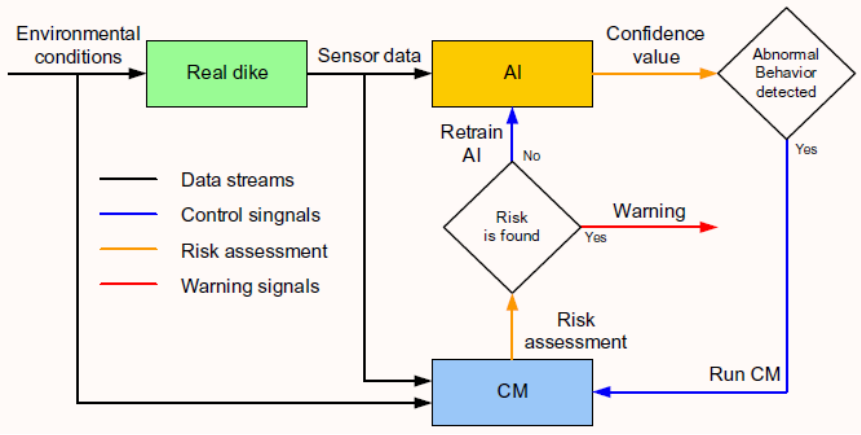


Figure 5: AI and CM...

Flood early warning system: design, implementation and computational modules.

In [33] a prototype of an flood early warning system (EWS) is presented as developed within the UrbanFlood FP7 project. This system monitors sensors installed in flood defenses, detects sensor signal abnormalities, calculates failure probability of the flood defense, and simulates failure scenarios. All of this information is made available online as part of a DSS to help the relevant figure of authority make an informed decision in case of emergency or routine assessment.

Some requirements that must be taken into account in the design of an EWS include:

- Sensor equipment design, installation and technical maintenance.
- Sensor data transmission, filtering and analysis.
- Computational models and simulation components.
- Onteractive visualization technologies.
- Remote access to the system.

Thus it is clear that the development of an EWS or DSS consists of much more than the development of the software components, but must also take into account the installation of hardware and the transmission of information between components of the system. These many interacting components are shown in Figure 6 along with a description.

A clustering approach for structural health monitoring on bridges

In [15] a clustering based approach is presented to group substructures or joints with similar behaviour and to detect abnormal or damaged ones. The presented approach is based on the simple idea that a sensor located at a damaged substructure or joint will record responses that are significantly different from sensors at undamaged points on the bridge.

The approach was applied to data collected from 2,400 tri-axial accelerometers installed on 800 jack arches on the Sydney Harbour Bridge. An *event* is defined as a time period in which a vehicle is driving across a joint. A pre-set threshold is set to trigger the recording of the responses by each sensor, each event is then represented by a vector of samples X.

Prior to performing any abnormality detection the data is preprocessed. First each event data is transformed into a feature $V_i = |A_i| - |A_r|$ where A_i is the instantaneous acceleration at the ith sample and A_r is the "rest vector" or average of the first 100 samples. The event data is then normalised as $X = \frac{V-\mu(V)}{\sigma(V)}$.

After normalisation of the event data, k-nearest neighbours is applied for outlier removal. One might consider that outliers are useful in the detection of abnormal conditions, since they represent abnormal responses. However if outlying data per joint are removed, then a greater level of confidence can be had when an abnormal condition is detected knowing that the result is not based on any outliers. In this outlier removal step the sum of the energy

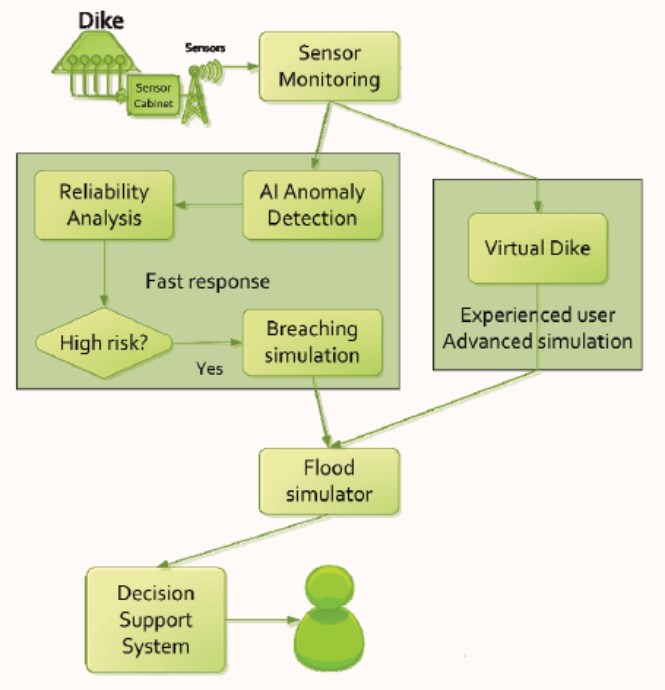


Figure 6: The *Sensor Monitoring* module receives data from the installed sensors which are then filtered by the *AI Anomaly Detector*. In case an abnormality is detected the *Reliability Analysis* calculates the probability of failure. If the failure probability is high then the *Breach Simulator* predicts the dynamics of the dike failure. A fast response is calculated beginning with the *AI Anomaly Detector* and ending with the *Breaching Simulator*. The *Virtual Dike* module is additionaly available for the purpose of simulation by expert users, but takes longer. The fast response and the response from the *Virtual Dike* module are both fed to the *Flood Simulator* which models the flooding dynamics, this information is sent to the decision support system to be made available to the decision maker.

in time domain is calculated for event data as $E(X) = \sum_i |x_i|^2$. Then for every iteration of k-nearest neighbours, the k closest neighbours to the mean of the enery of the joint's signals μ_{joint} is calculated.

The event data is then transformed from the time domain into a series of frequencies using the Fast Fourier Transform (FFT), such that the original vibration data is now represented as a sequence that determines the importance of each frequency component in the signal. After this transformation a distance metric is calculated for each pair of event signals, this metric is used for k-means clustering of the data for anomaly detection. The distance metric used is the Euclidean distance: $dist(X, Y) = ||X - Y|| = \sqrt{\sum(x_i - y_i)^2}$.

Two clustering methods were applied, event-based and joint-based. In the event-based clustering experiment it was known beforehand that joint 4 was damaged. All event data was clustered using k-means clustering with $K = 2$ which resulted in a big cluster containing 23,849 events and a smaller cluster of 4662 events mostly located in joint 4. The percentage of events per joint in the big cluster are shown in Figure 7 where joint 4 is clearly an outlier.

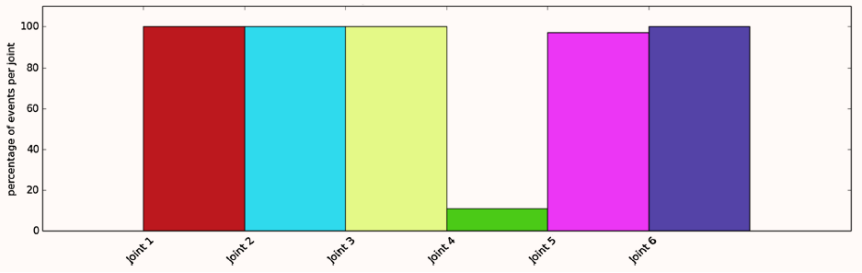


Figure 7: ...

A frequency profile of both the big and small cluster are shown in Figures 8 and 9. In case there is no knowledge of abnormal behaviour then this method can be used to separate outliers and obtain a profile of normal behaviour. In this research on SHB there was prior knowledge of a damaged joint. A frequency profile of an arbitrary joint and the damaged joint before and after repair is shown in Figure 10. The difference of the damaged profile to the other two is clear, which indicates that there is sufficient information in frequency information from accelerometers to detect abnormal joints.

