Utilizing sensors for the purpose of building a decision support system for bridge maintenance

Jeremy Barisch-Rooney

July 10, 2019

Contents

1 Introduction **2**

2 Literature review **3**

2.1 The application of machine learning to structural health monitoring . . . 4
2.2 Neural Clouds for monitoring of complex systems . . . 6
2.3 Combining data-driven methods with finite element analysis for flood early warning systems . . . 8
2.4 Flood early warning system: design, implementation and computational modules. . . . 8
2.5 A clustering approach for structural health monitoring on bridges . . . 9
2.6 Conclusion . . . 14

3 Methods **14**

3.1 Simulated responses . . . 14
3.1.1 Neccessary data . . . 14
3.1.2 Finite element model . . . 14
3.1.3 Collected data . . . 14
3.1.4 Data analysis . . . 14
3.2 Damage identification . . . 14
3.2.1 Damage scenarios . . . 15
3.2.2 Test setup . . . 17
3.2.3 Data analysis . . . 17
3.2.4 Damage identification model . . . 17
3.3 Decision support system . . . 17

3.3.1 Sensor placement 17
3.3.2 Cost-benefit analysis 17
3.3.3 Uncertainty 17

4 Results **17**
4.1 Simulated responses 17
4.2 Damage identification 17
4.3 Decision support system 17

5 Conclusion **17**

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. Sensors can provide useful real-time information without the delay or cost of a manual maintenance check. How sensors can be utilized to build a decision support system (DSS) for bridge maintenance is the topic of this thesis.

Sensors on bridges can provide real-time measurements of the responses of the part of the bridge on which they are installed. Depending on the sensor-type this measured response may be translation, rotation, vibration or one of many other types of response. In this thesis the focus is on a single bridge, bridge 705 in Amsterdam. The reason bridge 705 was chosen is because a 3D finite element model (FEM) is available for the bridge, and a field test was conducted where known loads were applied to the bridge and the corresponding sensor measurements recorded. The FEM is useful so that sensor measurements for a known load can be simulated without having to conduct a field test, the measurements from the field test allow us to verify the accuracy of the data generated by simulation.

A DSS for bridge maintenance must provide information on the damage status of the bridge to the user of the system or policy maker. Thus it is necessary to transform the responses measured by the sensors into a report of the damage condition of the bridge. To accomplish this a condition classification model (CCM) is built which transforms sensor measurements into a condition report. The CCM is based on two methods referred to from now on as abnormal condition classification (ACC) and similar structure similar behaviour (SSSB).

The goal of ACC is to determine if the condition of the bridge has deviated from the normal range of conditions. To build an ACC system it is necessary to first find out what the range of sensor measurements are during normal operation of the bridge. This is achieved by applying a normal range of loading conditions to the FEM and recording the simulated sensor measurements. Then a one-class classifier can be applied to the simulated responses and be used to decide if any subsequent sensor measurements fall within the expected normal range of responses or not.

The SSSB method is based on the assumption that similar structures should behave in a similar manner when subjected to the same load. Bridge 705 in Amsterdam has seven spans each with the same dimensions, ignoring the small differences due to construction and time in operation. To develop an SSSB system loads must be ”driven” across the bridge in the FEM, then an analysis must be performed on the difference between sensor measurements from sensors at equivalent positions on each substructure.

The research question that this thesis answers is: how can sensors be utilized to build a DSS for bridge maintenance. The structure of this thesis and how the research question is answered is as follows. First a review of relevant literature and background material is presented. The DSS is then introduced at a high-level, showing how the separate components interact. The components of the DSS are examined in detail, with a large focus on the condition classification model that determines if sensor measurements represent an abnormal condition of the bridge. An analysis is presented of which sensor types and what sensor placement is optimal for detecting such an abnormal condition. A finite element model is used to simulate sensor measurements in order to address the lack of available data. Due to the safety requirements of any bridge, uncertainty measures for the damage estimates are calculated. Once the capabilities and limitations of the model are understood, an outline of a DSS is presented for policy makers which includes the model and a cost-benefit analysis is presented of the system. Finally (stretch-goal) an investigation is conducted into how such a system can be generalized to bridges other than bridge 705.

2 Literature review

This section contains a review of the most relevant material studied during this thesis work. The aim of presenting a review of this material is to place this thesis in context and to provide background information to the reader on techniques that are employed in this thesis. After reviewing the most relevant material this section closes with a more general overview of related material.

2.1 The application of machine learning to structural health monitoring

cite:worden2006application illustrates the utility of a data-driven approach to structural health monitoring (SHM) by a number of case studies. 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 simply to identify if a system has departed from normal (i.e. undamaged) condition. The more sophisticated problem of *damage identification* seeks to determine a greater level of information on the damage status, even to provide a forecast of the likely outcome of a situation. The problem of detection and identification can be considered as a hierarchy of levels as described in cite:rytter1993vibrational.

