NTNU
Norwegian University of
Science and Technology

Updating of numerical models for enhanced dynamic assessment of existing structures

Analysing parameters for model updating of a riveted steel railway bridge, based on the sensitivity method

Babæk Abnar
Annum Iftikhar Akhtar

Master of Science in Civil and Environmental Engineering
Submission date: June 2016
Supervisor: Anders Rönnquist, KT
Co-supervisor: Gunnstein Thomas Frøseth, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Institutt for konstruksjonsteknikk

FAKULTET FOR INGENIØRVITENSKAP OG TEKNOLOGI

NTNU – Norges teknisk-naturvitenskapelige universitet

MASTEROPPGAVE 2016

for

Babæk Abnar og Annum Akhtar

Oppdatering av numeriske modeller for bedret analyse av egenskaper til eksisterende konstruksjoner

Updating of numerical models for enhanced dynamic assessment of existing structures

Numeriske modeller av konstruksjoner er essensielle hjelpemidler for dimensjonering av nybygg, identifisering av ytre påvirkninger på konstruksjonen og vurdering av skader på eksisterende byggverk. Usikkerhet knyttet til utvalgte modellparametere og modelleringsvalg fører i visse tilfeller til at numeriske modeller ikke er i stand til å predikere responser med tilfredsstillende nøyaktighet. Målinger av responsen på eksisterende eller tilsvarende konstruksjoner kan brukes til å bestemme modellparametere og identifisere modellfeil for å bedre modellens evne til prediksjon i og utenfor måleområdet.

Oppgaven bør inneholde følgende temaer:

- Grunnleggende metoder for oppdatering av elementmodeller
- Implementering og sammenligning av forskjellige metoder for modell oppdatering
- Identifisering av aktuelle oppdateringsparametere gjennom konstruksjon og sensitivitetsanalyse
- Systemidentifikasjon og valg av responsvariabler for oppdatering
- Analyse av oppdaterte responsvariabler og oppdateringsparametere

Studentene velger selv hva de ønsker å legge vekt på

Besvarelsen organiseres i henhold til gjeldende retningslinjer.

Veileder(e): Anders Rönnquist og Gunnstein Frøseth

NTNU, 08.02.2016

Faglærer, Anders Rönnquist

Sammendrag

Oppdatering av en numerisk modell av en virkelig konstruksjon, er et viktig verktøy for å forbedre modellen, slik at den representerer konstruksjonen mer nøyaktig. En pålitelig og nøyaktig numerisk modell av en eksisterende konstruksjon er nødvendig, for å kunne vurdere den nåværende tilstand av konstruksjonen, og forutsi dens gjenværende levetid. Dette er spesielt relevant for eldre konstruksjoner, hvor rivning og ombygging av disse kan være svært kostbart, og konstruksjoner som bør bli bevart på grunn av høy kulturhistorisk verdi.

Ved å begynne med den teoretiske bakgrunnen, fører denne masteroppgaven leseren gjennom hele prosessen for sensitivitets-basert modelloppdatering av en konstruksjon. Den viser hvordan en bør tenke når man oppdaterer en modell, med som formål om å optimalisere prosessen.

For å vise modelloppdatering i praksis, har det blitt gjennomført et fullstendig studie, hvor to konstruksjoner har blitt oppdatert; en rigg og naglet jernbanebro av stål. Fokuset ligger hovedsakelig på viktigheten og effekten av ulike parameter valg, på oppdateringen. Det omfatter sensitivitetsanalyse for både riggen og broen, av parametere som er viktige for disse konstruksjonene. For oppdateringen, har et skript blitt laget i Python, som også er lagt ved som vedlegg for bruk og til nytte for de interesserte leserne.

Riggen er installert i Materialteknisk laboratorium ved NTNU, og er en representasjon av en gangbro. Analysen av denne er brukt som et enklere eksempel på modelloppdatering, for å lede gjennom den grunnleggende ideen, og påpeke forhold som er viktige å vurdere når en oppdaterer en konstruksjon.

Den naglede jernbanebrua, Lerelva Bro, er en fagverksbro bygget i 1919, og er en av de mange eldre bruene som er del av det norske jernbanenettet. Jernbaneverket, er interessert i å finne den gjenværende levetiden på denne broen, og derfor er en pålitelig og nøyaktig numerisk modell etterspurt. Den utførte analysen, leder gjennom en måte å tenke på når en oppdaterer modellen av en slik bro, viser eksempler på oppdateringer, og gir en mer praktisk forståelse av modell oppdatering.

Abstract

Updating a numerical model of a real structure is an important tool to enhance an existing model, such that it represents the structure more accurately. To assess the current state of a structure, and predict the remaining service life of it, it is necessary to have a reliable and accurate numerical model of the structure. This is especially relevant for old structures, demolishing and rebuilding of which may be very costly, and structures that have to be preserved because of their high cultural and historical value.

Starting with the theoretical background needed, this thesis leads the reader through the whole process of sensitivity-based model updating of a structure. It guides through a way of thinking when updating a model, aiming to optimize the process.

To show model updating in practice, a complete study has been carried out, where two structures have been updated; a rig and a riveted steel railway bridge. The focus lies mainly on the importance and effects of different parameter choices on the updating. It includes a sensitivity analysis of parameters that are important, for the rig as well as the bridge. For the updating, a script has been made in Python, which is also attached as appendix "A1-The script" for the use and benefit of the interested readers.

The rig is installed in the "Materialteknisk" laboratory at NTNU, and is a representation of a pedestrian bridge. The study is used as a simpler example to lead through the basic idea of model updating, and point out factors that are important to consider while updating a structure.

The riveted railway bridge, "Lerelva Bridge", is a truss bridge built in 1919, and is one of many old bridges that are part of The Norwegian railway system. The owner, "Jernbaneverket", is interested in determining the remaining service life of this bridge, and therefore a reliable and accurate numerical model is needed. The study conducted, leads through a way of thinking when going forward while updating such a bridge model, showing examples of updating, and giving a more practical understanding of model updating.

Preface and Acknowledgements

This master thesis is written in the spring semester of 2016, at the Institute of Structural Engineering. It is the final work of a 2-year Master's degree at the Norwegian University of Science and Technology (NTNU), and constitutes 30 credits. Professor Nils Erik Anders Rönnquist has been the main supervisor, and PhD-Candidate Gunnstein Thomas Frøseth the co-supervisor through the semester.

The reader is expected to have some prior knowledge in structural dynamics and Finite Element Method, in order to read and understand this thesis. The recommended prior knowledge is covered, among others, by the courses *"TKT-4192 Finite Element Methods in Strength Analysis*", "*TKT-4201 Structural Dynamics"* and "*TKT-4108 Dynamics, Advanced Course",* which are lectured at NTNU.

The goal of this thesis is to analyse dynamic responses of FE models, and the structural parameters that influence the responses. Our main motivation for this thesis was to acquire deeper and new knowledge within structural engineering and dynamics. In addition, we wanted to acquire some programming techniques, and use this in the thesis. It was important for us that the thesis contains practical aspects as well as theoretical, therefore the acquired knowledge was used to analyse a riveted steel bridge. We therefore see this thesis as a great opportunity and a great challenge for us.

We would like to thank our supervisors, Professor Nils Erik Anders Rönnquist and PhD-Candidate Gunnstein Thomas Frøseth, sincerely for their guidance, support and most importantly, motivation through the entire semester. Their genuine interest in this thesis has been very valuable for our accomplishments. We have learned a lot throughout this spring, and feel that we have achieved our goals. We also want to thank Ragnar Moen for assistance in the laboratory, and Bartosz Siedziako for modelling the bridge used in this thesis. At last, thanks to all the other professors and students who have discussed and helped us with the thesis, at the Institute of Structural Engineering, at NTNU.

Babæk Abnar   Annum Akhtar

Trondheim, June 2016

Contents

List of Figures

List of Tables

INTRODUCTION

1.1 Background

In civil engineering, numerical models are used to estimate the behaviour or response of a structure, to for example identify external effects or evaluate the damage on the structure. When making numerical models, the engineer would have to make approximations and simplifications due to the complexity of the real structure, and limitations set by the modelling tools. There will also be simplifications done by manufacturer of materials related to material properties, used in the model. At construction site, it is impossible to build the structure exactly as given in the drawings, hence causing more difference between the model and the real structure. This result in inaccurate parameters used in the model, such as material properties and dimensions, and cause inaccurate behavioural estimates of the structure.

Model updating may be used to correct the numerical model, such that it matches the real structure with a greater accuracy. This is done after the structure is built, by taking measurements of the real response of the structure, and comparing it to the one estimated by the numerical model. Some of the model parameters are chosen to be updated by such an updating procedure.

Choosing the correct parameters for updating is the key to a successful updating. There is no common method of going forward, since a good choice of parameters will be very different from structure to structure. This thesis investigates and discusses how one should go forward to choose these updating parameters, which factors one should take into account and how the different choices will affect the updating. Such that the true errors are corrected, and the updated model represents the real structure.

1.2 Area of Focus

The area of focus in this thesis has been chosen carefully, by a dialogue between the students and supervisors, where the interests of both parties were taken into account. Basically the topic would be as simple as, "*Updating of numerical models for enhanced dynamic assessment of existing structures"*, but to emphasise the chosen areas of focus, a more descriptive topic is formulated as:

Analysing parameters for model updating of a riveted steel railway bridge, based on the sensitivity method;

How to go forward for updating a structure?
How to choose parameters for the updating?
What are the effects of parameter choices on the updating results?

1.3 Limitations and Delineation

A complete updating of a riveted railway bridge is an extensive task that will acquire more time and resources than what is disposed for this thesis. Therefore, some aspects of model updating are either simplified or completely disregarded in this thesis:

- Damping is disregarded due to its small impact on the responses of steel structures
- Only one updating method is implemented; the sensitivity method.
- Manual model tuning is not done prior to the updating, assuming that the model already is very close to the real structure
- Only the given measured response is used for the updating, no further measurements are taken.

THEORY

2 Model Updating

2.1 Introduction

Today, very powerful computers and FE software are used when modelling civil engineering structures. Still, significant inconsistencies between the estimated and measured responses of the structure may occur. Model updating aims to correct FE models with such errors or deficiencies. The reason for errors is not only assumptions made during modelling itself, but also variances in for example material data provided by the manufacturer, or simplifications done at site during construction. The extent and significance of these errors and deficiencies vary greatly, from structure to structure and model to model.

A numerical model can estimate the behaviour and response of a structure at given external actions; called analytical response or results. A completed structure will have an actual response to the external actions; called measured response or results (Mottershead and Friswell, 1993). When discrepancies between analytical and measured results arise, one can update the numerical model, so that it will be able to represent the real situation in a satisfactory manner.

One way to reduce these discrepancies is to identify model and parameter errors and adjust these, until satisfactory accuracy in the analytical results is obtained. This obviously may be done by trial and error, but would be very time-consuming, and in some cases not possible. Thus, several methods have been developed which, using measured response of the real structure, update the numerical model. Discrepancies in the results are then reduced to a minimum efficiently.

Updating a numerical model of a structure requires measured data as an input. This limits model updating procedures to be performed only after the completion of the structure. The usefulness of an updated model can still be great. A close to exact numerical model may for example be used for long-term analysis of structures, such as estimating remaining life expectancy, detecting damage, analysing structural damage and analysing a substructure of a new structure. Another benefit is that it can also be used for educational purposes, for example, to learn why the initial assumptions and simplifications were wrong, and how to make better assumptions and simplifications in future modelling (Ren and Chen, 2010).

2.2 Sources of Error in the Model

Correcting the real errors in the model should be the main goal of model updating. Only then will the model be able to represent the real structure exactly. However, detecting the exact cause of error is very difficult.

In addition, not all the errors are possible to correct by updating. The features of the model that cannot be corrected by adjusting parameters are called model-structure errors. The analyst should be aware of all possible sources of error, and whether or not a model updating procedure would be able correct them. Mottershead et al. (2010) have listed up such errors in categories (1) and (2), as shown below:

(1) Idealisation errors resulting from the assumptions made to characterise the mechanical behaviour of the physical structure. Such errors typically arise from:

- simplifications of the structure, for example, when a plate is treated like a beam, which might or might not be erroneous depending on the length to width ratio of the plate and the frequency range to be covered
- inaccurate assignment of mass properties, for example, when distributed masses are modelled with too few lumped masses or when an existing eccentricity of a lumped mass is disregarded
- when the finite element formulation neglects particular properties, for example, when the influence of transverse shear deformation or warping due to torsion in beam elements is neglected
- errors in the connectivity of the mesh i.e. some elements are not connected or are connected to a wrong node
- erroneous modelling of boundary conditions, for example, when an elastic foundation is assumed to be rigid
- erroneous modelling of joints, for example, when an elastic connection is assumed to be rigid (clamped) or when an eccentricity of a beam or a plate connection is omitted from the model
- erroneous assumptions for the external loads
- erroneous geometrical shape assumptions
- a non-linear structure assumed to behave linearly

(2) Discretization errors introduced by numerical methods such as those inherent in the finite element method, for example:

- Discretization errors when the finite element mesh is too coarse so that the modal data in the frequency of interest is not fully converged
- truncation errors in order reduction methods such as static condensation
- poor convergence and apparent stiffness increase due to element shape sensitivity

If a numerical model holds idealisation (1) or discretization (2) errors, the updated model may not be able to reproduce measured response outside the response range. It is therefore important that the model is correctly discretised and holds minimal idealisation errors.

Category (3), below, shows examples of errors that are possible to correct by a model updating procedure:

(3) Erroneous assumptions for model parameters, for example:

- material parameters such as Young's modulus or mass density
- cross section properties of beams such as area moments of inertia
- shell/plate thicknesses
- spring stiffnesses or non-structural mass

However, even if there are no idealization or discretization errors, a wrong choice of updating parameters, may lead to the same problem. Again, since the selected parameters are not the real source of error, the updated model will not be able to reproduce dynamic properties of the structure outside the response range (Mottershead et al., 2010).

2.3 Updating Methods

There are many ways to update an FE model, but not all are equally popular or well established. Various updating methods have been verified through extensive research and applied successfully to full-scale industrial structures. However, there are essentially two main ways to update a finite element model; by a direct method or by an iterative method (Friswell and Mottershead, 1995).

Direct methods are among the first methods developed for model updating, and can be considered as global one-step methods. These methods are based on updating the entire global stiffness- and mass- matrices, without considering the physical parameters that build these. By using measured data, these matrices only, are updated so the model is able to reproduce the measured response exactly, within the response area. This is accomplished in one step, which makes these methods extremely effective.

Matrix update method is an example of direct methods, and is usually used to detect and localize damage in the structure. This may be accomplished by changing system matrices, to minimize the difference between analytical and measured responses. The analyst identifies damage in the structure, and localizes it by comparing the initial system matrices to the updated ones. It should then be clear where the error or damage is, and somewhat the magnitude of it (Friswell and Mottershead, 1995).
One weakness of direct methods is that the updated model is not physically correct, since the physical parameters are not updated. Therefore, the updated model is not able to predict responses outside the response area with small discrepancies. Another weakness is that the direct methods require high quality test date as well as many measurements in

order to conduct an updating procedure (Grafe, 1998). Maia and Montalvao Silva (1998) points out weaknesses and limitations with such methods, and prefer iterative methods for model updating (Marwala, 2010).

Many fields use system identification to get a correct representation of the process between data input and data output, including control engineering and biology. System identification is a method for representing a dynamic system with mathematical models, based on the information about the system's input and output signals (MathWorks, 2016). In structural engineering, According to Mottershead and Friswell (1993), Natke first used the term *direct* system identification to emphasise the absence of an initial model, and stated that model updating could be seen as an *indirect* system identification.

Unlike control engineering, model updating in structural engineering is used to modify the physical properties of the model. Mottershead and Friswell (1993), describes how this makes the physical meaningfulness of parameters a necessity in model updating, as opposed to system identification in control engineering.

Unlike direct, iterative methods are based on updating local physical parameters, such as geometric or material properties. The parameters are changed iteratively by comparing measured and analytical results for each iteration, such as natural frequencies. The process repeats itself for a convergence criterion is achieved. This can be a very computationally expensive method, but in return, you get a robust model that can represent reality also outside the response range (Grafe, 1998).

The two best-known methods for model updating is sensitivity method and response surface method. Sensitivity method is the most popular and is known as a very robust and efficient method (Brownjohn and Xia, 2000). It requires multiple simulations of the model for each iteration in the construction of the sensitivity matrix. If the FE model is very large, this can be very demanding. Ren and Chen (2010) propose using Response Surface Method in such cases. However, with today's powerful machines, it is conceivable that the FE model must be enormously large and with high degree of nonlinearity for this to be relevant.

2.4 Measured Response

2.4.1 Choosing Response Variables

Validation of a model is the first, and one of the key steps in model updating. This is done by comparing the analytical and measured results of the same response variable, within a given response range. There are several possibilities when choosing a response variable to be used for updating. Generally, they all fall under two main categories, dynamic or static.

It is shown, by Mottershead & Friswell (1993), that modal data, such as natural frequencies and mode shapes, obtained from measured frequency response, can be used as a target when adjusting parameters. Such dynamic response variables have successfully been used in parameter updating, also for correction of industrial scale FE models (Mottershead et al., 2010). The natural frequency residuals are defined as the difference between vector of measured, z_m, and analytical, z_j, natural frequencies as shown in Equation 1.

Equation 1

$$r_i := z_m - z_j$$

When deformations, are chosen as static response variables, one usually uses input force as target for updating. Bakhtiari-Nejad et al. (2005) did this in their study as they evaluated the analytical force, by multiplying the measured deformations with the analytical stiffness of the structure. The difference between the input force and the analytical force is then the residual, given by Equation 2:

Equation 2

$$r_i := f_1 - K \cdot u_m$$

Where f_1 is the input force vector, K is the global stiffness matrix of the structure and u_m is the vector of measured deformations.

Bakhtiari-Nejad et al. (2005) argues strongly for the use of static response variables for damage detection in structures, and uses deformation as the response variable successfully. One argument is that dynamic methods require considerably more accurate measurements of the mode-shapes, to eliminate false excitations. This can be difficult to achieve on an industrial-scale structure. Therefore, measured static response can be seen as more precise than dynamic response (Bakhtiari-Nejad et al., 2005).

To use static measurements, such as deformation, one must know the exact value of the applied load. This is done by loading the structure with specific, known loads. On industrial-sized structures such as a bridge, this requires the bridge to be closed for traffic, hence is unpractical. On the other hand, extracting dynamic responses does not require such loading of the structure, making it a more suitable choice.

Using dynamic response variables is common for model updating, and there is a lot of literature that supports this procedure, among others, Mottershead et al. (2010), Esfandiari et al. (2010), Rad (1997), Ren and Chen (2010). There are, however, some disadvantages of using dynamic response variables.

One of them is that dynamic response depends not only on stiffness, but also mass and damping of the structure. This can create difficulty in parameter identification, unlike static approaches, where the only parameter is stiffness.

On the other hand, this can also be a limitation when using static response variables; mass parameters will not affect the analytical results, although these parameters could be the source of discrepancies in the response of the numerical model.

When industrial scale models have been updated, often, dynamic response variables are preferred. For instance, Mottershead et al. (2010) uses natural frequencies in the model updating of a Lynx helicopter airframe. Brownjohn and Xia (2000) also uses natural frequencies when updating a cable stayed bridge.

Dynamic response variables are also successfully used in detecting damage in a structure, for example in Cawley and Adams (1979). Also, Marwala (2010), uses dynamic response variables, when comparing different methods for damage detection.

As discussed above, the use of both dynamic and static response variables is well documented. Sensible response variables should be chosen, based on what the updated model needs to represent, in correlation of what is practical. For example, if a dynamic analysis has to be performed on the updated model, natural frequencies may be selected as the response variable. However, how practical it is to take measurements will usually be the conclusive factor for which response variable is chosen.

2.4.2 Mode Pairing

After the response variables are chosen, one has to make sure that the analytical and measured data belongs to the same mode before validation. This is called modal paring. There are several techniques for modal paring, and one of them, Modal Assurance Criterion (MAC), is discussed and used in this study. This is a well know technique and is used in several case studies globally.

The MAC is generally given by Equation 3, where c is the reference, d is the degree of freedom, r is the mode number, T is transpose, cc is the complex conjugate and ϕ is the mode shape vector (Allemang, 2003).

Equation 3

$$MAC_{cdr} := \frac{\left[\left|\left(\phi_{cr}\right)\cdot\left(\phi_{dr}^{\ cc}\right)\right|\right]^2}{\left(\phi_{cr}\right)^T\cdot\left(\phi_{cr}^{\ cc}\right)\cdot\left(\phi_{dr}\right)^T\cdot\left(\phi_{dr}^{\ cc}\right)}$$

The MAC gets a value between zero and one, where zero means no correlation between the mode shape vectors at all. A MAC value equal to one means there is a 100% correlation between the two vectors. This makes it simple to pair modes and their associated responses.

Mottershead et al. (2010) accepts a MAC value greater than 0.75, as a good indication for pairing two modes. However, the MAC value might not be near the two extremes, making it difficult to see if there is a match.

This may happen as a result of too few, or not correctly placed measuring points on the structure. The mode shape vector will then not be able to describe the mode shape in detail, hence it will be difficult to distinguish between two similar mode shapes. This might result in more than one match to the same mode.

To avoid problems like this, one can use more measurement points, and place them in strategically chosen places, such that the MAC easily can distinguish between the mode shapes, and correctly pair the measured modes with their respective analytical modes mathematically. Otherwise, the analyst has to use engineering judgement to complete the pairing of modes. One may for example look at the frequencies to see if there is any indication there, or if any other mode matches clearly with one of the modes.

2.5 Sensitivity Method

2.5.1 Procedure

For updating in this study, the sensitivity method is used. The sensitivity method is one of the most successful methods for updating FEM models of engineering structures. It is an iterative procedure, which uses measurements from the vibrational test data of the real structure as an input. The aim of the procedure is to minimise the objective function, that represents the error in the FEM model, with respect to the measured data from vibration test of the real structure;

$$J(x) := \varepsilon_z$$

Here, the error (ε_z) in the model is taken as the difference between the measured data (z_m) and the associated FEM model (analytical) data (z_θ).

$$\varepsilon = z_m - z(\theta_i)$$

The relationship between the measured data and the parameters of the model that need to be corrected, is non-linear, but is linearized in the procedure by truncating the nonlinear terms of a Taylor series expansion;

Equation 4

$$\varepsilon := z_m - z(\theta_i) \approx r_i - G \cdot \Delta\theta_i$$

Where r_i, is the residual at the i-th iteration, $\Delta\theta_i$ is the parameter modification. The method primarily builds on the sensitivity matrix of the structure, G_i, which is given by

Equation 5

$$\mathbf{G}_i = \left[\frac{\partial z_j}{\partial \theta_k}\right]_{\boldsymbol{\theta} = \boldsymbol{\theta}_i}$$

Where j=1,2,….,q denotes the output data points and k=1,2,…,p is the parameter index. The values in G_i are actually the level of change in the structure's response, to a small change of the parameter value, i.e, how "sensitive" the structure is to that change. Each column of the sensitivity matrix contains the sensitivities to one particular parameter, for the respective modes, i.e, rows of the matrix.

By minimizing the objective function, and altering Equation 4, the following equation gives the required parameter change;

$$\Delta\theta_i := G^{-1} \cdot r_i$$

The sensitivity matrix will be rectangular if equation 1 is underdetermined or overdetermined, depending on the number of response variables and parameters chosen. In that case, the inverse sensitivity matrix can be calculated as the pseudo inverse of G:

$$G_{pseudoinverse} := \begin{cases} G^{-1} & \text{if } n = m \\ \left[\left(G^T \cdot G\right)^{-1} \cdot G^T\right] & \text{if } j > k \\ \left[G^T \cdot \left(G \cdot G^T\right)^{-1}\right] & \text{if } j < k \end{cases}$$

The calculated value of $\Delta\theta_I$, is then used as an input for the next iteration. This whole procedure continues until satisfactory level of accuracy is achieved for the model, i.e. minimal value of residual.

In order to emphasise specific modes, weighting of the residual vector is needed, by introducing a symmetric weighting matrix, We, making the objective function;

$$J(\boldsymbol{\theta}) = \boldsymbol{\varepsilon}_z^T \mathbf{W}_\varepsilon \boldsymbol{\varepsilon}_z$$

There are various ways of choosing the weighting matrix, but it should at least be able to account for the difference in amplitudes of the different modes. Hence, to increase the importance of lower modes, a good choice will be;

$$\mathbf{W}_\varepsilon = [\mathrm{diag}(\mathbf{z}_m)]^{-2}$$

An ill conditioned sensitivity matrix can be a problem and may cause divergence. The ill conditioned, noisy system of equations is typically a problem when vibrational measurements are used. In order to treat this, regularisation may be used. Regularisation puts a requirement of minimal parameter change, $\Delta\theta$, by introducing another weighing matrix W_θ and a regularisation parameter, λ. This weighing matrix is chosen to represent the uncertainties of the initial parameters.

Whether regularisation is needed or not, may be decided based on the condition number of the sensitivity matrix, given by:

$$\kappa(\mathbf{G}_n^T \mathbf{W}_e \mathbf{G}_n)$$

If the condition number is very low, the sensitivity matrix is well conditioned, and there is no need for regularisation. In this thesis conditioning problem has not occurred, hence regularisation has not been necessary.

2.6 Choice of Parameters

2.6.1 Introduction

The art of model updating lies primarily in the choice of parameters to be updated. Different choices of parameters will lead to different results, and therefore have a great influence on how successful the updating will be. Still, there is yet no known explicit, systematic or mathematical way of going forward. Instead, choosing parameters is based generally on a profound analysation of the structure and its behaviour, together with engineering judgement.