In joint-based clustering a pairwise map of distances is calculated between each pair of joint representatives. A joint representative is calculated as the mean of the values of all event data for one joint, after the outlier removal phase. Two experiments were conducted. One experiment consisted only

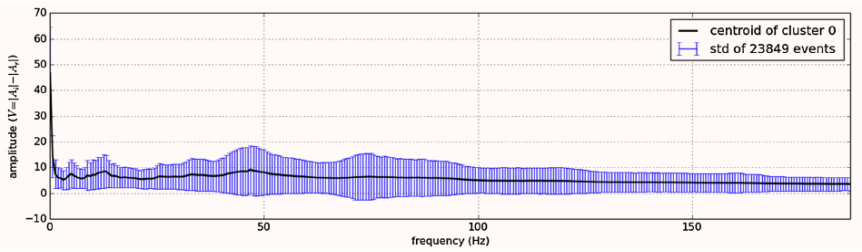


Figure 8: ...

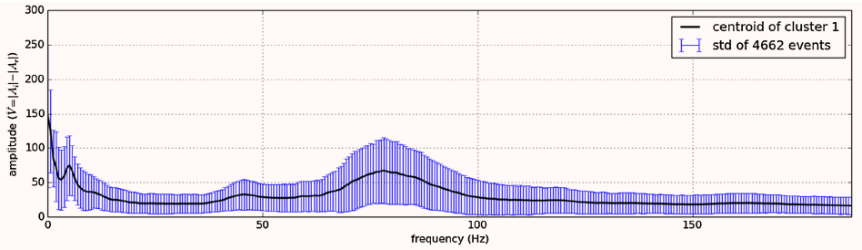


Figure 9: ...

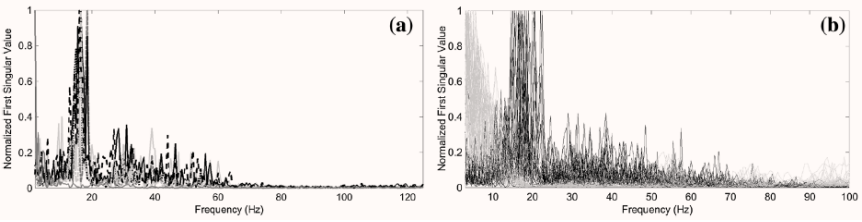


Figure 10: ...

of 6 joints, including the damaged joint 4. The clustering method detected the damaged joint as can be seen in 11. The second experiment was run on data from 71 joints. The resulting map can be seen in 12 which accurately detected the damaged joint 135. Damage was also detected in joint 131 but this result was not verified.

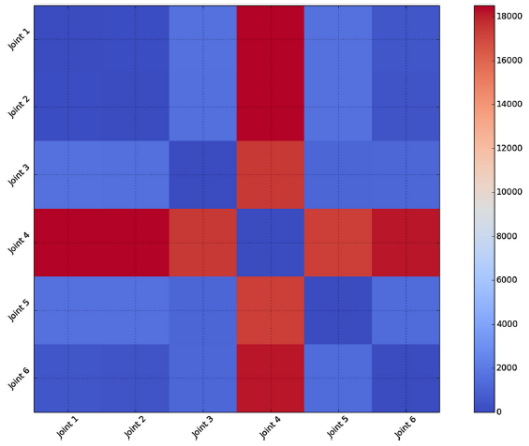


Figure 11: TODO:CAPTION

DSS

TODO: Overview of bridge DSS

Summary

TODO: conclude the literature review

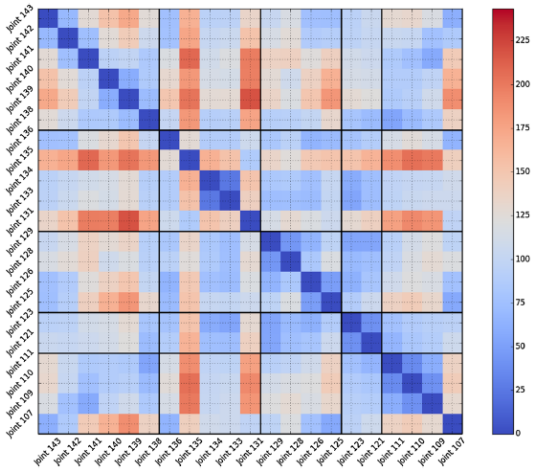


Figure 12: TODO:CAPTION

5 Methods

5.1 Simulation

This section describes the data collection system which was created to model a bridge in software and to collect data from simulating the bridge's response under a damage scenario and traffic scenario. Following a brief overview of how the data collection system operates, this section describes in detail the model of a bridge's geometry (`Bridge`), of a damage scenario and a traffic scenario, the FE software used to simulate a bridge's response, how the data collection system operates from input to output, a description of the collected data, validation of the model, and finally an overview of the assumptions that were made in modeling.

First a quick summary of the data collection system. A simulation scenario is defined as a combination of a damage scenario and traffic scenario. For a given `Bridge`, a number of FEMs are generated of the bridge in undamaged state, and simulations are run. In each simulation a unit load is placed at a different point on the bridge deck. Each point is chosen to be on a "wheel track", which is where a vehicle's wheels will be when the vehicle is later "driven" along the bridge. Vehicles are sampled according to the given traffic scenario and driven along the bridge on a traffic lane in discrete time steps. Using the principle of superposition, responses collected from the previous simulations can be summed together (one for each vehicle's wheel) to calculate a response at a requested point. A number of additional simulations must be run for the bridge in damaged state. This will all be explained more thoroughly in Subsection 5.1.6 but it is useful to present a brief overview in advance.

5.1.1 Bridge Model

A parametric model for describing no-prestress no-posttension concrete slab bridges was created for the programming language Python. The parametric model exists as the type `Bridge`.

A `Bridge` is parameterized by dimensionality, length, width, piers, lanes, material properties and parameters that define the mesh density. A `Bridge` can be declared as 2D or 3D, this defines if the resulting FEM will be 2D or 3D. The length and width define the area of the bridge deck. Piers define the position, size and angle of the piers which support the bridge deck. Lanes define where vehicles are allowed to drive on the bridge and the direction of traffic. Material properties determine the interaction between the bridge

elements and their behaviour under load. Mesh parameters define the density of the base mesh and how the mesh is built.

```
data Bridge {
  length   :: Float,
  width    :: Float,
  lanes    :: [Lane],
  sections :: [Section],
  piers    :: [Pier]
  }

bridge705 = Bridge {
  length   :: 102,
  width    :: 33.2,
  lanes    :: [Lane(4, 12.4), Lane(20.8, 29.2)],
  sections :: [Section],
  piers    :: [12.75, 15.3, 15.3, 15.3, 15.3, 15.3, 12.75]
  }
```

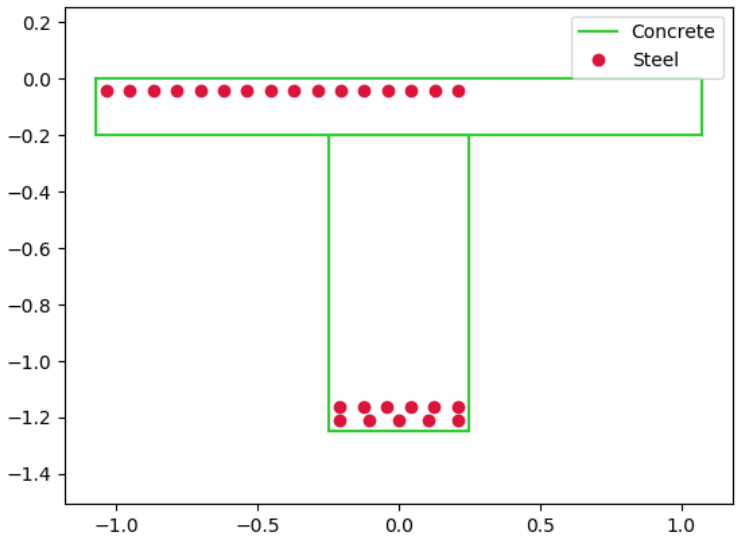


Figure 13: Cross section of bridge 705.

1. Discretization

    - Material properties may vary according to a continuous function on a real bridge while material properties in the FEM change at given discretization points.

2. Bridge 705 The nodes of the generated FEM which have degree's of freedom fixed are: the nodes which make up the abutments at the east and west ends of the bridge, and the nodes along the bottom of each the piers. All of these nodes are fixed in translation along the y (vertical) and the z (transverse) axes. In addition piers 8 - 15 are fixed in translation along the x (longitudinal) axis. Figure

    TODO: Figure of fixed and numbered nodes.