- 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.

This paper argues that ML provides solutions to these problems at upto level 3, and 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 1 shows the waterfall model (cite:bedworth2000omnibus) 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.

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

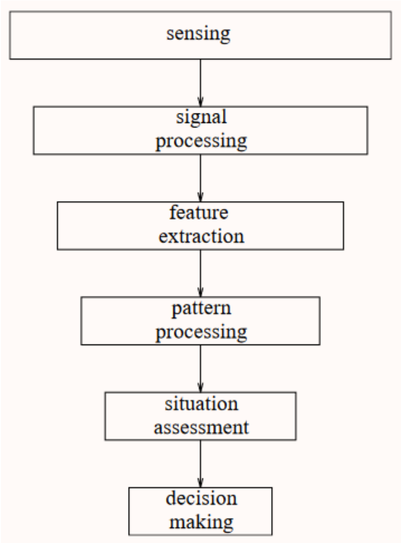

Figure 1: The *waterfall* model.

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.

2.2 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 cite:lang2008neural 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 2 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

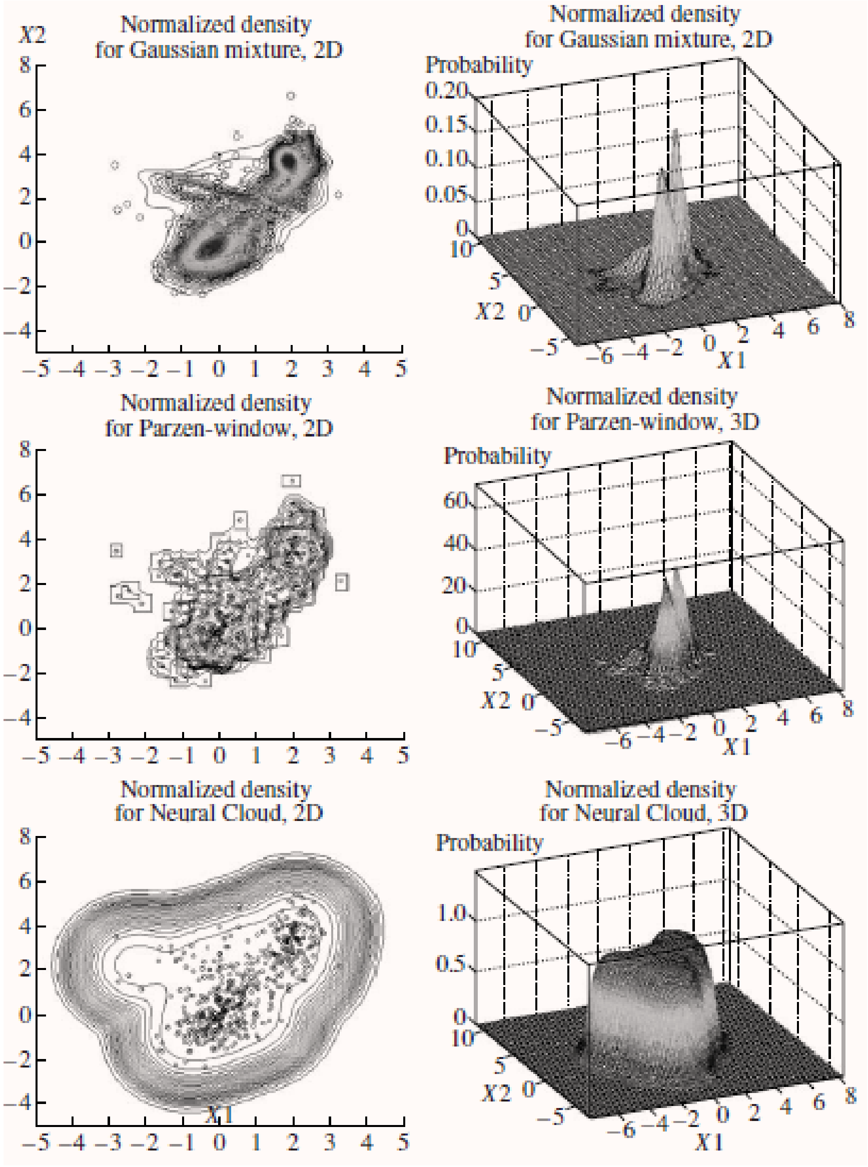

Figure 2: Comparison of Neural Clouds with other approaches, namely Gaussian mixture and Parzen-window. At the left side 2D contour line plots are pictures and at the right normalized density 3D plots.

NC classifier is continually being retrained with real-time data then it may not detect a gradual long-term change to the system.

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

In cite:pyayt2015combining 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-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 3 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.