The updated model might be used to extract or reproduce different types of structural data, for example to further analyse the structure. It is therefore important to know how well the data from the updated model actually represents the real structure.

By updating a model, it will always be possible to reproduce the measured data, as long as the equation is determined or overdetermined. However, one should know that the updated parameters are not true values, but estimated. This is because the measured data will always contain some noise, due to a number of random and systematic errors causing pollution. Mottershead et al. (2010) However, that is not possible to take into account or correct while choosing parameters for updating.

One should also keep in mind that even if the model is able to reproduce the measured data correctly, theoretically, not all the data would necessarily be exactly like the real structure. For instance, very large changes in parameters to compensate for other errors in the model, may lead to a model with different properties than the real structure. Therefore, one should be careful while using data from the updated model, such as structural properties, deformations or natural frequencies of other modes.

The goal of updating should not be to reproduce only the measured data gained from the real structure, but to be able to represent the structure as a whole, with all its properties and responses. That means, basically, to reproduce all the natural modes of the structure correctly, within and outside the measured set of response. To achieve that, there are various factors that should be considered, which are discussed below.

2.6.2 Sensitivity Analysis

If the updated parameter values are far from the initial values, they are probably far from the real structural properties as well. In that case, the model will not be able to represent all the other natural modes of the model correctly. To avoid that, the changes in parameter values should be kept as small as possible during the updating.

One way of doing that, is to choose parameters by their sensitivities. Basically, the sensitivity of a parameter is a way of measuring how much the response would change, given a small change in that parameter. i.e., the parameter's influence on the dynamic response of the structure. Therefore, by choosing the most sensitive parameters, one would ensure that the desired decrease in residual is achieved, with only small changes in parameter values.

To get a proper overview of the sensitives of parameters, it is conducted a sensitivity analysis of the possible updating parameters, prior to each updating. This involves calculating all the parameter sensitivities and comparing these to one another, while considering each natural mode and its amount of residual.

The sensitivity matrix calculated in Equation 5 contains the absolute sensitivities. If sensitivities of different types of parameters are to be compared, such as density and E-modulus, the relative sensitivities should be used;

$$G_{relative} := \left(\frac{\delta z_j}{\delta \theta_{k.0}} \right) \cdot \theta_{k.0}$$

Furthermore, one can also normalise the relative sensitivity matrix with respect to the response value;

$$G_{normalized} := z_j^{-T} \cdot \left(\frac{\delta z_j}{\delta \theta_{k.0}} \right) \cdot \theta_{k.0}$$

Such a representation of normalized relative sensitivities of the parameters, can be a good starting point of choosing the parameters to be updated. The low sensitivity parameters, which have no effect on the response variables of the structure, can then be "filtered out". While, it helps identify those parameters that must be modelled in a precise manner to achieve accurate results.

As mentioned, it will always be possible to get a solution to the updating problem, and reproduce the measured data with some combination of parameter values. However, whether the values make any sense or not is not a matter of course.

Even if highly sensitive parameters are chosen, it may lead to large enough changes in parameter values, such that the parameters loose their physical relevance. The reason might be that the chosen parameter try to compensate for various errors in the model. To cope with this, constraints should be set up in the form of upper and lower bounds for each parameter. These bounds are chosen based on engineering judgement of what a realistic, and hence allowable, value of each parameter can be.

2.6.3 Error Localization

In order to achieve a better result, such that the model represents the structure correctly, it is important to try to correct the real errors in the updating. Hence, another important factor to consider is the localization of error, and choosing a parameter that is able to represent that error. That can be very challenging of course, and requires good engineering skills and understanding of the structure.

There are various proposed methods of going forward. For example in Zang et al. (2012), an evaluation of Equivalent Element Modal Strain Energy (EEMSE) and Equivalent Element Modal Kinetic Energy (EEMKE) is used to localise the errors in the finite element model. A model of an existing aero engine casing is used as the *actual* structure, while the same model but with a zone assigned with reduced value of Young's modulus, is used as the analytical model for the updating. The results show that the error is correctly localised using EEMKE and EEMSE, and then a successful updating of only that part of the casing is done.

The scope of this case study however, does not allow such an evaluation to be conducted. Hence localisation of error is analysed qualitatively, based on engineering judgement alone.

An evaluation is conducted of the most probable locations of mistakes in the model, based on experience and what is typical for such a structure. These areas, such as foundations, where there might be many simplifications involved, should be included in the updating and hence the related parameters should be chosen. One might also eliminate some parameters based on how certain they are. See also *"2.2 Sources of Error in* the Model*".*

2.6.4 Number of Chosen Parameters

How many parameters are chosen for the updating procedure, plays an important role in model updating. In an industrial scale structure the error will obviously be related to several parameters. The intention of the user should be to correct as many parameters as possible to ensure that one, or a few updated parameters do not compensate for errors located elsewhere, but rather correct its own error. In this way it is more likely to get an updated model that can reproduce responses, both in the response range and outside it with great accuracy.

The number of chosen parameters should be less than the number responses that is to be evaluated, due to noise in the measurements, and because one might be interested in evaluating other responses outside the measuring range.

Obviously, if number of chosen parameters equals number of responses, the equation system will be determined and it will give one unique solution. This is a good approach if the user has succeeded in measuring every response in the relevant response range with no noise, and does not intend to evaluate the structure outside this range. However, as mentioned, there will always be some noise, and it would be extremely difficult to measure all the relevant responses of interest. For instance, if the user has measured five out of seven frequencies within a range, the user should select less than five parameters for updating, i.e. an overdetermined system, such that the updated model can represent the remaining two responses with satisfactory accuracy as well as the measured five. It is therefore important that the user choses a reasonable number of updating parameters, such that the updated model is able to reproduce other modes with satisfactory accuracy.

One way of controlling the updated model is by updating the model with only a few of the measured responses and then to use the other measured responses as control by comparing them to the analytical responses. This is called cross validation and is proven an efficient way of controlling the updated model. To do this, the user would need a great number of measured responses. For instance if the user only has five or six measured responses, it would be difficult for the user to choose enough parameters to correct the model in a satisfactory manner, without reaching the number of responses used in the updating while still saving a few measurements for later control.

Clustering of Parameters

While evaluating the parameter sensitivities, one might realise that many parameters are almost relatively equally sensitive in each mode. This can be observed by looking at the sensitivity matrix, where the relevant parameter sensitivities would have almost the same relation to one another in all the modes. If two or more parameters can be clustered, this implies that their change will affect the responses in the same way, not necessarily with the same magnitude.

One way of decreasing number of updating parameters, and the work or time required for the updating, is to cluster those parameters that effect the responses in a similar way, such that one single column of the sensitivity matrix represents all of them. In other words, clustered parameters behave as one single parameter. Mottershead et al. (2010) mention that one may determine whether the sensitivities of the parameters are close enough to cluster the parameters, is by the condition of the angle between their respective column vectors being less than 5 degrees;

Equation 6

$$\cos\gamma_{\alpha\beta} = \frac{\mathbf{g}_\alpha^T \mathbf{g}_\beta}{(\mathbf{g}_\alpha^T \mathbf{g}_\alpha \cdot \mathbf{g}_\beta^T \mathbf{g}_\beta)^{1/2}}$$

3 Finite Element Model

3.1 Introduction

Finite element method (FEM) is a numerical method used to predict the behaviour of a structure, or analyse it. This is done in a software by dividing a large complex structure into smaller pieces called finite elements. A mathematical model of each element predicts its behaviour. All these then added up as a prediction of the whole structure.

Finite element analysis (FEA) is the practical use of FEM; FEM is used to model the structure and then carry out an analysis in an FE software. A structure modelled in an FE software is called an FE model, and is a numerical representation of the real structure.

Scientists and engineers all over the world use FEA to solve complex problems in different fields, such as civil engineering. There are several benefits of carrying out an FEA, as a tool to predict structural behaviour. For instance, may FEA be used to optimise geometrical or material properties of a structure.

Several types of analysis can be carried out depending on the purpose of the analysis. Therefore there exist several different FE software; some custom-made for their industries and others with a high degree of generality, for example Abaqus. Obviously, a more general FE software is most probably also more complex and advanced, than a custom-made or simplified software.

Appropriate modelling is key to a successful updating, and puts different requirements to the model than for conventional analysis of a structure. If the residual values are very large, truncation of the Taylor series expansion to first order, in sensitivity method, may lead to divergence of the updating. Therefore, in order to achieve convergence, the initial model should be able to give response variables relatively close to the measured values. This is done by *Manual tuning* of the model, prior to updating.

A high level of detail in geometric and structural modelling is an important requirement to achieve physical significance of the updated model. Brownjohn and Xia (2000) first updated model of a curved-cable stayed bridge, with a relatively simplified deck. This lead to a model with maximum error of 15% in the frequencies, with a 100% change in six of the parameters, hence losing its physical relevance. The same bridge was then modelled with a more detailed deck and then updated. This time the maximum difference between measured and analytical frequencies was 10%, with only 30% change in the parameters at the most.

However, which parts of the structure should be modelled in detail and which can be simplified, requires engineering judgement and an understanding of their relevance to the dynamics of the structure. No matter how greatly detailed a part is modelled, if it does not have any effect on the dynamics of the structure, the updating will not be improved.

Since model updating updates physical parameters, it is necessary to represent the uncertainties in the model quantitatively. For example, a damaged part of the structure can be represented by “weak elements”. The parameters related to these elements, when updated, will then represent the extent of damage in that zone. The uncertainties of supports for example, can be represented with support springs. Updating their stiffnesses may lead to correct simulation of the boundary conditions.

3.2 Abaqus

The model of the bridge analysed later in this thesis, is modelled in the FE software, Abaqus. It has been decided to further model and carry out the analysis required for this thesis, also in Abaqus. As mentioned earlier, Abaqus is an FE software that gives the user a lot of freedom when analysing, because of its high generality and modelling capability. This among other reasons is why Abaqus is a popular software for academic and research purposes, and also in industrial problems. Abaqus is used in a vast variety of fields, such as structural engineering, biomechanics and fluid mechanics.

The main interactive space of Abaqus, Abaqus/CAE, is a complete environment where the modelling itself is done, analysis are submitted, jobs monitored and results evaluated. Abaqus/CAE consists of modules, where each module is used to define and create the different aspects of modelling, such as defining geometry and generating a mesh. Going through these modules, leads to the generation of a complete model, which may then be submitted for analysis. A subset of Abaqus/CAE is the Abaqus/Viewer, where all the results can be processed and displayed with the Visualization module.

The Abaqus finite element system includes various programmes designed for different types of analytical purposes. The three main programmes to work with are:

- Abaqus/Standard, a general-purpose finite element program;
- Abaqus/Explicit, an explicit dynamics finite element program;
- Abaqus/CFD, a general-purpose computational fluid dynamics program;

(Simula, 2013)

There are also various add-ons, which can be used to further extend the modelling possibilities in Abaqus/Standard and Abaqus/Explicit. For example, is Abaqus/Design used with Abaqus/Standard to perform design sensitivity analysis, while Abaqus/Aqua is aimed for analysis of underwater structures subjected to currents and wave actions. Abaqus co-simulation technique may also be used for coupling between Abaqus and a third-party analysis. All these available options make Abaqus an FE programme with a vast variety of modelling tools and techniques.

In addition to the GUI (graphical user interface) in Abaqus/CAE, there is an option of interacting with Abaqus directly through commands in the Abaqus Scripting Interface.

Abaqus uses the object-oriented programming language, Python, throughout the software, and gives user the option to directly communicate with the “brains” behind Abaqus/CAE; kernel. Abaqus Scripting Interface commands can be stored in a file as a script, which can then be run from within Abaqus/CAE. This possibility of using Python to communicate with kernel, further extends the modelling and analytical capabilities of Abaqus, making it even more flexible.

Analysis is carried out by the Abaqus/Standard or Abaqus/Explicit, with the use of input file from Abaqus/CAE. During the analysis, one can monitor the job from Abaqus CAE, and at the end, an output database is generated. All the commands executed in the Abaqus/CAE are stored in the replay file. The visualisation module is used to read the odb (output database) and display the results.

In order to understand how to interact with kernel through scripting, one has to understand how Abaqus works, where the files generated are stores and how to access the data. Since the modelling is already done, and this thesis focuses on the model updating itself, a script is only needed to carry out the updating and to work with the results generated. The data needed for this will be found in the ODB files generated when a job is executed.

The ODB (Output Data Base) contains two main types of data; the model data and results data. The tree below shows the paths to the different data stored in ODB.

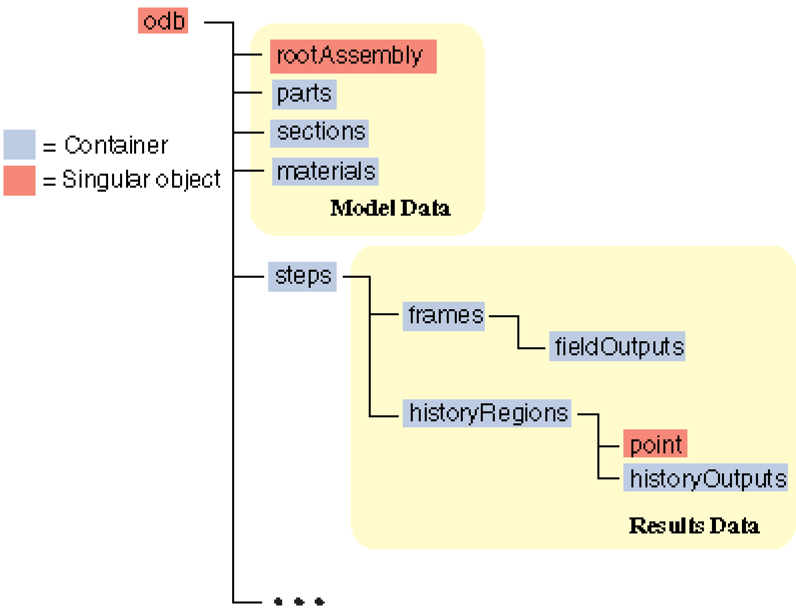


Figure 1- (Simula, 2013)

ODB can be used to extract information about the model itself, such as the section properties and material properties, or to extract the results from the analyses (Simula, 2013)

STUDY

4 Script for Updating

In order to carry out the sensitivity based iterative updating described earlier, a script of the procedure has been set up in Python. The script has been made reasonably general, to keep a wide range of options open, for updating parameters through the study. However, there still had to be some limitations. As described earlier, the choice of parameters to be updated is quite complex and there are many different types of parameters to choose from. It has been decided to include E-module, density, inertia and spring-stiffness in this script. Possible number of updating parameters of each type is actually unlimited in the script. It is believed that these four types of parameters cover the most important areas to be updated in a model, discussed under the chapter "2.6 Choice of Parameters". Cross sectional area, which is another commonly used type of parameter for updating, has been omitted. The reason is the complexity in the general script it would cause, and that it may easily be represented, by for example E-modulus and density.

The script is applicable to updating a model based on eigenfrequencies only, as the measured response variable. The measured frequencies is a necessary input, while all the analytical frequencies needed in the procedure, are calculated with Lanczos Eigensolver in Abaqus by the script. These are extracted from the odb files generated, for further use. Modal pairing is carried out by evaluating MAC after every eigenfrequency calculation. Hence, the measured mode shape vector is also needed as an input for the script.

The sensitivities in the sensitivity matrix is based on a small change in each parameter. This small change is chosen by the user himself, as a percentage of the initial/current parameter value, as an input. For evaluating the values of sensitivity matrix, updated Newton Raphson method has been used. At each iteration, the parameters are given a small change, and the resulting change in eigenfrequencies are calculated. This will give the sensitivity matrix as the tangent of the parameter-response function, at the iteration point. Weighing matrix has also been included in the script, with values recommended under the chapter: "2.5 Sensitivity Method, 2.5.1 Procedure"

As very large changes in the parameters can cause loss of physical meaning, an ill conditioned sensitivity matrix and divergence of solution, it is possible to set lower and upper bounds for each parameter to be updated in the script. Then, each parameter will only be updated within the corresponding region chosen by the user.

When the parameter has reached a bound, the updating might be trying to change the parameter further beyond it, to minimise the objective function. In such a case, even if there is a sensitivity, the script should not consider it while calculating the required parameter change, as it will not be used because of the restriction set by bounds. To account for that, when the parameter approaches its lower or upper bound in two consecutive iterations, the whole column representing it will be eliminated for the next iteration. It is advised to add such a parameter back when the solution has converged, and see if the parameter adjusts itself within the bounds.

The script has been made with the intention, that other interested readers also may use it in the future. Therefore, all the necessary inputs are arranged at the top of the script, with the corresponding explanation for the user, and is attached to the thesis as Appendix A1 as well as uploaded digitally with the thesis. This script may also be copy into a text editor and then run by the user.

Verification of the Script

To test the script, a simple spring-mass model has been made in Abaqus, shown in Figure 1. The model is a two degree of freedom system, with two point masses, M_1 and M_2. These are connected by three springs, with spring stiffness, k_1, k_2 and k_3. The reason for making such a simple model for testing is to verify the implementation of algorithm. The model has two degrees of freedom making it possible to analyse a model with more than one mode.

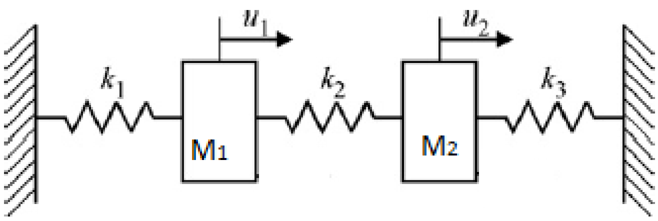


Figure 1, simple spring-mass model

The test was carried out by giving values to the five parameters mentioned above, and then calculating the two natural frequencies of the system. These natural frequencies were further treated as the *measured* frequencies of the model.

The second step was to give the model an *error* with respect to the initial model. For that, one of the parameters was given a different value than the initial one, and then new natural frequencies were calculated. These would then represent the analytical response of the model for the test. This gave a residual as the difference between *analytical* frequencies and *measured* frequencies of the model. Hence, the changed parameter was the source of discrepancies in the natural frequencies.

Next, the *measured* frequencies, calculated in the first step, were used as an input to the script for updating the *analytical* model. The expectation was of course, that the erroneous parameter would be changed back to its original value, after the updating

The test was run several times with different parameters. First, both spring and mass parameters were tested individually. Then two parameters were evaluated at the same time, one mass and one spring parameter. All tests showed that the chosen parameters changed back to their original values. As a result, the *measured* and *analytical* frequencies matched, making the residual equal to zero at the end of each updating.

The test is a theoretical case with fictive parameter values and no noise. This is why the residual became zero after updating. When real structures are analysed, one cannot expect zero residual in all modes. There will always be some difference in analytical and measured response, as a result of the expected noise in measurements.

5 Model Updating of Rig

The created script is implemented for updating of a rig structure in a lab. The rig is installed in “Materialteknisk” laboratory at NTNU for research and educational purposes, and is a representation of a pedestrian bridge.

The goal of this chapter is to illustrate the importance of choosing correct updating parameters, i.e. the real source of error in the responses. And what happens if the chosen parameters for updating are not the real source of error.

In addition, any defects in the script can be found and corrected, such that the script can be validated. This case is a more realistic case compared to the simple spring-mass case shown earlier, because this is an actual structure where actual measurements has been taken, it is therefore expected that the measurements will contain some noise.

5.1 Rig Description

The rig consists of a six-meter long and one-meter wide section (deck), point masses, and springs and cables as supports. Figure 2 and 3 shows that the rig is supported by three springs in series connected in four points in the horizontal transverse direction, by cables in horizontal longitudinal direction and in vertical transverse direction. The springs are assumed to be much softer than any other stiffness contributing component, making the rig’s primarily modes to be rigid body motion, i.e. it can be described with a lateral and a rotational degree of freedom. As shown in Figure 3, the rig has an evenly distributed mass in addition to two point masses, ½ M_P each, located in the middle of the longest edges. These masses are marked orange in Figure 3.

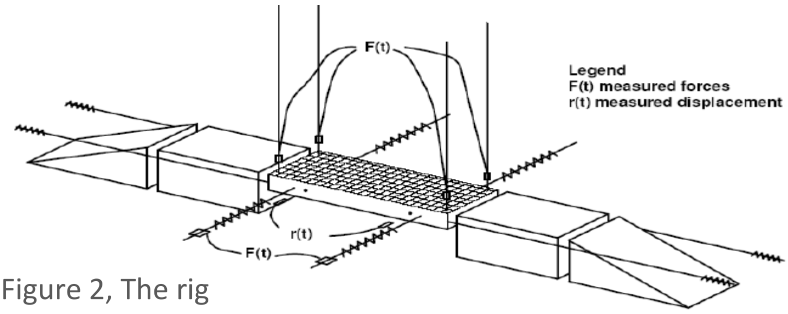


Figure 2, The rig

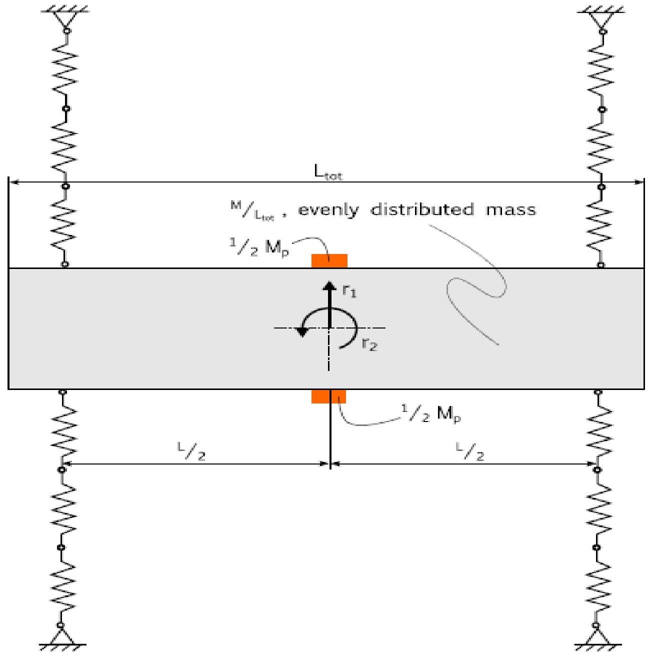


Figure 3, bird view of the rig

5.1.1 Parameter Data

The properties are given by:

- L_{TOT}= 6m
- L= 4.8m
- M_P= 310 Kg
- M=? (distributed mass)
- k_s=? (spring stiffness of one single spring)

At the outset, the spring stiffness and the distributed mass was unknown, so measurement needed to be taken. Measuring of the distributed mass was simply done by weighting the whole rig, and then subtracting M_p. The total mass of the rig was measured as 475Kg, which gives M=165Kg in total.

The measuring of the spring stiffness was done by stretching eight springs in series and measuring the deformation at different loadings. Then it was carried out a linear regression to calculate the stiffness of each spring. The measurements of the eight-spring system is shown in Table 1, and the linear regression is shown in Figure 4. From the function, it is clear that the spring stiffness for each spring is equal to 12 385N/m. Further, the line almost passes through the axis origin (zero force gives 0.04mm deformation), which suggest that the linearization is almost perfectly correct.

Table 1, Measurements for stiffness calculation of springs

Weight in Kg	Deformation in mm
0	0
15.1	97
26.1	165
43.7	277
51.2	325

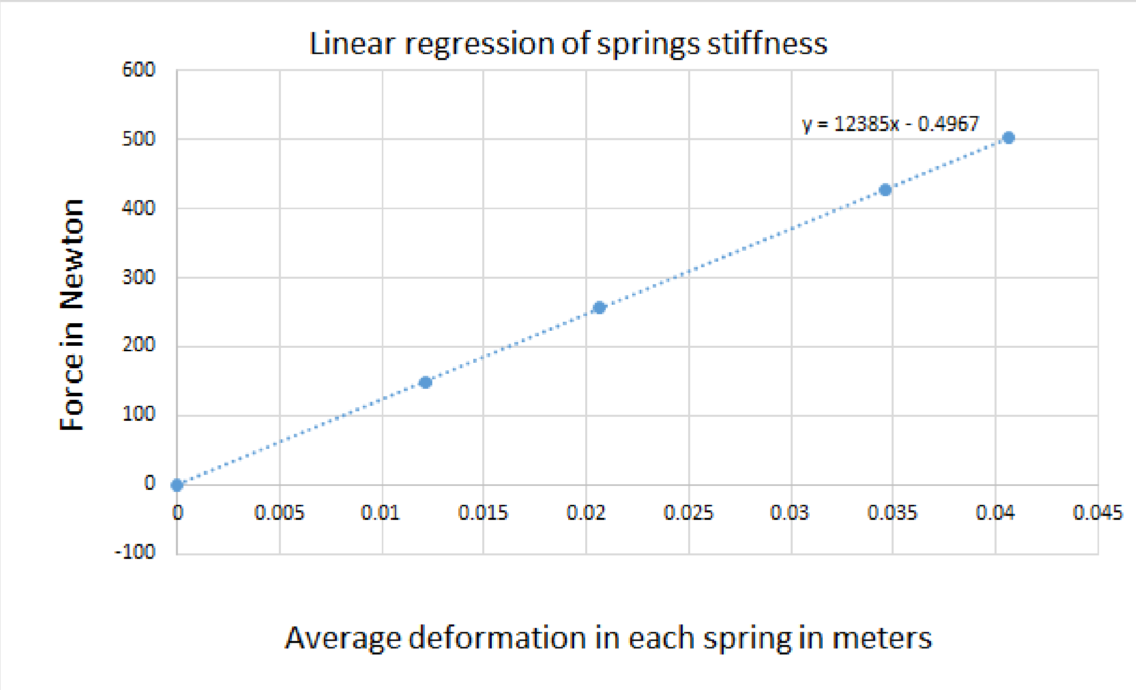


Figure 4, linear regression of measurements

5.1.2 Modelling of the Rig

The rig is analysed in Abaqus, here the section is modelled solid as a simplification. The reason for this is that shell elements have no volume, and adding mass to a part with no volume are problematic. Therefore, a solid section is chosen such that the material density could represent the distributed mass. The section is also constrained, such that it does not get strained, i.e. the section is infinitely stiff and it will be restricted to move only as a rigid body. The stiffness properties of the material assigned to the section would normally influence the global stiffness of the model, but in this case, it will not influence the global stiffness due to the constraint applied to the section. The same applies for the mesh, the mesh is somewhat coarse, with 60 (3x20) solid elements in the deck. However, as mentioned, because of the constraint applied to the deck, whether the mesh is fine or coarse does not influence the system stiffness as the deck is considered to not deform, i.e. infinitely stiff.