    TODO: Why is rotation fixed.

    TODO: Talk about rollers and different types of fixed pier.

5.1.2 Damage Model

The goal of the damage identification model is to detect or classify different damage scenarios as set up in an experiment. This Subsection describes how different damage types, as outlined in Subsection 3.3, are simulated.

Of the damage scenarios listed in Tables 3 and table:long-term-events]], four scenarios are selected for identification by the DIM in addition to one unlisted damage scenario. These scenarios are chosen due to the practicality of simulating them in a FEM of bridge 705.

Pier settlement can be simulated by displacing a pier by a fixed amount, this is achieved in practice by applying an increasing vertical force known as a *displacement load* to the deck until the desired displacement is achieved.

Abnormal loading conditions can be simulated relatively easily by applying the heavy loads in the FE simulation. Care must be taken regarding the axle configuration because extreme heavy loads typically have a different axle configuration than less heavy vehicles.

Cracked concrete can be simulated by reducing the value of Young's modulus for the cracked concrete section. In practice, Young's modulus is often reduced to $\frac{1}{3}$ of its original value ([34]). Simulating a crack zone in this manner makes sense because damage reduced the stiffness of a structure [11].

Corrosion of the reinforment bars can be simulated by increasing the size of the reinforcement bars TODO:WHY. Finally, a damage scenario is

considered where it is not the bridge that is damaged but rather a sensor is malfunctioning.

A *malfunctioning sensor* can be simulated by adding a significant amount of noise to the simulated sensor responses or adding a constant offset to the responses TODO:LITERATURE. From discussions with Sousa TODO:REF, detecting malfunctioning sensors is useful to accomplish.

5.1.3 Noise Model

Noise in a signal is considered to be the response from unwanted or unknown sources. In the case of bridges the following can be considered as noise: fluctuations in response caused by vehicles on another lane, random fluctuations caused by the sensor instrument itself, the effect of a change in temperature or indeed any other environmental effect. TODO: RED: Noise in sensors

A model is considered robust if its outputs are consistently accurate when input variables, whether foreseen or not, are changed. A damage identification model must be robust to noise in order for it to be considered usable in a real-life setting. In the damage identification experiments, that are laid out later in Subsection 5.2.1, the ability to identify damage in the presence of noise is considered.

Two forms of noise are modeled which we will refer to as background noise and thermal noise. Background noise is a catch-all term for short-term fluctuations and is modeled with a white noise process. For each sensor type a mean value and standard deviation of background noise is set. Thermal noise is modeled as a load that is applied in FE simulation. Thermal noise is considered sufficient to investigate the robustness of the damage identification model to long-term variations in response, whether these have daily or annual periods.

OpenSees does not directly support the application of a thermal load based on a user-input change in temperature (e.g. $+2\,°C$), thus the thermal load is calculated based on the theory of linear thermal expansion. Under the assumption of linear thermal expansion, the strain caused by a change in temperature ($\varepsilon_t hermal$) is linearly proportional to the change in temperature, $\varepsilon_t hermal \propto \delta T$.

The equivalent force applied to a beam can be decomposed into an axial force and a bending moment as is shown in Figure TODO fig:thermal-loading.

The combination of a change in temperature and a known coefficient of thermal expansion (CTE or α) of a material can be used to calculate the thermal load to be applied to a material as shown in Equation 1. First the

strain due to thermal expansion $\varepsilon_t hermal$ is calculated as the product of α and δT, then in combination with Young's modulus E the equivalent stress $\sigma_t hermal$ is obtained. The thermal load $F_t hermal$ to be applied to the cross section of an element in simulation is then determined as the product of the cross sectional area A and the stress. Note that the force applied to each of the shell element's nodes is $F_t hermal/2$ as there are two nodes sharing each of the four cross sections of the element, as shown in Figure. force per thermal load

$$\begin{aligned} \varepsilon_t hermal &= \alpha \cdot \delta T \\ \sigma_t hermal &= E \cdot \varepsilon_t hermal \\ F_t hermal &= \sigma_t hermal \cdot A \end{aligned} \tag{1}$$

5.1.4 Traffic Model

Distribution of passenger vehicles

`https://www.researchgate.net/publication/303809875_Emission_factors_for_alternative_drivelines_and_alternative_fuels`

A model of the normal traffic on bridge 705 is based on two datasets. A dataset was provided by TNO of vehicles recorded using Weight-in-motion (WIM) technology on the A16 highway in The Netherlands. This dataset will be referred to as the A16 dataset. Data was also used from the National Data Warehouse for Traffic Information (NDW). NDW provides a database of real-time and historic traffic data in The Netherlands. The dataset used from the NDW will be referred to as the NDW dataset.

The A16 dataset contains a number of columns, including time and date, lane the vehicle was travelling on, the vehicle type, vehicle speed, distance between axles and load per axle.

In the A16 data all vehicles are above 3500kg in weight and 7m in length. The A16 dataset was filtered so that neither the total weight nor the total length would exceed a z-score of 3 for that column respectively. The filtered data is shown in TODO filtered data.

In the A16 dataset only the distance between axles and the load per axle are considered, all other columns are ignored. All vehicles in the implemented traffic simulation travel at an equal speed of 40kmph thus the speed column is ignored. All vehicles are set to have an axle width of 2.5m, this is the axle width of Truck 1 from the experimental campaign. Setting the same axle width for all vehicles allows for the same set of unit load simulations to

be used to calculate responses for any vehicle travelling across the bridge, because they can then all travel along the same wheel tracks.

The wheel tracks that exist on a bridge are half an axle width (1.25m) either side of the center of each lane. The lanes on bridge 705 are both 8.4m wide and 4.2m from the center of the bridge in the transverse direction. Thus the lanes are separated from each other by 8.4m and the center of the lanes are at `z = +-8.4m`. The wheel tracks are located at `z = += 7.15` and `z = += 9.65`. The lanes and wheel tracks on bridge 705 are depicted in Figure TODO: lanes and wheel tracks figure.

The data collection system is parameterized by the distribution of the vehicles that drive over it. The system has as parameter a filepath `vehicle_data_path`, a column name `vehicle_pdf_col`, and at `vehicle_pdf` a list that describes the probability density function (PDF) of vehicles in terms of the data in that column. The parameter `vehicle_data_path` must point to a `.csv` file which contains descriptions of vehicles. This `.csv` file will be loaded as a Pandas `DataFrame` and should contain data as described in Table 5.

Table 5: Example of Pandas `DataFrame` containing descriptions of vehicles that will be sampled. "axle_load" is the load per axle in kilo Newton, "load" is the sum of these values. "axle_distance" is the distance in meters between each pair of subsequent axles, "distance" is the sum of these values.

load	axle_load	distance	axle_distance
225.55	[79.44, 101, 45.11]	.79	[6.02, 1.32]
...	...	...	...

For example, a Pandas `DataFrame` will be loaded from `vehicle_data_path`, then vehicles will be sampled from this `DataFrame` based on the PDF. A vehicle that is sampled from this `DataFrame` will have a speed of 40kmph, and an axle-width of 2m, the inter-axle distances and the axle weights are taken from the `DataFrame`.

1. Traffic To train a classifier to distinguish between normal and abnormal traffic conditions it is necessary to define normal traffic conditions and additional traffic conditions.

   Traffic is simulated by

5.1.5 FE Program

Two FE programs are used for the collection of sensor responses, OpenSees ([35]) and DIANA ([36]). OpenSees is used because it is open source software,

such that anyone can download and use the software without a licence. On the other hand is proprietary software, if you want to do research with Diana a licence must be purchased. The reason Diana is supported is because a verified 3D FEM of bridge 705 is available for Diana. In this thesis the Diana FEM is used in limited capacity for the verification of results obtained via OpenSees. The focus is instead on OpenSees because it is software that anyone with a laptop can use for free to extend this research. In addition it is useful to have two FE programs available, one (OpenSees) can be used to run less accurate but faster 2D FE simulations, allowing for a more rapid research cycle. The results can then be compared and verified against results from more accurate but also more computationally expensive 3D FE simulations (Diana). It is noted that the 2D model will ignore some aspects in the transverse direction of the bridge deck. For example the 3D model of bridge 705 has two lanes, but the 2D model ignores the concept of lanes entirely.