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

In cite:krzhizhanovskaya2011flood 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

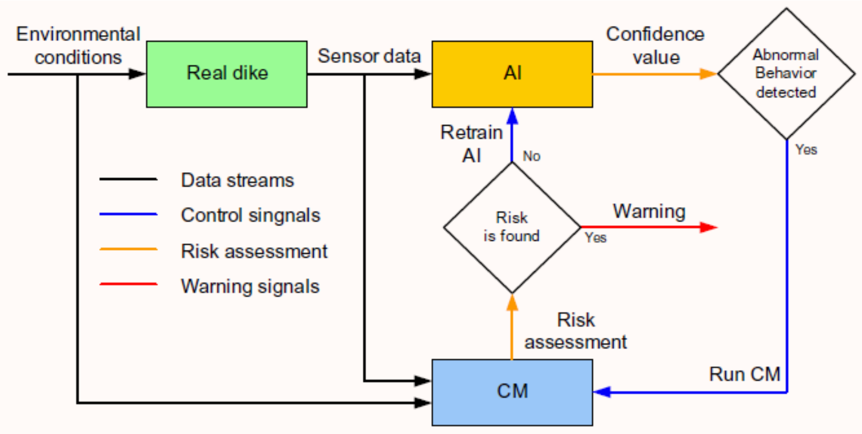

Figure 3: AI and CM...

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 4 along with a description.

2.5 A clustering approach for structural health monitoring on bridges

In cite:diez2016clustering a clustering based approach is presented to group substructures or joints with similar behaviour and to detect abnormal or

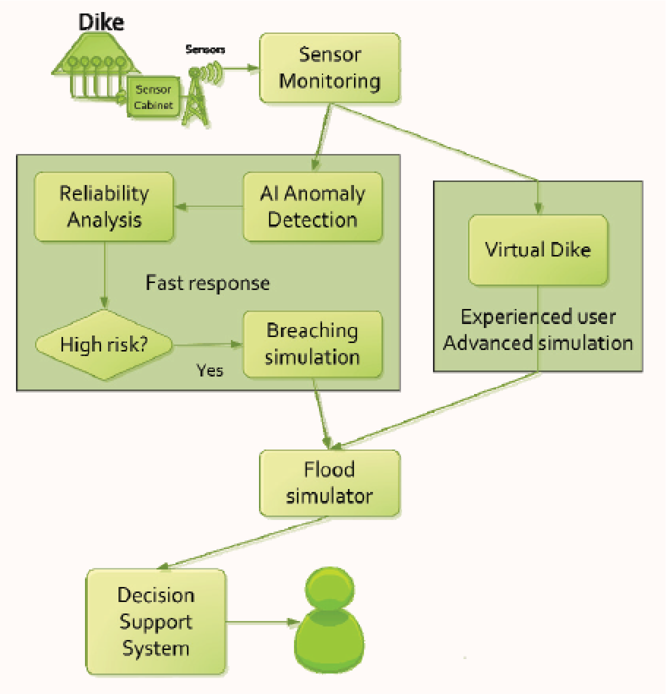

Figure 4: The *Sensor Monitoring* module receives data from the installed sensors which are then filtered by the *AI Anomaly Detector*. In case an abnormality is detected the *Reliability Analysis* calculates the probability of failure. If the failure probability is high then the *Breach Simulator* predicts 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.

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 $i$th 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 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 5 where joint 4 is clearly an outlier.

A frequency profile of both the big and small cluster are shown in Figures 6 and 7. In case there is no knowledge of abnormal behaviour then this method can be used to separate outliers and obtain a profile of normal

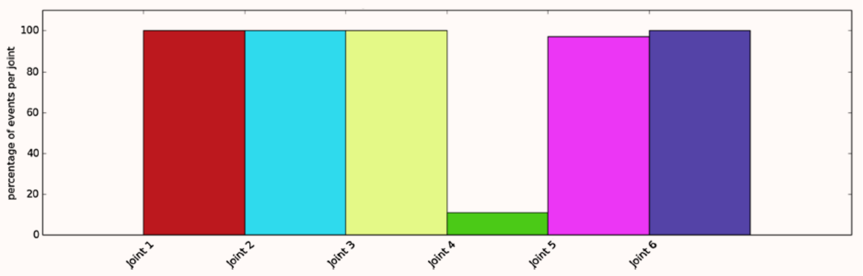

Figure 5: ...

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 8. 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.

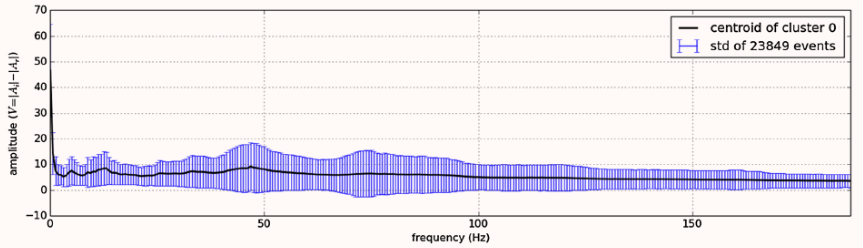

Figure 6: ...

