NTNU
Norwegian University of
Science and Technology

Modal Analysis and Damage Identification of the Hell Bridge Test Arena

Edvart Braathen Granhaug
Leander Anton Guust Van de Pontseele

Civil and Environmental Engineering
Submission date: June 2018
Supervisor: Anders Rönnquist, KT
Co-supervisor: Bjørn Thomas Svendsen, KT

Norwegian University of Science and Technology
Department of Structural Engineering

Sammendrag

Hensikten med Hell Bridge Test Arena er å evaluere ny sensorteknologi, overvåkningsalgoritmer og konstruksjonsmonitoreringsteknikker på en konstruksjon i kontrollerte omgivelser for å etablere pålitelige, kostnadseffektive og tids-effektive inspeksjonsmetoder. Siden Hell Bridge Test Arena er et prosjekt forventet å vare i mange år er formålet for denne avhandlingen avgrenset til å vurdere den globale dynamiske oppførselen til Hell Bridge Test Arena gjennom numerisk modellering og måledata, og videre å utforske en kjent skade i knutepunktet mellom en kryssdrager og en bærebjelke gjennom dynamisk analyse i søket etter nye enkle teknikker for skadeidentifisering. Den kjente skaden vil bli inspisert gjennom lokale målinger av påførte vibrasjoner og videre implementert i de numeriske modellene.

Den numeriske modellering ble gjort i Abaqus. Det ble laget to elementmodeller, en enkel bjelkemodell og en mer detaljert skallmodel. Den globale dynamiske oppførselen ble vurdert ved å gjennomføre en egenverdianalyse av begge modellene som videre ble sammenlignet med resultatene fra de ulike systemidentifiseringsmetodene. Systemidentifiseringsmetodene som ble brukt var Peak Picking og Frequency Domain Decomposition. De numeriske modellene ble videre brukt til å introdusere den kjente skaden i knutepunktet mellom kryssdrageren og bærebjelken på to forskjellige måter. Tilslutt ble simuleringer med påførte vibrasjoner utført og sammenlignet med måledata gjennom de etablerte skadeidentifiseringsteknikkene.

Resultatene viser at en bjelkemodell er nøyaktig nok for formålet til denne avhandlingen og at skaden er godt representert ved å la bærebjelken bevege seg fritt vertikalt og rotere fritt om den lokale transversale bjelkeaksen. Videre viser resultatene at skadeidentifisering er mulig ved å bruke de enkle teknikkene Moving Standard Deviation, Cross Correlation og Normalization, alle i tidsdomenet.

Abstract

The purpose of the Hell Bridge test arena is to assess new sensor technology, monitoring algorithms and SHM techniques on a real structure in a controlled environment to establish reliable, cost-effective and time efficient inspection methodologies. As the Hell Bridge test arena is a project expected to last for many years, the purpose of this paper is narrowed down to assessing the global dynamic behavior of the Hell Bridge Test Arena using numerical modelling and full-scale measurements, and further to investigate a known damage in the joint connection between a stringer and cross girder by dynamic analysis in the search for new simple techniques for damage identification. The known damage was investigated by local measurements of imposed vibrations and further implemented in the numerical models.

The numerical modelling was performed in Abaqus. Two FE models were constructed, one simple beam model and one more detailed shell model. The global dynamic behavior was assessed by performing an eigenvalue analysis of the two models, which were compared to the results from the system identification methods. The system identification methods used were Peak Picking and Frequency Domain Decomposition. The numerical models were further used to introduce the known damage in the connection between stringer and cross girder in two different ways. Finally, numerical simulations of imposed vibrations were performed and compared to the full-scale measurements through the established damage identification techniques.

The results show that a beam model is sufficiently accurate for the purpose of this paper and that the damage is well represented by allowing the stringer of the damaged structural component to move free vertically and rotate free about the local transversal beam axis. Further the results show that damage identification is possible by the use of the simple techniques Moving Standard Deviation, Cross Correlation and Normalization, all techniques in the time domain.

Contents

1 Introduction

The railway system is an important part of the infrastructure in todays society. In Norway we have approximately 1000 steel railway bridges, from which most were constructed before 1960 [1]. Many of these bridges have considerable damages that should be investigated closer, but with todays methods and technology this requires extensive equipment, resulting in major costs and traffic downtown to cover all inspections. As this is a global problem, not only for railway bridges, the field of structural health monitoring (SHM) has grown rapidly throughout the last decade, but there is still big room for improvement [2]. This has lead to the motivation of the Hell Bridge Test arena where the ultimate goal is to assess new sensor technology, monitoring algorithms and SHM techniques on a real structure in a controlled environment to establish reliable, cost-effective and time efficient inspection methodologies.

For the Hell Bridge test arena project, one module of the retired steel riveted Hell Bridge has been donated to perform full-scale research on, see figure 1.1. The module has been lifted on shore and given new fundaments. From visual exceptions, there has been observed possible serious errors in two of four stringers at mid span. Unfortunately, the relevant structural components are located below the bridge deck, making them hard to access. To access the state of health of these structural components, this has lead to the motivation for investigation of new, simple, techniques for damage identification from imposed vibrations. As the first step in a general approach to develop damage detection techniques is to find indication of damage as a basis for determining further inspection, this is in good accordance with the ultimate goal of the project.

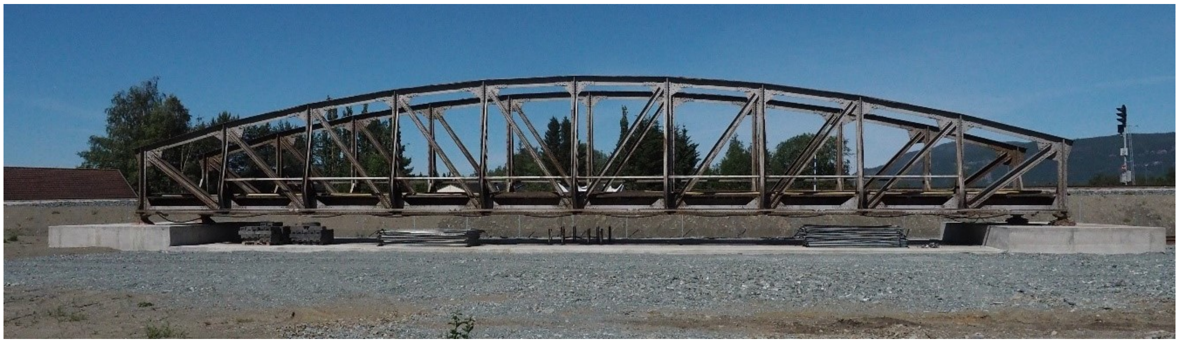


Figure 1.1: Hell Bridge testarena

Since the Hell Bridge test arena is a project planned to last many years, the purpose of this paper is narrowed down to assessing the global dynamic behavior of the Hell Bridge Test Arena using numerical modelling and full-scale measurements, and further to investigate a known damage in the joint connection between stringer and cross girder by dynamic analysis in the search for simple methods for damage identification.

Two different numerical models of the module, with different levels of details, will be established in Abaqus. The first one will be a simple beam model and the second a more detailed shell model. Verification of the numerical models is done by comparing the results from the eigenvalue analysis to the natural frequencies found from the system identification methods. The experimental data, in the form of acceleration response time series, will be induced by the use of a modal hammer with the purpose of investigating the known damages in the bridge deck. Further the observed damage in the connection joint between stringer and cross girder of the bridge will be implemented in the numerical models in different ways. By modelling the damage in the numerical model, we can evaluate its impact on the global response of the structure, and hopefully quantify the severity of them. Evaluation of the different ways of implementing the damage in the numerical models will be done by comparison of the damage identification techniques applied on full scale measurements and numerical simulations.

This thesis is divided into 9 chapters and 4 appendices. In the following a short presentation of each chapter is given.

Chapter 2 gives a theoretical introduction to the theory relevant for the work done in this thesis. The areas touched are structural dynamics, damping, modal analysis, and structural health monitoring.

Chapter 3 gives an introduction to the load carrying system of the Hell Bridge and the instrument setups used for operational and experimental data collection.

Chapter 4 presents the natural frequencies extracted with the different system identification methods.

Chapter 5 presents the two FE models constructed with assumptions and simplifications made.

Chapter 6 presents the application and evaluation of the different damage identification techniques.

Chapter 7 gives a thorough introduction to the damaged structural component, how it's implemented numerically and the results of the damage identification techniques applied on numerically simulated acceleration response. The chapter ends with a discussion of the results.

Chapter 8 presents the conclusion made by the authors, based on the work presented in previous chapters.

Chapter 9 proposes further work.

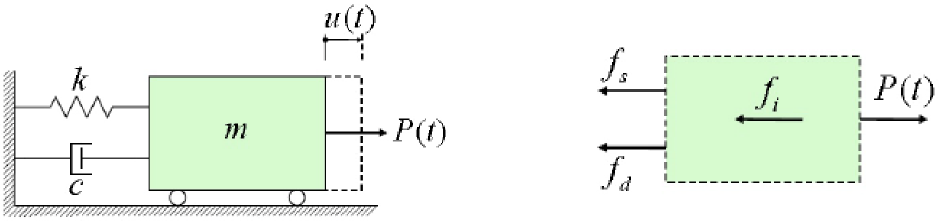
2 Theory

2.1 Structural dynamics

Structural dynamics are usually concerned with the vibration of structural members induced by forces that vary with time, resulting in a time dependent response of the structure. All structures have their own unique natural frequencies and if subjected to a load with close to, or the same, natural frequency as one of the structures natural frequencies, resonance may occur creating large response amplitudes relative to the static response. To consider the inherent dynamic properties of a structure, knowledge about the different aspects of vibrations is required and is presented in the two following under sections.

2.1.1 Single degree of freedom systems

The simplest model of a vibrating mechanism is that of a single degree of freedom (SDOF) system [3].



(a) General SDOF system.

(b) Free body diagram of the SDOF system.

Figure 2.1: The figure shows a SDOF system and it´s free body diagram [3].

The system in figure 2.1 consist of a mass, m, a viscous damper, c, and a linear spring, k, and is subjected to an applied force $P(t)$ resulting in the horizontal translation described by the single degree of freedom, $u = u(t0)$.

Free oscillations
If we regard the SDOF system after is has been subjected to an externally applied load but still is in motion, i.e. monitor it's time dependent response for $P(t) = 0$, the system will experience free vibration. The forces acting on the body are then:

$$\begin{aligned} f_t &= m\ddot{u} && \text{Inertial force} \\ f_d &= c\dot{u} && \text{Damping force} \\ f_s &= ku && \text{Spring force} \end{aligned} \tag{2.1}$$

The equation of motion for the system can then be represented by the following second order differential equation:

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = 0 \tag{2.2}$$

The natural circular frequency $[rad/s]$ of the system is given as:

$$\omega_n = \sqrt{\frac{k}{m}} = 2\pi f_n \tag{2.3}$$

Where f_n is the natural frequency in Hertz. The damped natural frequency is relevant when the system contains damping ($c \neq 0$):

$$\omega_d = \omega_n \sqrt{1 - \xi^2} \tag{2.4}$$

The response will decay exponentially and approach zero with damping present. In the equation above ξ is the damping ratio, which is the ratio between damping, c, and the critical damping $c_{cr} = 2m\omega_n$ [6]:

$$\xi = \frac{c}{c_{cr}} \tag{2.5}$$

As shown in figure 2.2, the obtained response is highly affected by the damping ratio. For < 1, we say that the system is underdamped and we have no oscillatory behavior. For $\xi = 1$, we say that the system is critically damped, also here we have no oscillatory behavior and critical damping is what returns the system back to its static equilibrium in the shortest time. Finally, for $\xi < 1$, we say that the system is underdamped and the response becomes oscillatory.

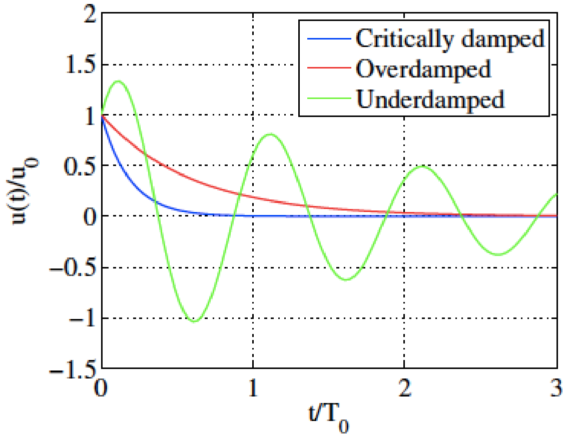


Figure 2.2: Shows the free vibration of an underdamped, a critically damped and an overdamped system

The damping ratio ξ of a system can be determined by measurements or by an educated guess. Over the years, many measurements have been made on a wide range of structures, so that a rational approach would be to utilize data collected for a similar structure to the one being analyzed. For most engineering structures where no special devices are added for damping, a damping ratio higher than 15 percent, i.e. $\xi = 0.15$, is rarely observed [4, p. 454].

Forced oscillations

If we now regard the system while still under loading, i.e. $P(t) \neq 0$, the equation of motion becomes:

$$m\ddot{u}(t) + c\dot{u}t + ku(1) = P(t) \tag{2.6}$$

The frequency ratio is given by the ratio between the frequency of the applied load and the natural frequency of the system:

$$\beta = \frac{\omega}{\omega_n} \tag{2.7}$$

In figure 2.3, the Dynamic Magnitude Factor (DMF) against the frequency ratio β for a forced SDOF system is shown. DMF is the ratio between the dynamic response amplitude and the corresponding static amplitude that would be obtained with a static load ($\omega = 0$). Figure 2.3 shows that when $\beta \to 1$, i.e. when the frequency of the externally applied load approaches the systems natural frequency, large response amplitudes occur. This phenomenon is what we call resonance, and without sufficient damping continued forcing at the resonance frequency may lead to unstable excitation, increasing unconditionally [6].

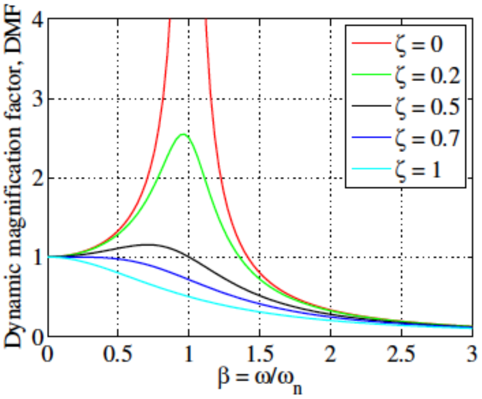


Figure 2.3: Shows the resonance phenomenon by a plot of Dynamic Magnification Factor versus frequency ratio for different damping ratios.

2.1.2 Multi degree of freedom systems

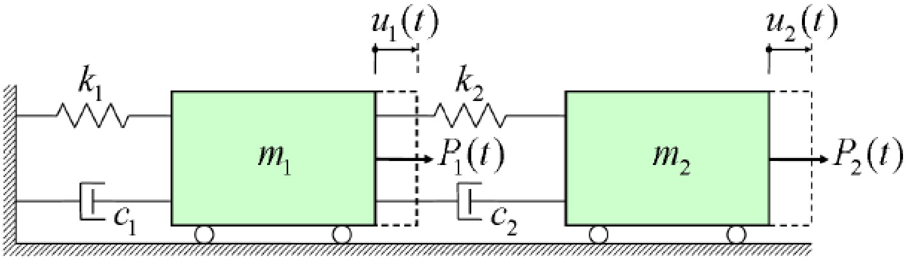


Figure 2.4: Shows a schematic drawing of a multi degree of freedom system [3].

Even with considerable simplifications, real structures are rarely simple enough to be modelled as a single degree of freedom system. This is solved by introducing the multi degree of freedom (MDOF) system, which results in more complex motions of the system with n coupled equations of motion, n being the number of degrees of freedom. The uncoupling of the coupled equations of motion by modal techniques is presented later in this chapter. What the uncoupling does is provide the opportunity to study the system as a set of n simple oscillators, each one describing a characteristic vibration of the system.

First the establishment of MDOF systems is presented with basis in the two degree of freedom system shown in figure 2.4. By establishing a free body diagram for the two masses, all forces acting on them can be accounted for:

$$\begin{aligned} &\left.\begin{aligned} f_{t1} &= m_1\ddot{u}_1 \\ f_{t2} &= m_2\ddot{u}_2 \end{aligned}\right\} \quad Inertial forces \\ &\left.\begin{aligned} f_{d1} &= c_1\dot{u}_1 \\ f_{d2} &= c_2(\dot{u}_2 - \dot{u}_1) \end{aligned}\right\} \quad Damping forces \\ &\left.\begin{aligned} f_{s1} &= k_1 u_1 \\ f_s2 &= k_2(u_2 - u_1) \end{aligned}\right\} \quad Spring forces \end{aligned} \tag{2.8}$$

These equations can be written on matrix form: $\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}$

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \end{Bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} P_1(t) \\ P_2(t) \end{Bmatrix} \tag{2.9}$$

which can be written more compactly as:

$$[\mathbf{M}][\ddot{u}] + [\mathbf{C}][\dot{u}] + [\mathbf{K}][u] = [\mathbf{P}(t)] \tag{2.10}$$

where $[\mathbf{M}]$ is the mass matrix, $[\mathbf{C}]$, is the damping matrix, $[\mathbf{K}]$ is the stiffness matrix, $\mathbf{P}$(t) is the vector of forcing functions and $\mathbf{u}$ is the displacement vector. With larger systems, the most common way to solve the matrix equation in equation (2.10) is by use of the finite element method, in which a real structure with infinitely many DOFs is discretised into a finite number of elements which are interconnected at a limited number of nodes which have a finite number of DOFs. Next, the mass matrix, damping matrix, stiffness matrix and force vector have to be established to get the equation of motion. At last, the time varying response is approximated by solving the equation of motion. Several methods exist for solving this second order differential equation, from which modal decomposition is one. Modal decomposition is discussed below, but first, some brief theory about natural frequencies and mode shapes is presented.