OpenSees stands for the *Open Sysem for Earthquake Engineering Simulation*, it is "an open source software framework for creating applications for the nonlinear analysis of structural and soil systems using either a standard FEM or an FE reliability analysis. It is object-oriented by design and—in addition to achieving computationally efficiency—it's designed to be flexible, extensible, and portable" [37].

DIANA (**DI**splacement **ANA**lyzer) is developed by DIANA FEA BV which is a spin-off company from the Computational Mechanics department of TNO Building and Conctruction Research Institute in Delft, The Netherlands. DIANA is a FE software package that is dedicated to problems in civil engineering, including structural and geotechnical, and engineering related to tunnelling, earthquake, and oil and gas.

TODO: Image of the 705 Diana model.

5.1.6 System Details

The goal of the data collection system is to translate a `Bridge`, along with a `TrafficScenario` and `BridgeScenario`, into a time series of responses. This subsection details how that translation takes place.

The data collection system transforms a `Bridge` into a FEM for OpenSees. The resulting FEM is a 2D or 3D model depending on the dimensionality of the `Bridge`. In each case the FEM takes the form of a `.tcl` file (written in the TCL language). A `.tcl` file for Opensees consists of a sequence of commands for declaring a structure's geometry, material properties, and other settings of a FE simulation. For example, a `.tcl` file created from a `Bridge` will consist of a number of `node` and `element` commands, where nodes are

points in space with degrees of freedom and elements are a mathematical relation of how degrees of freedom relate between nodes. In the case of the FEMs built from a `Bridge`, four nodes are connected by a *shell* element. Shell elements are used when the thickness is significantly smaller than the other dimensions. In the case of bridge 705's deck the length is 102.75m, width is 33.2m, and thickness is varying from 0.5m to 0.739m.

Under the `HealthyScenario` for a `Bridge`, a number of simulations are run the first time that a response is requested to a point load or vehicle. For each wheel track a number of simulations are run. The number of simulations per wheel track is specified by the system parameter `il_num_loads`. In each of these simulations a load of unit intensity `I` is placed at a point on the wheel track and responses of the bridge are recorded. The responses are translation from each node, and stress and strain from each element. Thus in summary, for each of the `il_num_loads` simulations per wheel track, the responses from the bridge are recorded. Each of these simulations we will call a unit load simulation, and the responses to such a simulation, unit load responses.

Unit load simulations are simulations that must only be run once, and then the principle of superposition can be used to determine the response to a vehicle under the `HealthyScenario`, based on the unit load responses. Furthermore, the response to traffic (multiple vehicles on the bridge) can be calculated simply by summing the response to each vehicle on the bridge. The use of the principle of superposition to calculate the response to a vehicle is introduced in Listing 1. This calculation can however be phrased as a linear algebra problem for which computers are typically optimized. The calculation of the response at many points to many vehicles over a series of time steps using matrix multiplication is shown in Listing 2.

`il_num_loads` number of unit load simulations are run per wheel track. And there are a finite number of responses collected from each unit load simulation, as determined by the mesh density. To explicitly state an important point: the unit load responses, which are used to calculate a response at a point `P` to a vehicle, are the responses at the recorded point closest to `P`, and the unit load simulations from which these responses are taken are those for which the unit load is closest to each of a vehicle's wheels position on the bridge. Thus the parameter `il_num_loads`, and the parameters that define the mesh density, determine the discretization step of the model and thus the accuracy of the responses which are calculated.

`il_num_loads` number of unit load simulations are run per wheel track. Then for any point on the bridge, the response at that point can be calculated to a load on one of that wheel tracks. The function of the response at a point

```
response = 0
p = Point(x=35, y=0, z=25)
for wp, wi in vehicle:
    unit_load_simulation = sim_with_unit_load_closest_to(wp)
    ru = unit_load_simulation.response_at(point)
    response += ru * (wi / ul)
```

Listing 1: Using the principle of superposition to calculate the response to a vehicle from unit load simulations. When requesting the response at a point `p` to a vehicle on a bridge, the vehicle is first decomposed into loading positions `wp` and intensities `wi`, one position and one load intensity for each of the vehicle's wheels. Then for each wheel position `wp`, the unit load simulation is selected where `wp` is closest to the unit load applied in that simulation. From this unit load simulation, the response `ru` at the recorded point closest to point `p` is considered. Thus the response `ru` is the response to a load at one of the vehicle wheel's positions, except not to the wheel's load but instead to a load of unit intensity, thus `ru` must be multiplied by `wi / ul` where `ul` is the unit load intensity.

```
$ traffic_t0 = [
    1, 2, 0, 0,
    1, 2, 0, 0,
    3, 3, 0, 0,
    3, 3, 0, 0]
$ traffic = [
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 3, 3]
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 3, 3, 0]
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 3, 3, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
$ points = [
    [0, 0, 0, 0]
    [0, 0, 0, 0]
    [0, 0, 0, 0]
    [0, 0, 0, 0]
]
$ print(np.matmul(traffic, points))
```

Listing 2: Response to traffic using matrix multiplication

due to a changing load is called an influence line, which is commonly used in structural engineering to describe a response function. Figure 14 contains a number of influence lines. Each influence line shows the displacement of the bridge deck at a different point on the wheel track at `z = -9.4m`, as a unit load is moved along the same wheel track.

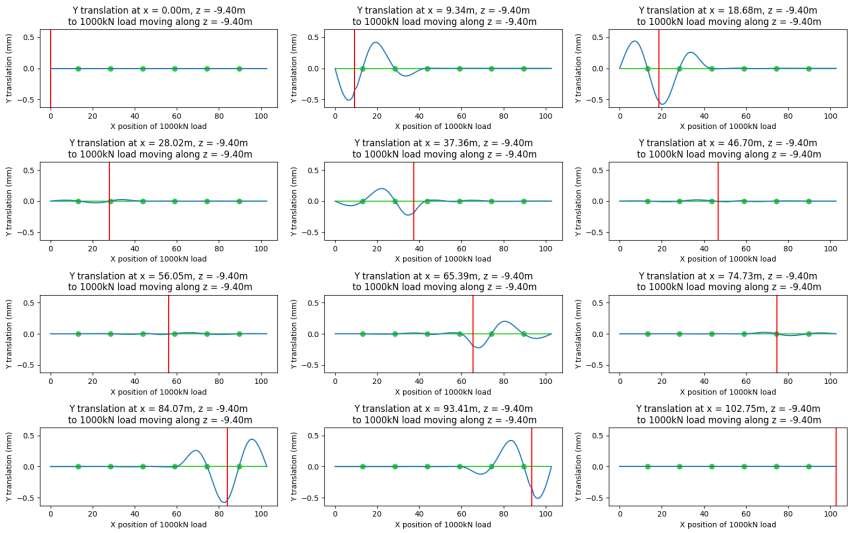


Figure 14: Displacement of the bridge deck at different points on the wheel track at `z = -9.4m`, in each influence line plot a unit load is moved along the same wheel track. The red vertical line depicts the position of the load.

Furthermore we can stack the influence lines for a number of points against each other, flipping each influence line by 90 so it is vertical. For example, we can consider a number of equidistant points along a slice in the longitudinal direction of a bridge, and for each of these points consider the response to a load moving along the same slice. Figure 15 shows such a matrix for `z = -9.4m`. Each column of the matrix is an influence line, each row shows the response along the bridge deck for `z = -9.4m` for a different loading position.

Another of the damage scenarios is pier displacement. To calculate responses to a load under this damage scenario, all of the unit load simulations need to be run again for this damage scenario. The name of the pier displacement damage scenario in the data collection sytem is `PierDisplacement`. `PierDisplacement` specifies a displacement in meters of one of a bridge's piers.