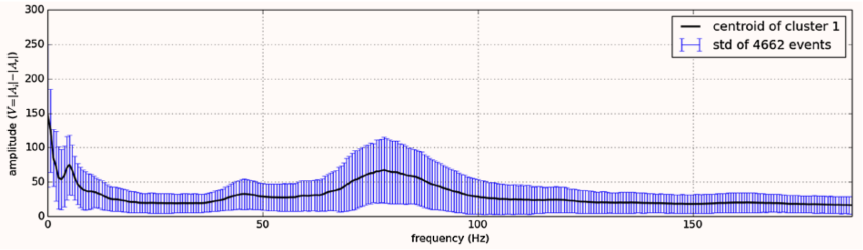

Figure 7: ...

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

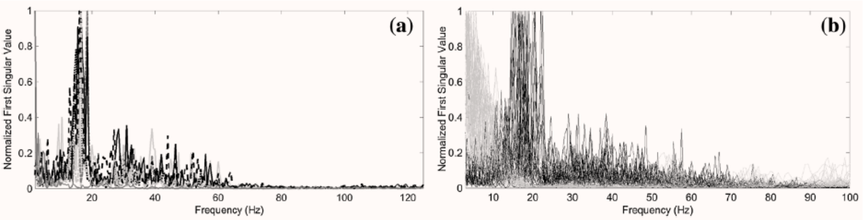

Figure 8: ...

removal phase. Two experiments were conducted. One experiment consisted only of 6 joints, including the damaged joint 4. The clustering method detected the damaged joint as can be seen in 9. The second experiment was run on data from 71 joints. The resulting map can be seen in 10 which accurately detected the damaged joint 135. Damage was also detected in joint 131 but this result was not verified.

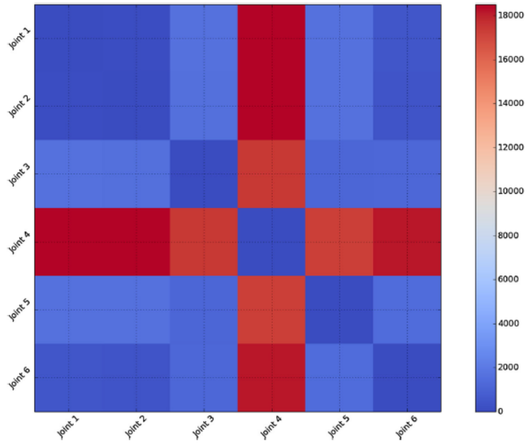

Figure 9: TODO:CAPTION

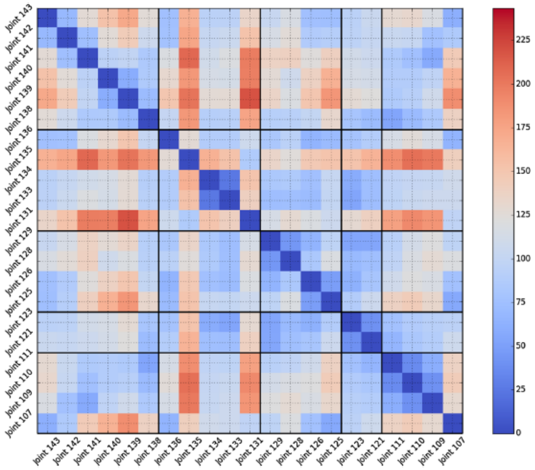

Figure 10: TODO:CAPTION

2.6 Conclusion

3 Methods

3.1 Simulated responses

3.1.1 Neccessary data

3.1.2 Finite element model

3.1.3 Collected data

3.1.4 Data analysis

3.2 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.

3.2.1 Damage scenarios

The goal of the damage identification model is to identify damage in a number of selected damage scenarios. Damage scenarios can be classified as short-term or long-term events. Short-term events 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 events 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 1 and 2 cite:sousa2019tool outline some of the predominant types of damage due to short-term and long-term events respectively.

Table 1: 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
Excessive vibration	Earthquake
Impact	Impact pressure by water and debris during floods

Table 2: 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

Of the damage scenarios listed in Tables 1 and 2, 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 (cite:li2010predicting).

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.

3.2.2 Test setup

3.2.3 Data analysis

3.2.4 Damage identification model

3.3 Decision support system

3.3.1 Sensor placement

3.3.2 Cost-benefit analysis

3.3.3 Uncertainty

4 Results

4.1 Simulated responses

4.2 Damage identification

4.3 Decision support system

5 Conclusion