2.1.3 Natural frequencies and mode shapes

The natural frequencies of a structure are the frequencies at which a system tends to oscillate in the absence of any driving or damping force. The oscillatory mode shapes are the characteristic deformed shapes of the structure when oscillating at the natural frequencies. An example of the first three natural frequencies and corresponding mode shapes of a simply supported beam is shown in figure 2.5.

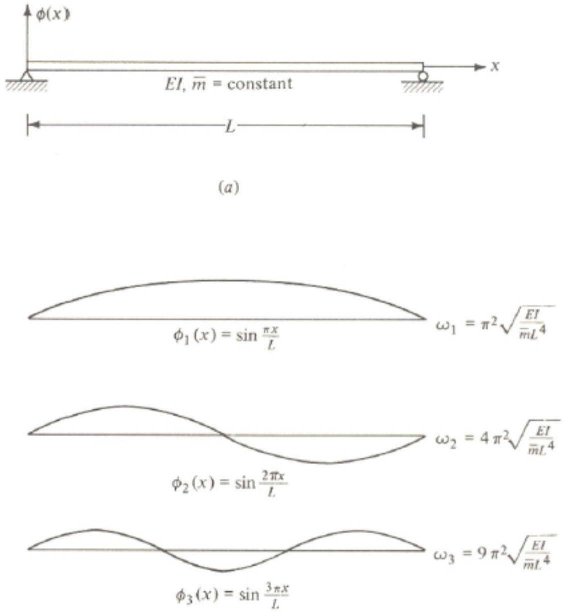

$\phi(x)$
x
$EI, \bar{m}$ = constant
L
(a)
$\phi_1(x) = \sin\frac{\pi x}{L}$
$\omega_1 = \pi^2\sqrt{\frac{EI}{\bar{m}L^4}}$
$\phi_2(x) = \sin\frac{2\pi x}{L}$
$\omega_2 = 4\pi^2\sqrt{\frac{EI}{\bar{m}L^4}}$
$\phi_3(x) = \sin\frac{3\pi x}{L}$
$\omega_3 = 9\pi^2\sqrt{\frac{EI}{\bar{m}L^4}}$

Figure 2.5: Shows the first three mode shapes of a simply supported beam.

When studying the dynamic response of a structure, its modes of oscillation are extremely important because of the previously mentioned resonance phenomenon. The different natural frequencies and mode shapes, also referred to as eigenfrequencies and eigenmodes respectively, of a system can be found by applying modal analysis. First, the natural frequencies of the system are calculated from the following equation:

$$([\mathbf{K}] - \omega_n^2[\mathbf{M}])[\mathbf{\Phi}] = 0 \tag{2.11}$$

Where $[\Phi]$ contains the eigenvectors of the system, containing its eigenmodes. As an eigenvector equal to zero is a trivial solution and therefore not interesting, equation (2.11) is satisfied by requiring the determinant of the dynamic stiffness matrix, $([\mathbf{K}] - \omega_n^2[\mathbf{M}])$, to be equal to zero. For every degree of freedom there will be a solution with a corresponding eigenmode. By calculating the dynamic stiffness matrix with the obtained natural frequencies and further solving for the eigenvector, the eigenmodes can be found. Since the dynamic stiffness matrix is singular when calculating the eigenmodes, one of the entries in the eigenvector has to be set equal to unity to determine the other entries. The eigenmodes therefore only show relative displacement.

2.1.4 Modal decomposition

When using modal decomposition to solve the equations of a MDOF system the mode shapes obtained from a modal analysis are used to decouple the set of differential equations into a set of linearly independent differential equations of SDOF oscillators [5]. This is done by presenting the displacement vector $\mathbf{u}(t)$ as a linear combination of the mode shape vectors, ϕ_n. The coefficients of the linear combination vary with time and are generalized displacement $q_i(t), i = 1, 2, ..., n$:

$$u(t) = \phi_1 q_1(t) + \phi_2 q_2(t) + .. + \phi_n q_n(t) = [\Phi]\mathbf{q}(t) \tag{2.12}$$

The modal mass, damping and stiffness matrices are obtained by inserting equation (2.12) into equation (2.10) and pre-multiplying with the transpose of the mode shape matrix:

$$[\mathbf{\Phi}]^{\mathbf{T}}[\mathbf{M}][\mathbf{\Phi}]\{\ddot{\mathbf{q}}\} + [\mathbf{\Phi}]^{\mathbf{T}}[\mathbf{C}][\mathbf{\Phi}]\{\dot{\mathbf{q}}\} + [\mathbf{\Phi}]^{\mathbf{t}}[\mathbf{K}][\mathbf{\Phi}]\{\mathbf{q}\} = [\mathbf{\Phi}]^{\mathbf{T}}\{\mathbf{P}(\mathbf{t})\} \tag{2.13}$$

often more compactly written as:

$$[\mathbf{M}^*]\{q\} + [\mathbf{C}^*]\{\dot{q}\} + [\mathbf{K}^*]\{q\} = \{\mathbf{P}^*(t)\} \tag{2.14}$$

Because the mode shape matrix is orthogonal with respect to both the mass and stiffness matrices of the system, all modal matrices (matrices with the superscript * in equation (2.14))are diagonal [6]. This yields n modal equations on the form:

$$\ddot{q}_n + 2\xi_n\omega_n\dot{q}_n(t) + \omega_n^2 q_n(t) = \frac{P_n^*(t)}{M_n^*} \tag{2.15}$$

Here $\ddot{q}_n(t)$, $\dot{q}_n(t)$ and $q_n(t)$ is generalized acceleration, velocity and displacement respectively, ξ_n is the damping ratio, ω_n is the natural circular frequency, $P_n^*(t)$ is the modal force and M_n^* is the modal mass, all for the n^{th} mode of vibration. With knowledge about the external forces the modal displacements $q_n(t)$ can be determined by solving equation (2.15) as an ordinary 2^{nd} order differential equation. At last, the physical displacements can be calculated by the relation in equation (2.12).

2.2 Damping

Damping is a desirable property of structures which reduces the dynamic response, i.e. brings a structure back to its state of equilibrium after its subjected to dynamic loads. The physical causes of damping are very complex. Shortly explained we can say that during vibration, potential energy turns into kinetic energy, and vice versa, and part of the energy is lost by plastic material deformations and/or is changed to other forms of energy such as acoustic and thermal energy. In this way, the energy supplied by a dynamic load is irreversibly dissipated into the environment, and this energy dissipation is what we call damping [7, chapter 5].

Extensive research of damping in bridges has been performed by a number of researchers, even damping of railway bridges in particular. *In Dynamics of Railway Bridges* [7, chapter 5], Frýba discusses the difference between viscous, hysteretic and Coulomb damping models, where he concludes that due to the relatively low damping values associated with railway bridges, the practical difference between the models is negligible. Further the author recognizes the possible applications of non-viscous damping models in dynamic simulations of railway bridges, but as there is a need to limit the scope and extent of this thesis and that the viscous damping model is relatively easy implemented in the finite element code, the choice is made to only look further into viscous damping models, more specific Rayleigh damping.

2.2.1 Viscous damping

Once again consider the equation of motion for an elastic multi degree of freedom (MDOF) system with linear viscous damping as below:

$$[\mathbf{M}]\{\ \ddot{\mathbf{u}}\}(t) + [\mathbf{C}]\{\dot{\mathbf{u}}\}(t) + [\mathbf{K}]\{\mathbf{u}\}(t) = \mathbf{0} \tag{2.16}$$

When assuming a viscous and frequency dependent damping, the most popular method to solve the equation of motion is using modal analysis. When solving the equation of motion by modal analysis, the equation of motion is first solved for zero damping and damping values are directly assigned to each mode afterwards. Damping ratios can be assigned using the Caughey serie, of which Rayleigh damping is a special case. Rayleigh damping is widely available in commercial codes and is often applied in dynamic analysis due to its simplicity and characteristics.

Classical Rayleigh damping expresses the system damping matrix, **C**, as a linear combination of the mass and stiffness matrices;

$$[\mathbf{C}] = a_0[\mathbf{M}] + a_1[\mathbf{K}] \tag{2.17}$$

where a_0 is the mass proportional damping coefficient and a_1 is the stiffness proportional damping coefficient, both real scalars with $1/sec$ and sec units respectively. The damping ratio for Rayleigh damping is given by equation (2.18) and a_0 and a_1 are given by equation (2.19) and (2.20).

$$\xi_n = \frac{a_0}{2\omega_n} + \frac{a_1\omega_n}{2} \tag{2.18}$$

$$a_0 = 2\omega_m\omega_n\frac{\omega_m\xi_n - \omega_n\xi_m}{\omega_m^2 - \omega_n^2} \tag{2.19}$$

$$a_1 = 2\frac{\omega_m\xi_m - \omega_n\xi_n}{\omega_m^2 - \omega_n^2} \tag{2.20}$$

Equation (2.18) and figure 2.6 show that for mass proportional damping, the damping ratio is inversely proportional to the frequency while for stiffness proportional damping it is directly proportional with the frequency. In this regard it is important to note that the dynamic response of a structure generally includes contribution from all N modes even though only a limited number of modes are included in the uncoupled equations of motion. Thus, only using mass proportional or stiffness proportional damping for an MDOF system in which the frequencies of the significant modes span a wide range is not suitable as the relative amplitude of the different modes will be seriously distorted by inappropriate damping ratios [8]. This is why we combine the mass and stiffness proportional damping, which is called Rayleigh Damping, after Lord Rayleigh who first suggested it.

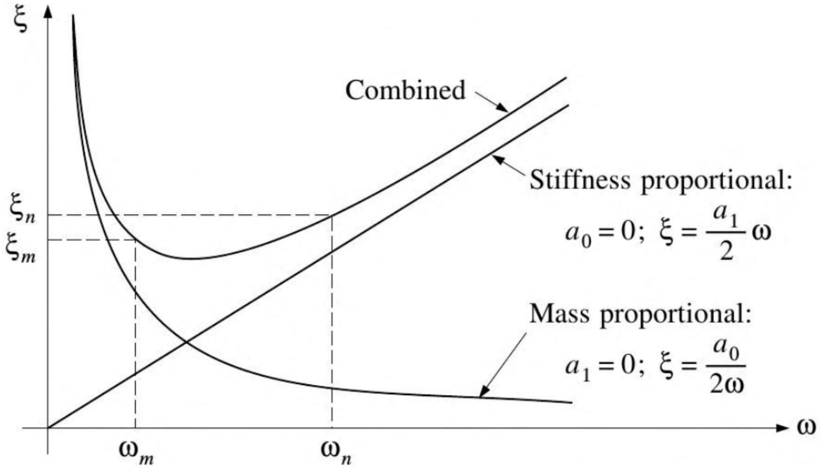


Figure 2.6: Shows the relationship between damping ratio and frequency for Rayleigh damping [8].

Because detailed information about the variation of damping ratio with frequency rarely is available, the same damping ratio is normally applied to both control frequencies, i.e. $\xi_m = \xi_n = \xi$. For this case, equation (2.19) and (2.20) can be simplified to equation (2.21) and (2.22) respectively.

$$a_0 = \frac{2\xi\omega_m\omega_n}{\omega_m + \omega_n} \tag{2.21}$$

$$a_1 = \frac{2}{\omega_m + \omega_n} \tag{2.22}$$

When applying Rayleigh damping in practice it is generally recommended that ω_m takes the value of the fundamental frequency of the MDOF system, i.e. the first natural frequency, and that ω_n takes the value of one of the higher frequencies of the modes that contribute significantly to the dynamic response [8, chapter 12]. This approach ensures that the desired damping ratio is obtained for these two modes and that the modes with frequencies between the two specified frequencies will have somewhat lower damping ratios. Further, all modes with frequencies lower than ω_m and higher than ω_n will have damping ratios that increase when moving further away from the chosen frequencies, see the combined curve in figure 2.6. The end result of this approach is that the response of very high frequency modes are efficiently eliminated by their high damping ratios.

2.3 Modal analysis

In general, modal analysis is the task of estimating the modal parameters; natural frequencies, damping and mode shapes of a structure. The modal parameters will depend on geometry, material properties and boundary conditions, and they can be extracted by the use of operational modal analysis (OMA) or experimental modal analysis (EMA).

The use of experimental tests to gain knowledge about the dynamic response of civil engineering structures is a well-established practice which can be dated back to the 20^{th} century [9]. Assuming that the dynamic behaviour of a structure can be described as a combination of modes, each one characterized by the modal parameters mentioned above, EMA identifies those parameters from measurements of the applied force and the vibration response. EMA has been applied in many fields such as automotive engineering, aerospace engineering, industrial machinery and civil engineering. Due to the large size and low frequency range of most civil engineering structures, applying controlled and measurable excitation on these structures becomes a challenging and complex procedure which requires expensive and heavy devices. This is the main reason that the community of civil engineers more recently has focused on the opportunities provided by OMA [9].

OMA can be defined as the modal testing procedure that allows the experimental estimation of the modal parameters of a structure from measurements of the vibration response only. Since OMA requires only measurements of the dynamic response under operational conditions it is also known under other names such as ambient vibration modal identification or output-only modal analysis. The idea behind OMA is to replace the artificial excitation used in EMA by the natural and freely available excitation due to ambient forces and operational loads, such as wind and traffic loads [9]. With OMA, we then only need devices to measure and store the excitation history, which previously was regarded as disturbance. This approach then becomes more economical, both in terms of structural down time and use of less equipment.

Although OMA is very applicable for most cases, there are some drawbacks; The low amplitude of vibrations in operational conditions requires very sensitive, low-noise sensors and a high performance measurement chain [9]. Also, since the engineer has no control on the applied excitation, the identification of closely spaced modes can become troublesome.

In this paper two techniques, both in the frequency domain, were implemented to extract the modal parameters of the Hell Bridge. The chosen methods were Peak Picking (PP) and Frequency Domain Decomposition (FDD). Before giving a theoretical introduction to these two methods, some basic theory about mathematical tools for random data analysis has to be established.

2.3.1 Mathematical tools for random data analysis

All theory presented in this section is taken from *Operational modal analysis of civil engineering structures* [9].

2.3.1.1 Fourier series and Fourier transform

The Fourier transform is a fundamental tool in signal analysis because of its important properties giving a simple relationship between time and frequency domain. The relationship provides the opportunity to transform complex calculations into simple multiplications.

The idea behind the Fourier analysis is that any signal can be written in terms of a linear combination of sinusoidal functions at different frequencies. This can be understood by accounting for the relation between sinusoidal functions and complex exponentials and that both are orthogonal functions, i.e. they fulfill the following general conditions ((2.23) and (2.24))

$$\int_a^b f_u(t)f_v^*(t)dt = 0, \qquad u \neq v \tag{2.23}$$

$$\int_a^b f_u(t)f_v^*(t)dt \neq 0 < \infty, \qquad u = v \tag{2.24}$$

Where f_u and f_v are complex functions and the superscript $*$ means complex conjugate. This decomposition, originally developed for periodic functions, can be extended to nonperiodic functions, such as random signals, by assuming that they are periodic functions with period equal to the duration T of the signal. For a nonperiodic signal $x(t)$ the Fourier transform (2.25) and the inverse Fourier transform (2.26) are given by:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft}dt \tag{2.25}$$

$$x(t) = \int_{-\infty}^{+\infty} X(f)e^{i2\pi ft} \tag{2.26}$$

From (2.25), by the relationship between complex exponentials and sinusoidal functions, it is shown that any signal $x(t)$ can be decomposed in a sum of sinusoidal functions. In practical applications, when the signal $x(t)$ is digitally recorded and analyzed, it is recorded with a constant sampling frequency, i.e. the time increment between each sampling point is constant. As a consequence, only discrete time and frequency representations are considered, and the expression of the Fourier transform has to be changed accordingly. By assuming that the signal $x(t)$ has been sampled at N equally spaced time instants and that the sampling frequency has been properly selected, the Discrete Fourier Transform (DFT) can be derived, given by:

$$X_k = \sum_{n-0}^{N-1} \chi_n e^{\frac{-i2\pi kn}{N}} \qquad k = 0, 1, 2, ..., N-1 \tag{2.27}$$

Because the evaluation of the DFT requires N^2 operations, the Fast Fourier Transform (FFT) has been developed with the goal to reduce the number of operations. Given that the number of data points equals a power of two, the FFT reduces the number of operations to $N * log_2 N$.

2.3.1.2 Spectral Density Functions

Given a pair of sample records $x_k(t)$ and $y_k(t)$ of finite duration T from a stationary random process, their Fourier transforms are:

$$X_k(f,T) = \int_0^T x_k(t)e^{-i2\pi ft}dt \tag{2.28}$$

$$Y_k(f,T) = \int_0^T y_k(t)e^{-i2\pi ft}dt \tag{2.29}$$

and the one-sided auto- and cross-spectral density functions, which are more commonly used than the two sided in applications, are given by:

$$G_{xx}(f) = 2\lim_{T\to\infty}\frac{1}{T}E[|X_k(f,T|^2] \qquad 0 < f < +\infty \tag{2.30}$$

$$G_{yy}(f) = 2\lim_{T\to\infty}\frac{1}{T}E[|Y_k(f,T)|^2] \qquad 0 < f < +\infty \tag{2.31}$$

$$G_{xy}(f) = 2\lim_{T\to\infty}\frac{1}{T}E[X_k^*(f,T)Y_k(f,T)] \qquad 0 < f < +\infty \tag{2.32}$$

Where E denotes the expected value. In practical applications, Welch procedure, based on the direct computation of the FFT of the records and the estimation of the power spectral densities (PSDs) in agreement with (2.30)-(2.32), is a common procedure applied to obtain the PSD estimates. The Welch procedure is computationally less demanding than other methods, but requires some operations on the signal in order to provide quality estimates. According to (2.30)-(2.32), the one-sided auto-spectral density function can be estimated by dividing a record into n_d contiguous segments, each of length $T = N^{*}\Delta T$, applying the Fourier transform to each segment and then computing the auto-spectral density by the following equation:

$$\hat{G}_{xx}(f) = \frac{2}{n_d N \Delta t} \sum_{i=1}^{n_d} |X_i(f)|^2 \tag{2.33}$$

The number of averages n_d, determines the random error of the estimates and the number of data points N in each segment determines the frequency resolution of the resulting estimates. N is often referred to as the block size for the computation of each FFT. By dividing the record into n_d contiguous segments, the estimates are now based on records if finite length which introduces error. To eliminate these errors, specific strategies are required.