The boundary conditions are applied to a reference point, which is located in the middle of the section, with degrees of freedom only as translation in the transverse direction and as rotation in the horizontal plane, i.e. about the vertical axis. Further, it is assigned two point masses (two times 1/2 M_p) as shown in Figure 3. The supports are modelled as single springs at two points as shown in Figure 5. Each of these springs would have to represent six springs, i.e. two parallel sets with 3 springs in series in each set. That gives **Feil! Fant ikke referansekilden.**, where, k_s is the spring stiffness of one physical spring and k_{eq} is the equivalent spring stiffness used in the FE model. This gives k_{eq} equal to 8 256N/m.

Equation 7

$$k_{eq} := \frac{2}{3} \cdot k_s$$

Figure 5, FE model of the rig

5.2 Response Variables

The response variables are chosen as natural frequencies, mostly because this will be the response variables in the next case study as well. Then the educational purpose of this case study would be greater in conjunction with the next case study. The measured frequencies are obtained by exiting the rig manually in the two modes and extracting deformation data in time-series at the connection points of the springs. The translation and rotation in the midpoint is then derived from the measured data.

It is then done a Fast Fourier transform of the data, presented as amplitudes in a frequency domain. Then the Fourier amplitudes of all four time-series are plotted against frequencies such that it is obvious which frequencies that gives the biggest amplitudes, i.e. the natural frequencies. This is presented in Figure 6, here it is clear that there are two main modes, the first natural frequency is 0.97323601Hz and the second is 2.0517052Hz. That the four curves lie almost exactly upon each other indicates that the measurements are done almost correctly, however it should be kept in mind that there will be some noise associated with these measurements, even though this is done in relatively unnoisy environment at a laboratory.

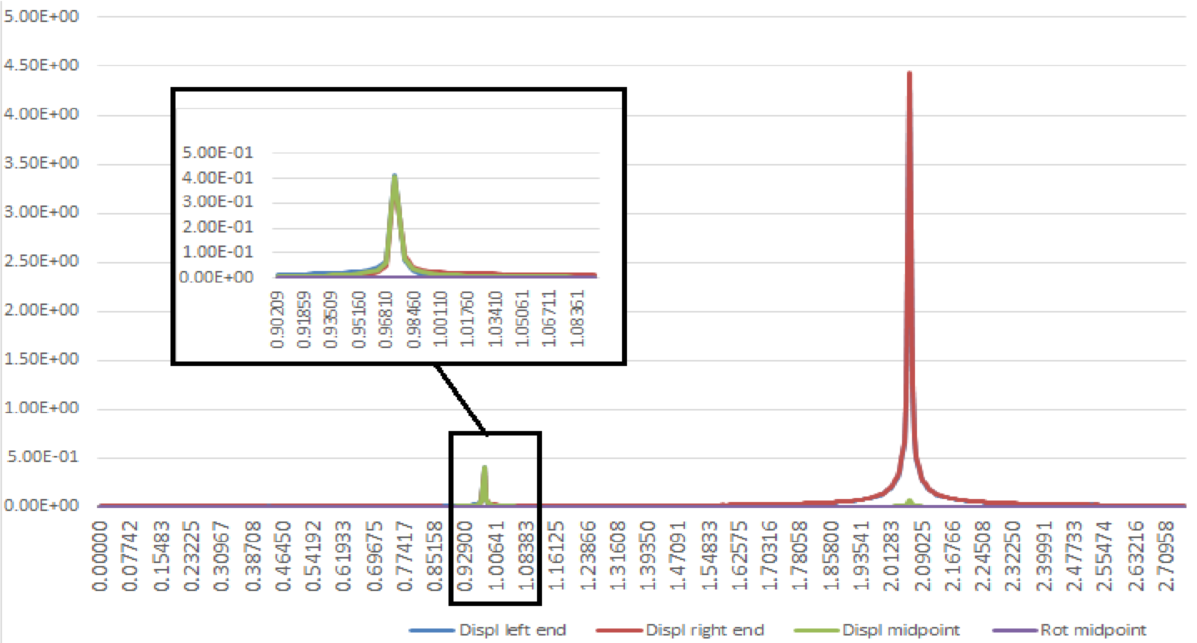


Figure 6, Fourier amplitudes vs frequency

Mode 1 and 2 are displayed in Figure 7 and Figure 8 respectively. It is shown that mode 1 is purely translation in the horizontal transverse direction, and that mode 2 is pure rotation in the horizontal plane, i.e. about the centre point of the section.

Figure 7, Mode 1 The Rig

Figure 8, Mode 2 The Rig

5.2.1 Residual

The residual is defined as measured natural frequencies subtracted analytical natural frequencies as shown in Equation 1. The measured and analytical frequencies, as well as the residual and percentage of error is displayed in Table 2.

Mode number	Measured Frequency in Hz	Analytical Frequency in Hz	Residual in Hz	Error in %
1	0.97323601Hz	0.93841Hz	0.03493Hz	3.59 %
2	2. 0517052Hz	2.0178Hz	0.0339Hz	1.65%

Table 2, Residual case study: The Rig

5.2.2 Modal Pairing

The pairing of the two modes is considered not to be necessary for analyzation and updating of the rig, this is because it is easy to distinguish between the two different modes and their corresponding frequencies. Therefore, the Mac values have not been evaluated.

5.3 Parameters

Choosing of parameters should generally be a profound analysation including engineering judgement and a sensitivity analysis, evaluating which parameters that could be the source of discrepancies and whether or not change in these parameters would correct the model to represent the measured data with satisfactory accuracy. In this case of updating the rig, there are only three relevant parameters; the spring stiffness, the point masses and the distributed mass represented by material density of the deck. Therefore, it has not been conducted any extensive analysation when choosing parameters. However, the sensitivities are evaluated and possible error sources have been discussed to get a better understanding of the structure and the errors. Further, all of these three parameters have been evaluated.

5.3.1 Identification of Error Sources

Before choosing updating parameters, it is important to identify the possible sources to discrepancies in the response. This will give a better understanding of the structure, about where the error might be located, whether or not it is a small or big error, and whether these errors contribute to negative or positive residual. It is therefore discussed which assumptions and simplifications are done regarding modelling of the rig, and what effect of these assumptions and simplifications have on the response.

- Mass:
  Even though it is used a modern digital weighing scale, which rounds off to the nearest ten kilograms it could still be some error in the measurements. It is assumed that the equipment that was used are well calibrated, and therefore the error in mass is minimum.

- Simplification of point mass:
  The point masses are obviously modelled as point masses, but in reality, these are metal weights, which are distributed along certain length. This will not make any difference in mode 1, rigid body translation, but in mode 2, rigid body rotation these masses would have an impact on the inertia and therefore this simplification will reduce the analytical natural frequency in this mode, compared to the reality.

- Since the rig is modelled to be rigid, one could think that whether the mesh is coarse or fine will not matter on the behaviour, but it is easy to forget that when distributed mass is represented by density the mass will be lumped to the nodes of the elements. With a coarse mesh, this could lead to some discrepancies.

- Stiffness of the springs:
  Previously it is shown how the spring stiffness is obtained. First, it is assumed that every single spring used have the same stiffness. This could be an error source, but it is expected to be extremely small, and therefore a fair assumption. The measurements done for calculation of the spring stiffness will have two error sources: one associated with the weighting scale, and the other when measuring the deformations manually by hand. There is related a big uncertainty especially to the latter simplification, which could contribute to a faulty model and needs to be kept in mind.

- Stiffness of the section:
  Because the springs are very soft compared to the section, the section is assumed infinitely stiff, this constrains the model to rigid body movements only. In reality, the section would, of course, be much stiffer than the springs, but not infinitely stiff. This assumption, even though it is expected to be a very good approximation, would contribute to a stiffer analytical system compared to reality.

- Stiffness contribution of horizontal longitudinal cables:
  The horizontal longitudinal cables are assumed to have stiffness only in the longitudinal direction. This is true for movement of the section in the longitudinal direction, but as soon as the rig is deformed in the horizontal transverse direction, the cables are no longer longitudinal, but deformed such that they have a stiffness contribution also in the transverse direction due to nonlinear geometry effects. These effects could, of course, be small if the cables are very long, but it should be kept in mind that this could be a big source of error. The simplification of neglecting such effects would make the model softer than in reality. It is therefore expected that the stiffness of the springs will be increased after updating, as a compensation for the missing stiffness contribution from the cables in the transverse direction due

to the nonlinear effects. This, of course, also relates to the vertical transverse cables. In Figure 9 it is shown how the longitudinal cables get deformed as the rig deforms, here in mode 1, transverse horizontal rigid body deformation. Clearly, the nonlinear effects due to deformations of the cables would influence the global stiffness in the transverse direction.

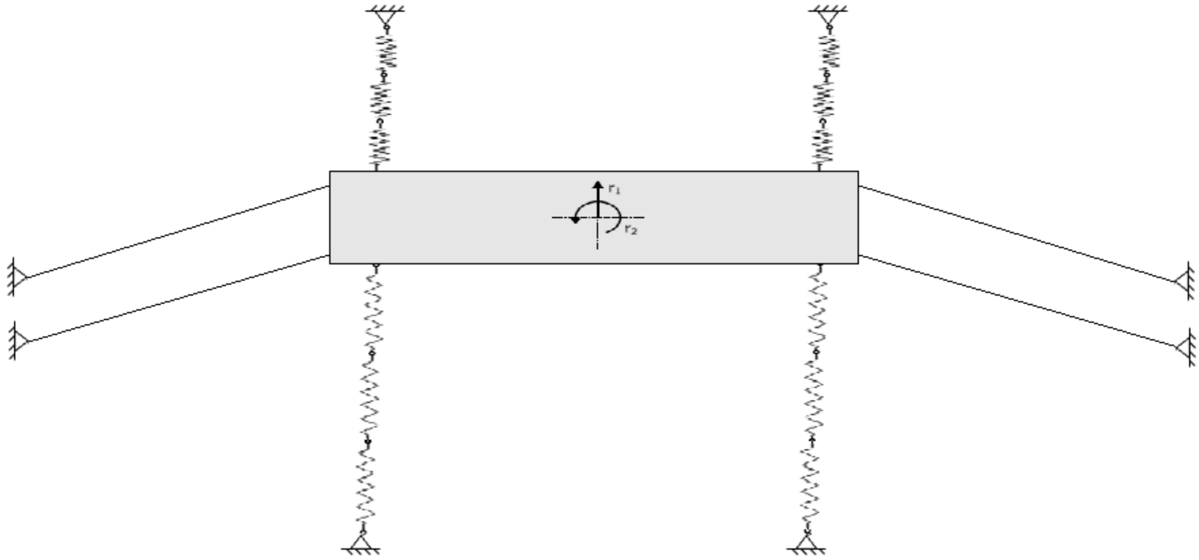


Figure 9, Deformed shape of the rig in mode 1

The analysation of identifying error sources points at the simplification of neglecting nonlinear transverse stiffness contribution from the longitudinal and vertical cables, as the main source of error. There is also big uncertainties associated with the measurements of both spring stiffness and total mass, but mostly spring stiffness, because manual measurements were taken. There are also other uncertainties, but they are considered rather negligible, and will not be evaluated further.

5.3.3 Sensitivity Analysis

To determine the sensitivities of each parameter, the parameters has been given one percent positive change. The sensitivities have then been normalized, as shown in *"2.6.*2 Sensitivity Analysis*",* such that the relative sensitivities of the different parameters can be compared. The sensitivities are also scaled, such that the highest sensitivity is equal to one. Figure 10 shows the absolute value of the scaled normalized sensitivities of all three parameters in the two modes. The sensitivity of the point mass and the density is obviously negative, because an increase in mass leads to a reduction in frequency. Because of this, the figure shows the absolute values such that it would be easier to compare the sensitivities of the different parameters with each other.

The figure shows that the springs are the most sensitive parameter in both modes, with almost equally sensitivity. The point mass obviously have a considerable sensitivity in mode

1, rigid body translation, and low sensitivity in mode 2, rigid body rotation. The reason for this parameter's sensitivity in mode 2 is the eccentricity of the point mass from the midpoint of the rig in the lateral direction. This gives rise to inertia effects in the longitudinal direction when the rig rotates in the horizontal plane. The figure also shows that the material density have a great sensitivity in mode 2.

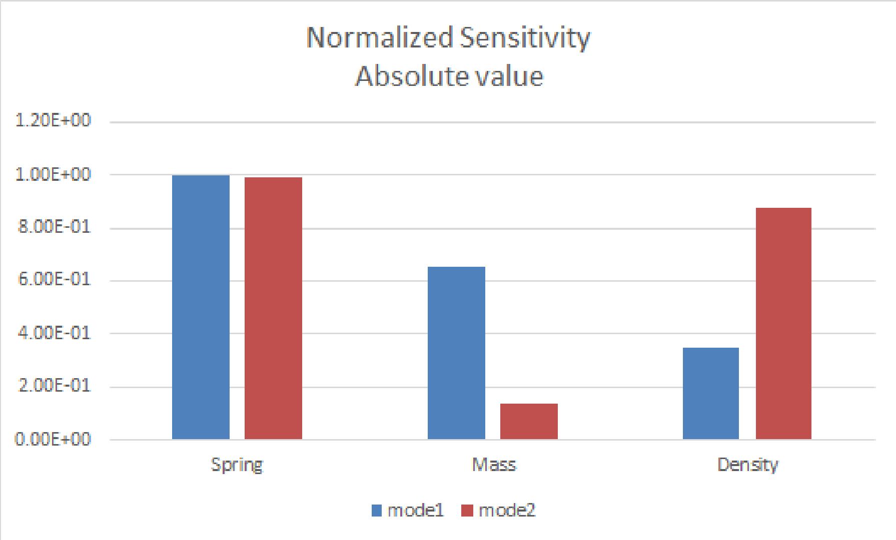


Figure 10, Sensitivities The Rig

5.3.4 Chosen Parameters

The analysation done when choosing updating parameters leads to spring stiffness as the best choice. Mostly because this parameter holds large error compared to the other parameters, but also because of its high sensitivity. This parameter can also compensate for the error related to the cables.

Nevertheless, it is expected that other parameters as well hold some error. Therefore, the two other parameters have also been evaluated for educational purposes. Lastly, it has been conducted an updating which combines two parameters. It is in total conducted four updating procedures on the rig.

Obviously, due to noise in the measurements of frequencies, the overdetermined equation system with two modes and only one parameter will not give one unique solution that satisfies both equations. Therefore, there will still be some error in response after updating in the three first updating procedures. For the same reason, in the last updating procedure there are two modes and two parameters, which will give one unique solution, regardless of any noise in the measurements.

5.4 Updating

5.4.1 Updating 1: Spring Stiffness

The spring stiffness has been evaluated as the updating parameter, nine iterations were conducted and solution has clearly converged as early as in the second iteration. As shown in Figure 11, both residuals decreases during the procedure, especially the residual in mode 2, which is also shown in Table 3. Even though the sensitivities of the spring stiffness in the two modes are almost identical initially, this could change as the parameter value changes, this has clearly happened since the residual in mode 2 has decreased faster than in mode 1. The sum of the absolute value of the residual is also shown in Figure 11 to indicate how the total error in the system develops; this value has decreased from 0.06883Hz to 0.02299Hz, which is a reduction of almost 67%.

Table 3, Result of updating1 of the rig

Mode number	Measured Frequency	Initial Analytical Frequency	Initial residual	Initial Error	Updated Analytical Frequency	Updated residual	Updated Error
1	0.9732Hz	0.9384Hz	0.0349Hz	3.59%	0.9574Hz	0.0157Hz	1.61%
2	2.0517Hz	2.0178Hz	0.0339Hz	1.65%	2.0590Hz	-0.0073Hz	-0.36%

Figure 12 shows how the spring stiffness develops during the procedure; the initial value was 8 256.67N/m and the final value became 8 597.87N/m. This is an increase of only 4% of the initial value, which is acceptable because the parameter does not lose its physical value. An increase of the stiffness is expected since the springs in the model not only represent themselves but also the stiffness contribution from the longitudinal and vertical cables when the system is deformed.

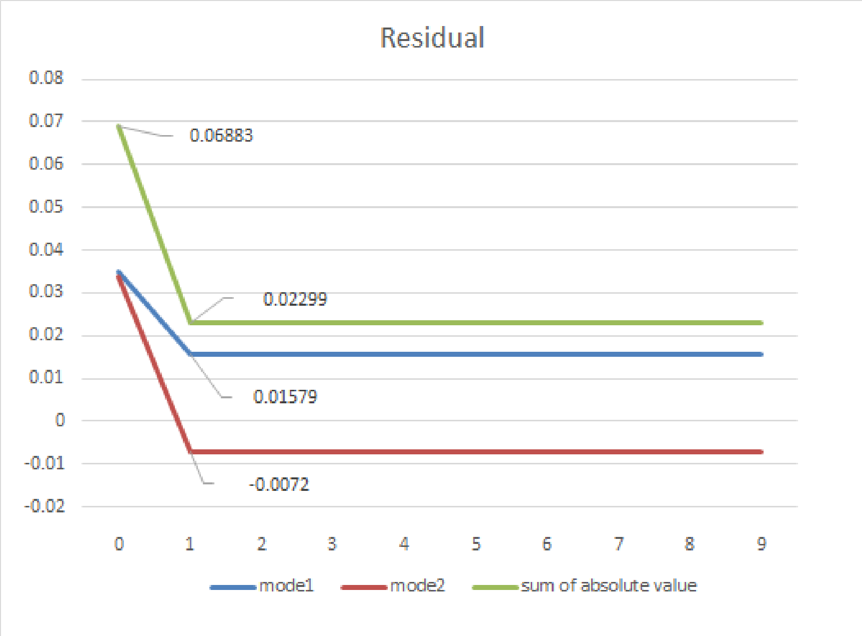


Figure 11, Residual development during updating1 of the rig

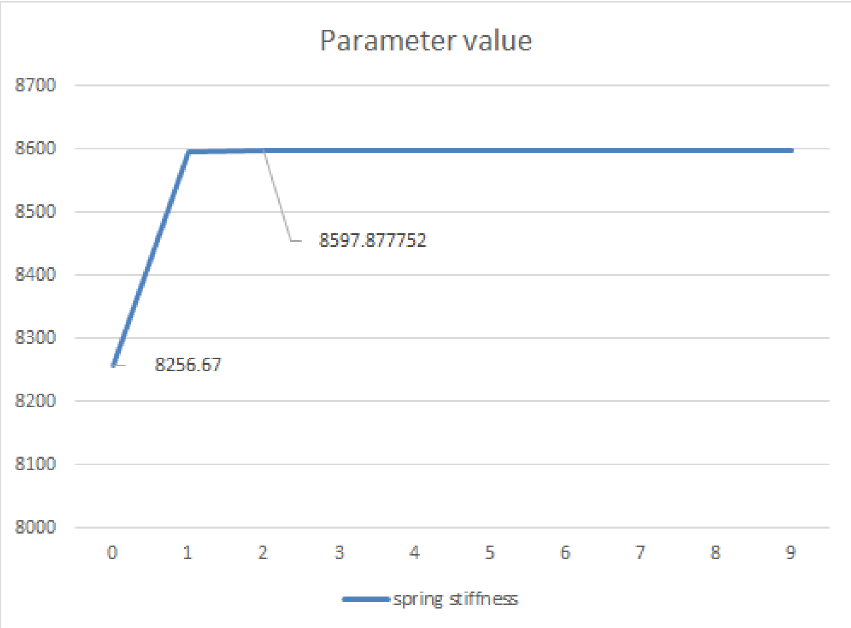


Figure 12, Spring stiffness change during updating1 of the rig

5.4.2 Updating 2: Point Mass

The updating procedure converge after two iterations only, but nine are shown in the figures. Figure 13 shows the development of the residual of both modes and the sum of the absolute value of both modes. It shows a considerably decrease in error, especially in mode 1, which make sense because of the parameter's sensitivity in this mode is larger than in mode 2. The total residual has decreased from 0.06883Hz to 0.02359Hz, this is a reduction of almost 66%, which is nearly the same as the previous updating, but the percentage error is rather similar in the two modes, unlike the previous updating. These results are shown in

Table 4.

Mode number	Measured Frequency	Initial Analytical Frequency	Initial residual	Initial Error	Updated Analytical Frequency	Updated residual	Updated Error
1	0.9732Hz	0.9384Hz	0.0349Hz	3.59%	0.9799Hz	-0.0067Hz	-0.69%
2	2.0517Hz	2.0178Hz	0.0339Hz	1.65%	2.0686Hz	0.0169Hz	0.82%

Figure 14 shows the development of the point mass value during the updating procedure. The point mass have an initial value at 155kg and a updated value at 135kg, this is a reduction of 13%, which is too much considered the small simplifications related to this parameter. However, a change in this parameter could also correct the error in frequency caused by other parameters, then the reduction in mass make sense because this change is partially a compensation of the neglected transverse stiffness from the cables.

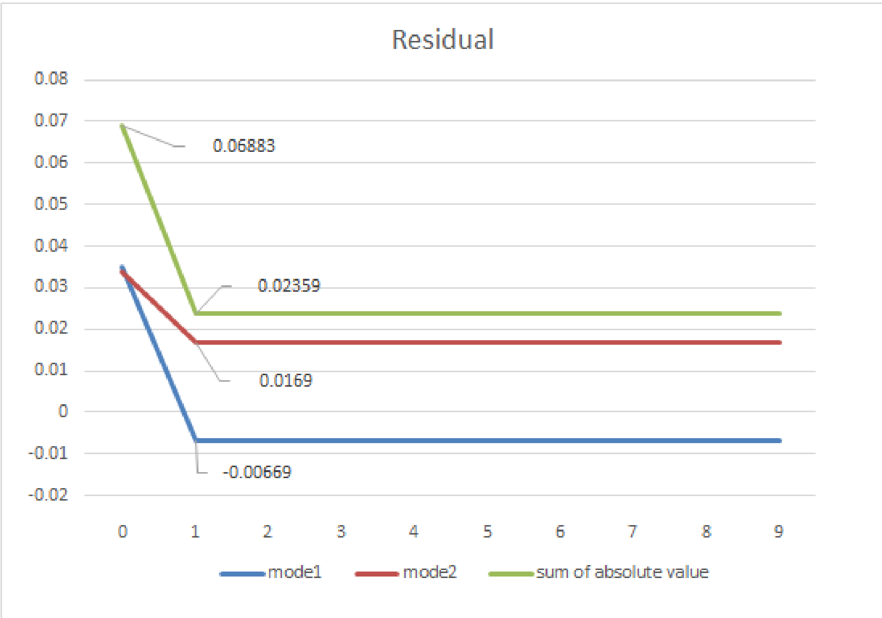


Figure 13, Residual development during updating2 of the rig

Table 4, Result of updating 2 of the rig

Mode number	Measured Frequency	Initial Analytical Frequency	Initial residual	Initial Error	Updated Analytical Frequency	Updated residual	Updated Error
1	0.9732Hz	0.9384Hz	0.0349Hz	3.59%	0.9799Hz	-0.0067Hz	-0.69%
2	2.0517Hz	2.0178Hz	0.0339Hz	1.65%	2.0686Hz	0.0169Hz	0.82%

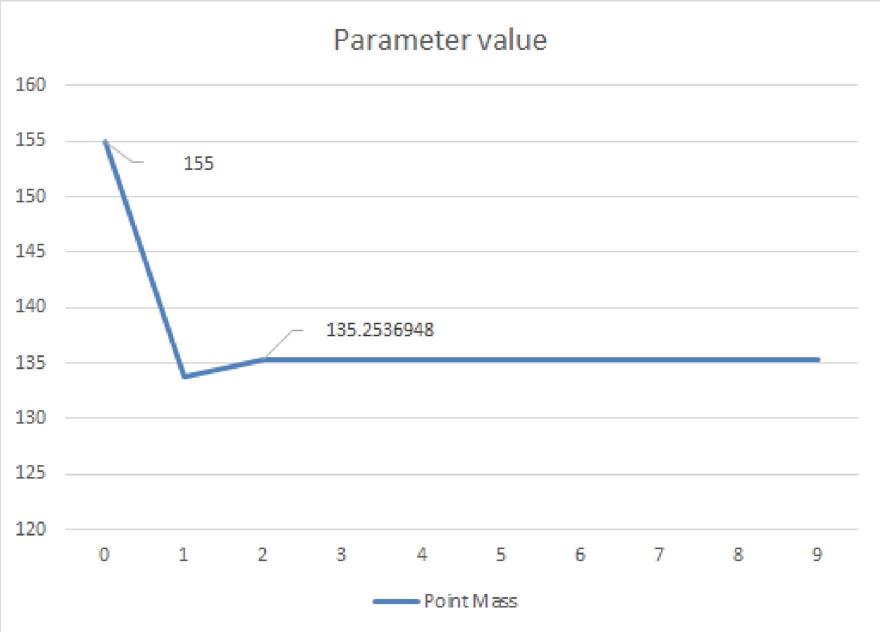


Figure 14, Point mass change during updating2 of the rig

5.4.3 Updating 3: Density

Again, the updating procedure converges after two iterations. Figure 15 shows the development of the residual of both modes and the sum of the absolute values of both modes. It shows a considerably decrease in error in mode 2, which was expected because of the parameter's high sensitivity in this mode. It was however expected a larger reduction of residual in mode 1 than what have occurred. Table 5 shows that the initial error in this mode was 3.59% and the updated error became 2.85%, which is not a good improvement. The total error has decreased from 0.06883Hz to 0.03372Hz; this is a reduction of 51%, which is not a good enough improvement.