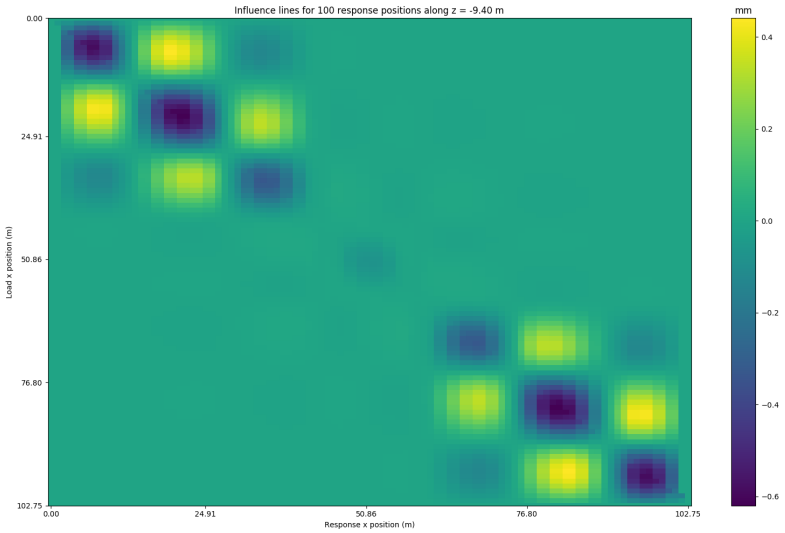


Figure 15: A number of vertical influence lines stacked together. Each influence line (column) shows displacement at a different point on the wheel track at `z = -9.4m`. Each column of the matrix is an influence line. Each row shows the response along the bridge deck for `z = -9.4m` for a different loading position. This image shows how, closer to the center of the bridge, the bridge does not suffer as much displacement.

When creating a FEM of a `Bridge` under pier displacement for OpenSees, each of the bottom nodes of the piers under displacement are not fixed for y translation (to allow for the displacement of the piers to occur). An important step when creating a FEM under this damage scenario for OpenSees is to set the method of integration with the `integrator` command. Under the undamaged scenario the integrator used is `LoadControl`, which specifies that, among other things, the predictive time step of the simulation is driven by the loads applied. In the case of pier displacement the `DisplacementControl` integrator is used instead, this is used to specify that in an analysis step, the displacement control algorithm will seek the time step that will result in a specified increment for a particular degree of freedom of a specified node. For example the command `integrator DisplacementControl 1 2 0.1` specifies that the displacement control algorithm will seek an increment of 0.1 at node 1 in the second degree of freedom.

When running a pier displacement simulation the `DisplacementControl` command is used to specify that the central bottom node of the pier should

be displaced by 1m. A load is placed on this node, though the load intensity is ignored by the `DisplacementControl` algorithm, the load intensity is instead increased until a displacement of 1m is reached. Figure 16 shows a contour plot of the displacement of the deck of bridge 705 due to a single pier being displaced by -1m.

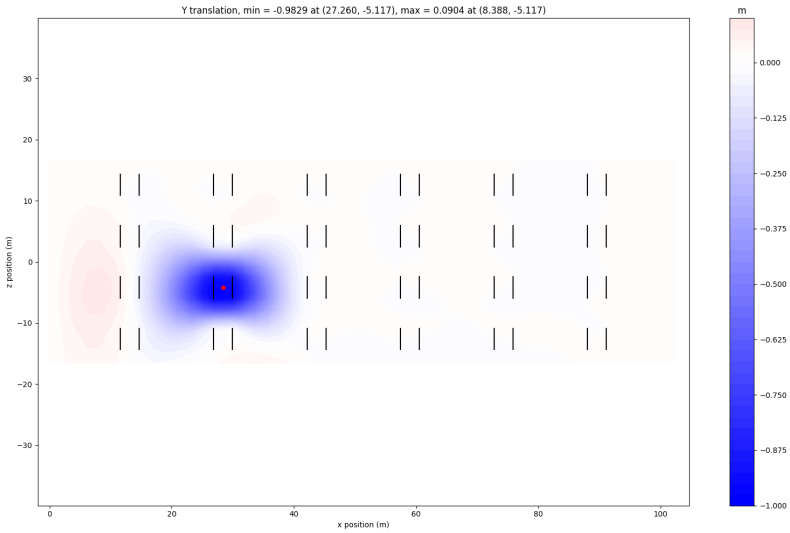


Figure 16: A contour plot of the displacement of the deck of bridge 705 due a pier being displaced by 1m. The node onto which a load is applied, and the same node that is watched by the `DisplacementControl` algorithm until the specified displacement of 1m is reached, is indicated by a red circle. This node is the central bottom node of the pier indicated by vertical black bars on either side of the red circle. The maximum displacement on the bridge deck is slightly less than 1m, this is because the piers are not infinitely stiff but have some elasticity.

Due to the linear elastic accumption made when modeling, only one pier displacement simulation needs to be run per pier. One simulation is run for each pier, until that pier has been displaced by unit amount, one meter in the case of this data collection system. After these simulations have run, the response at any point on the bridge can be calculated due to any combination of piers being displaced by different amounts, as outlined in Listing 3.

1. Meshing

```
response = 0
p = Point(x=35, y=0, z=25)
for vehicle in traffic:
    for wp, wi in vehicle:
        unit_load_simulation = load_sim_closest_to(wp)
        ru = unit_load_simulation.response_at(point)
        response += ru * (wi / ul)
```

Listing 3: Calculation of the response

5.1.7 System Interface

In recent years there is an increased emphasis on reproducable research within the scientific community. Reproducable research can be more easily verified by peers than research which must be reimplemented. If research is accomplished through code, for example as simulations are, then the research can be verified be downloading the software and running it.

If the parameters of the research are not buried deep in the code but instead "lifted" to the boundaries of the system, then that research can be considered not just reproduceable but extensible.

Furthermore if the software is presented as composable functions instead of scripts, then this allows for the reuse of the research, whereby a researcher can compose some of the functions in a manner which was not done in the original research.

Reuse of software aligns with the *don't repeat yourself* (DRY) principle of software engineering. Violations of DRY are creatively referred to as WET, or *write every time*. The downside of WET solutions are that each implementation has its own bugs, whereas in DRY solutions the bug fixes and optimizations are shared by all contributors and users. To aid reuse and avoid repitition, software must be made easy to use, which is the very next tip after the DRY principle in the book The Pragmatic Programmer [38].

> Make It Easy To Reuse
>
> If it's easy to reuse, people will. Create an environment that supports reuse.
>
> – The Pragmatic Programmer page number

Keeping with these principles, an effort has been made for the research in this thesis to me not just reproduceable but extensible. A system for running traffic simulations on concrete slab bridges and analyzing results is published

on the Python Package Index (PyPI) [39] under the name `traffic-sim` where you will also find the documentation. Two examples showing the use of the system are presented in Listing 4 and Listing 5, for `traffic-sim` installation instructions see the documentation on PyPI.

```
# example.py
from traffic_sim import PointLoad
```

Listing 4: Contour load of a point load in `traffic-sim`.

```
# example.py
from traffic_sim import PointLoad
```

Listing 5: Animation of a truck in `traffic-sim`.

5.1.8 Collected Data

The outputs of the system are time series of responses from sensors distributed across the bridge model, these time series of responses we call *events*. Events are labelled by simulation scenario and simulation time.

5.1.9 Validation

The FEM generated for the purpose of data collection needs to be validated, otherwise an analysis of the data would offer little value. In this Subsection the data collection system is set to bridge 705 and the generated FEM for OpenSees validated against the previously validated FEM of bridge 705 for Diana. In addition the generated FEM is validated directly against the measurements from the experimental campaign.

The density of the generated FEM's mesh is controlled by a number of parameters, as outlined in Subsection 5.1.1. As the number of nodes in the FEM increases, the expectation is that the error ϵ of responses from the OpenSees simulation will decrease to zero. ϵ is calculated for the FEM of bridge 705 as the mean difference of the maximum response recorded with OpenSees to the maximum recorded with Diana, for a number of loading positions. $\epsilon = \sum_{p=1}^{n} \left| r_{op} - r_{dp} \right| \frac{1}{n}$, where r_{op} is the maximum response recorded with OpenSees under a 100kN concentrated load at position p and r_{dp} is the maximum response recorded using the verified Diana model with the same load applied.