Without going into details, the procedure of dividing the record into n_d segments allows the energy at a certain frequency to spread to nearby frequencies, causing large amplitude errors. This phenomenon is known as leakage and the effects of it may introduce significant distortions in the estimated spectra, in particular in the presence of data characterized by narrow bandwidth. However, leakage effects do not occur when the analyzed data is periodic with a period equal to the record length and this is used to produce a correct spectrum. Thus, in order to suppress the leakage problem, data are made periodic by tapering them by an appropriate time window, leading to the elimination of discontinuities at the beginning and end of the analyzed record. The most common choice of window is known as the Hanning window. The Hanning window minimizes the leakage effects, but the use of it to compute the PSD estimates implies a loss factor of 3/8, introducing the need for a rescaling to achieve the correct magnitude.

Another drawback of the application of the Hanning window is that the half power bandwidth increases, which may affect the damping estimates. The increase in half power bandwidth is avoided by increasing the length of each segment until each FFT provides the same bandwidth with tapering that would have occurred without it. For a given number of averages n_d, the increase in the length of the tapered segments implies an increase in the total record length. If the data is limited, an increase in the length of the tapered segments is possible at the expense of the number of averages n_d, but this introduces an increased variability in the PSD estimates. This problem can once again be solved by dividing the total record into partially overlapping segments.

2.3.2 Frequency domain techniques

In the following couple of sections, two possible methods for modal parameter identification in the frequency domain are presented.

2.3.2.1 Peak picking

The Peak picking method, also known as the Basic Frequency Domain (BFD) technique, is a very popular method for identifying the modal parameters of civil engineering structures subjected to ambient vibration loading, reason being its implementation simplicity and processing speed [10].

The method is based on the fact that the frequency response function (FRF) goes through extreme values around the structures natural frequencies. This results in distinct peaks in the FRF and the values at where these peaks occur is a good estimate for the natural frequencies of the system. In the output only-case, the input-force dependent FRF can't be determined, as we don't know the input, and is simply replaced by the power spectral density function. From the plots of the power spectral density estimates the natural frequencies are determined by manually picking the peaks.

When estimating the mode shapes, we use a single mode approximation close to the natural frequency, meaning that we assume that the dynamic response at resonance is only dominated by one mode. If we have modes with close natural frequencies this will be a bad assumption and probably yield poor results for the mode shapes [10]. Choosing one of the accelerometer locations as reference point, $\varphi(x1)$ in equation (2.34), the modal value at any given point compared to the modal value of the reference point can be identified by the relationship in equation (2.34). The sign of the modal value at a given point is determined by the sign of equation (2.35), where $G_{r_1 r_i}$ cross spectral density between the reference point ant the point we are looking at, r_1.

$$\frac{G_{r_i r_i}(\omega_k)}{G_{r_1 r_1}(\omega_k)} = \frac{\varphi(xi)^2}{\varphi(x1)^2} \tag{2.34}$$

$$Re(G_{r_1 r_i}(\omega_k) \tag{2.35}$$

As the modal amplitude of reference appears in the denominator, it should not be zero. This is done by avoiding the location of the reference point close to a node of a mode that is important to identify [11].

In principle, the PP method should only be applied to evaluate the natural frequencies and mode shapes of a structure. The half-power bandwidth method is sometimes applied to get damping estimates from the spectra, but a number of studies have shown that the accuracy of this method is poor [10][].

Although peak picking is a popular choice for identifying modal parameters there are some relevant theoretical drawbacks for this method:

- Picking the peaks is a subjective task, meaning that the results obtained from the same data may depend on the experience of the user, especially if the peaks are not distinct [10].
- We don't obtain real mode shapes but operational deflection shapes. For cases with low damping and well separated peaks this is no major drawback, as the operational deflection shapes are very similar to the actual mode shapes for these cases [10][9].

2.3.2.2 Frequency domain decomposition

The frequency domain decomposition (FDD) method is an extension of the peak picking method. The goal with the method is to overcome the limitations of the PP method related to separating closely spaced modes and providing reliable damping estimates (the latter not relevant for this paper) [9].

The first step of the FDD method is constructing the spectrum matrix of each set up, with the number of lines equal to the number of measurement points in each set up and with as many columns as the number of reference points chosen. Each column in the PSD matrix contains the cross spectra between the structural response at all the measured points and the corresponding response of the reference point [11]. It can be shown that under the assumption of white noise excitation, low damping and orthogonal mode shapes for close modes, the singular value of the spectrum matrix, in the vicinity of each resonant frequency, are auto spectral density functions of a single degree of freedom (SDOF) system with the same frequency and damping as the different structure vibration modes. Therefore, the spectral matrix

($\mathbf{G_{yy}}$) is decomposed, at each frequency (ω_i), in singular values and vectors using the singular value decomposition (SVD) algorithm, described in equation (2.36).

$$\mathbf{G_{yy}}(\omega) = \mathbf{VSV^H} \tag{2.36}$$

Here, $\mathbf{S}$ is the singular value diagonal matrix, $\mathbf{V}$ is the orthogonal matrix of the singular vectors and $(\bullet)^{\mathbf{H}}$ refers to the transpose conjugate. The singular vectors, meaning the columns in $\mathbf{V}$, are orthogonal to each other [12].

A way of understanding the response signals $u(t)$ is from their decomposition into participations from the different modes Φ_i expressed via the modal coordinate $\mathbf{q}(t)$ (presented in section 2.1.4). Using equation (2.36) in the expression of the correlation matrix, $\mathbf{C_{yy}}(\tau)$, of the response we get:

$$\mathbf{C_{yy}}(\tau) = \mathbf{\Phi C_{qq}}(\tau)\mathbf{\Phi^H} \tag{2.37}$$

And further applying the Fourier transform in equation (2.37) gives:

$$\mathbf{G_{yy}}(\omega) = \mathbf{\Phi G_{qq}}(\omega)\mathbf{\Phi^H} \tag{2.38}$$

Where $\mathbf{G_{qq}}(\omega)$ is the spectrum matrix of the modal coordinates. We observe that equation (2.36) and equation (2.38) have the same form and it can be understood that the singular vectors present estimations of the mode shapes and the corresponding singular values present the response of each of the modes expressed by the spectrum of each modal coordinate. The assumptions are that $\mathbf{G_{qq}}(\omega)$ is a diagonal matrix, meaning that the modal coordinates are uncorrelated, and that the mode shapes, the columns in $\mathbf{\Phi}$, are orthogonal [12].

2.4 Structural Health Monitoring

Structural Health Monitoring (SHM) is the process of accessing the state of health of a structure and predicting its remaining life time [2]. To successfully develop and implement SHM, a good understanding of diverse disciplines such as sensor technology, material technology, modeling aspects and computing technology is necessary. In this paper the focus is on computational techniques that support the SHM process, but a thorough understanding of the whole process is still needed in order to achieve good results.

The general motivation for SHM is that all man-made structures have finite life spans and the degrading process starts as soon as they are put into service. Processes such ass corrosion, fatigue, erosion, wear and overloads degrade these structures until they are no longer fit for their intended use. Depending on a structures value, cost of repair and consequence of failure, a number of actions can be taken;

- Wait until failure and dispose the structure
- Wait until failure and repair the structure
- Examine it periodically and determine whether or not maintenance is needed

Bridges and other structures where the consequences of failure are critical, with human lives at risk, fall into the latter category. All structures that fall under this category have regular inspections performed by skilled engineers to assess the health of the structures and systems [2]. These inspections are expensive, faults are normally not found, but still, in lack of better ways to do them, they are very important. Unfortunately, they are also subject to human error, meaning that some unnecessary maintenance is performed and some faults are not discovered.

SHM has the objective of providing the tools for constant or periodic monitoring of critical structural components, in order to determine the need for remedial action and to prevent catastrophic failures. SHM potentially offers increased safety, since faults are addressed before they reach critical levels, avoids human error and reduces ownership costs by both removing expensive manual inspections and performing maintenance when its needed, instead of pre-planned precautionary maintenance. Because of the potential benefits of implementing SHM are huge, a great amount of research is in progress worldwide into developing and improving SHM methods [2].

Benefits of SHM include:

- Optimal use of the structure by minimizing downtime and avoiding catastrophic failure
- Replaces periodic and scheduled maintenance inspection with performance-based (or condition-based) maintenance or at least reduces the present maintenance labor
- Improves safety and reliability

A SHM system consists of both hardware and software elements. For this paper, the hardware elements are the sensors (accelerometers) and the associated instrumentation, while the software components consist of the damage modelling (in FE models) and damage detection procedures. We can broadly divide SHM into five levels:

- Confirming the presence of damage
- Determination of location and orientation of the damage
- Evaluation of the severity of the damage
- Possibility of controlling or delaying the growth of damage
- Determining the remaining life in the structure

SHM can be performed in a short-term or long-term perspective, depending on the application. Short-term SHM is damage detection based on information obtained from the structure through inspection, i.e. not removing manual labor. Long-term SHM obtains information from the structure, normally in service, over a long period of time, typically months or years. For this paper, short-term SHM is applied, so the determination of location is done manually by visual inspection and possible ways to confirm the presence of damage is further investigated. The remaining three levels of SHM are not in the scope of this paper.

2.4.1 Damage identification

The first step in a general approach to find new inspection methods for damage detection is to find indication of damage as a basis for determining further inspection. The basic principle of damage detection algorithms is that the structural characteristics parameters are functions of the physical properties of a structural system, consisting of mass, damping and stiffness. Hence, damage in those physical properties will cause change in the structural characteristics parameters. In other words, damage will somehow change the stiffness, mass and energy dissipation of the structure, which in turn will change the global dynamic properties.

For this study, it is believed that this change of stiffness can be reflected by analyzing time series obtained from imposed vibrations. In the following sections some brief theory about the four analysis techniques investigated in this paper is presented. The theory presented is thoroughly covered in a great number of books and articles, but a general reference is made to the books *Random vibrations: Theory and practice* [13], *Operational modal analysis of civil engineering structures* [9] and *Modal analysis* [14].

2.4.1.1 Time domain

2.4.1.1.1 Moving standard deviation

Moving standard deviation provides å measure of dispersion, in other words; a quantification of the variation in a given data set. In the case of considering a data set of acceleration history, a, made up of N scalar observations, the standard deviation is defined as:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{n} |a_i - \mu|^2} \tag{2.39}$$

Where μ is defined as the mean of a:

$$\mu = \frac{1}{N}\sum i = 1^N a_i \tag{2.40}$$

In the analysis performed, the unbiased normalization factor $N-1$ is used instead of N for the sample standard deviation. For the results presented in this paper, the choice of window length was set to 21 data points, i.e. the standard deviation values for each point calculated include 21 elements. This choice is discussed and accounted for in section 6.1.1.

2.4.1.1.2 Cross correlation

In signal processing, cross correlation is a measure of similarity between two waveforms as a function of a delay (time-lag) applied to one of them. Consider two sampled waveforms $x(i)$ and $y(i)$, where $i = 0, 1, 2 \ldots N-1$. The cross correlation, r, for all delays $d = 0, 1, 2 \ldots N-1$ is defined as:

$$r(d) = \frac{\sum_i [(x(i)-\mu_x)(y(i-d)-\mu_y)]}{\sqrt{\sum_i (x(i)-\mu_x)^2}\sqrt{\sum_i (y(i-d)-\mu_y)^2}} \tag{2.41}$$

Where μ_x and μ_y are the mean values of the x and y waveforms, as defined in equation (2.40). In this study, the cross correlation is used to measure the similarity of two stationary acceleration response time series, $A_1(t)$ and $A_2(t)$. The cross correlation function can then be defined as:

$$R_{A1A2}(\tau) = E[A_1(t)A_2(t+\tau)] \tag{2.42}$$

Where $E[A_1(t)A_2(t+\tau)]$ is the expected value of the cross correlation function between the acceleration response time series A_1 and A_2 for varying time lag, $\tau = t_2 - t_1$.

2.4.1.1.3 Normalization

Normalizing sampled waveforms provides a simple method to compare trends in waveforms of varying amplitude. In this study, the sampled waveforms are in the form of acceleration response time series, A, and the normalization is performed by the following equation:

$$\mathbf{A}_{norm} = \frac{\mathbf{A}}{|max(\mathbf{A})| + |min(\mathbf{A})} \tag{2.43}$$

Where $\mathbf{A}_{norm}$ is the normalized acceleration record, $|max(\mathbf{A})|$ finds the absolute value of highest value in the acceleration response time series and $|min(\mathbf{A})|$ finds the absolute value of the lowest value in the acceleration response time series.

2.4.1.2 Frequency domain

2.4.1.2.1 Frequency response function

In experimental modal analysis, frequency response function (FRF) is a frequency based measurement function used to identify the modal parameters of a physical structure. Equation (2.44) defines the FRF for a single-input, single-output (SISO) case where X, or $X(\omega)$, represents the response of the system and F, or $F(\omega)$, represents the external load.

$$H_{XF} = \frac{G_{XF}(\omega)}{G_F(\omega)} \tag{2.44}$$

Here, G_{XF} is the cross spectral density between the output response and the input force, G_F is the auto spectral density of the input force and H_{XF} is the FRF of the response caused by the force. It should be noted that the theory presented here is given in terms of angular frequency ω (rad/s), whereas the results presented in section 6.2.1 and 7.5.1.2.1 are given in terms of frequency f (Hz).

3 Hell Bridge

The original Hell Bridge, figure 3.1, was a steel riveted railway bridge in Trondheim, Norway, crossing Stjoerdal river. It was built over a span of four years, finished in 1902, as a part of Troenderbanen between Trondheim and Steinskjer on Nordlandsbanen. Its service life time as a part of the Norwegian railway system lasted for 114 years, but was replaced by the new Hell Bridge when it was finished in 2016. Parts of the original Hell Bridge will continue to serve for many years to come, now as a full scale dynamic project. The project is named Hell Bridge Test Arena and is a project expected to last for about 10 years. The end of its total service life time will be marked when it in the end is loaded to complete failure.

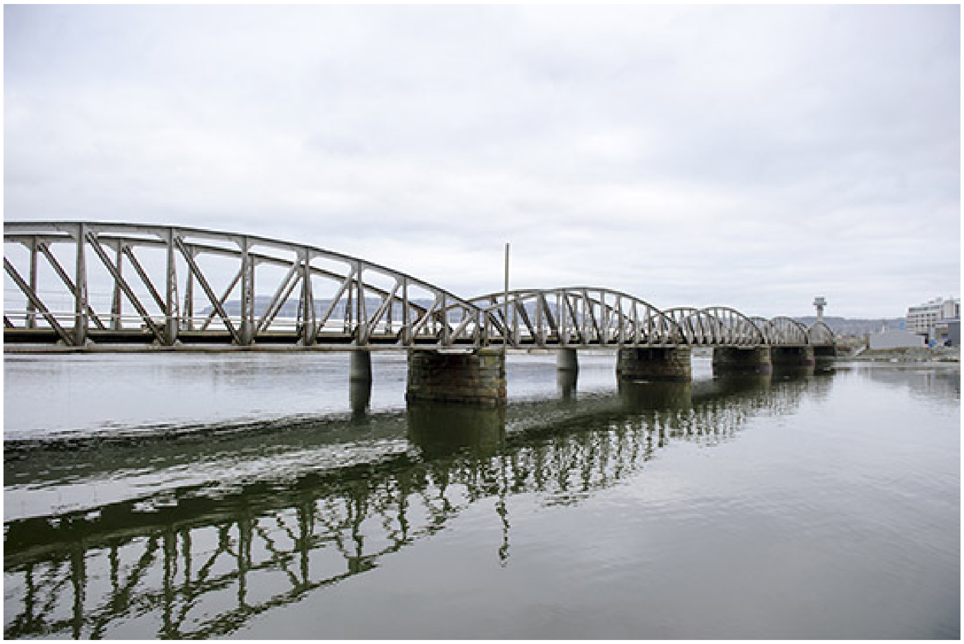
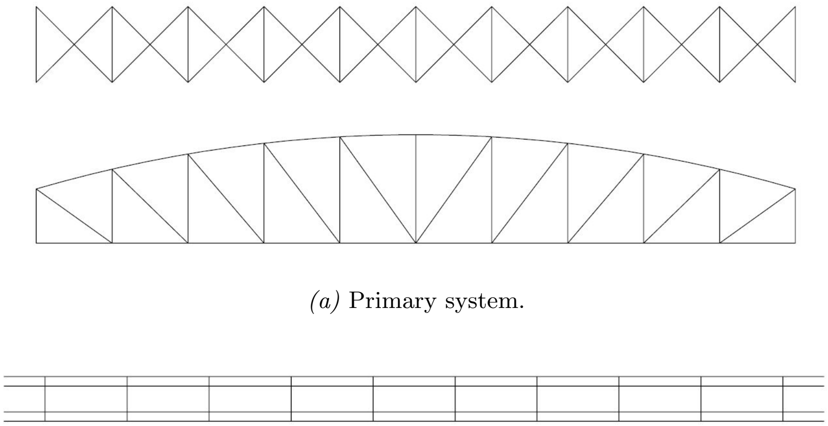


Figure 3.1: Hell Bridge when still in normal service.

The bridge was an open deck pony Parker truss bridge with five identical spans, or modules, of 35 meters and width of 4.5 meters. Each span can, if simplified, be looked at as a separate statically determined structure as the only thing connecting the different spans originally were the tracks and the expansion joints. The exact design of the bridge was unique, but it shares many structural details with other bridges in both Norway and Europe. In fact, the majority of the 1000 railway bridges constructed in Norway before 1960 have/had similar open deck riveted details as the Hell Railway Bridge [1].