Figure 16 shows the development of the material density during the updating procedure. The density have an initial value at 91.7kg/m^3 and an updated value equal to 87.7kg/m^3. This is a reduction of only 4%, which is a small change. Obviously, a bigger change in this parameter could minimize the error in mode 1, but then the error in mode 2 would have increased.

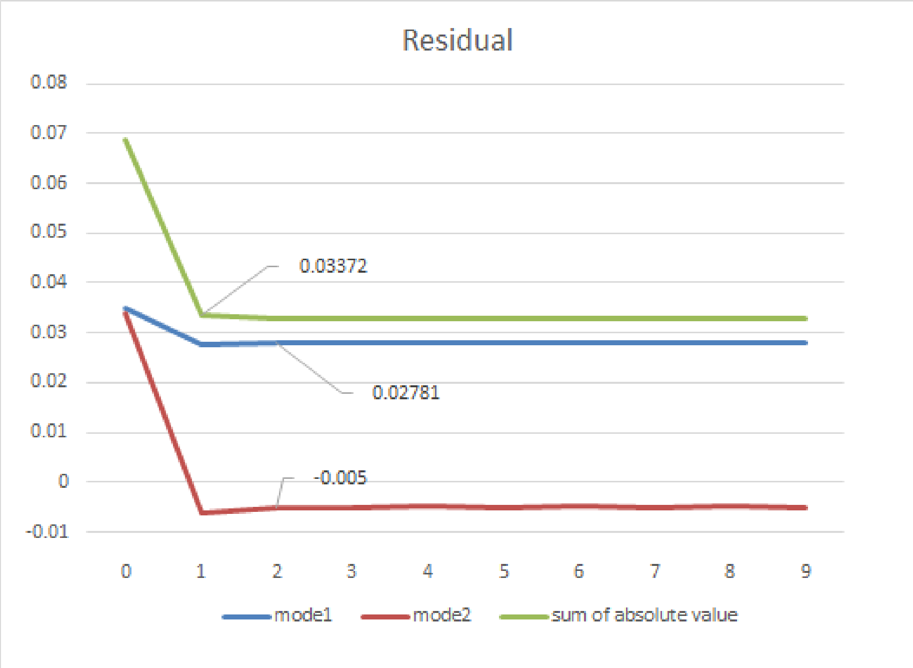


Figure 15, Residual development during updating3 of the rig

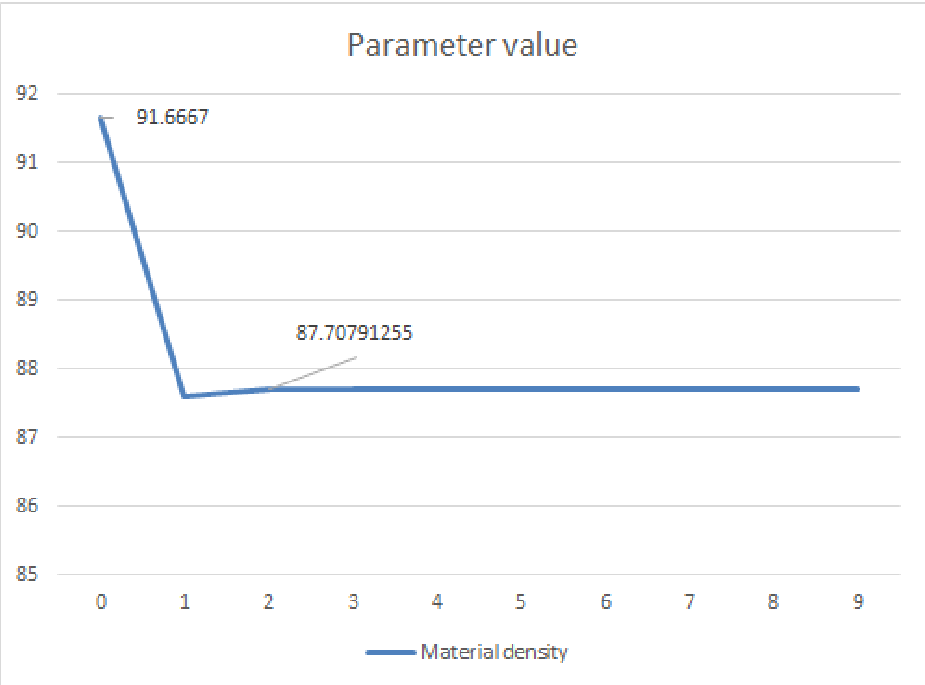


Figure 16, Density change during updating3 of the rig

Table 5, Result of updating3 of the rig

Mode number	Measured Frequency	Initial Analytical Frequency	Initial residual	Initial Error	Updated Analytical Frequency	Updated residual	Updated Error
1	0.9732Hz	0.9384Hz	0.0349Hz	3.59%	0.9729Hz	0.0278Hz	2.85%
2	2.0517Hz	2.0178Hz	0.0339Hz	1.65%	2.0518Hz	-0.0051Hz	-0.25%

5.4.4 Updating 4: Spring Stiffness and Point Mass

In this updating procedure two parameters have been evaluated, spring stiffness and point mass. When choosing two updating parameters while there are only two measured frequencies, the residual will become zero. This is because two equations with two unknowns is a determined system that will produce a unique solution. Therefore, regardless of which two parameters one would choose to combine in this procedure the residual would become zero. Still, it has carefully been decided which parameters to update, such that the true error source is corrected without too much change in the parameter value.

The reason for choosing spring stiffness and point mass as parameters is that the previous updating procedures shows that they are able to correct one mode each, respectively mode 2 and mode 1, i.e. the parameter's sensitivity is dominant in each mode. Even though Figure 10 shows that the spring stiffness have almost the same sensitivity in both modes, it should be kept in mind that the sensitivities can change as the parameters change.

Figure 17 shows the development of the residual during this procedure, in both modes and the sum of the absolute values. The results in the figure and Table 6 are shown for iteration number two because at this iteration the results are satisfactory, at further iterations the residual equals exactly to zero.

How the parameters change during the updating procedure is shown in Figure 18. The percentage change of each parameter is shown instead of the actual parameter value because the parameters have different units. As expected, the system becomes stiffer and lighter, i.e. a slight increase of spring stiffness, 2.36%, and a decrease in point mass, -7.47%. Both of these parameter changes has the same impact on the system: to increase the natural frequency in both modes. This is mostly to compensate for neglected stiffness contributions of both the longitudinal and vertical cables.

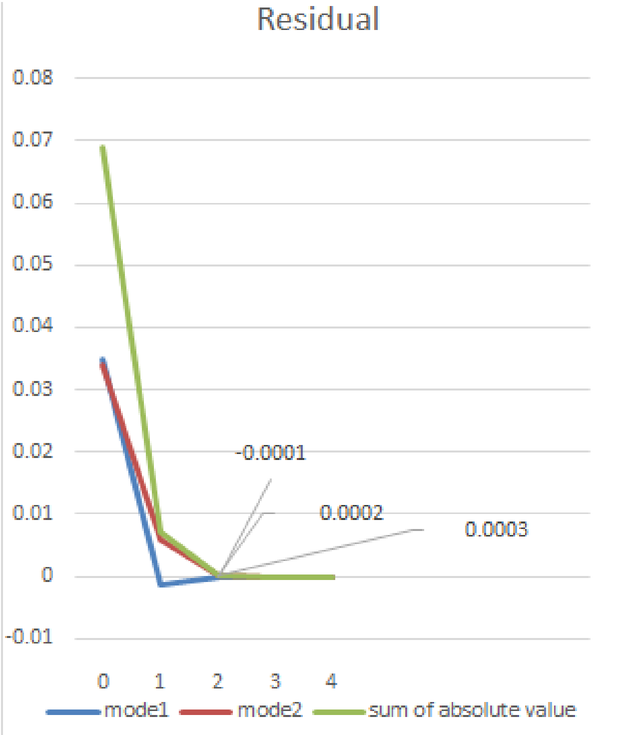


Figure 17, Residual development during

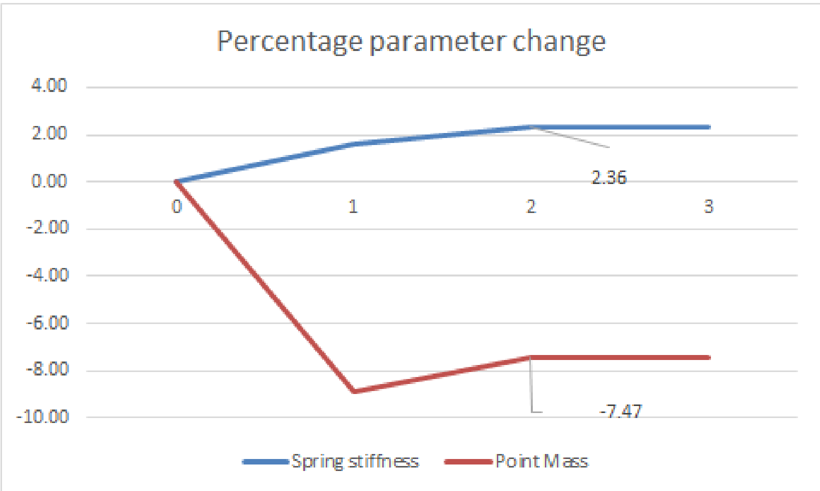


Figure 18, Parameter change during updating4 of the rig

Table 6, Result of updating4 of the rig

Mode number	Measured Frequency	Initial Analytical Frequency	Initial residual	Initial Error	Updated Analytical Frequency	Updated residual	Updated Error
1	0.9732Hz	0.9384Hz	0.0349Hz	3.59%	0.9733Hz	-0.0001Hz	-0.010%
2	2.0517Hz	2.0178Hz	0.0339Hz	1.65%	2.0519Hz	0.0002Hz	0.009%

5.5 Discussion

Every updating procedure converges, and the parameters changes as expected. This implies that the implemented algorithm is correct.

Initially, the analytical frequencies were lower than the measured frequencies. This complies with the conclusion that the analytical system stiffness in the transverse direction was too low. Hence the springs, which have stiffness in this direction, were chosen as parameters in the first updating. This updating improved the analytical model in both modes, but results in mode 1 was not satisfactory, which implies that there are other error sources that should be evaluated.

Further, both point mass and distributed mass were evaluated as parameters in updating number two and three respectively. Both parameters were reduced to increase the natural frequencies. Updating with distributed mass as parameter improved mode 2 very well, but mode 1 still had a large error. While updating with point mass as parameter the updated model became very close to representing the measured data with satisfactory accuracy in both modes. The only problem was the big change in the parameter, which implies that change in this parameter compensates for modelling error in other parameters, in addition to correcting its own error.

The results from the three first procedures confirms that more than one parameter holds any error. However, they illustrate the importance of choosing right updating parameters. Even though the residual reduces in all procedures, the user should always be critical to the parameter changes as the changes could be a consequence of other errors.

It was therefore conducted a fourth updating procedure with two parameters. This updating produced an analytical model that reproduces the measured data exactly, for the reasons mentioned previously. The most satisfactory about this procedure is that there were small changes in the parameters. It is expected that another constellation of updating parameters would have caused greater parameter changes, which would imply that the right parameters were chosen in updating number four.

5.6 Conclusion

The verification of the script is complete. Even though the script needed modifications during this case study it is verified and ready for further use.

The residual were reduced significantly in all four updating procedures. However, as discussed, this case study clearly illustrates that an updated model which reproduces the measured data correctly is not necessary a correct model. The user should therefore always post process the updated parameter values and evaluate whether these are logical or not. Because the goal is to obtain a correct model and not only a model that can reproduce the measurements.

After processing the data acquired during this case study, it is concluded that the simplification of neglecting the transverse stiffness contribution of both the longitudinal and vertical cables is the main cause of error. The parameter changes compensates mainly for this error, even though there are errors related to other parameters as well. Therefore, the user should always model the structure as detailed as possible in order to obtain a more correct model initially. This makes it easier to localize the errors in the model, hence the user can carefully choose the correct updating parameters.

If this rig structure is to be further analysed to, for instance, analysing higher modes or to determine the exact parameter values. It is recommended to model these cables as well, such that the true error in the model can be located and corrected.

6 Model Updating of the Bridge

Figure 19, Lerelva Bridge. Photo: Gunnstein Frøseth

6.1 Bridge Description

"Lerelva Bridge", shown in Figure 19 is a railway bridge in Trondheim, Norway. The bridge was built in 1919 and crosses the river, "Lerelva". This bridge is a part of "Størenbanen", which is the northernmost part of "Dovrebanen". In addition, "Rørosbanen" also connects to this railway. These two railways are the only two that connects Trondheim to Oslo and rest of southern Norway, which makes "Størenbanen" one of Norway's most busy and profitable railways (Aas, 2004).

The Norwegian railway system is distinguished by its aging railway and railway structures, such as this bridge. Demolishing and rebuilding a new bridge before the lifetime expires is too costly and not environmental friendly. Many of these railway bridges also have great historical and cultural value. Therefore, the owner, Jernbaneverket, is interested in determining the remaining service lives of these structures.

The bridge is a truss bridge made of riveted steel plates. The complex riveted connections makes it almost impossible to model this bridge with great precision. This is a challenge that has to be overcome when modelling the bridge.

The bridge is 5 meters wide, 25 meters long and has a small slope in the transversal direction. The bridge is very old, and therefore it shows signs of wear and tear, like rust and small damages. Over the years, there has been done improvements to the bridge when damage is detected. However, these improvements have not caused substantial changes to the structure, compared to the drawings. Hence, the original drawings may still be used as the bases of the structure.

The bridge is modelled in Abaqus by Bartosz Siedziako, an PhD-Candidate at the department of structural engineering at NTNU. The bridge is modelled as simply supported, and shell elements are used to make the sections. The x, y and z-axis are respectively in the transverse, vertical and longitudinal directions of the bridge.

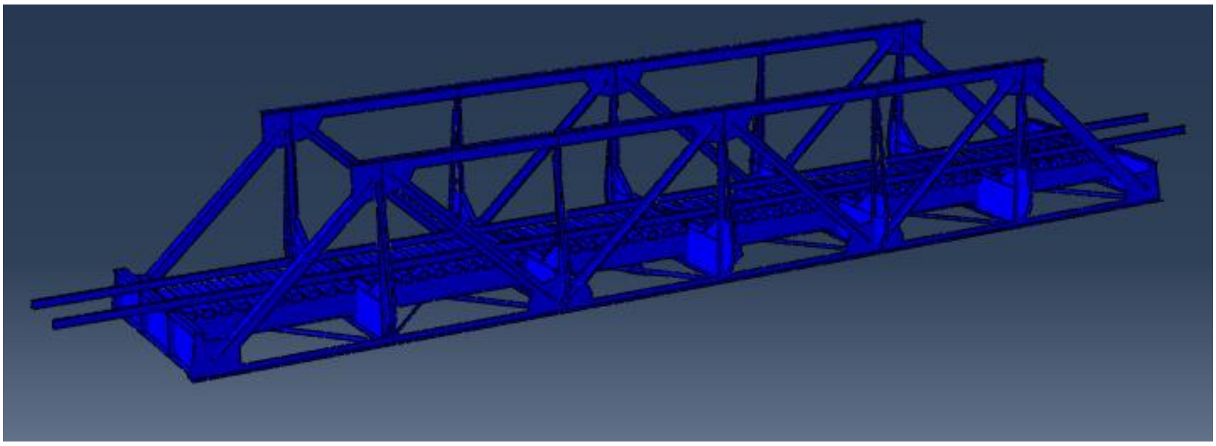


Figure 20, FE model of the bridge

6.2 The Structural System of the Bridge

As mentioned earlier, in order to carry out a successful updating of the model, choosing the correct parameters to be updated is essential. In this study, it has been decided to go forward by using a combination of engineering judgement and sensitivity analysis, to localize the error and update the model in a reasonable way. To be able to do that, there is a need to understand the structure and its load bearing system. This may help identify the components of the structure involved in the different vibrational modes, and hence lead to localization of the errors in the model.

The rails (cc 1500 mm) directly carry the vertical load from the trains, and then transfer it through transversal wooden sleepers, which lie upon two longitudinal steel beams (cc 1800 mm). These beams lie beneath the rails, but with some gap in between, which gives rise to a little bending action in the sleepers, but mostly shear. The longitudinal beams carry the load through bending and shear to the cross beams as two point-loads.

The ends of the cross beams are connected to the vertical parts of the semi-through truss, which is the main loadbearing component of the bridge. The truss then distributes the load to the supports at each end. The truss is made up of elements arranged to make triangular shapes, and transmit the stresses in the form of compression or tension only. It consist of vertical, diagonal and longitudinal (top and bottom cord) parts.

The height of the truss hinders the global vertical bending mode of the bridge, as the top and bottom cords work as a pair, each exposed to compression or tension. The vertical parts in the truss, which are also made up as a truss-system, prevent global lateral bending of the truss, and is the main stiffness contributor in related modes. The twisting modes are greatly influenced by the stiffness of the top cord and as well as the vertical parts of the truss.

An upper and bottom bracing system helps achieve in-plane stiffness of the bridge. The upper plan bracing system is attached to the top of the longitudinal beams, with the same connection that connects the cross and longitudinal beams, forming a plan truss. The transversal parts of the upper bracing are top cords of vertical intermediate bracings. The upper bracing also prevents lateral buckling of the longitudinal beams.

At the bottom, there is another plan bracing attached beneath the transversal cross beams and bottom cords of the semi-through truss, forming an in-plane x-shape. The plan bracings contribute mainly to the stiffness in global transversal direction, and therefore resist horizontal forces on the bridge, such as wind loads. It will therefore play an important role in the modes involving in-plane motion of the bridge.

6.3 Response Variables

The response variables used for the railway bridge study are the natural frequencies of the structure. The reason is primarily due to the simplicity in taking measurements of the structural response. In addition, natural frequencies as the response variable gives a greater variety of updating parameters to choose from, such as the option of mass.

PhD-Candidate Gunnstein Frøseth, at the department of structural engineering at NTNU, took the measurements of the bridge. A wireless measuring system with nine accelerometers were placed at nine different points, referred to as MAC-points by the modeller. The placements of these MAC-points are shown in Figure 21, marked as red dots. Data-driven Stochastic Subspace Identification, also referred to as SSI, was used to find the mode shapes and natural frequencies. It was extracted five mode shapes and natural frequencies.

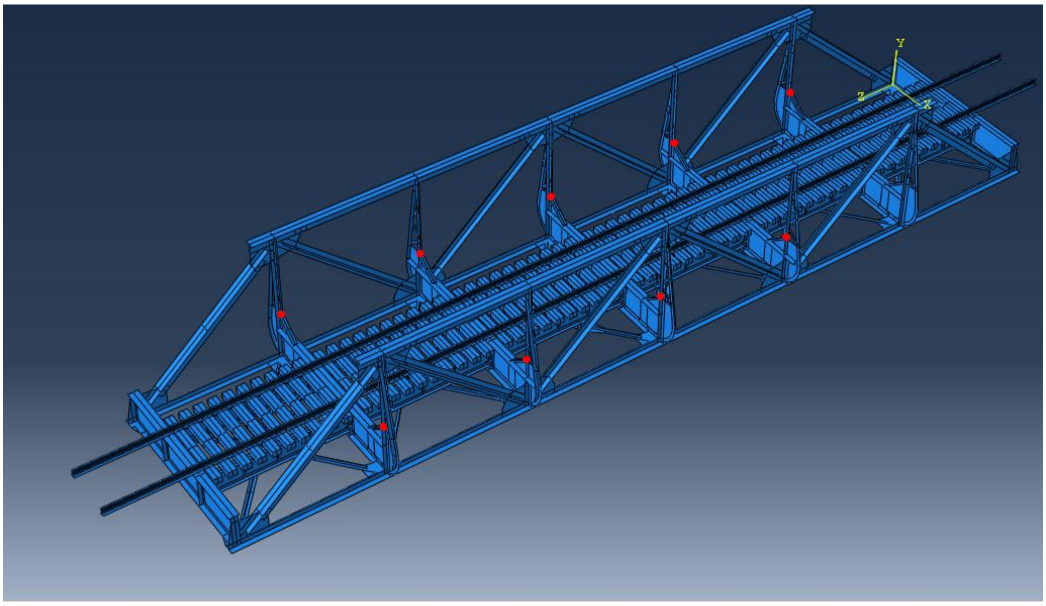


Figure 21, Placement of MAC-points

6.3.1 Modal Paring

For modal paring the modal assurance criterion (MAC) is used. This criterion is described under the chapter, 2.4.2 Mode Pairing. From the measurements taken, five modes with their corresponding mode shape vectors and frequencies were identified. These measured modes had to be paired to the analytical modes, in order to correctly compare each measured response with its respective analytical response.

Figure 22 shows the initial MAC-values calculated with the measured modes, m1 to m5, and compared to the ten first analytical modes, a1 to a10. It can be seen that every measured mode has a good match with an analytical mode.

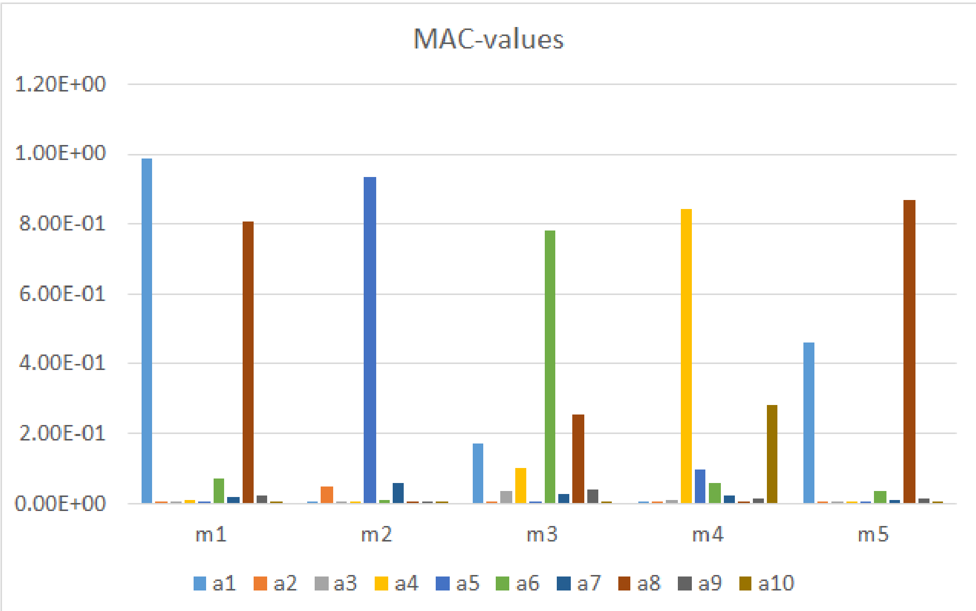


Figure 22, MAC-values

The figure shows that measured mode 1 (m1), has an almost perfect match with analytical mode 1 (a1), but also a very good match with analytical mode 8 (a8). The respective MAC-values are 0.984897832 and 0.80626667. The reason for this is that modes, a1 and a8, are quite similar in shape, and there are not enough measuring points on the structure to distinguish between them. Therefore, the mode shape vectors are not descriptive enough, to mathematically decide which one of these two analytical modes should be paired with the measured mode, m1. This is described under the chapter 2.4.2 Mode Pairing.

Even though the paring of m1 with a1 has the highest MAC value, this is further evaluated to ensure a correct pairing. To decide which of the analytical modes m1 should be paired with, the frequencies for m1, a1 and a8 are found, and are shown in Table 7.

Table 7, initial residual and error in the bridge

Mode number	Measured Frequency in Hz	Analytical Frequency in Hz	MAC value	Residual in Hz	Error in %
1	5.25305	5.7624	0.985	-0.50935	-9.7%
2	9.4757	9.6545	0.935	-0.1788	-1.9%
3	9.7178	9.8453	0.781	-0.12754	-1.3%
4	9.8444	9.3337	0.844	0.51069	5.2%
5	11.1790	12.216	0.869	-1.0371	-9.3%

The large difference between the frequencies for m1 and m8, indicates that these should not be paired. It can also be seen in Figure 22, that m5 has a high MAC value with a8, but a much lower with a1; 0.868703373 and 0.459453226 respectively. This suggests that m5 pairs with a8, and therefore a8 cannot be paired with m1. Hence, m1 is paired with a1. Rest of the measured modes have a very clear pairing, as can be seen in Figure 22.