Figure 17 shows ϵ for $n = 4$ as a function of model size. The Figure shows that for the $n = 4$ chosen loading positions that ϵ decreases as the model size increases. Four loading positions were chosen to cover various points of interest on the bridge deck. Each of the $n = 4$ loading positions is defined in Table **??** with the reason for including that position.

Get accurate values from Diana for ϵ calculation

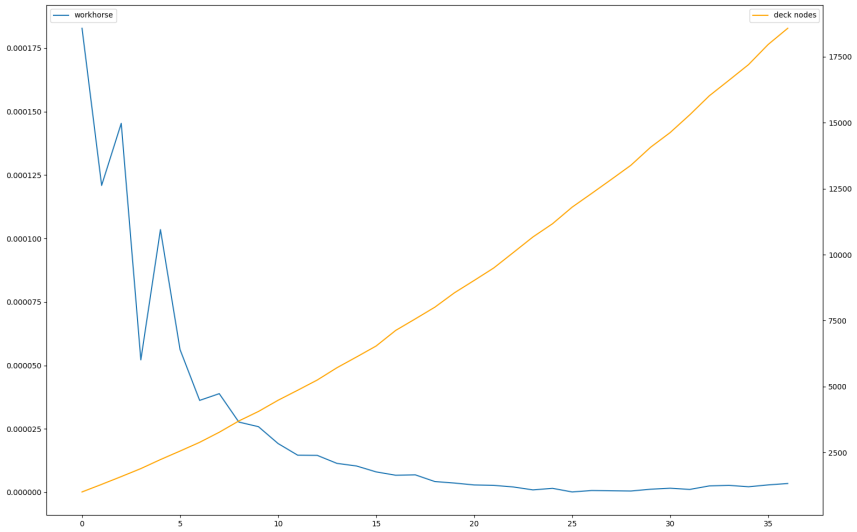


Figure 17: TODO:CAPTION

the comparison of the FEM of bridge 705 generated for OpenSees and the previously validated Diana model. Each point is given in meters along the longitudinal (x) and transverse (z) directions of the bridge. Each loading position is assigned a unique label so it can be referred to, and a reason why that position was chosen. The exact position chosen is the position of a node in the Diana model which fulfils the reason in the "Reason" column.

X (m)	Z (m)	Label	Reason
34.955	29.226	A	On a wheel track, between piers
51.251	16.6	B	At center of bridge, symmetry expected in response
92.406	12.405	C	On a wheel track, close to pier
101.765	3.974	D	On a wheel track, close to abutment

Verify the model is deterministic ϵ

An experimental campaign was carried out by TNO where two trucks were driven on bridge 705 in Amsterdam and sensor responses measured.

Table 6: Comparison of maximum displacement between simulations of bridge 705 with the validated Diana model and a high-density OpenSees model. The first column determined the position of a 100kN point load applied in simulation, the position is defined in Table **??**. The second and third columns show the maximum displacement in millimeters for the Diana and OpenSees models respectively, due to the 100kN load placed at the labeled point.

Point	Diana (mm)	OpenSees (mm)	Diff. (mm)	Diff. / Diana
A	0.1015	0.1082	0.0067	0.0660
B	0.0030	0.0032	0.0002	0.0667
C	0.0116	0.0121	0.0005	0.0431
D	0.0058	0.0066	0.0008	0.1379
		Mean	0.0021	0.0784

Table 7: Comparison of minimum displacement between simulations of bridge 705 with the validated Diana model and a high-density OpenSees model. The first column determines the position of a 100kN point load applied in simulation, the position is defined in Table **??**. The second and third columns show the minimum displacement in millimeters for the Diana and OpenSees models respectively, due to the 100kN load placed at the labeled point.

Point	Diana (mm)	OpenSees (mm)	Diff. (mm)	Diff. / Diana
A	-0.2810	-0.3221	0.0411	0.1463
B	-0.1375	-0.1431	0.0056	0.0407
C	-0.0464	-0.0673	0.0209	0.4504
D	-0.0210	-0.0258	0.0048	0.2286
		Mean	0.0181	0.2165

Sensors were installed by TNO aswell as by other companies. The sensors measured strain, displacement and acceleration. The measured responses allowed the FEM of bridge 705 for Diana to be verified by comparing the responses in simulation with Diana against the measured responses.

For the verification of the generated FEM against both the FEM of bridge 705 for Diana, and against the measured responses, only one of the trucks (truck 1) is considered. This is because the measured responses and the responses from Diana for truck 1 were available from TNO in a simple format.

The specification of truck 1 is shown in Figure 18 on the left. The plot on the left shows the size of the wheels, however in simulation the force from each wheel is represented by a point load as shown on the right in the same Figure.

In the experimental campaign, truck 1 was parked at 13 positions on the bridge deck and the responses from a number of sensors measured. The positions of the truck are shown in Figure 19. Each of the sensors are listed in Table 8, furthermore the positions of the sensors across the bridge deck is shown in Figure TODO REF

Use all strain sensors for verification

Figure of sensor positions

When bridge 705 was built

Amn important point is that the FEM of bridge 705 for Diana and the generated FEM are based on a blueprint of the bridge and make the assumption that bridge 705 is in perfect healthy condition. However since bridge 705 was built N years ago, it is more likely that some imperfections now exist in the structure. Both the generated FEM and the FEM for Diana are based on the assumption of perfect health and thus the measured responses are not expected to exactly match.

5.1.10 Parameter Selection

Increasing mesh density increases the accuracy of responses but also increases run time of the FE simulation. Figure TODO shows the run time of a FE simulation of bridge 705 as a function of the FEM size. Based on these values a base mesh density of X * Z was chosen. time reduction and ϵ. Recall from Subsection 5.1.1 that the generated mesh of the bridge deck will have greater density in most areas but may also have reduced density in areas away from the applied load.