3.1 Load carrying system

The load carrying system of the bridge can be divided into a primary and secondary system, see figure 3.2. The primary system consists of two identical planar trusses designed to carry the vertical loads, connected by a lower truss designed to take the horizontal forces, see figure 3.2 (a). The secondary system consist of 30 cross girders of two different lengths and 40 continuous longitudinal girders of equal length, see figure 3.2 (b). Being an open deck riveted bridge, the bridge does not have an upper truss connecting the top part of the planar trusses, making up the side walls. Instead all vertical beams, except from the four on the ends of the bridge, have stiffeners going out of the plane of the planar trusses, see figure 3.3. These stiffeners increase the in and out of plane bending stiffness of the planar trusses severely, but how they work will not be discussed further as this is beyond the scope of this paper.



(a) Primary system.

(b) Secondary system.

Figure 3.2: Load carrying system.

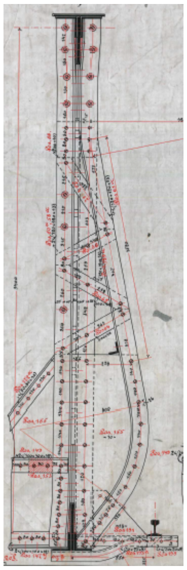


Figure 3.3: Structural detail of stiffener.

Originally, when the bridge was still in service, there were railway tracks resting on top of the wooden sleepers. The loading from the train passages was thereby transferred from the railway tracks to the wooden sleepers, and onto the longitudinal girders. From the longitudinal girders the forces were transferred to the horizontal connector girders and out to the vertical hangers. Through the vertical hangers, the forces were distributed throughout the truss and into the fundaments at every module´s ends. The effect of increased mass and stiffness from the railway tracks, which were permanently removed when the module was moved to the test arena, will be discussed in chapter 4.

3.2 Damage

In similarity with many other bridges, the Hell Bridge has an open deck construction where the bridge girder consists of stringers and cross girders. From the load distribution process presented in the previous section, it is shown that the longitudinal girders, also referred to as stringers, are a critical part in the load distribution process, and therefore a proper connection between stringers and cross girders is very important.

In the secondary load carrying system of the Hell Bridge, the connections between stringers and cross girders consist of angle bars. For the stringers subjected to further investigation, one of the joint connections is severely damaged and significant vertical movement is experienced when the stringer is subjected to load. In the continuation of this thesis, the stringer including its joint connection is referred to as the damaged structural component, whereas the other stringers including their joint connections are referred to as undamaged structural components. A thorough introduction to the damage is presented in chapter 7.

3.3 Monitoring system

For this project, both operational and experimental data is available. Whereas the operational data is only used to extract the natural frequencies and mode shapes of the system, the experimental data is also used for damage identification, which was its main purpose. All collected data is in the form of acceleration response time series. The operational data is collected from when the bridge was still in service, with no knowledge of input forces. The experimental data was produced with the use of a modal hammer on the Hell Bridge Test Arena, providing information about both input forces and output acceleration. For each case, a different monitoring system was set up, represented in the two following sections.

3.3.1 Operational data

During the measurement period in 2016, a total of 20 accelerometers were mounted on the bridge. The location of all accelerometers are shown in figure 3.4 and all discussions about accelerometers in this section are referred to this figure.

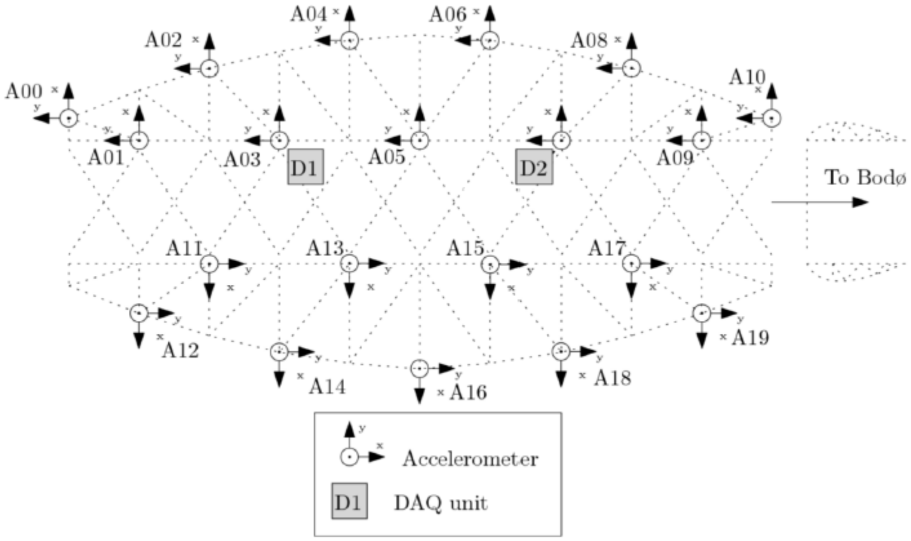


Figure 3.4: Overview of accelereometer locations

There are in total four different data files with sampling frequency 400 Hz and a duration varying between 1.5 and 2 hours, giving plenty of data to work with. The two first data files are from 9^{th} of July 2016, a cloudy day with strong gusted winds. Work was also being conducted by the entrepreneur on the south side of the bridge, where a large mobile crane and two diggers were working on the expansion joint for the new bridge. The two last data files are from 9^{th} of August 2016, a silent sunny day with no work around the bridge and almost no wind.

Before starting to process the acceleration response time series to extract the modal parameters of the bridge, the acceleration vs time was plotted for all three directions, x, y and z, for every accelerometer. This gave a total of 60 plots per data file, and was done to check if all plots looked reasonable and assure that the connection between the accelerometers and the channels were ok.

The top left part of figure 3.5 shows a good example of how such a plot might look like, with the four distinct peaks each represent a train passage. At first glance it might look like there is not much information in the record between train passages, but as displayed in the top right plot of figure 3.5, where we only plot the acceleration record up to the first train passage, this is not the case. As expected the amplitude of the acceleration is much lower for wind loads than train loads.

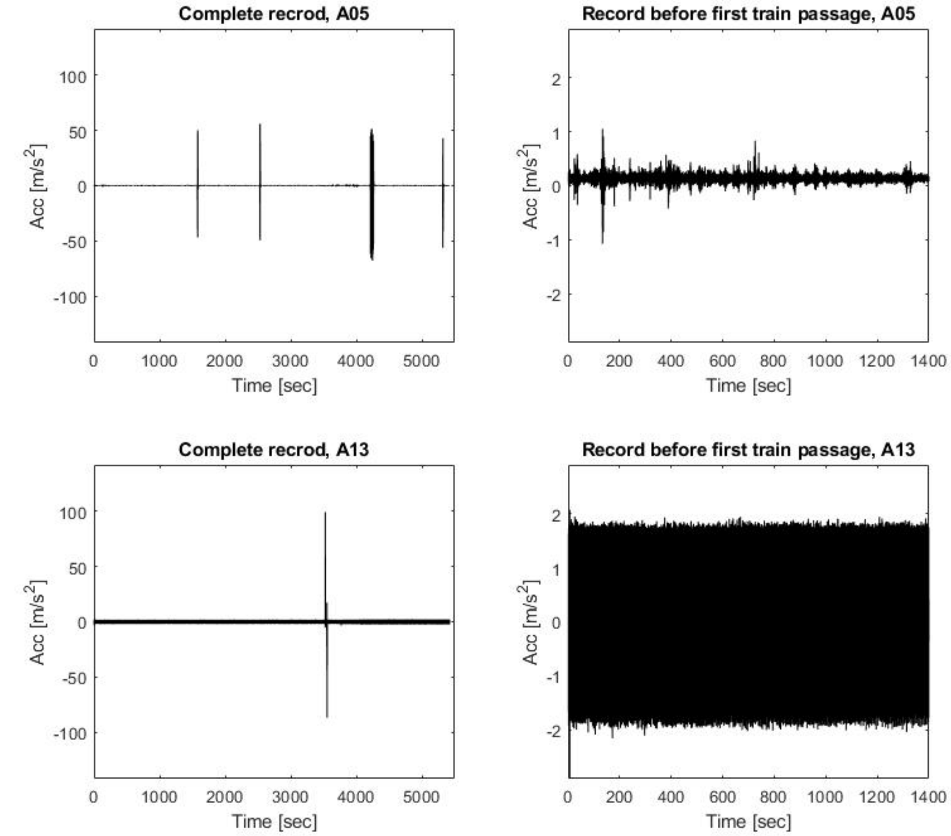


Figure 3.5: Acceleration plot in transversal direction for accelerometer A05 and A13

From plotting the acceleration history, it was found that there was no, or a bad, connection between the accelerometer and the channel for multiple channels, giving what we call a dead channel. For example, the plot for accelerometer A13, which should give somewhat the same information as accelerometer A05, looked like the one displayed in the bottom left of figure 3.5. One can clearly observe that there is only one peak, when there should be four, and for the rest of the time series there is some periodic noise, indicating a dead channel, see bottom right of figure 3.5.

Twenty accelerometers and almost 8 hours of data is a lot of information to process, but to give a good estimation of the different mode shapes, a high number of accelerometers is needed to be able to catch how the bridge moves, especially as the mode number increases. Preferably, all accelerometers would therefore be used to extract the modal characteristics of the bridge. Unfortunately, all 3 channels in both accelerometer A13 and A14 were dead for all data files. There were also some dead channels for accelerometer A11-A12 and A14-A19. For simplicity and reduction of possible error using channels that don't represent how the actual bridge behaved, only accelerometer A00-A10 were used to extract the modal parameters of the bridge. This leaves us with 11 accelerometers, only representing one side of the bridge, but this should still be enough to extract the natural frequencies and obtain mode shapes estimates without limiting the results.

3.3.2 Experimental data

The main objective with this test procedure is to use a modal hammer and a set of accelerometers to identify damage by performing measurements on the damaged and undamaged structural components. This is performed by determining the modal characteristics for relevant components in the bridge girder. The testing was performed using the rover hammer configuration method. This involves using several accelerometers that are located at fixed locations, see figure 3.6. Hits with the hammer were performed at different specified impact locations, also displayed in figure 3.6 [15].

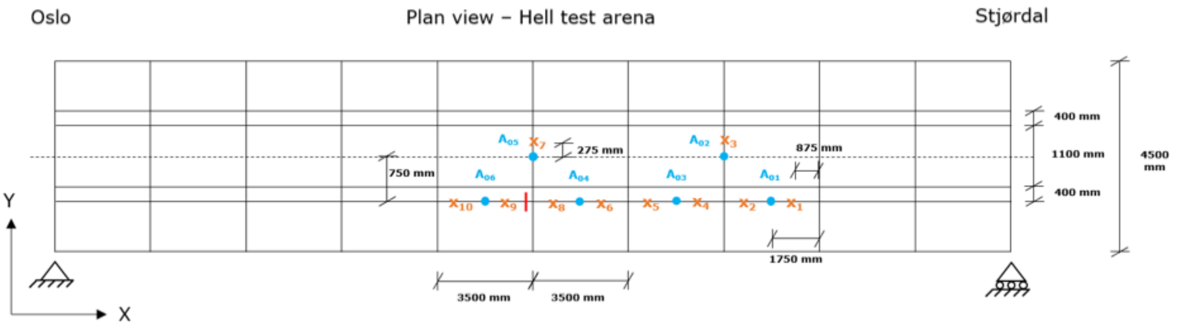


Figure 3.6: Overview over location of accelerometers, impact locations and location of damage [15].

All hits were performed directly on the beam, adjacent to the closest accelerometer. Accelerometer A05 and A02 were placed according to the global coordinate system, shown in the bottom left of figure 3.6, with the z-axis pointing out of the plane, towards the reader. The remaining four accelerometers were placed upside down on the bottom flange of the stringers due to restricted accessibility. Hence, for these accelerometers the following yields:

- $X_{LOCAL} = X_{GLOBAL}$
- $Y_{LOCAL} = -Y_{GLOBAL}$
- $Z_{LOCAL} = -Z_{GLOBAL}$

Three hits per location were performed, resulting in a total of 30 hits. Length per acceleration record is minimum 30 seconds, i.e. five seconds before the impact and 25 seconds after the impact. The length of the measured signal should be long enough to allow it to decay. The hardest hammer tip, the red tip, was used because it should result in the best possible impulse shape. Double-hits shall be avoided, i.e. the hammer tip shall not touch the impacted area twice. The data was collected at a sampling rate of 2048 Hz.

The damaged location subjected to further investigation is marked with a read line adjacent to impact location X9, see figure 3.6. Figure 3.6 also clearly displays the symmetric set up that is used. This is done so that we can compare results from the damaged structural component to results from the undamaged structural components as we would expect them to behave similarly if there was no damage present.

A quick check of the collected experimental data was also done by plotting the force vs time for the hammer sensor and acceleration vs time for all accelerometers and impact locations. The plots showed that nearly all hits exceeded the sensitivity of the sensor on the hammer tip. FRF's can therefore not be computed for these data sets as the full input is not known and PSD estimates have to be used instead. To see how this would effect the results, FRF and PSD estimates for the same acceleration output were plotted and compared. The results are shown in figure 3.7 where the FRF and PSD estimates for accelerometer A01 and A03 are compared for impact location X3.

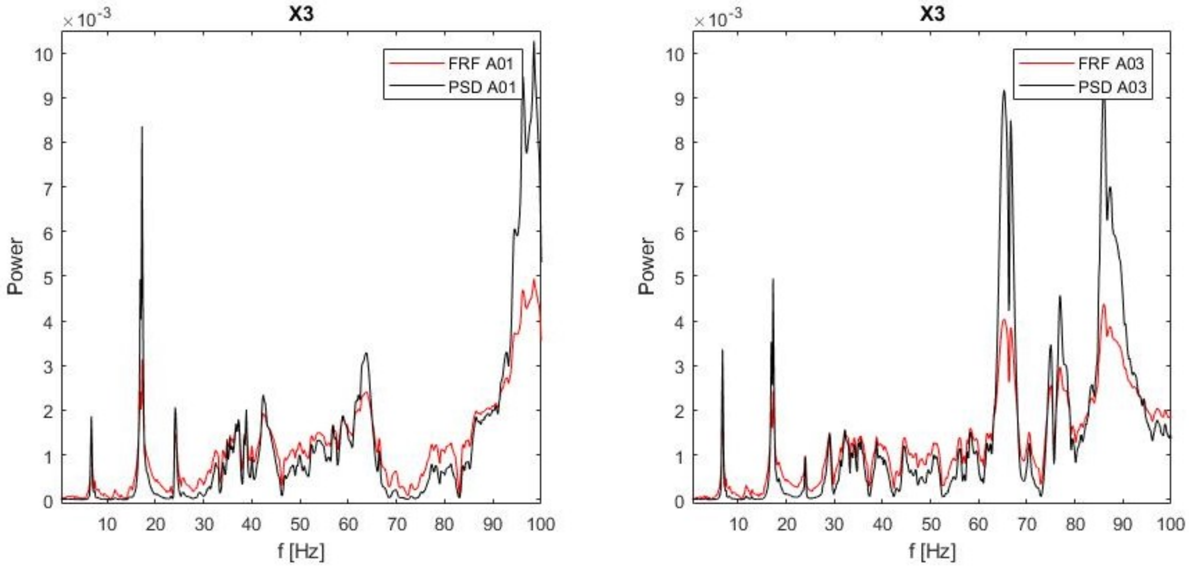


Figure 3.7: Comparison of FRF and PSD estimates for accelerometer A01 and A03, impact location X3

The observation is made that the curves in each respective plot look very similar for an impulse input. The only clear difference is the scaling of the magnitude of the different peaks, but the curves provide the same information. It is therefore concluded that for the case where the input force is in the form of an impulse, there is no practical difference between the FRF and PSD estimates. The fact that we don't have knowledge about the full input is therefore no significant drawback and the results presented in the following chapters are presented as if both input and output is known.

It was also found that many of the acceleration records where the impact location was directly adjacent to the sensor, exceeded the sensitivity of the sensor. All records were the sensitivity of the sensor was exceeded were not used in the methods described in chapter 6.

4 System identification

This chapter presents the application of the different system identification methods used in this paper.

The implementation of both frequency domain methods, PP and FDD, were performed by the use of Matlab. The PP method was simply implemented by using the built in function pwelch [16]. What this function does is give Welch's power spectral density estimate based on the inputs; acceleration response times series, sampling frequency, length of segments, number of overlap samples and number of discrete Fourier transform (DFT) points to use in the PSD estimate. The input acceleration and the sampling frequency have to be given, but the remaining three parameters can be left blank as there are built in default values, see Matlab Documentation for more information [16]. Changing these three parameters may help to smoothen out the plot and make the peak picking process easier. Unless stated otherwise, the default values were used.

The FDD method was implemented by using the FDD-script available on mathworks, constructed by *Mohamma Farshchin* [17]. This script takes acceleration records from an excel-file and sampling frequency as input, and returns a plot of the first singular values of the PSD matrix. The excel-file has one column per accelerometer and as many rows as sampling points. The number of columns in the PSD matrix will be the same as in the input excel-file. From the returned plot of the first singular values the peaks have to be picked manually. When the peaks are picked, the script returns the value at where each peak occurs, representing the natural frequencies, and their respective mode shapes. The scripts used for PP and FDD can both be found in Appendix B.

4.1 Operational modal analysis

The PP method was implemented first, so the discussion below is based on results from this method. When processing the operational data, the whole 1.5-2-hour long acceleration files were originally used as input, but this provided inconsistent results for the PSD plots. This made the task of picking the peaks hard, so to obtain better and more consistent results, three new approaches were made to the operational acceleration data used as input in the Matlab script:

- Take out acceleration data a few seconds before the train passes over the bridge and end it a few seconds after the acceleration plot is back to "normal". "Normal" being the acceleration the bridge experiences from wind loads, typically well below 2 m/s^2 in this paper.
- Take out the acceleration data between train passages, giving the bridge response from nature loads.
- Take out the acceleration data when the train has passed the bridge, but the bridge is still experiencing considerable acceleration, well above "normal".