6.3.2 Mode Shapes

- The first identified mode is the first lateral bending mode, this is shown in Figure 24. As shown, both the bridge deck as well as the trusses bends.
- The second identified mode is shown in Figure 23. This is the first vertical bending mode.
- The third identified mode is third bending mode of the side walls, i.e. the trusses. the trusses vibrated parallel, which gives rise to inertia effects that causes vibration in the bridge deck as well. This mode is shown in Figure 25.
- The fourth identified mode is the fourth bending mode of the trusses. This mode is shown in Figure 26. It illustrates how the trusses vibrates in opposite direction. This

causes equal and inertia effects in opposite directions, which means no vibration in the bridge deck, hence this mode causes no reaction forces in the supports.

- The fifth identified mode is a roll mode, i.e. bending of the bridge deck in one transverse direction, and bending of the trusses in the other transverse direction. This is shown in Figure 27.

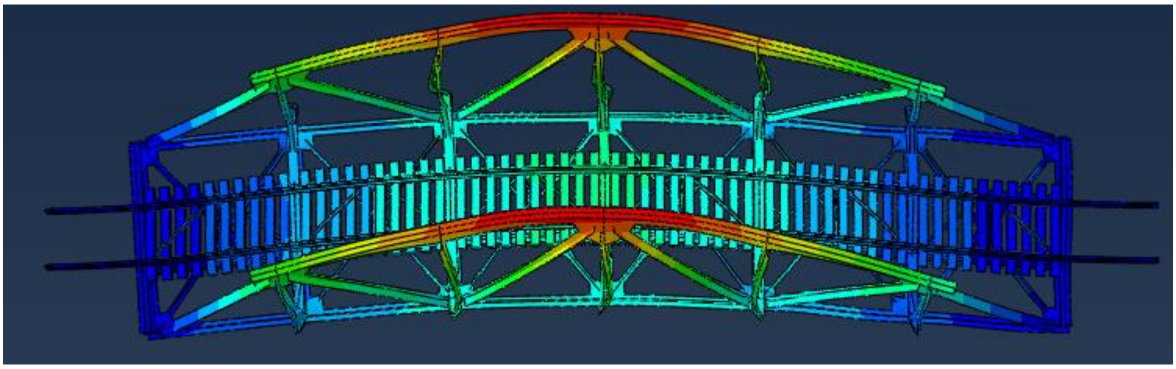


Figure 24, First lateral bending mode, Mode 1, 5.25305Hz

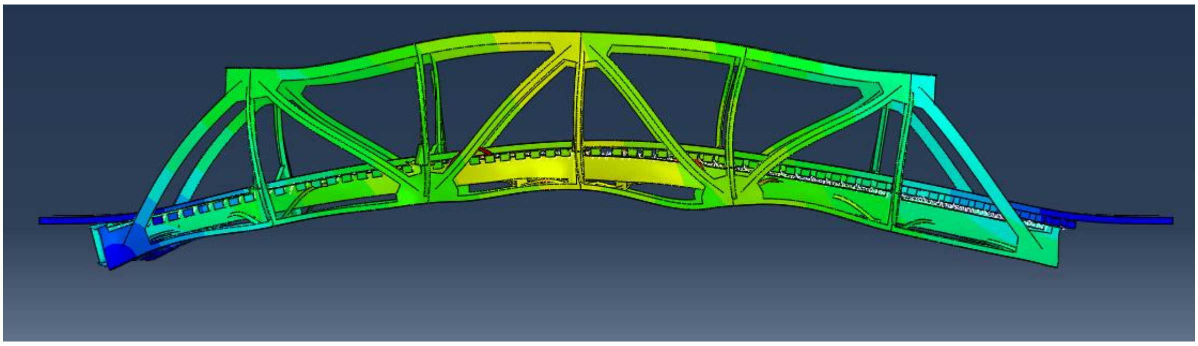


Figure 23, First vertical bending mode, Mode 2, 9.4757Hz

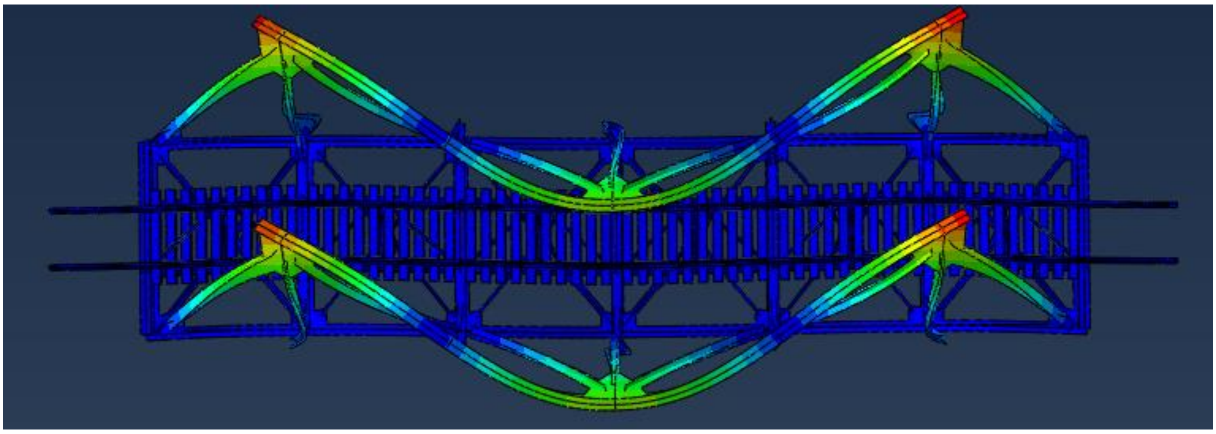


Figure 25, Third truss bending mode, Mode 3, 9.7178Hz

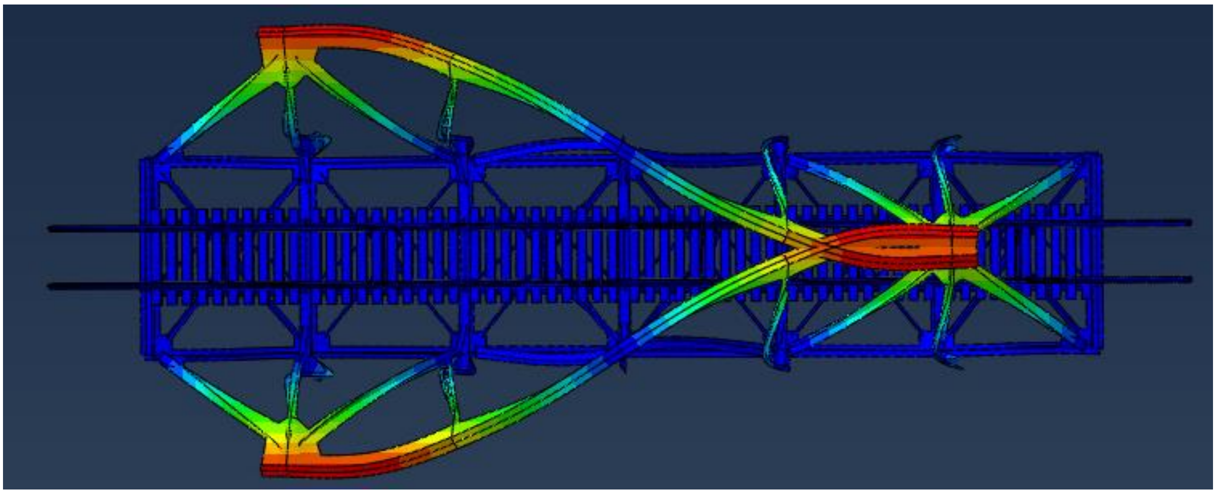


Figure 26, Fourth bending mode of truss, Mode 4, 9.8444Hz

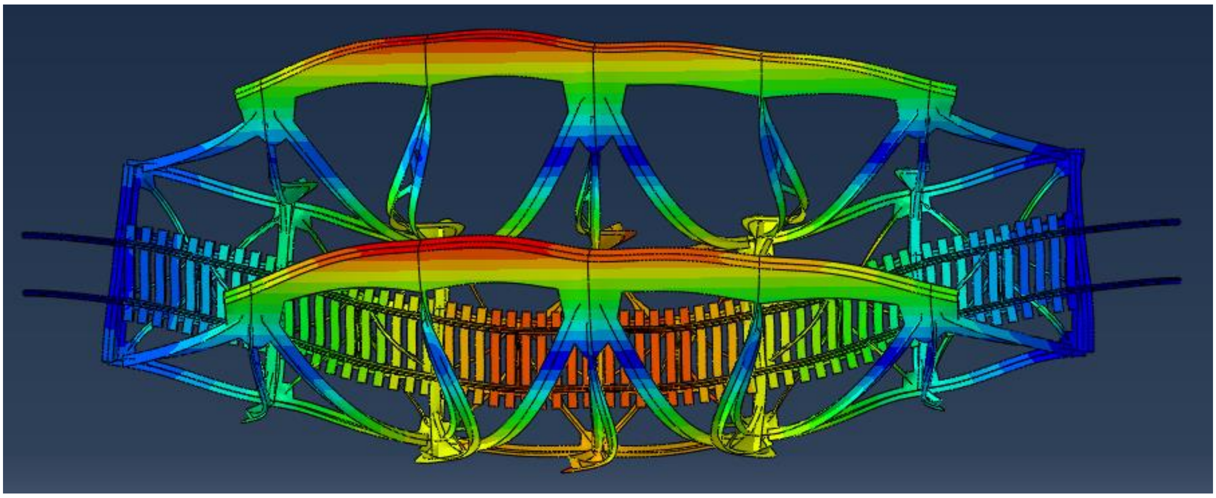


Figure 27, Roll mode, Mode 5, 11.1790Hz

6.4 Choosing Parameters

6.4.1 Introduction

The choice of updating parameters for the bridge, is mainly based on the theory discussed under *"2.6 Choice* of Parameters*"*. As discussed, the updated model should be able to represent, not only the modes within the range of measured response, but also outside of it. Therefore, while updating the bridge it is wise to have an overdetermined system with a greater number of measurements than updating parameters. In this case, it is recommended to use three updating parameters at the most, with the five measured modes.

In addition, to be able to compare sensitivities of different types of parameters, such as E-modulus and density, the sensitivities are normalized. Otherwise, the sensitivities have different units, and hence would not be comparable.

The bridge is divided into smaller parts, each of which is assigned an individual material, to be analysed further before the updating. This is done to reduce the total number of parameters in the model. In addition, this will restrict the same structural parts to change equally during the updating. If for instance every cross beam is evaluated individually, and there are different changes in all of them, it would be illogical, as all of them are actually equal. To start with, the division was based on the main parts of the structural system; truss, upper plan bracing, bottom plan bracing, intermediate transversal bracing, longitudinal beams and cross beams. Later, it was decided to subdivide the truss into smaller regions; diagonals, upper cord, bottom cord and vertical parts.

Further, localisation of errors and choosing the parameters representing these is an important factor for a successful updating. Therefore a complete discussion on possible sources of errors in this type of a railway bridge, based on experience and engineering judgement is done. For this specific case, the pictures taken by Gunnstein Frøseth while inspecting the bridge are also used to locate the possible sources of errors in the bridge by inspection. Here it is an overview of what may be typical causes of error in this structure.

6.4.2 Error Localization

Connections

The stiffness of each joint may vary from the designed value, as it is difficult to achieve the exact stiffness in a real structure. The bridge has many such riveted joints, which are difficult to construct and model exactly, and can cause large discrepancies. One of the connections is shown in Figure 28. This figure illustrates the complexity of the connections.

The model assumes full stiffness in all the connections, as if they are welded, which is not the reality. This might therefore be a significant source of error in the response of the model.

Figure 28, Top and end connection in truss. Photo: Gunnstein Frøseth

The difference in stiffness will cause errors in the overall stiffness of the model, and hence the dynamic properties.

Material Properties

Material properties, which are given to the engineers by suppliers, might deviate from the real values and could be a source of error in the model. Even if the discrepancies in the material properties such as density and E-modulus may be relatively small, it might have a large effect on the dynamics of the structure, as it is spread throughout the whole model.

Density of wood is obviously difficult to estimate as it depends on various factors, such as species, humidity, impregnation and age. The wooden sleepers in this bridge are very old and estimating their exact density is difficult. Therefore, it should be taken into account that there might be an error in their density used in the model.

Supports

The bridge is modelled simply supported, with four supports; two at each end. The two at one end, are modelled as boundary conditions, with the constraint of no deformation, i.e. pinned. The other two at the other end, have translational degree of freedom in the longitudinal direction, i.e. a roller support. There are no constraints applied related to rotation in the supports.

Obviously, the bridge is not supported exactly like this in reality. This is an idealization error made by the modeller of the bridge, and has an impact on the responses of the structure. These boundary conditions underestimate longitudinal stiffness in the roller support and rotational stiffness in all four supports. In addition, they overestimate the stiffness in every other direction in each support. It is not easy to say whether these errors together, increase or decreases the global stiffness of the structure as a whole, or whether it is significant or not.

Wear and Tear

As the bridge was built in 1919, most probably there will be various parts that have been worn out in all these years, because of, for example rust and fatigue. After inspecting the pictures of different parts of the bridge, it can be seen that there is a lot of such damage there. As seen in Figure 29, the bottom bracing is definitely worn out in the connection, which would affect the overall stiffness in the transverse direction.

Figure 29, damage in connection of bottom bracing. Photo: Gunnstein Frøseth

This is the case for several other structural parts of the structure. For instance the crossbeams, which is shown in Figure 30. Such wearing will make the structure weaker in those areas, and again effect the dynamic response of the whole structure. The model does not account for those reductions in stiffness as it is modelled based on the original drawings of the bridge. Hence, this may be a cause of difference between the measured and the analytical response of the structure.

Figure 30, damage in top flange of crossbeams. Photo: Gunnstein Frøseth

Element Sections

These days steal sections are in most cases rolled, but in 1919, this was not the practice. The sections were made of steel plates connected by connection plates and rivets. The FE model of this bridge does not include these connecting plates or the rivets; hence, there is some mass missing in the model. The connections between the plates are modelled as if they were welded together, as a rolled section, unlike the reality. This causes a higher stiffness in the connections of the plates in the FE model. This simplification effects both, global stiffness and mass properties of the structure, hence the dynamic response. Figure 28, 29 and 30, show how the sections are made by plates riveted together.

Extra Mass

When making the model, there will most probably be some mass, which has not been included. The reason may be the choice of neglecting it by the modeller, as it is assumed insignificant, or simply that the modeller does not know that it exists. Mass of the structure obviously plays a central role in the dynamic properties of the structure, so there will always be some influence on the response if some of the mass is not included. However, the extent to which it affects the response can vary based on how large it is.

On this bridge, after inspection, it has come to knowledge that there are some parts on the bridge slab, which are not included in the model. For instance, another set of rails, a wooden inspection deck on each side of the bridge deck and the fact that some of the sleepers are longer in reality than what has been modelled.

Figure 31 shows the inspection deck, another set of rails, and the steel joints connecting the rails to the sleepers. While Figure 32 shows that one sleeper is longer than what is modelled, this regards several sleepers as they carry the inspection deck. The simplifications of neglecting these masses causes a lighter model than reality, hence higher frequencies.

Figure 31, extra set of rails and inspection deck. Photo: Gunnstein

Figure 32, longer sleepers. Photo: Gunnstein Frøseth

6.5 E-Modulus as Parameter

6.5.1 Sensitivity Analysis

As it is desired to update only the errors in the model, the updating is not done throughout the whole structure, but at specific regions. As mentioned, the structure is divided into groups with individual materials, hence every group has individual E-moduli. Some of these are then chosen for updating based on a sensitivity analysis. The normalized sensitivities of these nine regions are shown in Figure 33.

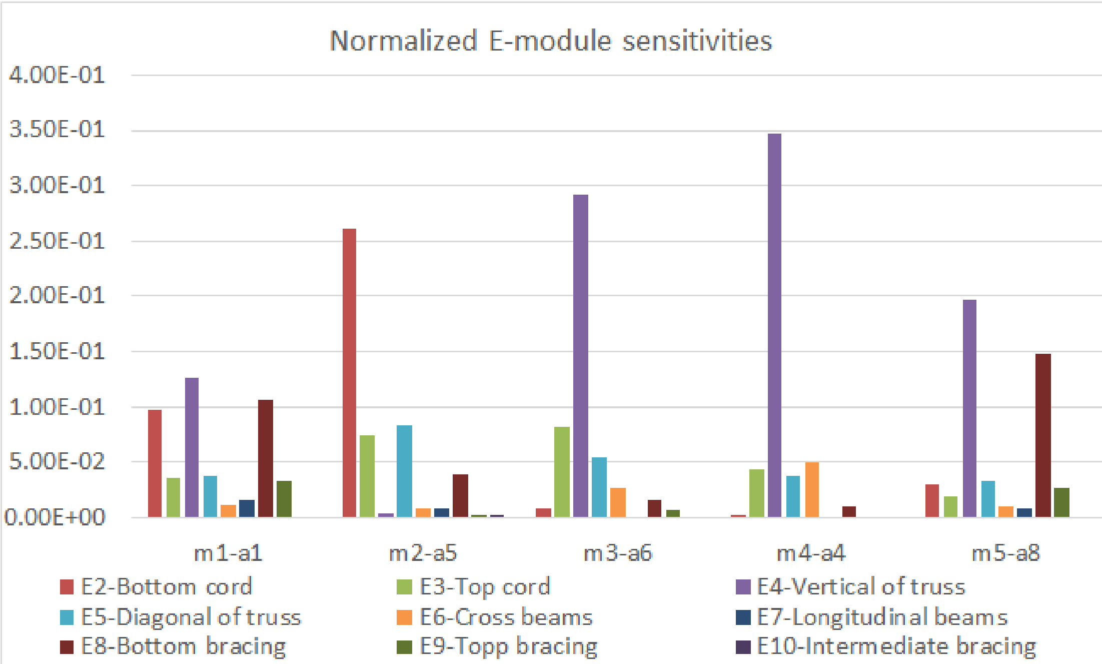


Figure 33, Normalized E-moduli sensitivities of main structural parts with subdivided truss

It is clear that the truss plays an important role in all the mode shapes, but a closer look reveals that the top cord and diagonals of the truss are only a little sensitive in every mode, while the bottom cord and vertical parts are very sensitive in most of the modes.

6.5.2 Updating 1

In the first updating of the model, it has been decided to use only E-modulus of the components as the updating parameters. An updating of E-modulus can compensate for most of the possible errors throughout the structure, hence represent the measured data well, and gives therefore a good starting point. In addition, a small change in E-modulus could give a relatively large change in dynamic properties of the model, as it applies throughout the whole structure, as discussed in the chapter 2.6 Choice of Parameters. E-modulus changes, will also be a good choice to represent stiffness differences generally in the model, such as joints. Which is very relevant for such a riveted steel bridge.

Parameter choice

In order to choose which structural parts that should be updated, the residual were evaluated in conjunction with the sensitivities of the parameters in each mode, such that the model could be changed to minimize the residual in the measured modes, i.e. represent the measurements accurately.

The 4th mode stands out from the rest, as it has a positive residual. Unlike the other modes, this mode requires the stiffness to be increased to achieve higher frequency. To achieve that in this mode only, a parameter that has highest sensitivity in this mode should be chosen for updating. E6, cross beams, if increased will increase the frequency mostly in the fourth mode. The consequently small increase of frequencies in the other modes can be compensated by a stiffness-reduction of parameters that are more sensitive in the other modes than the 4th mode.

Obviously, E4 is the most sensitive parameter in mode 4. Without large increase in the parameter value, E4 could correct the frequency error in mode 4. However, it is the most sensitive parameter in mode 3 as well, hence the frequency in this mode would increase as E4 increases. Consequently, a very large reduction in another parameter would be necessary to compensate for the frequency increase in mode 3. E6 on the other hand, if E4 is not relevant, is the most sensitive parameter in mode 4, but only the third most sensitive in the third mode, this is the reason for choosing E6 and not E4 in this updating.

Every mode, but mode 4, require a decrease in frequency to match the measured frequencies. E8, bottom bracing, has a larger sensitivity than W6 in all the other modes, except mode 3 and 4. A decrease in this parameter will decrease all frequencies, mostly in mode 5 and 1 (the lateral bending modes), which is very good considering the large residual of these modes.

The only mode that might not get the desired decrease in frequency based on the expected changes of these parameters is mode 3. In order to reduce the frequency in this mode without critically influencing the frequency of mode 4, the best option seems to be the top cord, E3. Even though mode 4 is sensitive to a change in E3, it is expected that an even larger increase of E6 will compensate for the frequency reduction in mode 4 caused by a reduction in E3.

Bounds

The bounds were applied the same to all three parameters. The updating procedure were conducted in three parts, each part with different bounds. The upper and lower bounds in all three parts of the procedure are shown in Table 8.

Table 8, upper and lower bound for updating 1

	Upper bound	Lower bound
Part 1	315 000 N/mm^2	105 000 N/mm^2
Part 2	367 500 N/mm^2	52 500 N/mm^2
Part 3	Infinitely	0

Results

The results of the first updating are shown in Table 9, while Figure 34 and Figure 35 show how the residuals and the total residual changes throughout the procedure respectively.

Table 9, Results from updating 1

Mode	Measured frequency	Initial residual	Initial error	Initial MAC	Updated residual	Updated error	Updated MAC
1	5.25305	-0.50935	-9.7%	0.985	-0.06025	-1.1%	0.989
2	9.4757	-0.1788	-1.9%	0.935	0.0171	0.2%	0.944
3	9.7178	-0.12754	-1.3%	0.781	-0.08474	-0.9%	0.821
4	9.8444	0.51069	5.2%	0.844	0.05239	0.5%	0.897
5	11.1790	-1.0371	-9.3%	0.869	0.1429	1.3%	0.948

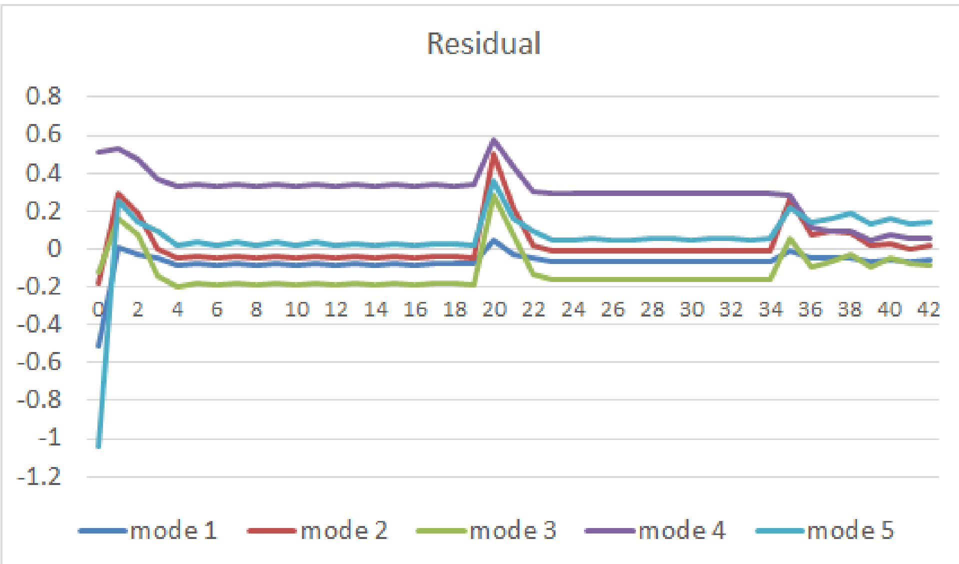


Figure 34, residual of each mode in updating 1

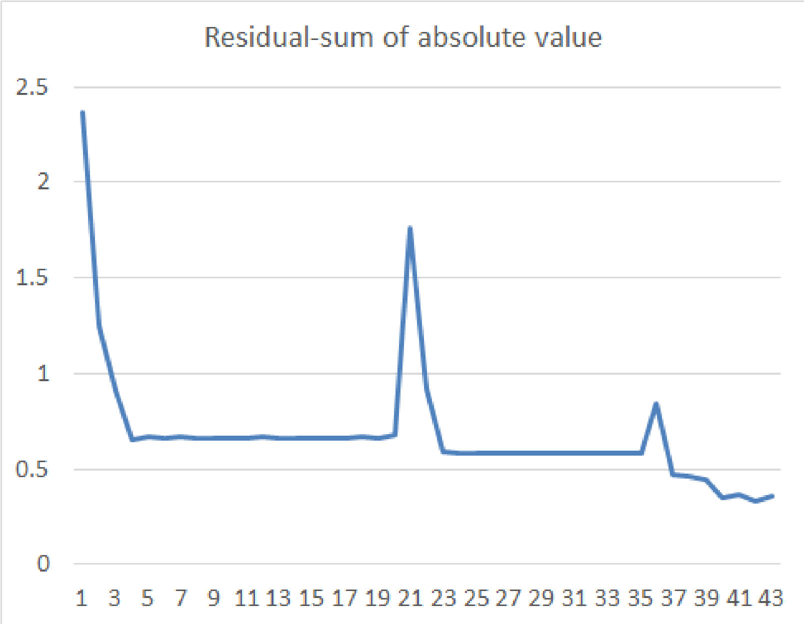


Figure 35, sum of absolute value of residual updating 1

Interpretations of Results

From the Figure 34 it is clear that the solution converges at the fourth iteration, with unsatisfactory high residual in mode 3 and 4, this is due to the bounds applied. E6 reaches its upper bound immediately as it is shown in Figure 36, and limits the rest of the procedure to be updated by only two parameters, which have great difficulties of correcting five modes. It was expected a large increase in this parameter, but it was also expected a better correction of error with such a large change. However, it was of interest to see whether this combination of parameters could successfully change the model to represent the measured data. The bound were therefore increased, and the procedure was continued.