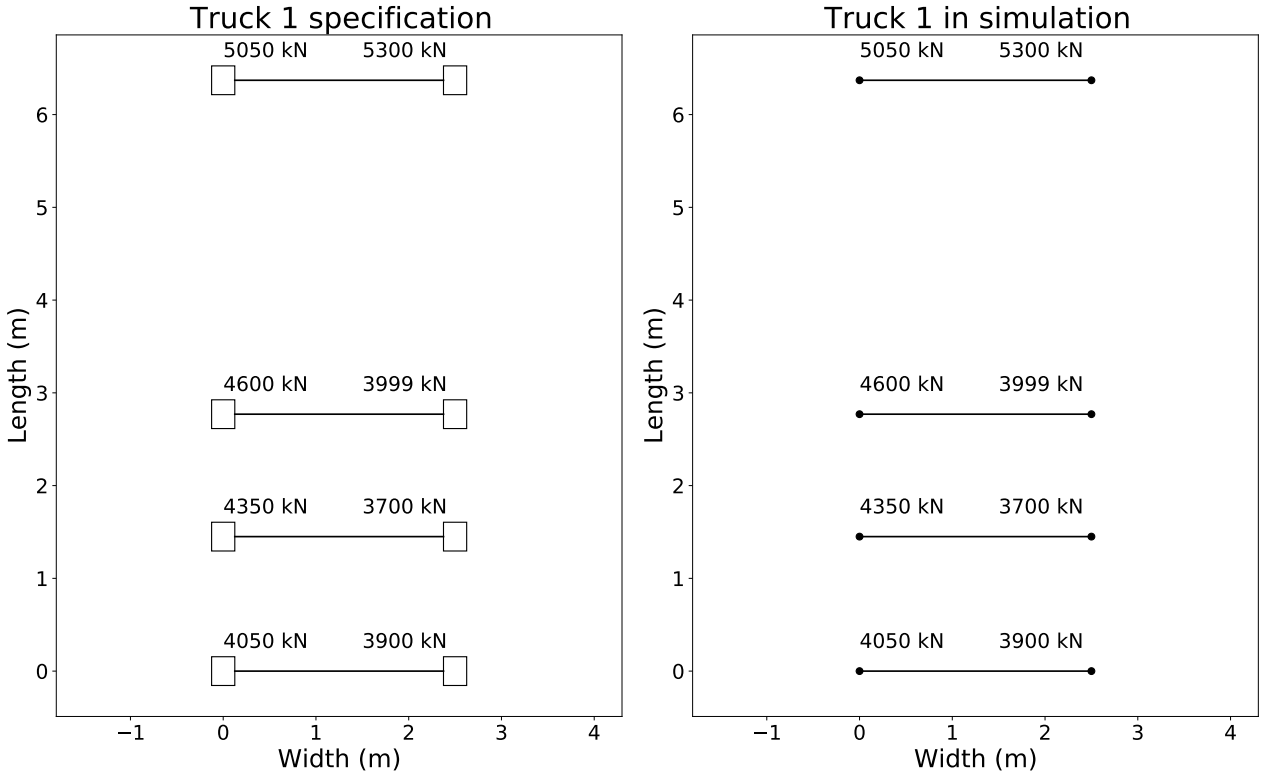


Figure 18: Specification of truck 1 from the experimental campaign. The plot on the left shows the distance between each axle in meters, the axle width in meters, the weight per wheel in kilogram, the total weight per axle in kilogram and the total weight of the truck in kilogram. The plot on the right is generated by the model of truck used in simulation, here the force from each wheel is represented by a point load.

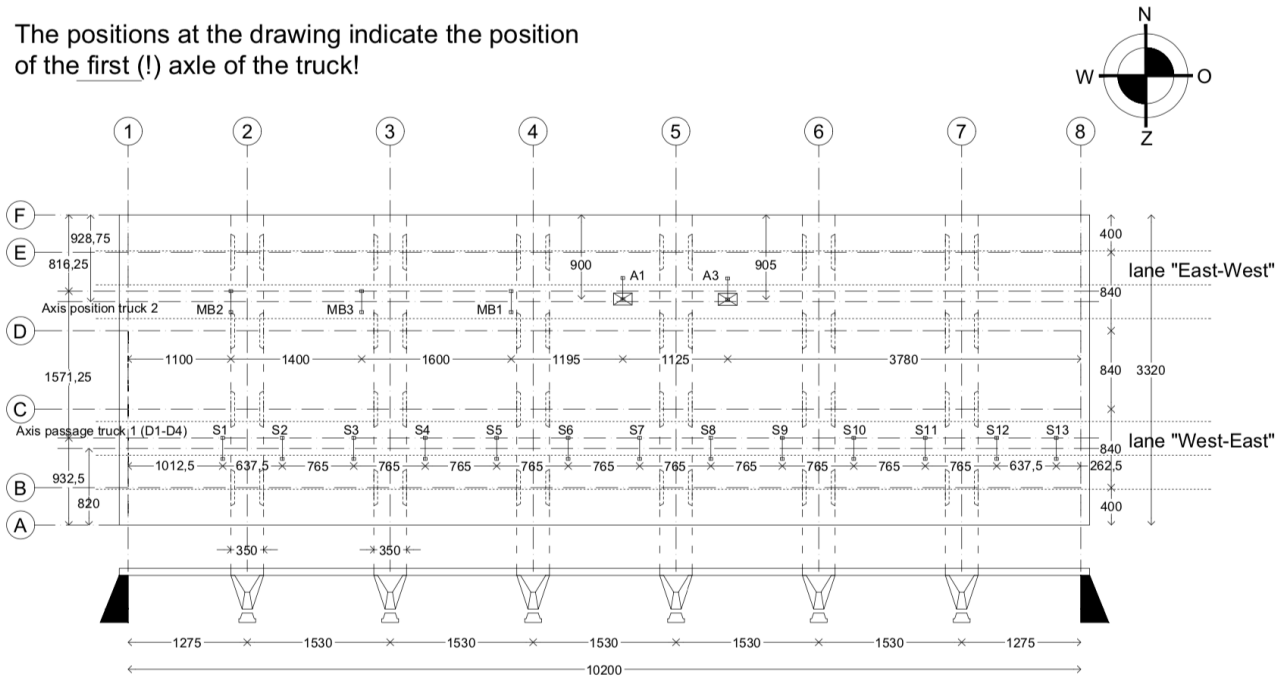


Figure 19: Positions of the front axle of truck 1 during the experimental campaign. The positions are labeled as `S1` to `S13`. In this image the distance 1012.5mm is shown from the west abutment of the bridge to position `S1`. It should be noted that there is an additional 375mm to the west-most end of the bridge, this is the overlap of the abument and the bridge. This image is provided by TNO.

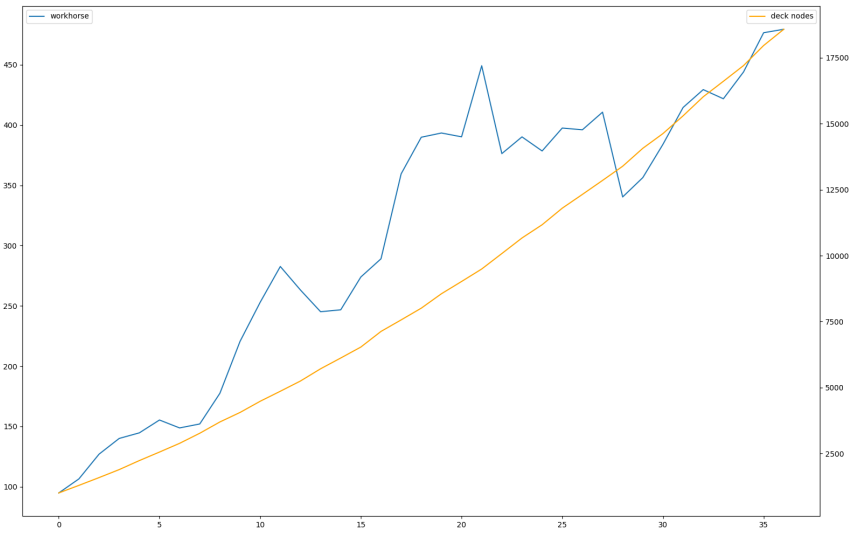


Figure 20: TODO:CAPTION

5.1.11 Model Assumptions

- All vehicles drive at the same speed.
- All vehicles drive along the center of a lane.
- All vehicles have the same axle-width.
- Vehicles arrive at a bridge according to a poisson process.
- measurements from the verified campaign only for a small subset (positions and existing healthy/damage state/)
- the model is linear elastic
- bridge scenarios
- The behaviour of a bridge captured in FE simulation is sufficiently close to the real behaviour of a real bridge that the analysis techniques explored on the simulated data can also work on real data.

  This assumption is verified by (A) applying the analysis techniques explored on real data in addition to the simulated data and (B) verifying the collected responses against sensor measurements collected in real life.

Note that the accuracy of the responses depends on the discretization density of the FEM. This is a trade-off of time versus accuracy which can be chosen by the user. Discretization of the FEM is covered in Section 1. The accuracy of the FEM is shown to converge for bridge 705 in TODO: Convergence plot.

- The simulated noise that is applied to responses from FE simulation is sufficiently close to noise from sensors in real life that the analysis techniques explored on the simulated data can also work on real data.

  This assumption is verified by (A) applying the analysis techniques explored on real data in addition to the simulated data with varying levels of noise and (B) verifying the simulated noise is comparable to the noise from measurements collected in real life as shown in noise.

5.2 Damage Identification

In this section the process of building the damage identification model is described. First there is an introduction to the damage scenarios that it is desirable for the model to identify, followed by a description of the setup for testing iterations of the model. After this an analysis is presented of the sensor responses with respect to the useful information in different sensor types for each damage scenario. Finally the damage identification model that is built is discussed.

5.2.1 Experiments

5.2.2 Feature extraction

5.2.3 Damage Identification

5.2.4 Sensor Placement

5.2.5 Other Bridges

6 Results

6.1 Simulation

6.2 Anomally Detection

7 Appendix

bridge 705 model. Each node corresponds to a loading position in Table **??**. Providing this information allows for the contour plots from the Diana model to be reproduced, by anyone with a licence for the Diana software. The Diana model can either be requested from the thesis author, or found in the `traffic-sim` repository.