The goal of the two first approaches was to see if there was any difference in the natural frequencies extracted from when the bridge was subjected to train loads vs wind loads. If so, this would possibly explain why the results were inconsistent. Both approaches gave much smoother plots, but only the second approach, where the acceleration was mainly induced by wind, gave more consistent results. It was found that the natural frequency estimates from when the bridge was only subjected to wind loads were somewhat higher than the ones estimated from when the bridge was also subjected to train loads. The reason for this explained by the fact that when the train is on the bridge it becomes a part of the structure giving mainly additional mass, but probably also some additional stiffness. From the simple formula shown in equation (4.1) it's easy to see that with a big increase in mass and a small increase in stiffness, the natural frequencies of the system will be lowered.

$$\omega = \sqrt{\frac{\mathbf{K}}{\mathbf{m}}} \tag{4.1}$$

This observation explains why using the whole 1.5-2-hour acceleration records as input provided inconsistent results; the results depend on the train passing the bridge. This also explains why the results obtained from using train load induced acceleration as input were inconsistent, while results from using wind load induced acceleration as input were consistent.

From the discussion above it's natural to draw the conclusion that only using the wind load induced acceleration would be best. This is true in terms of consistency of the results, but the problem is that wind loads are normally very low frequent. For a fairly short and stiff bridge, such as the Hell Bridge, wind loads will therefore excite the first couple of modes so much stronger that it's hard to localize higher frequencies from the PSD plots. Results showed that it was possible to localize peaks up to about 15-20 Hz from the PSD plots when using the wind induced acceleration only, while the PSD plots from the train induced acceleration presented peaks in the whole range up to 200 Hz. By the Nyquist-Shannon sampling theorem, 200 Hz is the highest possible frequency to detect when a sampling rate of 400 Hz is used [18].

This gave the motivation for the third approach described, where the goal was to use the bridge response from train loads, without the additional mass and stiffness from when the train was on top of it. The results obtained from this approach seemed to correlate better with the natural frequencies found from only wind loads, and higher modes were also excited. The problem was that as we had no information about the speed and length of the different trains passing the bridge, it was a troublesome procedure to find exactly when the train had passed the bridge and the choice was made to not go further with this approach.

For the PP method, a combination of the two first approaches was used. The second approach to extract the lower natural frequencies of the bridge with greater precision and the first approach to extract the natural frequencies of modes that were not excited as well by wind loads, well aware that these results are somewhat altered, depending on the train passing. Figure 4.1 shows an example of the PSD estimates for accelerometer A05, located at the center of the bridge, for both transversal and vertical direction based of acceleration data from between train passages.

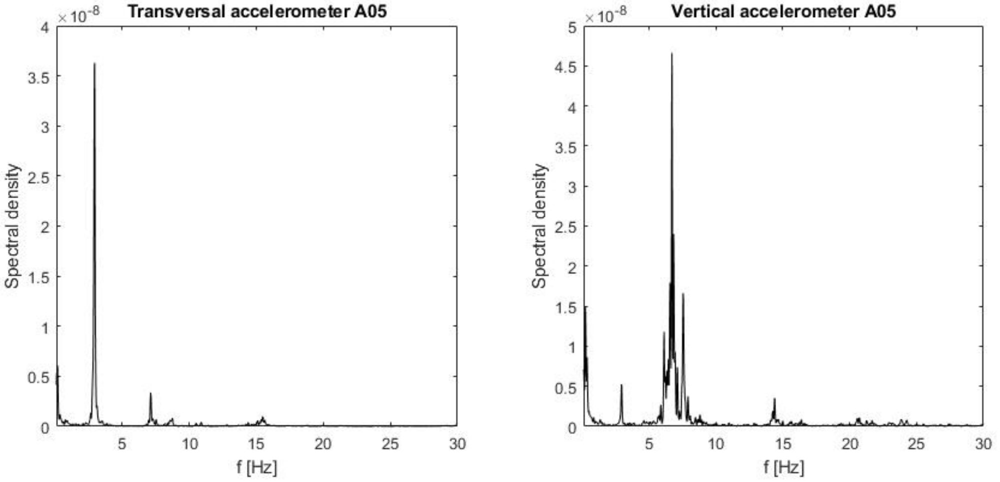


Figure 4.1: Shows plots of the PSD estimates for accelerometer A05 in both transversal and vertical direction based of acceleration from between train passages.

For the FDD method on the other hand, where the peaks were picked based on a plot from the first singular value of the PSD matrix, only the wind induced acceleration records were used. This choice was made because the information provided from this approach was sufficient to localize the first 14 modes. While the modal estimates from PP were fair, at least for the 10 first modes, the modal estimates from FDD were excellent for all 14 identified modes, presented in table 4.1 (see next section). An example of the first singular values of the PSD matrix plotted, and the mode shapes estimated for the chosen peaks is shown in figure 4.2 and 4.3 respectively. The singular values and their corresponding singular vectors are ordered in singular value descending order, meaning that the first singular value will be the largest. By applying peak picking to the first singular values, 6 significant modes from which three pure vertical bending modes, see green dots in figure 4.2, were identified in the frequency range 0-25 Hz. The plot is based of input vertical acceleration in the girder, therefore only representing modes with significant vertical motion in the girder.

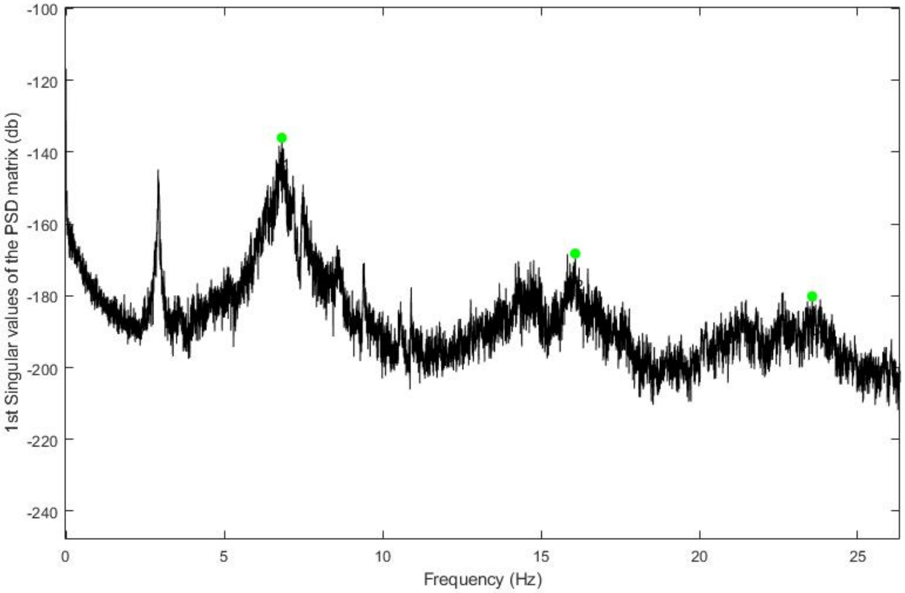


Figure 4.2: 1^{st} singular value of PSD matrix plotted, three first vertical bending modes picked (green dots).

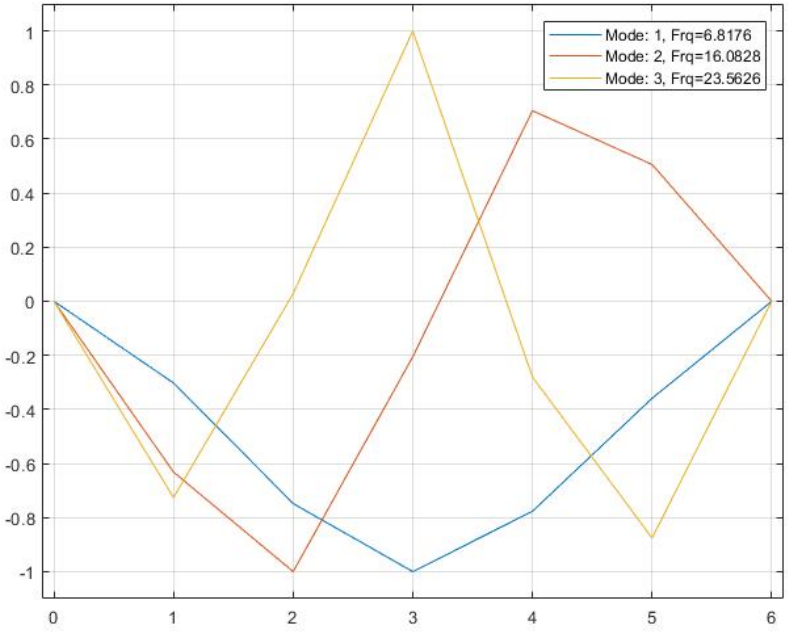


Figure 4.3: Mode shape estimates for the three first vertical bending modes using FDD.

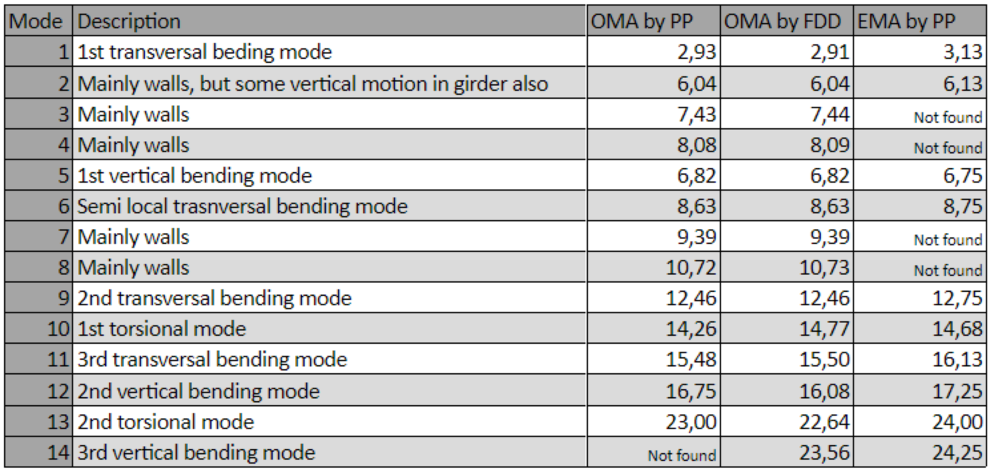
The plots in figure 4.3 are based of 7 points, therefore not very smooth, but what modes the different peaks represent becomes very clear. The displacement is known to be zero at both ends, while number 1-5 on the x-axis represent the location of accelerometer A01, A03, A05, A07 and A09 respectively (see figure 3.4, section 3.3.2 for accelerometer locations).

4.2 Experimental modal analysis

As shown in section 3.3.2, only 6 accelerometers, placed in the girder, were used for the experimental data, while 20 accelerometers, placed in both side walls and girders, were used for the operational data. Because of the high number of evenly distributed, and symmetric, accelerometers throughout the bridge for the operational data, this data gives more information than the experimental data. Not only does it give us information about both the girder and the side walls, but it also gives us the opportunity to derive good mode shape estimates. With good mode shape estimates it becomes apparent what peaks, representing natural frequencies, from PP and FDD that match the modes found in the FEA.

Although system identification was already performed using the operational data, PP was also performed on the experimental data. This was done because the experimental data was collected from the bridge after it was moved to its new fundaments, as it stands today. The numerical models are based on the bridge after it was moved to its new fundaments, i.e. with one span and no rails. As the accelerometer set up for the experimental data only covers about 40% of the bridges length, it's impossible to derive good mode shape estimates, but with the operational modal analysis already performed, there is little to no doubt of which mode shape the different peaks represent.

The results displayed in table 4.1 show that the effect of removing the rails is significant. Comparison of results from PP on operational data vs PP on experimental data clearly shows that the rails lower the natural frequencies of the bridge. From equation (4.1) in section 4.1, this indicates that that mass contribution from the rails is bigger than the stiffness contribution. Table 4.1 also shows that the results obtained with PP and FDD for the operational data are very similar, as one would expect, and that one mode not found with PP is found with FDD. Only the modes with significant motion in the girder are found for PP with the experimental data as input.



Mode	Description	OMA by PP	OMA by FDD	EMA by PP
1	1st transversal beding mode	2,93	2,91	3,13
2	Mainly walls, but some vertical motion in girder also	6,04	6,04	6,13
3	Mainly walls	7,43	7,44	Not found
4	Mainly walls	8,08	8,09	Not found
5	1st vertical bending mode	6,82	6,82	6,75
6	Semi local trasnversal bending mode	8,63	8,63	8,75
7	Mainly walls	9,39	9,39	Not found
8	Mainly walls	10,72	10,73	Not found
9	2nd transversal bending mode	12,46	12,46	12,75
10	1st torsional mode	14,26	14,77	14,68
11	3rd transversal bending mode	15,48	15,50	16,13
12	2nd vertical bending mode	16,75	16,08	17,25
13	2nd torsional mode	23,00	22,64	24,00
14	3rd vertical bending mode	Not found	23,56	24,25

Table 4.1: Shows results for PP applied on both operational and experimental data, and FDD on operational data. Frequencies are given in Hertz [Hz].

As Both FE models presented in chapter 5 were constructed with the goal to replicate the dynamic behavior of the Hell Bridge Test Arena in the best way possible, the models are updated to fit the natural frequencies estimated from the experimental data for all modes that can be found with the monitoring set up. Four out of the 14 modes presented in table 4.1 have little to no movement in the girder. Hence, natural frequency estimates for these four modes can't be found from the experimental data, so the natural frequencies estimated from the operational data are therefore used for these modes when finding the best fitting for the FE models. The main difference between the bridge when the operational data was collected vs when the experimental data was collected is that the rails were removed when collecting the latter. It could be argued that the rails mainly add mass, and some stiffness, to the modes with significant motion in the girder. The four modes with little to no motion in the girder presented in table 4.1, i.e. mode 3, 4, 7 and 8, should therefore not be affected much by the removal of the rails. The natural frequency estimates from the operational data for these four modes should therefore be very close to the ones found if the same monitoring set up was used when collecting the experimental data. When presented with the "measured" natural frequencies or the results from the system identification methods in chapter 5, the presented values will be the ones found from PP method on experimental data, supplemented with the ones found with FDD on operational data if not found from experimental data.

5 Finite element models

Two FE models were constructed for this paper, one simple beam model, and one more detailed shell model. Both FE models and the choice of numerical damping are presented in this chapter.

The steel in the bridge is assumed to have the material properties; $E = 210GPa$, $\nu = 0.3$ and $\rho = 7850kg/m$. Both FE models were updated to better fit results from the system identification. The updating was done by changing the E-modulus of different structural parts and is also presented in this chapter.

5.1 Beam model

5.1.1 Element types

The element library in Abaqus contains several types of beam elements. A "beam" element in this context is an element where assumptions are made so that the problem is reduced to one dimension mathematically, i.e. the primary solution variables are functions of position along the beam axis only [19]. For such assumption to be reasonable, a beam must be a continuum in which we can define an axis such that the shortest distance from the axis to any point in the continuum is small compared to typical lengths along the beam axis. This assumption can be justified when establishing a numerical model of the Hell Bridge and a beam model is therefore introduced.

The Abaqus user manual states; "Timoshenko beams allow for transverse shear deformation. They can be used for thick ("stout") as well as slender beams" [19]. These elements in Abaqus are formulated so that they are efficient for thin beams, where transverse shear deformations are negligible and Euler-Bernoulli theory is accurate, as well as for thick beams where the transverse shear deformations should be accounted for. Because of this, the Timoshenko beam elements are the most effective beam elements in Abaqus [19]. The use of truss elements and Euler-Bernoulli elements could be justified for several parts in the model, and would most likely decrease the run time of the analysis somewhat, but only B31 Timoshenko beams were used for simplicity. A combination of Timoshenko beam elements, Euler-Bernoulli beam elements and truss elements would also possibly alter the results of the eigenvalue analysis somewhat, but the difference from only using Timoshenko beam elements should not be drastic.

The beam sections of the primary system of the Hell Bridge are mostly composed of plates, L-profiles, and nails, making up different profiles. All beam sections in the beam model were described by the Abaqus beam cross-section library, classified as I-profiles, T-profiles, L-profiles, rectangular profiles or generalized. To fit the beam cross-section library, all I- and T-profiles with varying thickness of flanges and webs were evenly smeared out. The resulting basic finite element model is presented in figure 5.1.