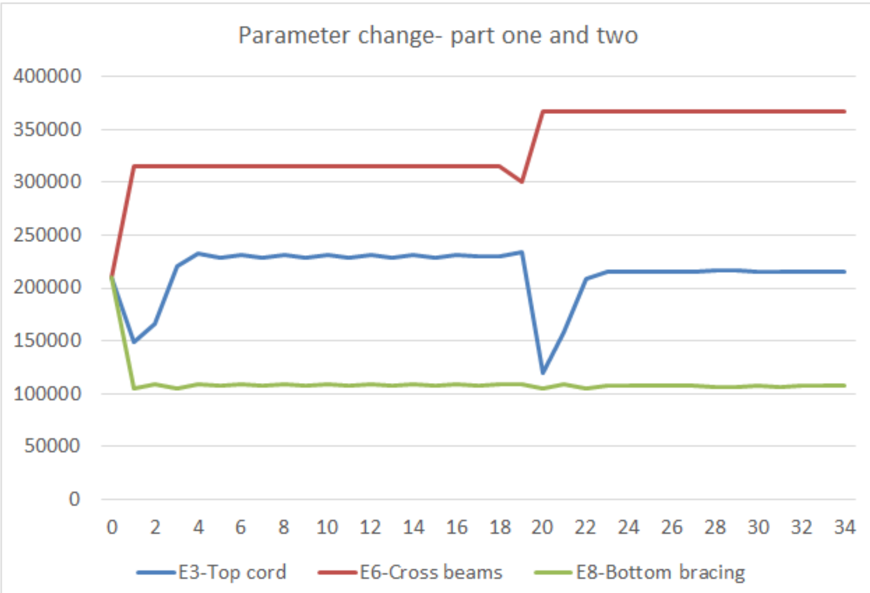


Figure 36, parameter change updating 1

Figure 36 shows that E6 reaches the upper bound quickly this time also. This improves the total error somewhat, as shown in Figure 35, but still not enough. Therefore the bound were repealed, such that the procedure could properly converge without any constraint. Obviously the residual decreased as the parameters changed greatly, Figure 35 illustrate this with the sum of the absolute values of the residuals.

The change in parameters throughout part 1 and 2 of the procedure is shown in Figure 36, and the parameter values after each part of the procedure in Table 10. Part 3 is not shown in the figure, as the change in E4 is to high too illustrate.

Table 10, parameter change updating 1

Parameter	Initial value (N/mm^2)	Part 1		Part 2		Part 3	
		Updated value (N/mm^2)	Change	Updated value (N/mm^2)	Change	Updated value (N/mm^2)	Change
E3 Top cord	210 000	229 720	9.4%	215 264	2.5%	141 151	-32.8%
E6 Cross beams	210 000	315 000 (bound)	50%	367 500 (bound)	75%	5 372 545	______
E8 Bottom bracing	210 000	108 060	-48.5%	107 001	-49%	100 753	-52%

As expected, E6 increases to correct the error in mode 4, while E8 decreases to correct the error in mode 1 and 5. The decrease in E3 is to compensate for the increase of frequency in this mode, caused by the increase in E6. Even though it was expected that the parameters would change in these directions, the magnitude of the change is greater than expected. In retrospect one could say that the sensitivity of E6 in mode 4 is too low to correct the corresponding error, but also as the parameters changes, the sensitivity of this parameter in this mode reduces, which leads to a greater increase in E6 to correct the error, than what could have been expected.

E8 was decreased significantly to improve mode 1 and 5, with a total reduction of about 52% after part three, however the changes in this part of the procedure are polluted by the large change in E6. A closer look at Figure 34 reveals that mode 1 and 5, the two modes that are mainly sensitive to this parameter, have lowest error after part one of the procedure. It is therefore more relevant to evaluate the parameter change after this part of the procedure, i.e. -49%.

The large stiffness reduction in the bottom bracing is obviously not only a correction of stiffness in this part, but also a compensation of other errors. This suggests that either the bridge is modelled too stiff in transversal direction, or the parts moving in transversal modes are too light. The latter may be due to the simplifications done by for example neglecting some of the mass on the bridge deck. Another reason might be the low initial estimation of wood density, as discussed earlier. The stiffness reduction may also compensate for too stiff modelling of connections, as a simplification.

As expected, this procedure was able to correct every mode in the model satisfactory, such that the model represent the measurements with high precision, however the changes in the parameters are too large. This suggest that the real modelling errors were not corrected, but that parameters were changed to compensate for the real errors. The updated model is then able represent the measured data with good accuracy, but the responses outside tis range will probably not be represented correctly due to the large parameter changes, hence the lack of physical relevance. Therefore, the user should always choose parameters with relatively high sensitivity.

The goal of the user should be to correct parameters that are the real sources of error, and not parameters that can, by changing them, make a model that is able to successfully represent only the measurements, and no other responses.

6.5.3 Updating 2

This updating procedure is somewhat a continuance of the previous procedure, still, only E-modulus of the structural parts is evaluated when choosing updating parameters, for the same reasons as before. However, in this analysation parameters have been chosen based on the lessons learned from the previous updating. The most sensitive parameters have been chosen for updating, since this gives an updated model with minimum change in the parameter values, hence a physically correct model.

Chosen Parameters

As Figure 33 shows, the most sensitive parameter is E4, the vertical parts of the truss. This parameter is the most sensitive parameter in mode 4,3,5 and 1, in that order, but with almost no sensitivity in mode 2. In this mode, the most sensitive parameter is E2, the bottom cord. This parameter also have some sensitivity in mode 1. These two parameters together have high sensitivity in all modes, which is a requirement to change the frequencies in every mode with small parameter changes. When the third parameter was chosen, the main factor was the residual and error, as mode 5 had the highest residual it was desirable to choose a parameter with high sensitivity in this mode, thus E8, bottom bracing , was chosen as the third parameter.

How the parameters change during the procedure depends on the residual and the parameters sensitivity. Because of high negative residual in mode 5, it is expected that the parameter with high sensitivity in this mode would change significantly to correct this error. The residual is negative, i.e. the system is stiffer in this mode than what the measurements suggest, and therefore the E-modulus of the bottom bracing was expected to be reduced, as in the previous updating procedure. A reduction in this parameter would also give rise to reduction of frequency in mode 1 as well because of high sensitivity in this mode, and some reduction in mode 2.

Further mode 4 has a positive residual, i.e. the model is too soft in this mode, and hence an increase of stiffness is required by the updating to correct this error. E4 is the only parameter with high sensitivity in this mode therefore an increase of the E-module in the vertical parts of the truss is expected. An increase of this parameter would indeed correct the frequency error in mode 4. However, as the parameters increases, other modes would be affected as well, i.e. frequencies in mode 1, 3 and 5 will increase as this parameter increases. This will cause a larger decrease in E8 to decrease the frequencies in mode 1 and 5 as a compensation, but since E8 is not sensitive in mode 3, it will not decrease the frequency in this parameter. It is therefore expected higher frequency in mode 3 than initially, while all the other modes should be improved.

E2 should change too much to correct the error in mode 2, initially it looks like the parameter value would decrease to correct the negative residual, but a large decrease in E8 may cause such frequency reduction in mode 2 that E2 may be increased to compensate for this. Regardless, the changes should be minimal.

Bounds

As shown in the previous updating procedure, E8 decreases not only to represent stiffness error in E-modulus of the bottom bracing, but also other errors. Therefore, the lower bound is set very low, such that this parameter should be able to decrease and represent the other errors. The upper bound is also set high, but the physical relevance of the parameters can be evaluated after the procedure.

Table 11, Upper and lower bound for updating 2

Upper bound	Lower bound
315 000 N/mm^2	70 000 N/mm^2

Results

The updating procedure converged at the fourth iteration. The change in residual for each mode is shown in Figure 38. It is clear how the residual in every mode, except in mode 3, is improved by this updating procedure, especially in mode 1 and 5, which had large residuals initially. Even though mode 4 was improved greatly, this mode and mode 3 hold the largest errors after the procedure.

Figure 37 shows the sum of the absolute value of the residuals. Initially the total residual was 2.37Hz, but after the updating it was reduced to 0.58378Hz, which is a reduction of over 75%. This is a significant improvement even though the error in mode 3 is increased.

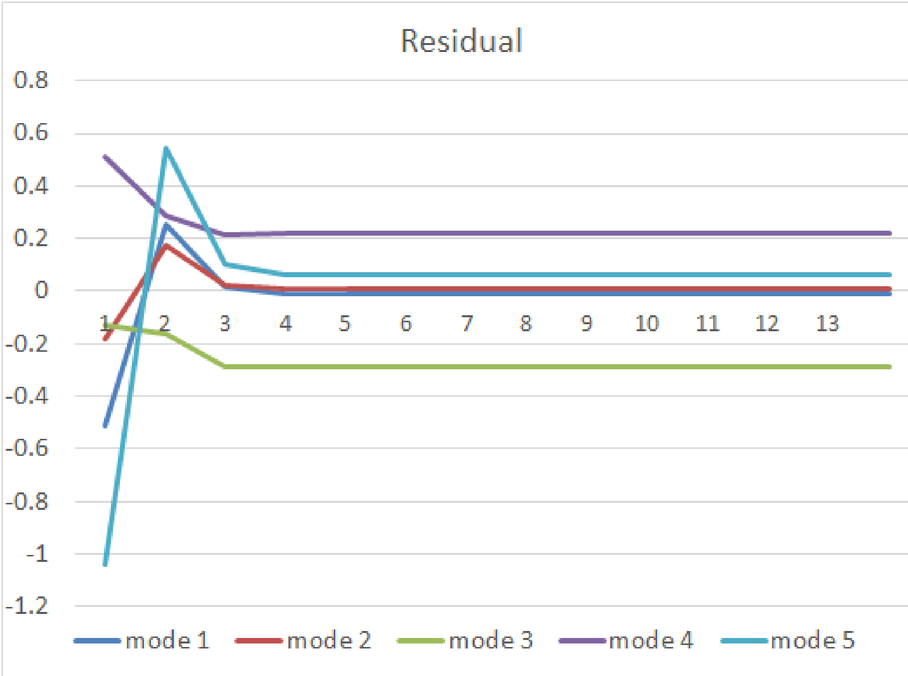


Figure 38, residual of each mode in updating 2

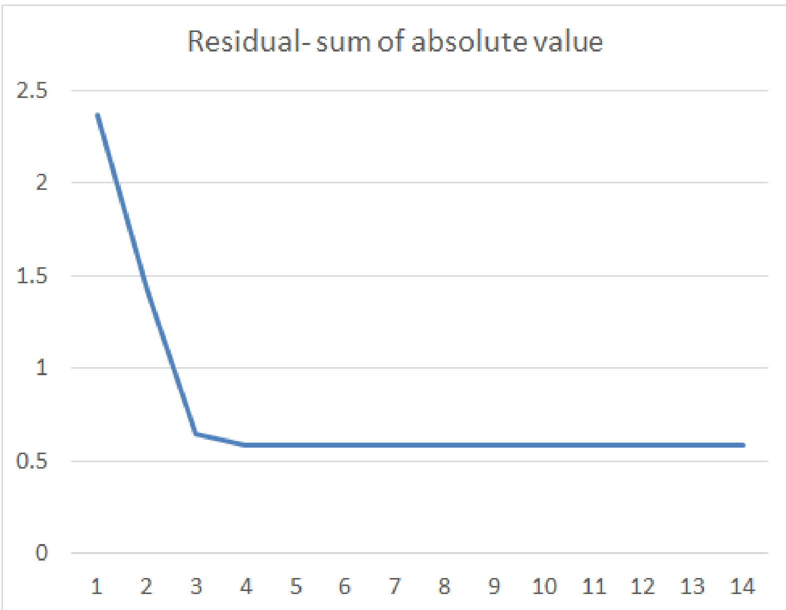


Figure 37, sum of absolute value of residual updating 2

Table 12, Results from updating 2

Mode	Measured frequency	Initial residual	Initial error	Initial MAC	Updated residual	Updated error	Updated MAC
1	5.25305	-0.50935	-9.7%	0.985	-0.00705	-0.1%	0.982
2	9.4757	-0.1788	-1.9%	0.935	0.0112	0.1%	0.922
3	9.7178	-0.12754	-1.3%	0.781	-0.28524	-2.9%	0.84
4	9.8444	0.51069	5.2%	0.844	0.21839	2.2%	0.85
5	11.1790	-1.0371	-9.3%	0.869	0.0619	0.5%	0.977

Interpretations of Results

The change in parameters throughout the procedure is shown in Figure 39, and the final values in Table 13. As expected, E8 was decreased significantly to improve mode 1 and 5, with a total reduction of about 55%. The large stiffness reduction in E8 is obviously not only a correction of stiffness in this part, but also a compensation of other errors as discussed under 6.5.2 Updating 1. However, the decrease in E8 is larger in this procedure than the previous, due to the compensation needed caused by an increase of E4.

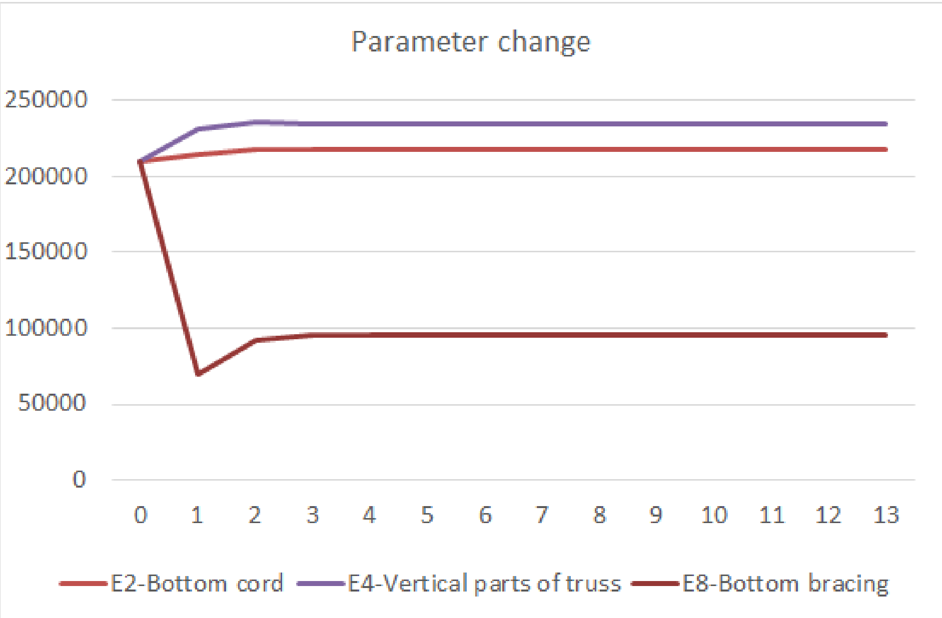


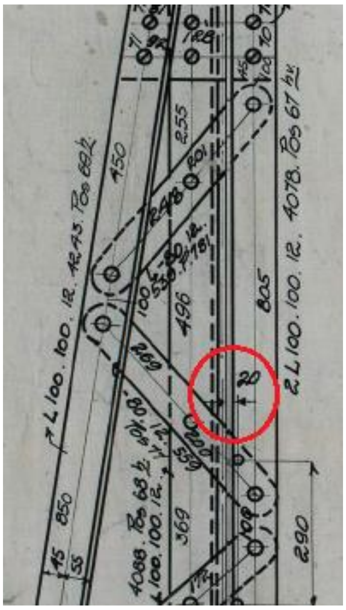
Figure 39, parameter change updating 2

Table 13 also shows that E4 is increased by almost 12%, as discussed previously this was expected. As predicted, an increase of this parameter caused an increase of negative residual in mode 3 as it is shown in Figure 38, this has happened because out of the three chosen parameters only this parameter is sensitive in mode 3 and in mode 4, none of the other parameters are sensitive in any of these two modes. Due to this fact and the fact that the initial residual in these two modes have opposite signs, as shown in Table 12, E4 cannot improve both these modes alone, i.e. increase of E4 will reduce the error in mode 4 and increase the error in mode 3, and vice versa.

Table 13, parameter change updating 2

Parameter	Initial value (N/mm^2)	Updated value (N/mm^2)	Change
E2 Bottom cord	210 000	217 931	3.8%
E4 Vertical of truss	210 000	234 469	11.7%
E8 Bottom bracing	210 000	94 978	-54.8%

It may however be discussed, that this increase of stiffness in the vertical parts, may not have any physical meaning, or logical explanation. It may be expected that such a truss, with many connections, would have a lower stiffness in reality than in the model. To be sure, the original drawings of the bridge were compared to the model dimensions, and it was found that a part in the verticals of the truss, was modelled to be 12mm, when the drawings show 20mm. This is shown in the figure to the right, cut out from the drawing attached as B1 in the Appendix. However, when it was corrected, there was almost no effect on the dynamic response of the model. Which indicates that the increase in stiffness is compensating for other errors in the model.



E2 is only slightly increased, by 3.8%, this is as discussed a consequence of the large decrease in E8. This slight change in this parameter, which is mainly sensitive in mode2, suggest that the error in mode 2 would not have been large if this parameter was excluded from the procedure. Thus another parameter could have been chosen, for instance E3 which is sensitive in mode 3, and could correct some of the error in this mode.

After the updating, mode 3 and mode 4 are the two modes with unacceptably high errors, the reason that these modes has not been corrected satisfactorily is, as discussed, that out of the chosen three parameters there are only one with sensitivity in these modes. To correct the error in these modes it is necessary to conduct an updating procedure with two parameters, where one is more sensitive in one mode and the other parameter is more sensitive in the other mode.

As the stiffness reduction of E8 could be a compensation for the simplification of neglecting mass, the mass properties in the relevant modes of the bridge should be evaluated as possible updating parameters for future analysis.

6.6 Final Sensitivity Analysis

The two updating procedures done previously, even though the updated model is capable of representing the measured data, suggest that the true error sources are not yet found, because of the large parameter changes. Due to this, the responses outside the measuring range are probably not represented well. Therefore, other parameters such as mass and supports should also be evaluated as updating parameters. The error is probably spread throughout the structure, such as related to mass and supports. Hence, a further sensitivity analysis is conducted of both mass and support properties of the bridge, to help choose parameters for any future updating of this bridge. This may also work as an example for other similar structures of which parameters should be considered.

6.6.1 Support Stiffness

Adjusting the Model

In the chapter *"*6.4.2 Error Localization*"*, the simplification of modelling the bridge as simply supported is discussed, and whether or not this is an idealisation error that could cause large discrepancies in the response. Therefore, springs in all three directions as well as three rotational springs are modelled as replacements for the boundary conditions in all four supports, such that supports can be evaluated as parameters. In this way the supports are represented quantitatively, such that they can be modified to fit the measurements. This is unlike boundary conditions, which are either fixed or free to move in a specified direction.

As mentioned, every support has been represented with six springs, three in translational directions and three rotational. The sensitivity of each spring was then evaluated. All the rotational springs, compared to the translational springs, had close to zero sensitivity in every mode, thus they were not evaluated any further. The remaining twelve translational spring parameters were replaced with four parameters: all four transverse springs were clustered to one parameter, Fixed X. All four vertical springs were clustered to one, Fixed. Both of the longitudinal springs in one end were clustered as Fixed Z, and the other two longitudinal springs at the other end, were clustered as Roller Z.

As a starting point for the updating, it was decided to tune the spring supports, such that the new model would represent the model made by Bartosz Siedziako initially, and have the same analytical results. This was first done manually by engineering judgement and a trial and error approach to estimate the spring stiffness roughly.

Since the bridge is initially modelled as simply supported, it is completely free to move in longitudinal direction at one end, i.e. Roller Z, equals zero. However, as discussed, this is not the case in reality, and therefore this parameters was modified to a value, one percent of the pinned supports. All other translational supports were initially modelled as fixed, i.e. absolutely no translation was allowed in the supports. This is also not the reality, as there will be some deformations at the supports, hence Fixed X, Fixed Y and Fixed Z were modified to be softer than initially.

The levels of decrease in stiffness in these supports, were carefully chosen, so that all the supports had some sensitivity. If any of the supports had been given a too large value of stiffness, the sensitivity analysis would result in zero sensitivity of these supports. This is because, a completely stiff support has actually a very high stiffness, and there will be a point, above which all values will act as completely stiff, hence making the structure insensitive to any further increase. This point was found and chosen by adjusting the stiffness of these springs manually, until the highest stiffness was found, that still gave a change in frequency when increased slightly.

Second part of the tuning, fine-tuning, was done with the use of the updating script. The analytical responses from the initial model were taken as target responses in the updating procedure, with all four support stiffness as updating parameters. The tuned model then produced the same results as the initial model, i.e. the spring stiffness in the tuned model now represented the boundary conditions of the initial model. The initial parameter estimations, after manual tuning and after the fine-tuning for all four springs are shown in Table 14.

Table 14, spring stiffness at supports

	Fixed X	Fixed Y	Fixed Z	Roller Z
Initially	infinity	infinity	infinity	0
Manual tuning	1e7	1e7	1e7	1e5
Fine tuning	5.8e6	4.2e7	1.0e8	0

Sensitivity Analysis

The objective of this chapter is to investigate to what extent the idealization error of modelling the bridge as simply supported will affect the dynamic responses of such a structure. This is done by evaluating the sensitivity of the support parameters in conjunction with error in measurements and whether or not a change in the supports stiffnesses are physically logical.

The sensitivities, which are shown in Figure 40 , were calculated based on a slight decrease in the fixed supports and a slight increase in the roller support, because these are the predicted errors in the model, thus expected changes if supports are chosen as updating parameters. The normalization of support sensitivities are done by multiplying with the next parameter value, rather than the initial value. Otherwise, the normalized sensitivity of Roller Z would be zero, and not true.

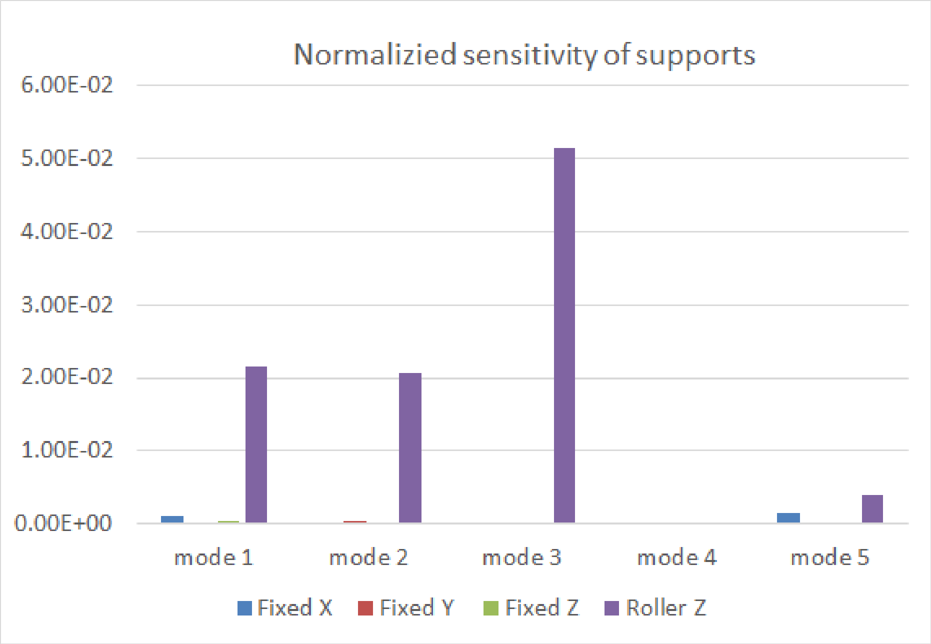


Figure 40, Normalized sensitivities of supports

Figure 40 shows the normalized sensitivities of all four supports. It is clear that Roller Z is the most sensitive parameter with highest sensitivity in mode 3. This parameter is equal to zero, but as discussed, there are certainly some stiffness in Roller Z in reality. If this parameter had some stiffness in the model, it would influence mode 1, 2 and 3 with an increase of frequency. This would again make the residuals larger, which means that the simplification of modelling a roller support made the analytical responses closer to the measurements. However, this does not mean that Roller Z is not relevant for updating, but rather a very relevant if there are any modes which is corrected by a change in another parameter that causes frequency reduction in mode 3, then an increase of Roller Z would correct the error. Even though the sensitivity is very low compared to the sensitivity of E-modulus, a greater relative change of Roller Z is accepted.

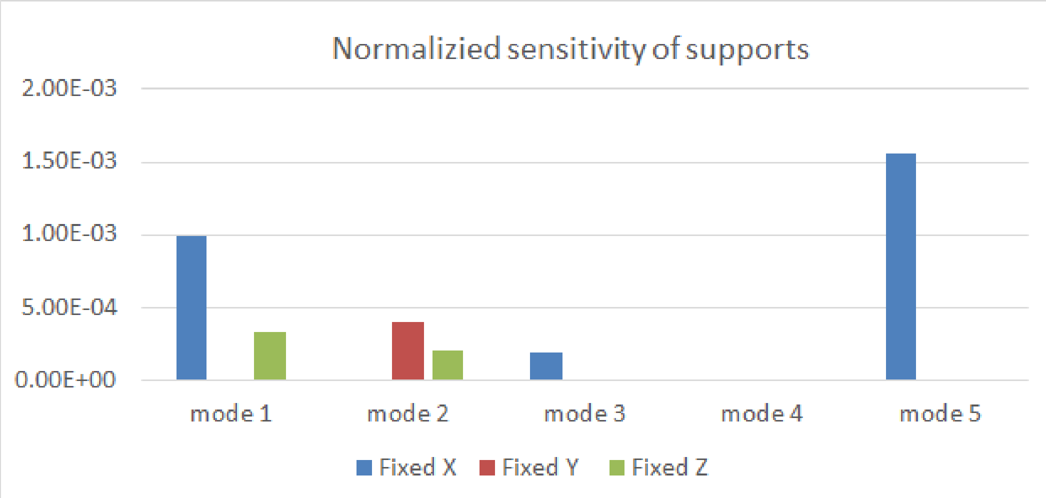


Figure 41, Normalized sensitivity of fixed supports

Because the sensitivities of the fixed supports are very low compared to Roller Z, they do not show in Figure 40, hence they are presented alone in Figure 41. The figure shows that Fixed X is most sensitive in the two transversal bending modes: mode 1 and 5. These modes holds the highest negative error, therefore a decrease in this parameter could correct some of that error. A decrease in this parameter is also logical, since it is probably modelled as too stiff. However, with such low sensitivity too large parameter change would be required to correct these errors. This implies that modelling error in Fixed X is not the only source of frequency error in mode 1 and 5, but rather a contributor to the error.