Label	Node identifier	X	Z
A	126'679	34.95459	26.24579
B	87'125	51.25051	16.6
C	53'166	89.98269	9.445789
D	54'603	102.5037	6.954211

Table 8: Sensors in the experimental campaign. Each sensor is given with an identifying label, position on the bridge deck, type of response the sensor is measuring, and company that installed the sensor.

Label	X	Z	Type	Company
0.49	0.465	A		
0.14	0.130	B		
0.162	0.180	C		
0.13	0.128	D		

References

[1] E. Aktan, S. Chase, D. Inman, and D. Pines, "Monitoring and managing the health of infrastructure systems," in *Proceedings of the 2001 SPIE conference on health monitoring of highway transportation infrastructure*, pp. 6–8, 2001.

[2] C. R. Farrar, W. Baker, T. Bell, K. Cone, T. Darling, T. Duffey, A. Eklund, and A. Migliori, "Dynamic characterization and damage detection in the i-40 bridge over the rio grande," tech. rep., Los Alamos National Lab., NM (United States), 1994.

[3] S. W. Doebling and C. R. Farrar, "Statistical damage identification techniques applied to the i-40 bridge over the rio grande river," tech. rep., Los Alamos National Lab., NM (United States), 1998.

[4] N. Stubbs, S. Park, C. Sikorsky, and S. Choi, "A global non-destructive damage assessment methodology for civil engineering structures," *International Journal of Systems Science*, vol. 31, no. 11, pp. 1361–1373, 2000.

[5] A. Pandey, M. Biswas, and M. Samman, "Damage detection from changes in curvature mode shapes," *Journal of sound and vibration*, vol. 145, no. 2, pp. 321–332, 1991.

[6] V. Dawari and G. Vesmawala, "Structural damage identification using modal curvature differences," *IOSR Journal of Mechanical and Civil Engineering*, vol. 4, pp. 33–38, 2013.

[7] M. Dilena, A. Morassi, and M. Perin, "Dynamic identification of a reinforced concrete damaged bridge," *Mechanical Systems and Signal Processing*, vol. 25, no. 8, pp. 2990–3009, 2011.

[8] S. Doebling, "Damage detection and model refinement using elemental stiffness perturbations with constrained connectivity," in *Adaptive Structures Forum*, p. 1307, 1996.

[9] A. Teughels and G. De Roeck, "Structural damage identification of the highway bridge z24 by fe model updating," *Journal of Sound and Vibration*, vol. 278, no. 3, pp. 589–610, 2004.

[10] H. Sohn and K. H. Law, "Bayesian probabilistic damage detection of a reinforced-concrete bridge column," *Earthquake engineering & structural dynamics*, vol. 29, no. 8, pp. 1131–1152, 2000.

[11] W. Yeung and J. Smith, "Damage detection in bridges using neural networks for pattern recognition of vibration signatures," *Engineering Structures*, vol. 27, no. 5, pp. 685–698, 2005.

[12] C. Peter, F. Alison, and S. Liu, "Review paper: health monitoring of civil infrastructure," *Structural health monitoring*, vol. 2, no. 3, pp. 0257–267, 2003.

[13] J. M. Brownjohn, "Structural health monitoring of civil infrastructure," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 589–622, 2006.

[14] K. Worden and G. Manson, "The application of machine learning to structural health monitoring," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1851, pp. 515–537, 2006.

[15] A. Diez, N. L. D. Khoa, M. M. Alamdari, Y. Wang, F. Chen, and P. Runcie, "A clustering approach for structural health monitoring on bridges," *Journal of Civil Structural Health Monitoring*, vol. 6, no. 3, pp. 429–445, 2016.

[16] G. Kerschen, P. De Boe, J.-C. Golinval, and K. Worden, "Sensor validation using principal component analysis," *Smart Materials and Structures*, vol. 14, no. 1, p. 36, 2004.

[17] J. Kullaa, "Sensor validation using minimum mean square error estimation," *Mechanical Systems and Signal Processing*, vol. 24, no. 5, pp. 1444–1457, 2010.

[18] K. H. Law, S. Jeong, and M. Ferguson, "A data-driven approach for sensor data reconstruction for bridge monitoring," in *2017 World Congress on Advances in Structural Engineering and Mechanics*, 2017.

[19] D. Mattern, L. Jaw, T.-H. Guo, R. Graham, and W. McCoy, "Using neural networks for sensor validation," in *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, p. 3547, 1998.

[20] X. Xu, J. W. Hines, and R. E. Uhrig, "Sensor validation and fault detection using neural networks," in *Proc. Maintenance and Reliability Conference (MARCON 99)*, pp. 10–12, 1999.

[21] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[22] S. Jeong, M. Ferguson, and K. H. Law, "Sensor data reconstruction and anomaly detection using bidirectional recurrent neural network," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*, vol. 10970, p. 109700N, International Society for Optics and Photonics, 2019.

[23] S. Doebling and C. Farrar, "Using statistical analysis to enhance modal-based damage identification," in *Proc. DAMAS*, vol. 97, pp. 199–210, 1997.

[24] X. Ye, X. Chen, Y. Lei, J. Fan, and L. Mei, "An integrated machine learning algorithm for separating the long-term deflection data of prestressed concrete bridges," *Sensors*, vol. 18, no. 11, p. 4070, 2018.

[25] H. Sohn, K. Worden, and C. R. Farrar, "Statistical damage classification under changing environmental and operational conditions," *Journal of intelligent material systems and structures*, vol. 13, no. 9, pp. 561–574, 2002.

[26] H. Klatter and J. Van Noortwijk, "Life-cycle cost approach to bridge management in the netherlands," in *Proceedings of the 9th International Bridge Management Conference, April 28-30, 2003, Orlando, Florida, USA*, pp. 179–188, 2003.

[27] H. Sousa, A. Rozsas, A. Slobbe, W. Courage, and A. Bigaj van Vliet, "Development of a novel pro-active shm tool devoted to bridge maintenance based on damage identification by fe analysis and probabilistic methods," 2019.

[28] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 3, p. 25, 2009.

[29] A. Rytter, *Vibrational based inspection of civil engineering structures*. PhD thesis, Dept. of Building Technology and Structural Engineering, Aalborg University, 1993.

[30] M. Bedworth and J. O'Brien, "The omnibus model: a new model of data fusion?," *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 4, pp. 30–36, 2000.

[31] B. Lang, T. Poppe, A. Minin, I. Mokhov, Y. Kuperin, A. Mekler, and I. Liapakina, "Neural clouds for monitoring of complex systems," *Optical Memory and Neural Networks*, vol. 17, no. 3, pp. 183–192, 2008.

[32] A. L. Pyayt, D. Shevchenko, A. P. Kozionov, I. I. Mokhov, B. Lang, V. V. Krzhizhanovskaya, and P. M. Sloot, "Combining data-driven methods with finite element analysis for flood early warning systems," *Procedia Computer Science*, vol. 51, pp. 2347–2356, 2015.

[33] V. V. Krzhizhanovskaya, G. Shirshov, N. Melnikova, R. G. Belleman, F. Rusadi, B. Broekhuijsen, B. Gouldby, J. Lhomme, B. Balis, M. Bubak, *et al.*, "Flood early warning system: design, implementation and computational modules," *Procedia Computer Science*, vol. 4, pp. 106–115, 2011.

[34] Y. Li, J. Walraven, N. Han, C. Braam, P. Hoogenboom, and I. L. Houben, "Predicting of the stiffness of cracked reinforced concrete structures," 2010.

[35] S. Mazzoni, F. McKenna, M. H. Scott, G. L. Fenves, *et al.*, "Opensees command language manual," *Pacific Earthquake Engineering Research (PEER) Center*, vol. 264, 2006.

[36] F. DIANA, "Diana 10.3 user's manual," 2019.

[37] F. McKenna, "Opensees: a framework for earthquake engineering simulation," *Computing in Science & Engineering*, vol. 13, no. 4, pp. 58–66, 2011.

[38] A. Hunt, *The pragmatic programmer*. Pearson Education India, 1900.

[39] P. Ranking, "Pypi python modules ranking."