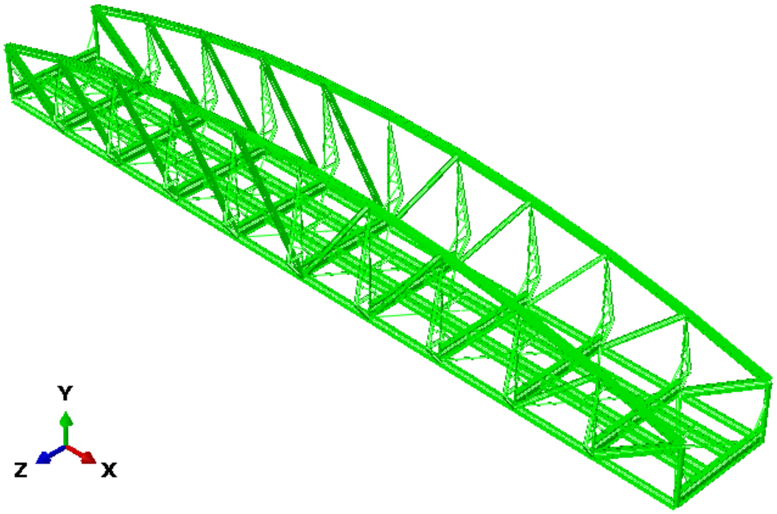


Figure 5.1: The figure shows the basic configuration of the Beam FE-model of the Hell Bridge

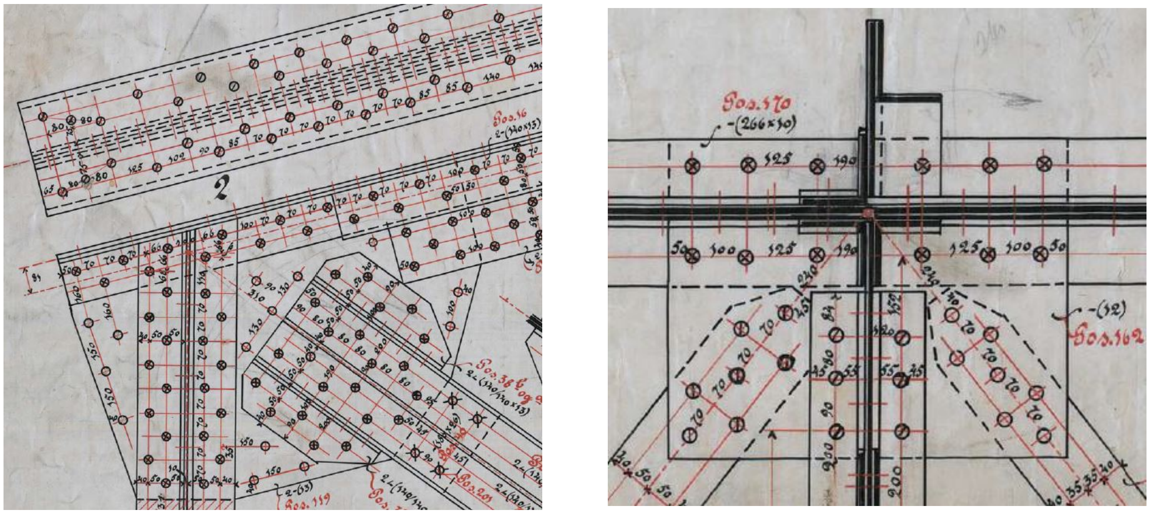
5.1.2 Service loads

The main contribution of the non-structural mass of the Hell Bridge test arena comes from the wooden sleepers resting on top of the stringers and cross girders. The estimate of the non-structural mass is therefore based on a rough estimate of the weight of the wooden sleepers only. From looking at the drawings of the Hell Bridge it is assumed that the average dimension of a wooden sleeper is 26x16x260 cm and that there are 130 wooden sleepers in total. From guidelines in the governing Eurocode for wooden sleepers and bearers, NS-EN 13145 [20], it is further assumed that the wood used is European oak with a density of 702 kg/m^3. From computation, this gives an approximate non-structural mass of 10 tons which is smeared out evenly over the stringers and cross girders. With the weight of all structural parts adding up to approximately 60 tons this gives an approximate total weight of 70 tons.

5.1.3 Boundary conditions and other constraints

The following discussion about the boundary conditions of the bridge is referred to figure 5.1, where the coordinate system in the bottom left corner of the figure has its origin in the back-left corner of the bridge. The bridge is assumed simply supported in its length direction, which is in good agreement with its new fundaments at the Hell Bridge Test Arena and also with the drawings. It is restrained in the x-direction in both corners at one end and restrained at all four corners in y- and z-direction.

There are a great number of joints in the Hell Bridge. For every single beam in every single joint, a choice has to be made; should the beams attaching form a completely stiff joint or should some degrees of freedom (DOFS), typically rotational DOFS, be unconstrained? It's important to realize that imposing constraints increases the stiffness of the system, which once again increases the natural frequencies of the system. A completely stiff joint does not exist in reality, but this is a fair simplification for most joints in the Hell Bridge. For the beam model, all beam connections but the diagonal beams in the side walls and the bracing in the girder, see figure 5.1, were fully constrained. Both the diagonal beams and the bracing were modelled as free to rotate out of their respective planes, with the remaining 5 DOFS constrained to follow their respective master nodes. These connections are made by thin plates, see figure 5.2 (a) and (b), so in reality they are not totally free to rotate, but without modelling a spring it´s the representation believed to be closest to reality.



(a) Connection joint with diagonal beam

(b) Connection joint with bracing

Figure 5.2: (a) shows the connection joint in all four corners at the top. (b) shows a connection joint in the bottom of the girder.

All constraints were imposed by choosing one master node at the end of a carefully chosen beam for every joint and the end of the remaining beams attaching were set as slave nodes. The slave and master nodes were then tied together with kinematic tie constraints, see under the keywords *Constraint, Kinematic and Tie* in the Abaqus keywords reference manual [19].

5.1.4 Natural frequencies

With the element model established according to the description in previous sections, a non-linear eigenvalue analysis was performed to extract the natural frequencies. The results showed that almost all modes with significant motion in the girder were too stiff, while all modes where mainly the walls were excited (little motion in girder), were too soft, when compared to the results of the system identification, see table 5.2. There are a number of possible reasons for this:

- Most profiles in this bridge are composed by different profiles and nails. The assumption that these profiles are fully bonded will overestimate their stiffness
- Simplifications made to profiles looking like I- and T-profiles to fit the Abaqus beam section library. In the model, theses profiles have the correct area, height and width, but the thickness is changed somewhat, resulting in approximate cross sectional properties.
- Most joints were modelled as completely stiff which makes the FE model stiffer than in reality
- The simplification for the stiffeners, presented in section 5.1.5. This simplification will most likely cause the stiffeners to be softer than they actually are.

To achieve a better fitting between the modal parameters obtained from the different system identification and the FE model, the model was updated by adjusting the stiffness of different parts in the model. The adjustments made are presented in table 5.1. The names of the parts in table 5.1 are shown in figure 5.3. Bottom stiffener is the black triangle in the bottom left of figure 5.3, while top stiffener is the rest of the stiffener.

Structural part	E-modulus	Percent of original E = 210 GPa
Frame	178.5 GPa	85%
Diagonal beams	189.0 GPa	90%
Stiffener bottom	420.0 GPa	200%
Bracing	157.5 GPa	75%
Stiffener top	252.0 GPa	120%
Inside stiffener	252.0 GPa	120%
Cross girder	231.0 GPa	110%

Table 5.1: Adjustments made to the tuned beam model

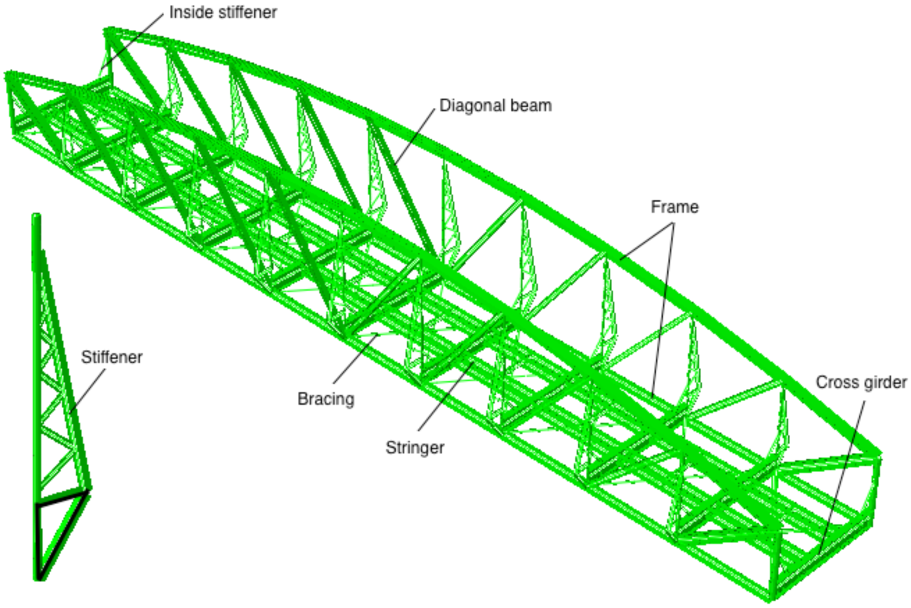
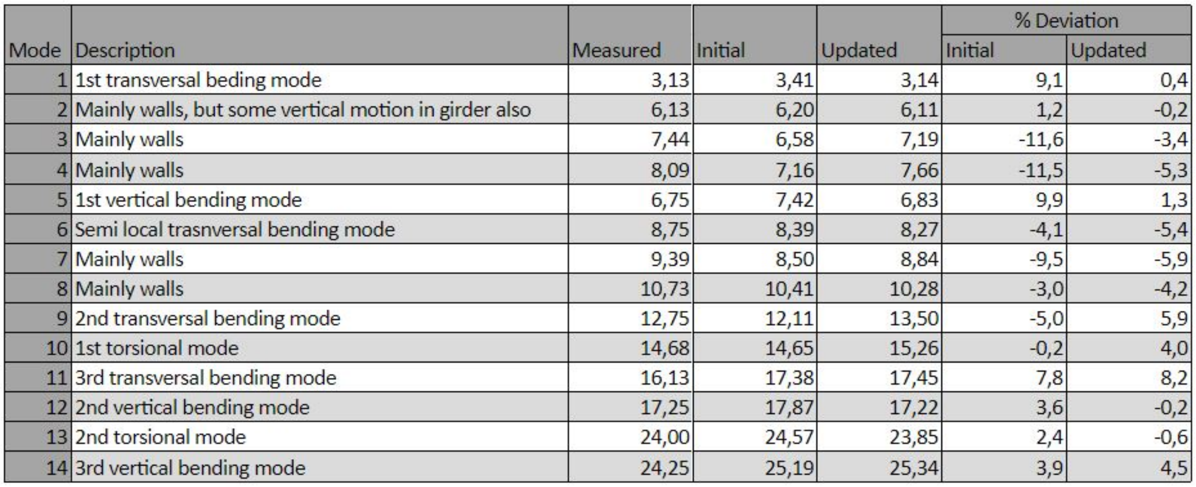


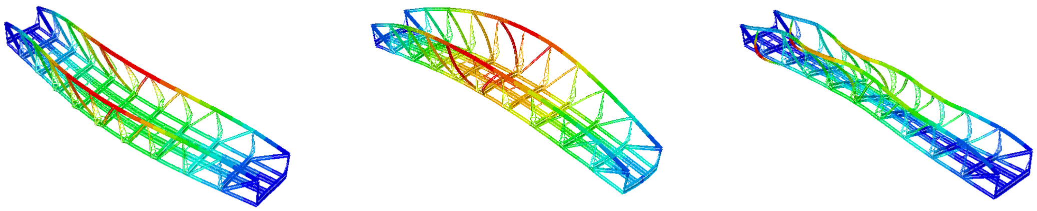
Figure 5.3: Names for different structural parts

Table 5.2 displays the 14 first natural frequencies found from the eigenvalue analysis and how they deviate from measured natural frequencies. All modes found from the system identification were also found in the FE model. The first transversal bending mode, first vertical bending mode and first torsional mode are shown in figure 5.4 (a), (b) and (c) respectively. The remaining modes can be found in appendix A.



Mode	Description	Measured	Initial	Updated	% Deviation	
					Initial	Updated
1	1st transversal beding mode	3,13	3,41	3,14	9,1	0,4
2	Mainly walls, but some vertical motion in girder also	6,13	6,20	6,11	1,2	-0,2
3	Mainly walls	7,44	6,58	7,19	-11,6	-3,4
4	Mainly walls	8,09	7,16	7,66	-11,5	-5,3
5	1st vertical bending mode	6,75	7,42	6,83	9,9	1,3
6	Semi local trasnversal bending mode	8,75	8,39	8,27	-4,1	-5,4
7	Mainly walls	9,39	8,50	8,84	-9,5	-5,9
8	Mainly walls	10,73	10,41	10,28	-3,0	-4,2
9	2nd transversal bending mode	12,75	12,11	13,50	-5,0	5,9
10	1st torsional mode	14,68	14,65	15,26	-0,2	4,0
11	3rd transversal bending mode	16,13	17,38	17,45	7,8	8,2
12	2nd vertical bending mode	17,25	17,87	17,22	3,6	-0,2
13	2nd torsional mode	24,00	24,57	23,85	2,4	-0,6
14	3rd vertical bending mode	24,25	25,19	25,34	3,9	4,5

Table 5.2: Shows the 14 measured natural frequencies, frequencies from the eigenvalue analysis of the beam model before and after updating and the deviation from the measured values. Frequencies are given in Hertz [Hz]



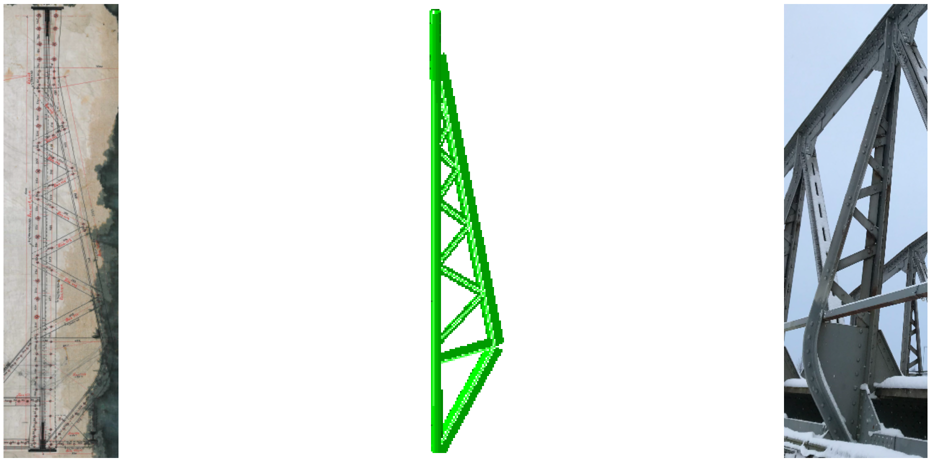
(a) 1^{st} transversal bending mode *(b)* 1^{st} vertical bending mode *(c)* 1^{st} torsional bending mode

Figure 5.4: Mode shape estimates from beam model

With the 14 first modes of the Hell Bridge represented in the FE model, with good correlation when compared to data from the full scale dynamic analysis, the model was evaluated to be sufficiently accurate to move on with modelling the known damage in the connection joint between stringer and cross girder.

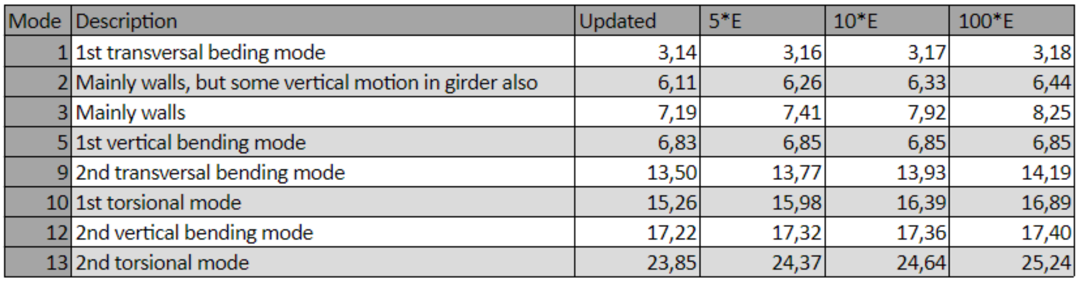
5.1.5 Sensitivity analysis

When establishing the beam model, a number of simplifications had to be made. One significant simplification is the one made to the outside stiffeners of the bridge. In reality, all vertical beams except the two beams furthest out on both sides, have outside stiffeners similar to the one shown in figure 5.5 (a). According to the drawings these stiffeners are rounded at the bottom and a 10-mm thick plate fills the bottom part, see figure 5.5 (a) and (c). In the FE model this is simplified to a triangle, made out of beams, shown in figure 5.5 (b). This simplification will possibly have a great impact on the mode shapes and natural frequencies from an eigenvalue analysis of the FE model. A sensitivity analysis was therefore performed to evaluate the effect of increased stiffness in the bottom part of the outside stiffeners. This was done by increasing the E-modulus of the bottom triangle in figure 5.5 (b) (also see black triangle in bottom left of figure 5.3). The results of the analysis are presented in table 5.3. Only 8 modes are shown as this is more than enough to show the trend.



(a) Stiffener from drawing *(b)* Simplified stiffener in FE model *(c)* Stiffener on the bridge

Figure 5.5: Figure (a) shows a drawing of one of the stiffeners, figure (b) shows the simplified version used in the beam model and figure (c) shows one of the stiffeners from the actual bridge.



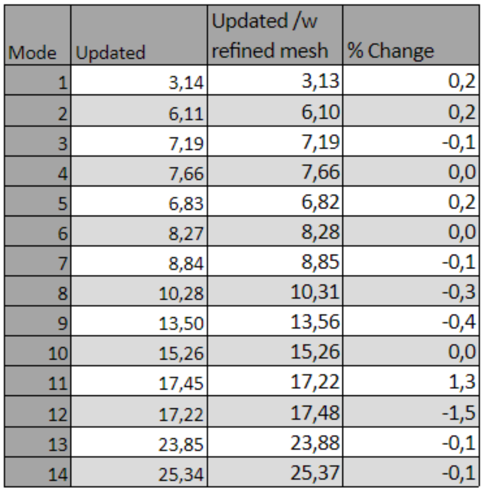
Mode	Description	Updated	5*E	10*E	100*E
1	1st transversal beding mode	3,14	3,16	3,17	3,18
2	Mainly walls, but some vertical motion in girder also	6,11	6,26	6,33	6,44
3	Mainly walls	7,19	7,41	7,92	8,25
5	1st vertical bending mode	6,83	6,85	6,85	6,85
9	2nd transversal bending mode	13,50	13,77	13,93	14,19
10	1st torsional mode	15,26	15,98	16,39	16,89
12	2nd vertical bending mode	17,22	17,32	17,36	17,40
13	2nd torsional mode	23,85	24,37	24,64	25,24

Table 5.3: Shows the effect of increasing stiffness of bottom triangle in stiffener on chosen modes. Frequencies are given in Hertz [Hz].

Table 5.3 shows that the increased stiffness has little effect on the vertical bending modes, moderate effect on the transversal bending modes and a big effect on torsional modes and modes where the main motion is in the side walls. The conclusion is drawn that with the simplifications made, the stiffness in the bottom part of the stiffener must be chosen with care, as it is a simplification with a considerate impact on the estimated natural frequencies. To model the outside stiffeners in a more accurate way, a shell model should be introduced, see section 5.2.