Parameter Fixed Y is the springs in the vertical direction. This parameter is the most sensitive in mode 2, the vertical bending mode. This mode have small error in frequency compared to other modes. This suggests that Fixed Y holds small or no error, which is logical because most probably the ground should have settled completely by now, and therefore the ground is very stiff. Hence, all the vertical deformations at the supports is caused by deformation in the abutments and in the connection between the abutment and the bridge. This parameter is not sensitive in any other mode, as mode 2 is the only mode with vertical action.

Fixed Z is sensitive in mode 1 and 2 only, but insignificantly. One would expect that this parameter was sensitive in mode 3 as well, just as Roller Z, which is most sensitive in this mode. However, it has been shown that the sensitivities change as the parameters change. Therefore, it is reason to believe that this parameter would be sensitive in mode 3 if the parameter were reduced.

It should be noted that none of the supports are sensitive in mode 4, i.e. a change in the support parameters will not affect the frequency in mode 4. This is because mode 4 is purely vibration of the truss, and no lateral or vertical movement of the bridge deck itself, i.e. no forces in the supports. Additionally, mode 4 is the only mode with positive error, i.e. modelled too soft, all other modes are too stiff. This clearly implies that the supports are modelled too stiff. A reduction in support stiffness could correct the frequency error in the other four modes without making mode 4 even softer and increase the error in this mode.

However, If only support parameters are chosen for model updating, the parameters would be changed significantly in order to correct the model due to the low sensitivities, hence the parameters would lose their physical meaning. In addition, the supports could be more sensitive to other modes, such as higher order modes, which would cause very large changes in the frequency of these modes. However, if numerous modes are identified by the measurements, the user may choose several parameters for updating. Only then, supports can be chosen as updating parameters as supplements to other updating parameters.

6.6.2 Density as Parameter

Introduction

Since mass properties of a structure impacts the dynamic responses significantly and there are some modelling errors related to the mass, this should be evaluate this as updating parameters. This chapter aims at evaluating the impact any sources of error may have on the response and determine which mass parameters that, if modelled inaccurate, may cause large discrepancies in the frequencies.

As discussed in the chapter "6.4.2 Error Localization" there are some missing mass in the bridge deck, mostly because non-structural parts are not modelled. In addition, the wood density is highly uncertain. Thus, density of all the structural parts are evaluated as parameters to represent mass.

The structure is divided into parts as described earlier in chapter *"6.4 Choosing Parameters"*, and the normalized sensitivities of all density parameters are established and presented in Figure 42, including wood density, which represent the mass of the sleepers. Obviously, the density sensitivities are negative, which means that an increase of density would decrease the frequency. The absolute values of the sensitivities are shown in the figures.

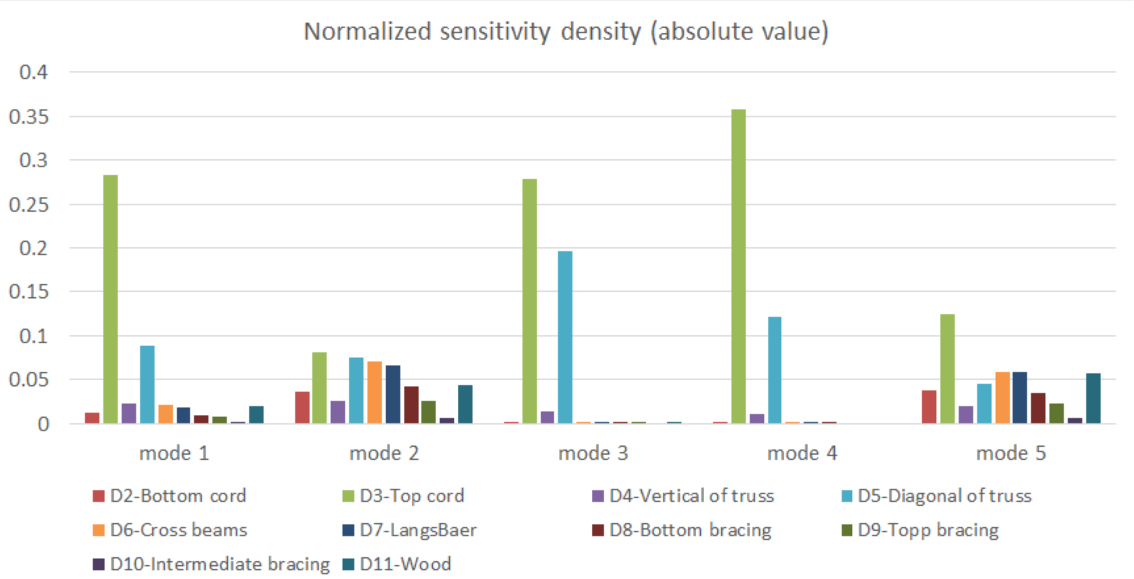


Figure 42, Normalized sensitivity of density

Several of the parameter's sensitivity vector have close to the same direction, i.e. they can be clustered to one parameter. Thus, parameters D6, D7, D8 and D9 were clustered to one parameter, D1-Clustered bridge deck, as the angle between the vectors were less than 5 degrees. Since all of these parameters are located in the bridge deck, it have physical meaning to cluster these parameters as one. Figure 43 shows the normalized sensitivity of the density of D1-Clustered bridge deck as well as well as the other parameters that were not clustered.

Sensitivity Analysis

The sensitivities are displayed in Figure 43. It shows that D1-Clustered bridge deck is sensitive in mode 1, 2 and 5, but mostly in mode 2, the vertical bending mode. All of these modes are bending/movement of the bridge deck, which gives rise to inertia effects of the mass located here. Unlike mode 1 and 5, mode 2 have low error in frequency, hence increasing this parameter to correct mode 1 and 5 would cause error in mode 2. This implies that modelling error in D1 is not the main source of frequency error in the model. Therefore, if choosing this parameter for updating, the user should also select other parameters that can correct either mode 1 and 5 or mode 2. It is known that there are missing mass in the bridge deck in the model, an increase in this parameter could represent the missing mass; hence an increase of this parameter is logical.

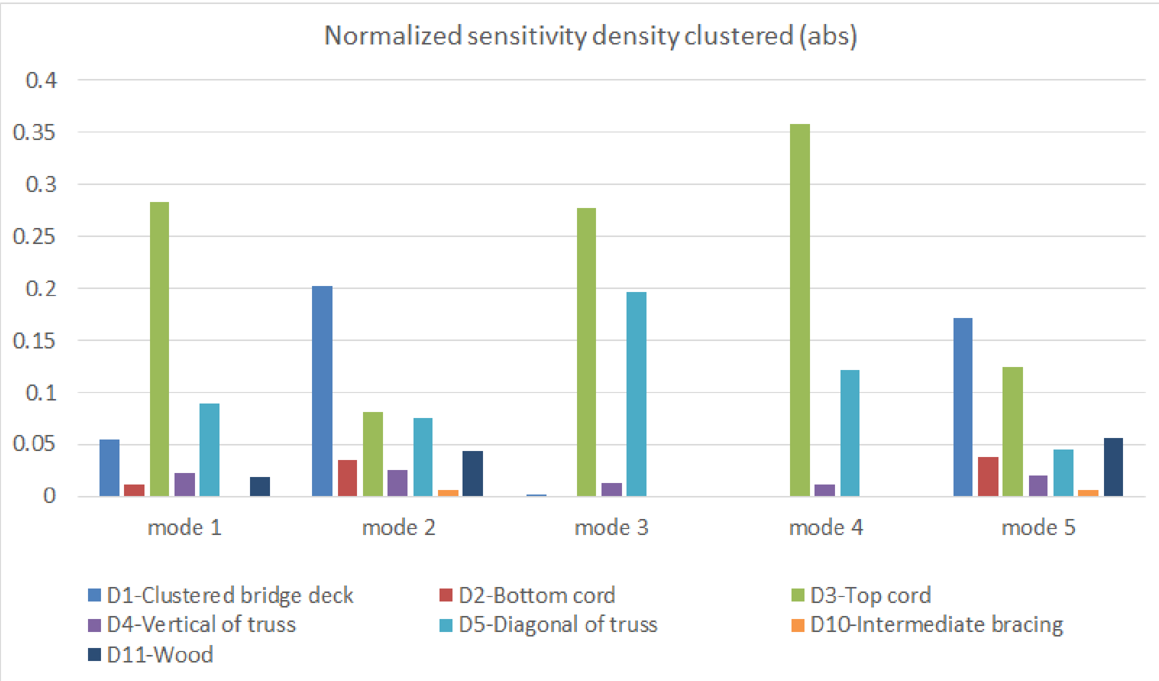


Figure 43, Normalized sensitivity of density with clustered bridge deck

As discussed, wood density is highly uncertain, therefore it is expected error in this parameter. The figure shows that D11 is sensitive in mode 5, 2 and 1, in that order. Again, an increase of this parameter would correct some of the error in these modes, but the sensitivity is very low, thus it requires a large change in D11 to correct the error in frequencies. However, there is most certainly some error related to the mass of the wooden sleepers. Therefore, this parameter should be evaluated as a supplement to other updating parameters that can correct the regarded modes.

Parameter D3-Top cord is the most sensitive parameter out of all density parameters. This is due to the transverse vibration of the truss, and the top cord's eccentricity to the rotational axis. This gives rise to high inertia effects, thus large changes in frequency as the parameter changes. It is not expected large error in this parameter. However, a small change in D3 could represent correction of mass related error in the whole truss. It is seen that the parameter is mostly sensitive in mode 4, but very sensitive in mode 1 and 3 as well. As discussed several times previously, the frequency error in mode 4 is positive, in contrast to the other modes. This means that the structure could be too heavy in this mode. Reducing the density of the top cord should correct discrepancies between measured and analytical response in this mode, however the frequency error in the other modes would increase. This indicated that this parameter alone is not the source of error in mode 1, 3 or 4.

The diagonals of the truss are sensitive in all modes, but mostly in mode 3. It is not expected any large errors in this parameter either, although like D3, change in D5 could also represent mass related error in the truss. This shows the importance of modelling such structural parts and their respective densities properly as a small modelling error, could cause large frequency error.

RESULTS AND DISCUSSION

This thesis leads through the whole process of model updating of a riveted railway bridge of steel, from the theoretical background to what to what consider when updating such a structure. It guides through a way of thinking when updating a model, aiming to optimize the process in the best way. Updating the rig first, and then the bridge, helps explain the importance and effects of choosing the different parameters in practise. Some of the most important factors that have come forward through the whole study are discussed further here.

Choosing parameters

Various, different types of parameters have been discussed and used, in the updating of the rig as well as the railway bridge. The results showed how the different choices effected the updating of these structures. One important lesson learned was how, neither sensitivity analysis alone nor error localization alone, would lead to a good choice of parameters. The key was definitely to combine these two, and use engineering judgement, to be able to somewhat predict the effects of the different parameter choices.

The complete sensitivity analysis carried out for both the structures, is a good starting point to be able to choose the updating parameters. Error localization was done qualitatively by using experience and engineering judgement. However, there are various quantitative techniques out there, which would be more robust and lead to the location of error more easily.

Based on engineering judgement and experience, one could say that in an old riveted bridge, one important area of error to consider is definitely the connections. It is too difficult to find the exact stiffness of these joints, and model them correctly. Therefore, it should be taken into consideration, that the chosen parameters are able to represent such an error. In this study it was believed that changes in E-moduli of components around the joints would be able to represent such errors, which is a just thought. However, because of other factors discussed here, these parameters had to compensate for other errors also, resulting in too large changes.

Ideally, all the components within some of the joints, such as all joints in the truss, could be assigned the same material, and be used as one updating parameter, that would only represent the joint stiffness. Other parameters could then be chosen to represent other possible error areas. The time of this study did not allow this, but it is recommended to consider this in other studies of riveted bridge model updating.

It was also noted, that using 4 parameters, when there were 5 measured modes for the updating, would definitely lead to a smaller residual. However, number of parameters were kept to only three, which lead to some larger residual, but most probably a more correct model that can represent the modes outside these 5 modes as well.

When the number of parameters is kept very low in a complex structure such as a bridge, it will be more difficult to be able to correct the real errors in the model, as was seen in the updating done here. The real errors are most probably located throughout the bridge, in the form of different stiffness- and mass-related properties of the structure. The small number and variety of updating parameters, will lead to some of the parameters changing too much, in order to compensate for these errors, hence loosing physical meaning. This was clearly seen in the updating of the railway bridge. It could therefore be discussed, that to achieve optimal updating results for this bridge, there is a need of collecting more test data.

Quality of response data for updating

Since the measured response data from the structure is considered correct, and the model is adjusted up against this, the reliability of the updated model depends greatly on the quality of measured data.

As it comes forward in updating of the bridge, mode 3 and 4, have 78.1% and 84.4% match, to their analytical counterparts respectively. This is lower than the other modes, and may suggest unreliability of data from these modes, including frequencies. Usually, higher modes will have a greater chance of being unreliable (Brownjohn and Xia, 2000), which is also why the first mode has the greatest match. However, in this case, mode 5 has a better match than these two modes, indicating some other reason might be behind a potential error when matching these.

Mode 3 and 4 are quite similar, considering the fact that both of these have sidewall/truss bending. The main difference lies in the direction of movement of the upper parts of the sidewalls, which is also where measuring points are not placed on the structure. This may indicate that the measuring points are not able to differentiate between or define these two modes in detail, hence causing the lower match. To avoid such a problem, measuring points should be placed very carefully, based on a dynamic analysis of the structure prior to updating.

When a structure is excited to collect data, it may involve a lot of noise. It can be discussed that in large structures, such as the railway bridge, it will be difficult to excite it with the very large loads needed to easily distinguish between the signal and noise. This may also contribute to the low match of some of the modes. The importance of using precise instruments, and planning the placements of the measuring points, seem to be important factors for a successful matching of modes, hence, successful updating.

FE modelling for updating

As it comes forward in the study, model updating puts a special requirement to the level of detail when modelling the structure. For instance, updating the railway bridge indicated some differences in the mass on the deck, and difference in density of wood, when compared to the real structure. Also, the fact that the riveted joints were modelled as fully stiff, while in reality there were riveted connection plates, contributing to increase in mass as well as a probable decrease in joint stiffness.

Such differences between the model and the real structure might not have any considerable effect in a usual static analysis. However, to be able to update a model successfully, such details should be taken into account. That, so the initial model is as close as possible to the real structure, as well as there is a greater variety of updating parameters to choose from. That will make it possible to correct the real errors, instead of other parameters adjusting themselves to compensate for many different errors, losing their physical meaning.

Modelling of supports

As discussed, detailed modelling of structural components, contributes greatly to the correct updating of the structure. In the updating of the railways bridge, it was considered that the model might have idealization errors related to the supports. The sensitivity analysis of supports, with the given five modes, indicated that the model was almost insensitive to the supports. However, it should be noticed that based on engineering judgement and experience, one could with great certainty say that there will be some discrepancies related to such a modelling of supports.

Even if the supports have low sensitivity in this railway bridge, they should be considered when updating with other modes for example in the future, as these may be more sensitive. Also when updating any other similar structure, supports would still be an important parameter to consider for updating.

The method used in this thesis to replace the boundary conditions with springs by tuning their stiffness to get the initial ones, may not be hundred percent reliable. This is because it was difficult to find a good initial stiffness of each spring, which would represent a hundred percent stiffness, and still have some sensitivity. Ideally, some rough estimations of the soil bearing should be done, and be used to decide the initial stiffness of the vertical springs. In addition the stiffness of the roller support in longitudinal direction, should also be evaluated and used as the spring stiffness in this direction.

CONCLUSION

- Sensitivity based model updating definitely is a good way of achieving a more correct model of a structure. This model may then be used for further analysis, and is very relevant for example to evaluate the remaining lifetime of the structure.
- Model updating requires a more detailed FE model, than that needed for static analysis of a structure for example. It has to be as close to the real structure as possible initially, in order to achieve convergence.

- Measured data should be taken in such a way, that it ensures a clear difference between the modes, and hence gives clearer match to the analytical modes. This should be done by carrying out a dynamic analysis of the structure, prior to updating, which would help placing the measuring points at logical locations.

- Number of measured modes should be kept as large as possible. This will allow a greater number of updating parameters, as well as possibility of cross verification.

- There should be carried out a complete sensitivity analysis and analysis of error localisation, prior to the updating, to choose the updating parameters.

- When there are very few parameters used in the updating, such that changes in these not only have to correct "their own" errors, but also compensate for other errors in the model, bounds should not be used. However, these parameters will then lose their physical meaning. If there are many parameters used for updating, bounds may be used to ensure that each parameter changes within a given range.

- In riveted railway bridges, connections will be a very probable cause of error in the model, and should be considered in the choice of updating parameters. In addition supports may also be a probable source of error in such a structure, and should be considered. With the given 5 modes in the updating of the bridge, the supports should not be included because of their extremely low sensitivity.

FURTHER WORK SUGGETION

The suggestions for further work is only related to a successful updating of the riveted railway bridge used in this study. However, all information may be used as a guide for the updating of similar bridges, generally.

It is suggested to take more measurements of the bridge. These measurements should be taken with measuring points on also, the upper parts of the sidewalls. It is believed that mode 3 and mode 4 will then be defined more clearly, hence giving a more reliable response data to be used in the updating. In addition, cross verification should be done, by using measurements of lower and higher modes, to ensure that the updating is done correctly.

It is also suggested to adjust the model by modelling some of the parts in more detail, such as the inspection deck. Also a more accurate density of wood should be used, based on considering the varying factors while taking the measurements, such as whether it has rained in that period or not.

In a future updating of the bridge, connections are advised to be considered as one individual updating parameter, by assigning all the parts in the connections the same material.

BIBLIOGRAPHY

AAS, H. 2004. Jernbanedrift er lønnsomt, men ikke på Meråkerbanen, Raumabanen og deler av Rørosbanen. Available: https://samferdsel.toi.no/article18772-341.html.

ALLEMANG, R. 2003. The Modal Assurance Criterion – Twenty Years of Use and Abuse. *Sound and Vibration*, 14-21.

BAKHTIARI-NEJAD, F., RAHAI, A. & ESFANDIARI, A. 2005. A structural damage detection method using static noisy data. *Engineering Structures,* 27**,** 1784-1793.

BROWNJOHN, J. & XIA, P. 2000. Dynamic Assessment of Curved Cable-Stayed Bridge by Model Updating. *Journal of Structural Engineering,* 126**,** 252-260.

CAWLEY, P. & ADAMS, R. 1979. The Location of Defects from Measurements of Natural

Frequencies. .

ESFANDIARI, A., SANAYEI, M., BAKHTIARI-NEJAD, F. & RAHAI, A. 2010. Finite Element Model Updating Using Frequency Response Function of Incomplete Strain Data. *AIAA Journal,* 48**,** 1420-1433.

FRISWELL, M. I. & MOTTERSHEAD, J. E. 1995. *Finite Element Model Updating in Structural*

Dynamics, Norwell, Kluwer Academic Publishers Group.

GRAFE, H. 1998. *Model Updating of Large Structural Dynamics Models Using Measured*

Response Functions. Doctor of Philosophy, University of London.

MAIA, N. M. M. & MONTALVAO SILVA, J. M. 1998. *Theoretical and experimental modal analysis,* Baldock, Hertfordshire, England, Research Studies Press.

MARWALA, T. 2010. *Finite-element-model updating*, Springer.

MATHWORKS. 2016. *System Identification Overview* [Online]. Available: http://se.mathworks.com/help/ident/gs/about-system-identification.html?s_tid=gn_loc_drop.

MOTTERSHEAD, J. E. & FRISWELL, M. I. 1993. Model Updating In Structural Dynamics: A Survey. Swansea: Department of Mechanical Engineering, University College of Swansea.

MOTTERSHEAD, J. E., LINK, M. & FRISWELL, M. I. 2010. The sensitivity method in finite element model updating: A tutorial. *Mechanical Systems and Signal Processing,* 25**,** 2275-2296.

RAD, S. Z. 1997. *METHODS FOR UPDATING NUMERICAL MODELS IN STRUCTURAL DYNAMICS.* Doctor of Philosophy, Imperial College of Science, Technology and Medicine.

REN, W.-X. & CHEN, H.-B. 2010. Finite element model updating in structural dynamics by using the response surface method. *Engineering Structures,* 32**,** 2455-2465.

SIMULA, D. S. 2013. *Abaqus/CAE User's Guide* [Online]. Available: http://129.97.46.200:2080/v6.13/books/usi/default.htm.

ZANG, C., MA, S. C. & FRISWELL, M. I. 2012. Structural model updating with an improved parameter selection method. *ISMA 2012.*