A sensitivity analysis was also performed for the choice of element size. In the presented beam model, a global element size of 0.25 m for the outside stiffeners, 0.5 m for diagonal beams and bracing, and 0.125 m for all remaining structural parts was chosen. To see how this choice effected the results of the eigenvalue analysis the whole mesh was refined by a factor of five, i.e a global size of 0.05 m for the outside stiffeners, 0.1 m for the diagonal beams and bracing, and 0.025 m for all remaining elements.

Table 5.4 shows the results for the first 14 modes from the eigenvalue analysis for the FE model, the same model with a refined mesh and the percent change.



Mode	Updated	Updated /w refined mesh	% Change
1	3,14	3,13	0,2
2	6,11	6,10	0,2
3	7,19	7,19	-0,1
4	7,66	7,66	0,0
5	6,83	6,82	0,2
6	8,27	8,28	0,0
7	8,84	8,85	-0,1
8	10,28	10,31	-0,3
9	13,50	13,56	-0,4
10	15,26	15,26	0,0
11	17,45	17,22	1,3
12	17,22	17,48	-1,5
13	23,85	23,88	-0,1
14	25,34	25,37	-0,1

Table 5.4: Displays the effects of refining the mesh by a factor of 5. Frequencies are given in Hertz [Hz]

As the table displays, the changes are marginal. When using finer mesh, the run time of the analysis increases, so the small changes presented in table 5.4 support that the choice of element sizes in the beam model are sufficiently small.

5.2 Shell model

Abaqus user manual reads: “Shell elements are used to model structures in which one dimension, the thickness, is significantly smaller than the other dimensions”[19]. As most profiles in the Hell Bridge are composed by different slender plates, L-profiles and nails, an FE model constructed by shell elements can also be justified. An FE model with beam elements is already presented in the previous section, but construction of a shell model allows for the valuable comparison of two different models. As a shell model generally allows for a more detailed representation, this also provides the opportunity to model damage in a different way than in the beam model, if necessary.

Figure 5.6 shows the constructed shell model and the coordinate system used with its origin in the back-left corner. The model consists of 399 part instances, from which 381 are made of shell elements and 18 are made of beam elements. The parts have been drawn individually, then partitioned into regions for constraint and mesh assignment. The beam elements are used in the truss of the outside stiffeners as this makes the procedure of modelling the stiffeners significantly easier and should not have any effect on the results relevant for this paper.

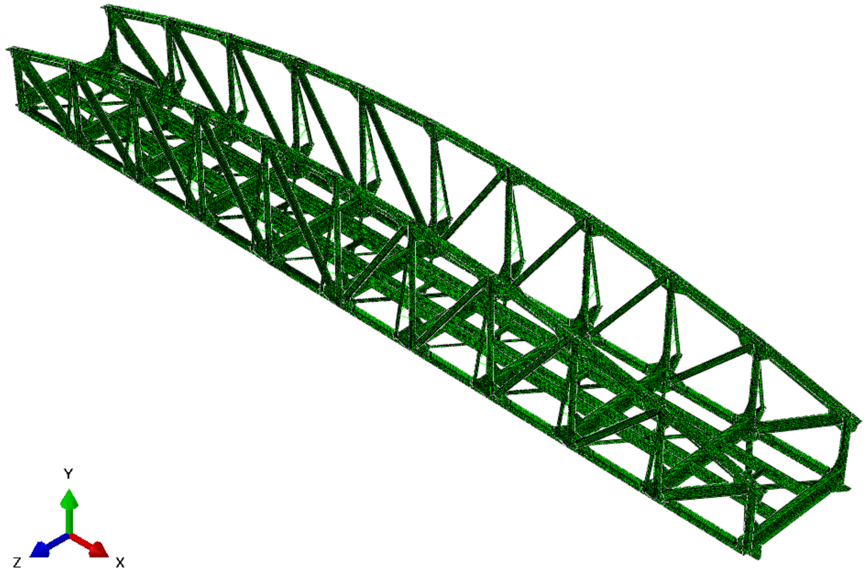


Figure 5.6: The figure shows the basic configuration of the Shell FE model of the Hell Bridge.

5.2.1 Constraints, boundary conditions and service load

A typical joint on the Hell Bridge is shown in figure 5.7. Due to the large number of bolts used in the connections, all connection are modelled as completely stiff. This is done by using tie constraints for coinciding shell elements where all slave nodes within a user specified tolerance are constrained to follow their respective master surfaces. The default value provided by Abaqus was used for the distance tolerance, which is 10% of the typical master facet diagonal length. As recommended in the Abaqus user manual, the surfaces that coincide with most other surfaces were chosen as the master surface. If present, all connection plates were therefore chosen as master surfaces in the tie constraint formulations for this model. For joints with no connection plate, the part with the greatest stiffness was assigned master surface.

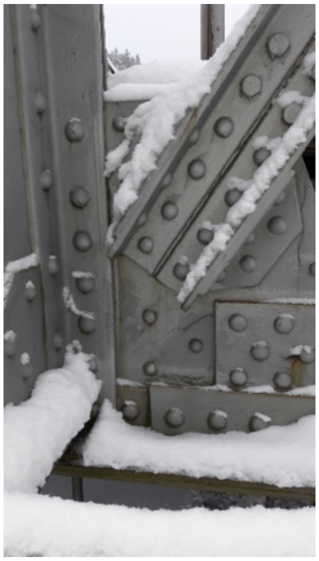


Figure 5.7: A typical plated connection on the Hell bridge

The same boundary conditions were used for the shell model as for the beam model, i.e. simply supported in its length direction.

The same rough estimate of non-structural mass was also added to the shell model, once again evenly smeared out over the stringers and cross girders. It should be mentioned that the weight of the structural parts of the shell model is approximately 66 tons, considerably higher than for the beam model, giving an approximate total mass of 76 tons.

5.2.2 Element types and mesh

When making the choice of element type in a shell model, it's important to have the goal of the numerical model in mind. For this paper the goal has been to get a good estimates of the dynamic behavior of the bridge, and measure nodal displacement and acceleration accurately. For this purpose, a 4-node element with reduced integration should be sufficient. The use of reduced integration reduces the computational time of the analysis and avoids the error source of overestimated stiffness, which regular integration methods typically tend to do [21]. The effect of reduced integration method vs regular integration method on the natural frequency estimates is presented in the sensitivity analysis in section 5.2.4.

As defined in the Abaqus user manual, for a problem where the shell thickness exceeds 1/15 of the element length, thick shell theory is advised [19]. Abaqus provides a high number of general-purpose conventional shell elements, which can be used for both thick and thin plate theory. These elements use thick shell theory, also known as shear-flexible theory or Mindlin Theory, as the shell thickness increases and become discrete Kirchhoff thin shell elements as the thickness decreases. Since the Abaqus user manual states that general-purpose shell elements should be the users first choice from the element library and that the Hell Bridge consist of both thick and thin shells, a general-purpose conventional shell element should be a good choice [19]. The choice of element therefore landed on the S4R and S3R shell elements, both classified as general-purpose conventional shell elements. The 3-node S3R shell elements were used for all areas where the 4-node S4R shell elements were not compatible with the geometry.

Since neither S4R or SR3 tend to get distorted, no emphasis was put on avoiding distortion when assigning mesh regions. Something to note when using tie constraints is that the master surface must be larger than the slave surface for the surfaces to tie together properly. For this reason, three main element sizes were used in the shell model. The elements were created using the uniformly edge seeding tool in Abaqus CAE, creating elements with different sizes at each seeding; 50mm, 65mm and 75mm respectively.

5.2.3 Natural frequencies

From the eigenvalue analysis of the shell model, the same 14 modes found in the beam model were also found in the shell model, only at somewhat different frequencies. Initially with a uniform E-modulus of 210 GPa, all vertical modes with significant vertical and transversal motion in the girder were too stiff when compared to the natural frequency estimates from the system identification. Similarly, all modes with the main motion in the side walls (little motion in girder) were found to be too soft. There are several possible reasons for this, similar to the ones discussed for the beam model:

- Most profiles in this bridge are composed by different profiles and nails. The assumption that these profiles are fully bonded will overestimate their stiffness
- Simplifications of different profiles to make the modelling process easier. The simplifications made are in such a way that the cross sectional area and the second moment of inertia about the strongest bending axis remained unchanged.
- The simplifications made to the outside stiffeners geometries, presented in section 5.2.4. As the plate in the bottom part of the stiffeners are somewhat represented in the model, this should provide a better representation than the beam model, but the simplifications made are still expected to cause the stiffeners to be softer than in reality.

As for the beam model, the shell model was updated to fit the natural frequencies estimates from the system identification better. This was once again done by adjusting the stiffness of different structural parts, presented in table 5.5. The parts with adjusted stiffness are highlighted in figure 5.8.

Structural part	**Elastic modulus**	**Percent of original E = 210 GPa**
Bracing	168.0 GPa	80%
Stiffener plates	420.0 GPa	200%
Main girders	189.0 GPa	90%

Table 5.5: Adjustments made to the tuned shell model

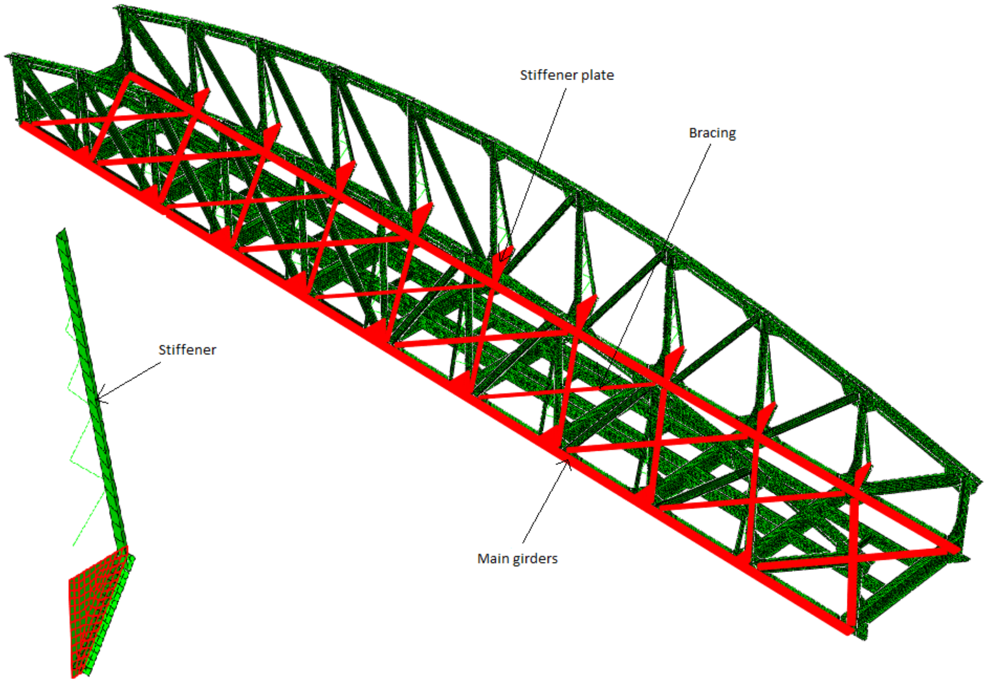
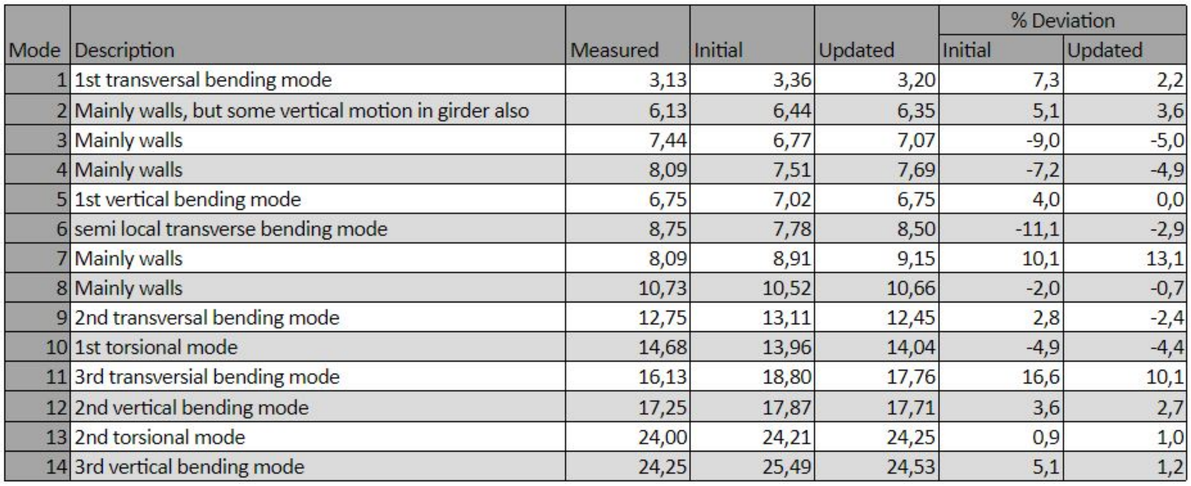


Figure 5.8: Shows structural names of tuned parts in the shell model

Table 5.6 displays the 14 first natural frequencies found from the eigenvalue analysis of the shell model and how they deviate from the results of the system identification. The table displays that a fairly good fitting is achieved. Further discussion and comparison of the two numerical models will be presented section 5.4. The mode shape estimates from the shell model are displayed in appendix A

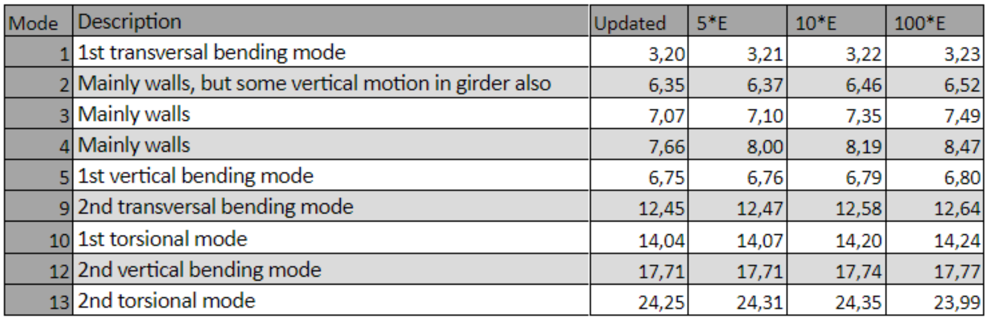


Mode	Description	Measured	Initial	Updated	% Deviation	
					Initial	Updated
1	1st transversal bending mode	3,13	3,36	3,20	7,3	2,2
2	Mainly walls, but some vertical motion in girder also	6,13	6,44	6,35	5,1	3,6
3	Mainly walls	7,44	6,77	7,07	-9,0	-5,0
4	Mainly walls	8,09	7,51	7,69	-7,2	-4,9
5	1st vertical bending mode	6,75	7,02	6,75	4,0	0,0
6	semi local transverse bending mode	8,75	7,78	8,50	-11,1	-2,9
7	Mainly walls	8,09	8,91	9,15	10,1	13,1
8	Mainly walls	10,73	10,52	10,66	-2,0	-0,7
9	2nd transversal bending mode	12,75	13,11	12,45	2,8	-2,4
10	1st torsional mode	14,68	13,96	14,04	-4,9	-4,4
11	3rd transversial bending mode	16,13	18,80	17,76	16,6	10,1
12	2nd vertical bending mode	17,25	17,87	17,71	3,6	2,7
13	2nd torsional mode	24,00	24,21	24,25	0,9	1,0
14	3rd vertical bending mode	24,25	25,49	24,53	5,1	1,2

Table 5.6: Shows the 14 measured natural frequencies, frequencies from the eigenvalue analysis of the shell model before and after updating and the deviation from the measured values. Frequencies are given in Hertz [Hz].

5.2.4 Sensitivity analysis

Like for the beam model, the vertical stiffeners in the shell model were simplified. The main simplifications regard the geometry of the stiffeners. As previously presented in section 5.1.5, the bottom of the stiffeners are rounded in reality. In the shell model the stiffeners are simplified to have a triangular shape, shown in figure 5.8. The plates in the model are also simplified to have a triangular shape and are smaller than in reality. These simplifications will most likely reduce the stiffness of the stiffeners and affect the global stiffness of the bridge. A sensitivity analysis was therefore performed to evaluate the effect of increased stiffness in the bottom part of the outside stiffeners. This was done by increasing the E-modulus of the "stiffener plate", highlighted in the bottom left part of figure 5.8. The results are presented in table 5.7, where the plate stiffness is increased from 2 times original stiffness in the updated model to 5, 10 and 100 times the original stiffness of 210 GPa. Only 9 modes are shown as this is enough to show the trend.