APPENDIX

A1-The script

```
#!/usr/bin/env python

#!/usr/bin/python

# -*- coding: utf-8 -*-


# Model name

ModelName='gtf_model_2_6'


# Step name

StepName='freq'


# name of the initial odb file

odbname='Job-1'


# Name of the initial job

JobName='Job-1'


import numpy

# meassured natural frequency

# values in a list, e.g three modes with their respective
natural frequency: wm=[2,5,13]

# values in Hz

wm=[5.2530499999999997,9.4756999999999790,9.71776000000000
0176,9.8443900000000640,11.1789000000000050]
```

```
zm=numpy.array(wm)

We=numpy.diag(zm**-2)


# names of the spring or dashpot elements of which the parameters should be evaluated.

# names in list, e.g ['name-1','name-2','name-3']

# if no spring or dashpot elements should be evaluates, then springDashpotsName=[]

springDashpotsName_s=[]

# maximum and minimum values, i.e. upper and lower bounds of every input element

Maximum_s=[]

Minimum_s=[]


# names of the mass/inertia elements of whitch the parameters should be evaluated.

# names in list, e.g ['name-1','name-2','name-3']

# if non  ass/inertia elements should be evaluates, then inertiaName=[]

inertiaName_m=[]

# maximum and minimum values, i.e. upper and lower bounds of every input element

Maximum_m=[]

Minimum_m=[]


# names of the materials of whitch the density parameter should be evaluated.

# names in list, e.g ['name-1','name-2','name-3']

# if non material density should be evaluates, then inertiaName=[]
```

```
MaterialName_dd=[]

# maximum and minimum values, i.e. upper and lower bounds of
every input element

Maximum_d=[]

Minimum_d=[]


# names of the materials of whitch the elasticity modulus
parameter should be evaluated.

# names in list, e.g ['name-1','name-2','name-3']

# if non material elasticity modulus should be evaluates, then
inertiaName=[]

MaterialName_ee=[]

# maximum and minimum values, i.e. upper and lower bounds of
every input element

Maximum_e=[]

Minimum_e=[]


#Measured modal eigenvectors in the MAC-points(x,y, in point
1, x,y,z in point 2, etc.)

Mm1=[4.71799999999999972e-01,1.61920000000000084e-
01,6.83350000000000164e-03,4.94740000000000131e-01,-
2.11650000000000048e-01,9.26669999999999937e-
02,8.18919999999999813e-01,2.70050000000000123e-01,-
4.92650000000000323e-02,7.95490000000000304e-01,-
3.48389999999999775e-01,1.19709999999999970e-
01,1.00000000000000000e+00,3.20749999999999796e-
01,1.95420000000000034e-02,9.91049999999999865e-01,-
3.58430000000000264e-01,2.97870000000000098e-
02,8.73129999999999617e-01,3.09819999999999843e-
01,8.61039999999999998e-02,8.56199999999999611e-01,-
3.53119999999999894e-01,-4.05430000000000265e-
02,5.77439999999999533e-01,2.24319999999999916e-
01,6.77050000000000129e-02]

Mm2=[1.17590000000000001e-01,4.59770000000000120e-
01,1.66199999999999866e-01,6.57760000000000113e-
```

```
02,5.14820000000000554e-01,1.53249999999999973e-
01,1.17230000000000009e-01,7.65900000000000252e-
01,1.04840000000000026e-01,8.81030000000000076e-
02,8.65260000000000291e-01,1.89099999999999903e-
01,7.93800000000000611e-02,9.28509999999999465e-
01,1.97739999999999991e-01,-4.63119999999999900e-
02,1.00000000000000000e+00,2.50599999999999894e-
01,4.22259999999999965e-02,7.84200000000000082e-
01,1.68099999999999994e-01,3.26270000000000322e-
02,8.65729999999999995e-01,2.73289999999999774e-01,-
6.63090000000000336e-03,4.91740000000000104e-
01,2.46969999999999951e-01]

Mm3=[7.51279999999999478e-01,1.90209999999999902e-01,-
2.76729999999999968e-02,1.00000000000000000e+00,-
2.92669999999999857e-01,3.63379999999999808e-
01,4.10650000000000150e-01,-1.17360000000000060e-
01,1.98110000000000083e-01,4.11090000000000110e-
01,6.05240000000000141e-02,3.09709999999999853e-02,-
1.49340000000000005e-01,-3.42469999999999966e-
01,2.24560000000000096e-01,-9.38999999999999736e-
02,2.00169999999999868e-01,-4.69769999999999966e-
03,3.74499999999999996e-01,1.42170000000000047e-
02,3.16099999999999921e-02,4.20879999999999764e-
01,1.20199999999999943e-02,3.22979999999999893e-
01,8.89630000000000319e-01,1.36110000000000088e-
01,3.08410000000000175e-01]

Mm4=[-4.85999999999999876e-01,-5.60469999999999786e-03,-
5.67210000000000058e-02,7.26180000000000477e-01,-
8.25140000000000401e-02,2.62259999999999931e-01,-
3.16290000000000156e-01,9.81449999999999613e-
02,6.97320000000000509e-01,2.85119999999999846e-
01,1.21789999999999955e-
01,1.00000000000000000e+00,6.07979999999999788e-02,-
5.36350000000000212e-02,9.38419999999999486e-02,-
9.83109999999999561e-02,5.52049999999999713e-02,-
4.48810000000000075e-03,3.43679999999999855e-01,-
8.64230000000000329e-03,7.46510000000000069e-01,-
2.43930000000000079e-01,-1.09179999999999993e-
01,9.43089999999999839e-01,6.03450000000000419e-
01,8.36719999999999638e-02,5.62779999999999470e-01]

Mm5=[3.39310000000000005e-01,-1.92459999999999922e-
01,5.25440000000000043e-02,4.05949999999999775e-
01,1.63229999999999862e-01,1.10570000000000016e-
```

```
01,8.79920000000000355e-01,-4.21989999999999762e-
01,5.74090000000000170e-02,8.27459999999999733e-
01,2.93949999999999891e-01,-1.97530000000000111e-
01,1.00000000000000000e+00,-5.04290000000000160e-01,-
9.96250000000000522e-02,9.55539999999999450e-
01,3.10670000000000017e-01,9.11809999999999816e-
02,9.03549999999999643e-01,-4.01710000000000115e-01,-
2.99130000000000070e-01,8.45690000000000528e-
01,2.60259999999999913e-01,2.86470000000000024e-
01,4.82360000000000110e-01,-2.46620000000000059e-
01,4.43629999999999966e-02]


Mmm=list([Mm1,Mm2,Mm3,Mm4,Mm5])



# initial delta parameter guess in percentage of initial
parameter values

a1=0.05

a2=0.05

#Convergense criterion (procedure ends when residual is lower
than b)

b=0.01

#Input END



import numpy

import os

from odbAccess import *

import time

import math

from abaqusConstants import *
```

```
from odbMaterial import *
from odbSection import *

myodbpath=odbname + '.odb'
odb=openOdb(myodbpath)

Ps_1=[0]*len(springDashpotsName_s)
Ps_0=[0]*len(springDashpotsName_s)

for i_s in xrange(0,len(springDashpotsName_s)):
     Ps=mdb.models[ModelName].rootAssembly.engineeringFeatures.
springDashpots[springDashpotsName_s[i_s]].springStiffness
     Ps_1[i_s]=Ps

dPs_0=a1*numpy.array(Ps_1)
dPs_0_gs=a2*numpy.array(Ps_1)

Pm_1=[0]*len(inertiaName_m)
Pm_0=[0]*len(inertiaName_m)

for i_m in xrange(0,len(inertiaName_m)):
     Pm=mdb.models[ModelName].rootAssembly.engineeringFeatures.
inertias[inertiaName_m[i_m]].mass
     Pm_1[i_m]=Pm

dPm_0=a2*numpy.array(Pm_1)
dPm_0_gm=a2*numpy.array(Pm_1)
```

```
Pd_1=[0]*len(MaterialName_dd)

Pd_0=[0]*len(MaterialName_dd)


for i_d in xrange(0,len(MaterialName_dd)):

     Pd=mdb.models[ModelName].materials[MaterialName_dd[i_d]].density.table[0][0]

     Pd_1[i_d]=Pd


dPd_0=a2*numpy.array(Pd_1)

dPd_0_gd=a2*numpy.array(Pd_1)


Pe_1=[0]*len(MaterialName_ee)

Pe_0=[0]*len(MaterialName_ee)


for i_e in xrange(0,len(MaterialName_ee)):

     Pe=mdb.models[ModelName].materials[MaterialName_ee[i_e]].elastic.table[0][0]

     Pe_1[i_e]=Pe


dPe_0=a2*numpy.array(Pe_1)

dPe_0_ge=a2*numpy.array(Pe_1)


Am1=[0]*27

Am2=[0]*27

Am3=[0]*27

Am4=[0]*27

Am5=[0]*27
```

```
Am6=[0]*27

Am7=[0]*27

Am8=[0]*27


Amm=list([Am1,Am2,Am3,Am4,Am5,Am6,Am7,Am8])


MACpoint=['MACPOINT01','MACPOINT02','MACPOINT03','MACPOINT04',
'MACPOINT05','MACPOINT06','MACPOINT07','MACPOINT08','MACPOINT0
9']


for i in xrange(1,len(odb.steps[StepName].frames)):

     Frames=odb.steps[StepName].frames[i]

     displacement=Frames.fieldOutputs['U']


     for nMAC in xrange(0,9): #9=len(MACpoint)

          MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]


     MAC_point_displ=displacement.getSubset(region=MAC_point)


          for xyz in xrange(0,3):  #3=number of directions in
each point

               Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]




Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]

Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]

MAC_cdr_top=Am.T*Mm

MAC_cdr_bot=Am.T*Mm
```

```
MAC_cdr=Am.T*Mm
MAC_cdr_alt=Am*Mm.T

for im in xrange(0,len(Mmm)):
     Mm=numpy.array(Mmm[im])[numpy.newaxis]

     for ia in xrange(0,len(Amm)):
          Am=numpy.array(Amm[ia])[numpy.newaxis]
          MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2
          MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

     MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
          MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]

AmNumber=[0]*5
for i_Am in xrange(0,len(AmNumber)):
     AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)

wa=[0]*(len(AmNumber))
for i_w in range(0, len(AmNumber)):
     wa_i=(odb.steps[StepName].frames[AmNumber[i_w]+1].frequenc
y)
     wa[i_w]=wa_i

rt=numpy.array(numpy.array(wm)-numpy.array(wa))[numpy.newaxis]
r=rt.T

count=0
```

```
r_max_abs=max(abs(x) for x in r)


while r_max_abs>b:
     print 'r'+'_'+str(count)+'='+str(r)
     count=count+1


     nonzero_s=numpy.nonzero(dPs_0)[0]


     springDashpotsName=[0]*len(nonzero_s)
     for i_ss in xrange (0, len(nonzero_s)):

     springDashpotsName[i_ss]=springDashpotsName_s[nonzero_s[i_
ss]]


     Ps_1=[0]*len(springDashpotsName)
     Ps_0=[0]*len(springDashpotsName)
     for i_s in xrange(0,len(springDashpotsName)):

     Ps=mdb.models[ModelName].rootAssembly.engineeringFeatures.
springDashpots[springDashpotsName[i_s]].springStiffness
          Ps_1[i_s]=Ps


     dPs_0_gs=a1*numpy.array(Ps_1)


     gs=list([0]*(len(springDashpotsName)))


     for i_s in xrange(0,len(springDashpotsName)):
          Ps_0[i_s]=Ps_1[i_s]
          dPs_0_gs[i_s]=a2*Ps_0[i_s]
```

```
        Ps_1[i_s]=Ps_0[i_s]+dPs_0_gs[i_s]

    mdb.models[ModelName].rootAssembly.engineeringFeatures.spr
ingDashpots[springDashpotsName[i_s]].setValues(
        springStiffness=(Ps_1[i_s]))


        jsname = 'job_s' + str(i_s+1) + '_' + str(count)
        js = mdb.Job(name=jsname, model=ModelName)
        js.submit(consistencyChecking=OFF)
        time.sleep(.5)
        js.waitForCompletion()


        from odbAccess import*
        odbs=openOdb(jsname+'.odb')


        for i in xrange(1,len(odb.steps[StepName].frames)):
            Frames=odb.steps[StepName].frames[i]
            displacement=Frames.fieldOutputs['U']


            for nMAC in xrange(0,9): #9=len(MACpoint)

    MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]

    MAC_point_displ=displacement.getSubset(region=MAC_point)


                for xyz in xrange(0,3):  #3=number of
directions in each point
                    Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]
```

```
        Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]
        Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]
        MAC_cdr_top=Am.T*Mm
        MAC_cdr_bot=Am.T*Mm
        MAC_cdr=Am.T*Mm
        MAC_cdr_alt=Am*Mm.T

        for im in xrange(0,len(Mmm)):
             Mm=numpy.array(Mmm[im])[numpy.newaxis]

             for ia in xrange(0,len(Amm)):
                  Am=numpy.array(Amm[ia])[numpy.newaxis]
                  MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2

    MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

    MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
                  MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]

        AmNumber=[0]*5
        for i_Am in xrange(0,len(AmNumber)):
             AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)

        wa_s=[0]*(len(AmNumber))
        for iw in xrange(0, len(AmNumber)):

    wa_ss=(odbs.steps[StepName].frames[AmNumber[iw]+1].frequen
cy)
             wa_s[iw]=wa_ss
```

```
            dwa_s=numpy.array(wa_s)-numpy.array(wa)

            gs[i_s]=numpy.divide(dwa_s,dPs_0_gs[i_s])

        mdb.models[ModelName].rootAssembly.engineeringFeatures.spr
ingDashpots[springDashpotsName[i_s]].setValues(
                springStiffness=(Ps_0[i_s]))



        nonzero_m=numpy.nonzero(dPm_0)[0]

        inertiaName=[0]*len(nonzero_m)
        for i_mm in xrange (0, len(nonzero_m)):
            inertiaName[i_mm]=inertiaName_m[nonzero_m[i_mm]]

        Pm_1=[0]*len(inertiaName)
        Pm_0=[0]*len(inertiaName)
        for i_m in xrange(0,len(inertiaName)):

        Pm=mdb.models[ModelName].rootAssembly.engineeringFeatures.
inertias[inertiaName[i_m]].mass
            Pm_1[i_m]=Pm

        dPm_0_gm=a2*numpy.array(Pm_1)

        gm=list([0]*(len(inertiaName)))
```

```
    for i_m in xrange(0,len(inertiaName)):
        Pm_0[i_m]=Pm_1[i_m]
        dPm_0_gm[i_m]=a2*Pm_0[i_m]
        Pm_1[i_m]=Pm_0[i_m]+dPm_0_gm[i_m]

    mdb.models[ModelName].rootAssembly.engineeringFeatures.ine
rtias[inertiaName[i_m]].setValues(
        mass=(Pm_1[i_m]))


        jmname = 'job_m' + str(i_m+1) + '_' + str(count)
        jm = mdb.Job(name=jmname, model=ModelName)
        jm.submit(consistencyChecking=OFF)
        time.sleep(1.0)
        jm.waitForCompletion()


        from odbAccess import*
        odbm=openOdb(jmname+'.odb')


        for i in xrange(1,len(odb.steps[StepName].frames)):
            Frames=odb.steps[StepName].frames[i]
            displacement=Frames.fieldOutputs['U']


            for nMAC in xrange(0,9): #9=len(MACpoint)

    MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]


    MAC_point_displ=displacement.getSubset(region=MAC_point)
```

```
                    for xyz in xrange(0,3):  #3=number of
directions in each point
                        Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]



        Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]
        Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]
        MAC_cdr_top=Am.T*Mm
        MAC_cdr_bot=Am.T*Mm
        MAC_cdr=Am.T*Mm
        MAC_cdr_alt=Am*Mm.T

        for im in xrange(0,len(Mmm)):
            Mm=numpy.array(Mmm[im])[numpy.newaxis]

            for ia in xrange(0,len(Amm)):
                Am=numpy.array(Amm[ia])[numpy.newaxis]
                MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2

    MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

    MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
                MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]

        AmNumber=[0]*5
        for i_Am in xrange(0,len(AmNumber)):
            AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)
```

```
        wa_m=[0]*(len(AmNumber))
        for iw in xrange(0, len(AmNumber)):

    wa_mm=(odbm.steps[StepName].frames[AmNumber[iw]+1].frequen
cy)
            wa_m[iw]=wa_mm
        dwa_m=numpy.array(wa_m)-numpy.array(wa)

        gm[i_m]=numpy.divide(dwa_m,dPm_0_gm[i_m])


    mdb.models[ModelName].rootAssembly.engineeringFeatures.ine
rtias[inertiaName[i_m]].setValues(
            mass=(Pm_0[i_m]))

    nonzero_d=numpy.nonzero(dPd_0)[0]

    MaterialName_d=[0]*len(nonzero_d)
    for i_dd in xrange (0, len(nonzero_d)):
        MaterialName_d[i_dd]=MaterialName_dd[nonzero_d[i_dd]]

    Pd_1=[0]*len(MaterialName_d)
    Pd_0=[0]*len(MaterialName_d)
    for i_d in xrange(0,len(MaterialName_d)):

    Pd=mdb.models[ModelName].materials[MaterialName_d[i_d]].de
nsity.table[0][0]
        Pd_1[i_d]=Pd
```

```
    dPd_0_gd=a2*numpy.array(Pd_1)

    gd=list([0]*(len(MaterialName_d)))

    for i_d in xrange(0,len(MaterialName_d)):
        Pd_0[i_d]=Pd_1[i_d]
        dPd_0_gd[i_d]=a2*Pd_0[i_d]
        Pd_1[i_d]=Pd_0[i_d]+dPd_0_gd[i_d]


    mdb.models[ModelName].materials[MaterialName_d[i_d]].densi
ty.setValues(table=((Pd_1[i_d],
    ), ))

        jdname = 'job_d' + str(i_d+1) + '_' + str(count)
        jd = mdb.Job(name=jdname, model=ModelName)
        jd.submit(consistencyChecking=OFF)
        time.sleep(1.0)
        jd.waitForCompletion()

        from odbAccess import*
        odbd=openOdb(jdname+'.odb')

        for i in xrange(1,len(odb.steps[StepName].frames)):
            Frames=odb.steps[StepName].frames[i]
            displacement=Frames.fieldOutputs['U']

            for nMAC in xrange(0,9): #9=len(MACpoint)
```

```
    MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]

    MAC_point_displ=displacement.getSubset(region=MAC_point)

                    for xyz in xrange(0,3):  #3=number of
directions in each point
                         Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]



        Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]
        Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]
        MAC_cdr_top=Am.T*Mm
        MAC_cdr_bot=Am.T*Mm
        MAC_cdr=Am.T*Mm
        MAC_cdr_alt=Am*Mm.T

        for im in xrange(0,len(Mmm)):
            Mm=numpy.array(Mmm[im])[numpy.newaxis]

            for ia in xrange(0,len(Amm)):
                Am=numpy.array(Amm[ia])[numpy.newaxis]
                MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2

    MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

    MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
                MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]
```

```
            AmNumber=[0]*5
            for i_Am in xrange(0,len(AmNumber)):
                 AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)

            wa_d=[0]*(len(AmNumber))
            for iw in xrange(0, len(AmNumber)):

       wa_dd=(odbd.steps[StepName].frames[AmNumber[iw]+1].frequen
cy)
                 wa_d[iw]=wa_dd
            dwa_d=numpy.array(wa_d)-numpy.array(wa)

            gd[i_d]=numpy.divide(dwa_d,dPd_0_gd[i_d])


       mdb.models[ModelName].materials[MaterialName_d[i_d]].densi
ty.setValues(table=((Pd_0[i_d],
       ), ))
       nonzero_e=numpy.nonzero(dPe_0)[0]

       MaterialName_e=[0]*len(nonzero_e)
       for i_ee in xrange (0, len(nonzero_e)):
            MaterialName_e[i_ee]=MaterialName_ee[nonzero_e[i_ee]]

       Pe_0=[0]*len(MaterialName_e)
       Pe_1=[0]*len(MaterialName_e)
       for i_e in xrange(0,len(MaterialName_e)):

       Pe=mdb.models[ModelName].materials[MaterialName_e[i_e]].el
astic.table[0][0]
```

```
        Pe_1[i_e]=Pe

    dPe_0_ge=a2*numpy.array(Pe_1)

    ge=list([0]*(len(MaterialName_e)))
    for i_e in xrange(0,len(MaterialName_e)):
        Pe_0[i_e]=Pe_1[i_e]
        dPe_0_ge[i_e]=a2*Pe_0[i_e]
        Pe_1[i_e]=Pe_0[i_e]+dPe_0_ge[i_e]



    mdb.models[ModelName].materials[MaterialName_e[i_e]].elast
ic.setValues(table=((Pe_1[i_e], 0.3
    ), ))

        jename = 'job_e' + str(i_e+1) + '_' + str(count)
        je = mdb.Job(name=jename, model=ModelName)
        je.submit(consistencyChecking=OFF)
        time.sleep(1.0)
        je.waitForCompletion()

        from odbAccess import*
        odbe=openOdb(jename+'.odb')

        for i in xrange(1,len(odb.steps[StepName].frames)):
            Frames=odb.steps[StepName].frames[i]
            displacement=Frames.fieldOutputs['U']
```

```
            for nMAC in xrange(0,9): #9=len(MACpoint)

     MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]

     MAC_point_displ=displacement.getSubset(region=MAC_point)

                for xyz in xrange(0,3):  #3=number of
directions in each point
                         Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]

        Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]
        Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]
        MAC_cdr_top=Am.T*Mm
        MAC_cdr_bot=Am.T*Mm
        MAC_cdr=Am.T*Mm
        MAC_cdr_alt=Am*Mm.T

        for im in xrange(0,len(Mmm)):
            Mm=numpy.array(Mmm[im])[numpy.newaxis]

            for ia in xrange(0,len(Amm)):
                Am=numpy.array(Amm[ia])[numpy.newaxis]
                MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2

     MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

     MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
                MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]
```

```
        AmNumber=[0]*5
        for i_Am in xrange(0,len(AmNumber)):
            AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)

        wa_e=[0]*(len(AmNumber))
        for iw in xrange(0, len(AmNumber)):

    wa_ee=(odbe.steps[StepName].frames[AmNumber[iw]+1].frequen
cy)
            wa_e[iw]=wa_ee
        dwa_e=numpy.array(wa_e)-numpy.array(wa) #dette er en
vektor av endringene i frequens i alle moder, pga endring i en
(E-modul)-parameter


        ge[i_e]=numpy.divide(dwa_e,dPe_0_ge[i_e])


    mdb.models[ModelName].materials[MaterialName_e[i_e]].elast
ic.setValues(table=((Pe_0[i_e], 0.3
    ), ))

    # For-loop END E-modul


    # SENSITIVITETS MATRISEN
    Gs=zip(*gs)
    Gm=zip(*gm)
    Gd=zip(*gd)
```

```
Ge=zip(*ge)


G=numpy.column_stack([Ge])
G_pinv=numpy.linalg.pinv(G, rcond=1e-15)


if len(springDashpotsName)>0:
    if len(MaterialName_d)>0:
        if len(MaterialName_e)>0:
            G=numpy.column_stack([Gs,Gd,Ge])
        else:
            G=numpy.column_stack([Gs,Gd])
    else:
        if (MaterialName_e)>0:
            G=numpy.column_stack([Gs,Ge])
        else:
            G=numpy.column_stack([Gs])
else:
    if len(MaterialName_d)>0:
        if len(MaterialName_e)>0:
            G=numpy.column_stack([Gd,Ge])
        else:
            G=numpy.column_stack([Gd])
    elif len(MaterialName_e)>0:
         G=numpy.column_stack([Ge])


G_pinv=numpy.linalg.pinv(G, rcond=1e-15)
```

```
    dP_0=G_pinv.dot(r)


    dP_0=numpy.linalg.pinv(G.T.dot(We).dot(G) , rcond=1e-
15).dot(G.T).dot(We).dot(r)


    for i_s in xrange(0,len(springDashpotsName)):
        dPs_0[i_s]=dP_0[i_s]


    for i_m in xrange(0,len(inertiaName)):
        dPm_0[i_m]=dP_0[i_m+len(springDashpotsName)]


    for i_d in xrange(0,len(MaterialName_d)):

    dPd_0[i_d]=dP_0[i_d+len(springDashpotsName)+len(inertiaNam
e)]


    for i_e in xrange(0,len(MaterialName_e)):

    dPe_0[i_e]=dP_0[i_e+len(springDashpotsName)+len(inertiaNam
e)+len(MaterialName_d)]




    for i_s in xrange(0,len(springDashpotsName)):
        Ps_1[i_s]=Ps_0[i_s]+dPs_0[i_s]


        if Ps_1[i_s]>Maximum_s[i_s]:
            Ps_1[i_s]=Maximum_s[i_s]
```

```
        elif Ps_1[i_s]<Minimum_s[i_s]:
            Ps_1[i_s]=Minimum_s[i_s]
        else:
            Ps_1[i_s]=Ps_1[i_s]

    for i_s in xrange(0,len(springDashpotsName)):

    mdb.models[ModelName].rootAssembly.engineeringFeatures.spr
ingDashpots[springDashpotsName[i_s]].setValues(
        springStiffness=(Ps_1[i_s]))




    for i_m in xrange(0,len(inertiaName)):
        Pm_1[i_m]=Pm_0[i_m]+dPm_0[i_m]

        if Pm_1[i_m]>Maximum_m[i_m]:
            Pm_1[i_m]=Maximum_m[i_m]
        elif Pm_1[i_m]<Minimum_m[i_m]:
            Pm_1[i_m]=Minimum_m[i_m]
        else:
            Pm_1[i_m]=Pm_1[i_m]

    for i_m in xrange(0,len(inertiaName)):

    mdb.models[ModelName].rootAssembly.engineeringFeatures.ine
rtias[inertiaName[i_m]].setValues(
        mass=(Pm_1[i_m]))
```

```
for i_d in xrange(0,len(MaterialName_d)):
     Pd_1[i_d]=Pd_0[i_d]+dPd_0[i_d]

     if Pd_1[i_d]>Maximum_d[i_d]:
          Pd_1[i_d]=Maximum_d[i_d]
     elif Pd_1[i_d]<Minimum_d[i_d]:
          Pd_1[i_d]=Minimum_d[i_d]
     else:
          Pd_1[i_d]=Pd_1[i_d]

for i_d in xrange(0,len(MaterialName_d)):

mdb.models[ModelName].materials[MaterialName_d[i_d]].densi
ty.setValues(table=((Pd_1[i_d],
), ))



for i_e in xrange(0,len(MaterialName_e)):
     Pe_1[i_e]=Pe_0[i_e]+dPe_0[i_e]

     if Pe_1[i_e]>Maximum_e[i_e]:
          Pe_1[i_e]=Maximum_e[i_e]
     elif Pe_1[i_e]<Minimum_e[i_e]:
          Pe_1[i_e]=Minimum_e[i_e]
     else:
          Pe_1[i_e]=Pe_1[i_e]
```

```
          dPe_0[i_e]=Pe_1[i_e]-Pe_0[i_e]


     for i_e in xrange(0,len(MaterialName_e)):

     mdb.models[ModelName].materials[MaterialName_e[i_e]].elast
ic.setValues(table=((Pe_1[i_e],0.3
     ), ))


     jwname = 'job_' + str(count)
     jw = mdb.Job(name=jwname, model=ModelName)
     jw.submit(consistencyChecking=OFF)
     time.sleep(.5)
     jw.waitForCompletion()


     from odbAccess import*
     odb=openOdb(jwname+'.odb')


     for i in xrange(1,len(odb.steps[StepName].frames)):
          Frames=odb.steps[StepName].frames[i]
          displacement=Frames.fieldOutputs['U']


          for nMAC in xrange(0,9): #9=len(MACpoint)

     MAC_point=odb.rootAssembly.nodeSets[MACpoint[nMAC]]

     MAC_point_displ=displacement.getSubset(region=MAC_point)


               for xyz in xrange(0,3):  #3=number of directions
in each point
```

```
                    Amm[i-
1][nMAC*3+xyz]=MAC_point_displ.values[0].data[xyz]


     Mm=numpy.array([0.0]*len(Mmm))[numpy.newaxis]
     Am=numpy.array([0.0]*len(Amm))[numpy.newaxis]
     MAC_cdr_top=Am.T*Mm
     MAC_cdr_bot=Am.T*Mm
     MAC_cdr=Am.T*Mm
     MAC_cdr_alt=Am*Mm.T


     for im in xrange(0,len(Mmm)):
          Mm=numpy.array(Mmm[im])[numpy.newaxis]


          for ia in xrange(0,len(Amm)):
               Am=numpy.array(Amm[ia])[numpy.newaxis]
               MAC_cdr_top[ia][im]=(Mm.dot(Am.T))**2

     MAC_cdr_bot[ia][im]=(Mm.dot(Mm.T))*(Am.dot(Am.T))

     MAC_cdr[ia][im]=MAC_cdr_top[ia][im]/MAC_cdr_bot[ia][im]
               MAC_cdr_alt[im][ia]=MAC_cdr[ia][im]


     AmNumber=[0]*5
     for i_Am in xrange(0,len(AmNumber)):
          AmNumber[i_Am]=MAC_cdr_alt[i_Am].argmax(axis=0)
     for i_w in range(0, len(AmNumber)):

     wa_i=(odb.steps[StepName].frames[AmNumber[i_w]+1].frequenc
y)
```

```
            wa[i_w]=wa_i


        for i_w in xrange(0, len(AmNumber)):


        wa_i=(odb.steps[StepName].frames[AmNumber[i_w]+1].frequenc
y)
            wa[i_w]=wa_i


        rt=numpy.array(numpy.array(wm)-
numpy.array(wa))[numpy.newaxis]
        r=rt.T
        r_max_abs=max(abs(x) for x in r)


print 'End'
```

B1-Drawing of the Bridge