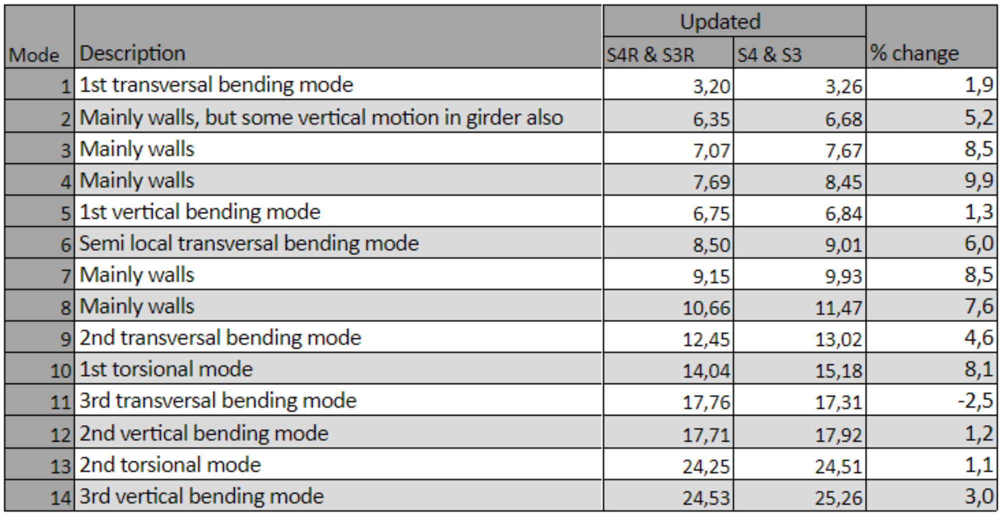


Mode	Description	Updated	5*E	10*E	100*E
1	1st transversal bending mode	3,20	3,21	3,22	3,23
2	Mainly walls, but some vertical motion in girder also	6,35	6,37	6,46	6,52
3	Mainly walls	7,07	7,10	7,35	7,49
4	Mainly walls	7,66	8,00	8,19	8,47
5	1st vertical bending mode	6,75	6,76	6,79	6,80
9	2nd transversal bending mode	12,45	12,47	12,58	12,64
10	1st torsional mode	14,04	14,07	14,20	14,24
12	2nd vertical bending mode	17,71	17,71	17,74	17,77
13	2nd torsional mode	24,25	24,31	24,35	23,99

Table 5.7: Shows the effect of increased stiffness of plate in bottom part of stiffener on chosen modes. Frequencies are given in Hertz [Hz]

Table 5.7 shows that the increased stiffness has little effect on the natural frequencies for the vertical, transversal and torsional bending modes with considerate motion in the girder. The only modes that are seriously affected by the stiffness increase are the modes where mainly the sidewalls are excited. The conclusion is drawn that the simplifications of the stiffeners are good ones as they mainly effect the stiffness of the sidewalls which easily can be tuned without it affecting the other global modes significantly.

A sensitivity analysis for the choice of elements was also performed. As previously discussed, the choice of element type should be made in accordance with the goal of the analysis. Since the model was constructed for lower order elements, S3 and S4, no emphasis was put on distortion when assigning mesh regions. The result of this is that the model becomes almost incompatible with higher order geometric elements which would require a very fine mesh in the regions to prevent element distortion. The sensitivity analysis therefore only presents the comparison of reduced integration method vs regular integration method. The results are presented in table 5.8, where the third column presents the natural frequencies from the eigenvalue analysis when using reduced integration method, the fourth column shows the same for regular integration method and the fifth column shows the percent change between the two. For both cases the 4-node element is used where the geometry allows for it, if not a 3-node element is used instead.



		Updated		
Mode	Description	S4R & S3R	S4 & S3	% change
1	1st transversal bending mode	3,20	3,26	1,9
2	Mainly walls, but some vertical motion in girder also	6,35	6,68	5,2
3	Mainly walls	7,07	7,67	8,5
4	Mainly walls	7,69	8,45	9,9
5	1st vertical bending mode	6,75	6,84	1,3
6	Semi local transversal bending mode	8,50	9,01	6,0
7	Mainly walls	9,15	9,93	8,5
8	Mainly walls	10,66	11,47	7,6
9	2nd transversal bending mode	12,45	13,02	4,6
10	1st torsional mode	14,04	15,18	8,1
11	3rd transversal bending mode	17,76	17,31	-2,5
12	2nd vertical bending mode	17,71	17,92	1,2
13	2nd torsional mode	24,25	24,51	1,1
14	3rd vertical bending mode	24,53	25,26	3,0

Table 5.8: Shows the effect of element choice on the natural frequency estimates in the shell model. Frequencies are given in [Hz]

The table supports the statement that regular integration methods tend to overestimate the stiffness of the system. From the 14 first modes found from the eigenvalue analysis, all natural frequencies but one increase, some as much as close to 10%. This goes to show that the type of element in a shell model has a great impact on the results of an eigenvalue analysis and should therefore be chosen with care, preferably with support in literature.

5.3 Damping in the finite element models

In section 2.2 some brief theory about damping was presented with further emphasis on the easily implemented Rayleigh damping method, used for the numerical models in this paper. The choice of Rayleigh damping is partially justified by results presented by Frýba in Dynamics of Railway Bridges and partially by the need to limit the scope and extent of this thesis.

Frýba presents data from a wide specter of different railway bridges showing a considerable spread in damping ratios. He further also presents a dependence between damping ratio and amplitude of displacement for the different bridges [7, chapter 5]. This shows that damping in structures is a highly complicated subject and that extensive research is most likely necessary to determine the damping of the specific bridge being analyzed. Regardless of this, as a guideline, Frýba recommends a damping ratio of $\xi_n \approx 0.013$ for steel railway bridges with spans larger than 20 m.

General recommendations for damping in dynamic analysis of railway bridges can also be found in the governing Eurocode, NS-EN1991-2 [22, chapter 6]. It states that only lower bound estimates of damping shall be used and recommends a damping of $\xi_n \approx 0.005$ for steel bridges with spans greater than 20 m.

Recall from section 2.2 that the Rayleigh coefficients, a_0 and a_1 are calculated from specifying two damping ratios, each with a natural frequency at where the respective damping ratios should apply. Generally, when performing dynamic analysis, it is important that all modes with considerable contribution to the dynamic response have a reasonable damping ratio. Overdamping of significant modes will underestimate the response of the structure and provide non-conservative results. For this thesis where local modes of the stringers also are very important for the acceleration response used in the different damage identification techniques, it should also be assured that these modes are assigned reasonable damping ratios. This means that for the most important global modes, we need to apply an appropriate damping ratio for natural frequencies in the range 20–160$\frac{rad}{s}$($\approx 3.1 - 25Hz$). Further from analyzing numerically simulated acceleration data in section 7.5.1.2.1, with the exact same set up as the experimental data, it is believed that the stringers have their first important local modes in the frequency range 125–600$\frac{rad}{s}$. Assigning appropriate damping values in such a wide frequency range is challenging with the simple Rayleigh damping method, but was attempted by finding a middle way.

This was done by assigning a damping ratio of 0.5%, i.e. $\xi_m = 0.005$, for the first natural frequency $\omega_m = \omega_1 = 2\pi rad * 3.13Hz = 19.67\frac{rad}{s}$. The upper frequency, also assigned a damping of 0.5%, was set equal to 250$\frac{rad}{s}$($\approx 38.2Hz$). Figure 5.9 shows that the damping ratios in the range does not exceed the damping value recommended by Fryba and that the minimum damping ratio is found to be approximately 0.28%. One might argue that the damping is somewhat low for several important modes in the range 50-150 $\frac{rad}{s}$, but as a higher damping in this range results in overdamping for important local modes it is not applied. The choice of damping is justified by the goal of the numerical analysis in mind; to find a good numerical representation of the damaged structural component from studying imposed vibrations.

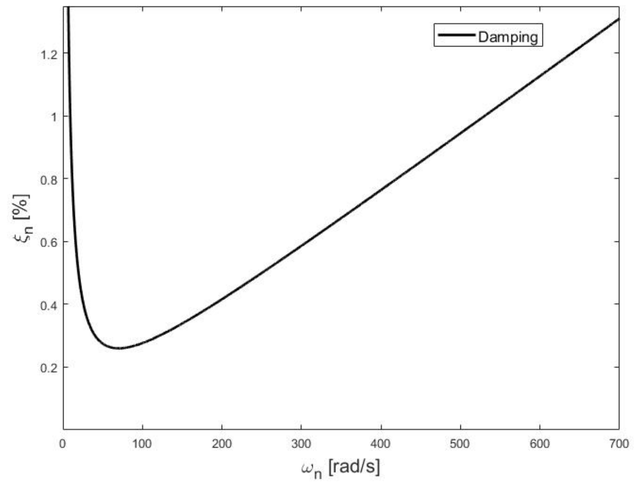


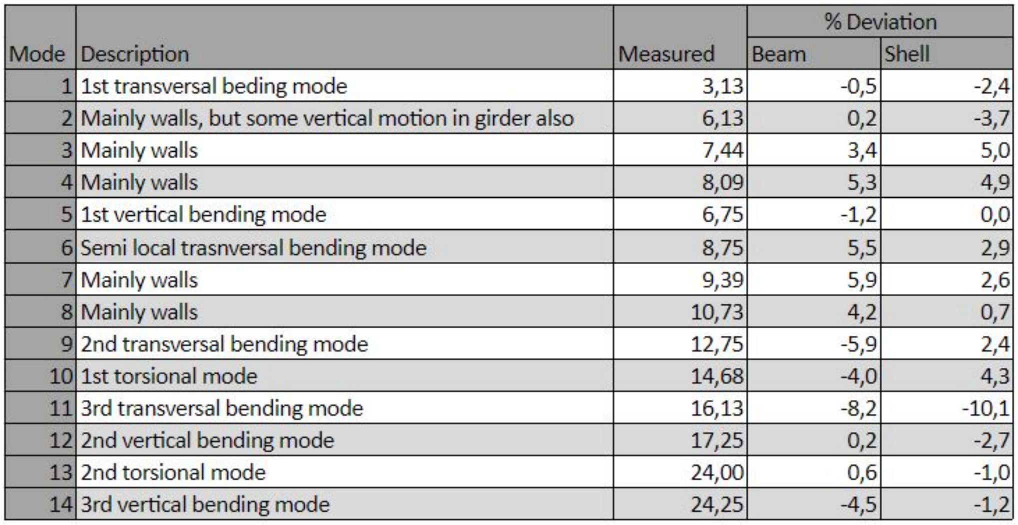
Figure 5.9: Shows the resulting damping ratios when a damping ratio of 0.5% is chosen in $\omega_1 = 19.67\frac{rad}{s}$ and $\omega_2 = 250\frac{rad}{s}$

The specified damping ratios and their respective natural frequencies resulted in the Rayleigh coefficients: $a_0 = 0.1824$ and $a_1 = 3.7e^{-5}$.

5.4 Comparison beam and shell model

The main difference between the beam model and the shell model are the elements used and the representation of the joints and outside stiffeners. The assumptions made for the beam model mathematically reduces the problem to a one-dimensional problem, while the assumptions made for the shell model mathematically reduces it to a two-dimensional problem. For both models, the shape of the stiffeners are simplified, but the shell model allows for a plate to be modelled in the bottom part, as in reality, while the beam model simplifies the plate as a beam. In the beam model, nearly all joints are modelled as completely stiff, while the shell model has included the plates in the connection joints. Other than this, the simplifications made for the two models are fairly similar.

In table 5.9 the 14 first natural frequencies found from the system identification is once again presented, now only with the deviation for the updated beam and shell model. As the table shows, there is no clear pattern in whether one model is stiffer than the other as this varies from one mode to the other. Both models fit the natural frequencies found from the system identification well, without one model sticking out as more accurate than the other.



Mode	Description	Measured	% Deviation	
			Beam	Shell
1	1st transversal beding mode	3,13	-0,5	-2,4
2	Mainly walls, but some vertical motion in girder also	6,13	0,2	-3,7
3	Mainly walls	7,44	3,4	5,0
4	Mainly walls	8,09	5,3	4,9
5	1st vertical bending mode	6,75	-1,2	0,0
6	Semi local trasnversal bending mode	8,75	5,5	2,9
7	Mainly walls	9,39	5,9	2,6
8	Mainly walls	10,73	4,2	0,7
9	2nd transversal bending mode	12,75	-5,9	2,4
10	1st torsional mode	14,68	-4,0	4,3
11	3rd transversal bending mode	16,13	-8,2	-10,1
12	2nd vertical bending mode	17,25	0,2	-2,7
13	2nd torsional mode	24,00	0,6	-1,0
14	3rd vertical bending mode	24,25	-4,5	-1,2

Table 5.9: Shows the 14 first measured natural frequencies and how the natural frequencies of the beam and shell model deviate from these values.

Since both numerical models are found sufficiently accurate for accessing the dynamic behavior of the Hell Bridge and that the procedure of establishing the shell model was significantly more troublesome and time consuming than the beam model, the conclusion is drawn that a rather simple beam model is the best choice for the purpose of finding the global dynamic behavior of the Hell Bridge test arena. It should be noted that the shell model was mainly constructed because it allows for more detailed modelling, which might be necessary to achieve a good representation of the known damage in the Hell Bridge. A detailed introduction to the damage and how this is implemented in the numerical models is thoroughly presented and discussed in chapter 7.

6 Damage identification

In this chapter, the use of modal characteristics for bridge monitoring is investigated. The objective is to establish a time and cost efficient method to identify severe structural faults. The techniques described are based on results obtained with a modal hammer and accelerometers, presented in chapter 3.3.2. Three out of the four methods presented below are previously investigated in the paper *Inspection of steel bridges by modal hammer from bridge deck only* [23]. These methods are further investigated in this chapter and a fourth method, normalization, is introduced.

As described in section 2.4.1, damage will cause a change in the stiffness, mass and energy dissipation of the structure, which for this paper is believed to be reflected by analyzing time series obtained from imposed vibrations. Response is measured from accelerometers located on the damaged and undamaged structural components, providing a basis for comparison. By performing analysis in both time and frequency domain, results are compared and it is established if it is possible to find indication of the damage.

Already before applying any techniques to the acceleration response time series, several observations indicated abnormal behavior of the damaged beam. The person performing the hits with the modal hammer states that he clearly felt, and could hear, the difference between hitting on the beams of the undamaged structural components vs the beam of the damaged structural component. All hits performed directly on the beam of the damaged structural component, i.e. impact location X10 and X9, also far exceeded the sensitivity level of the sensor attached to it, i.e. A06.

In this chapter it's important to have the monitoring set up of the experimental data fresh in mind. Figure 6.1 therefore displays the location of the damage and all sensors and impact locations discussed in the following sections.

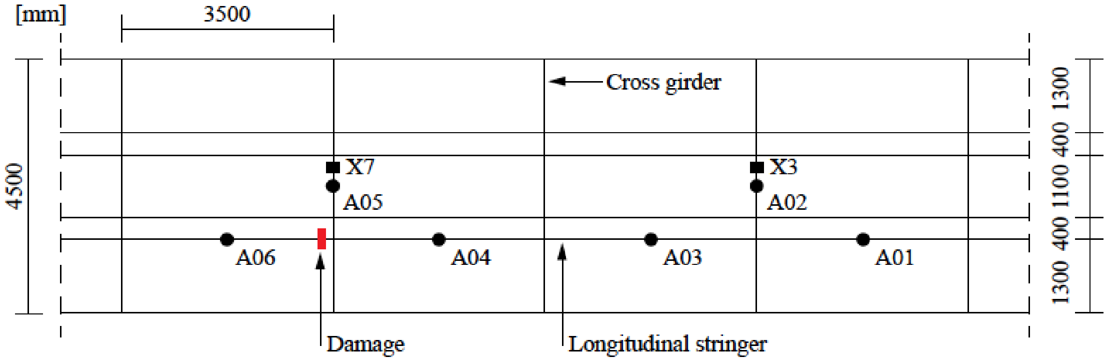


Figure 6.1: Overview over location of damage and all accelerometer and impact locations relevant in this chapter [23].

6.1 Time domain

6.1.1 Moving standard deviation

Figure 6.2 displays the moving standard deviation plots of the acceleration response of sensors A01, A03, A04 and A06 for impact location X7 and X3 with a window length of 21.

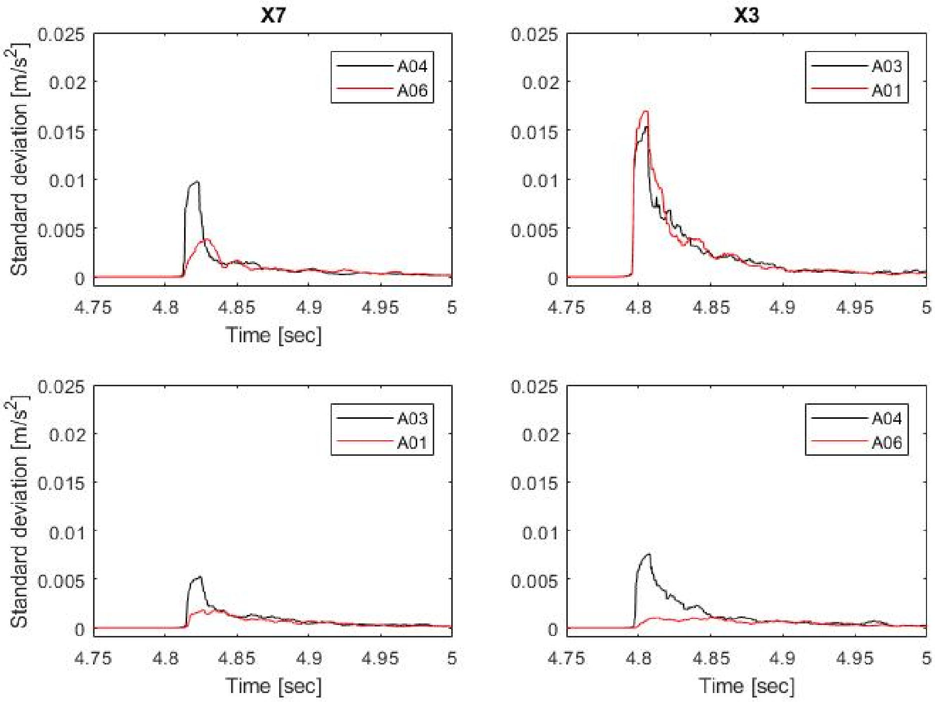


Figure 6.2: Moving standard deviation for relevant sensors, impact locations X3 and X7.

If we first consider the two top plots in figure 6.2 and assume that there is no damage present, we would expect the curves in each respective plot to be similar, i.e. the curves should follow each other closely. This means that the curves of the sensor pair A01 and A03 for impact location X3 should follow each other closely, and similarly for the curves of sensor pair A04 and A06 for impact location X7. The figure clearly shows that this is the case for sensor pair A01 and A03 with impact location X3. The curves of sensor pair A04 and A06 for impact location X7 on the other hand show a clear difference. The response of sensor A06 located on a damaged structural component appears to be significantly lower than the response of sensor A04 located on an undamaged structural component. These observations indicate that the moving standard deviation can be used to detect damage in the connection between stringers and cross girders. It should be noted that the distance between impact location and all four sensors considered is identical and fairly short.

Further only consider the two bottom plots of figure 6.2 where the impact locations are further away from the sensor pairs and the distance between impact location and the sensors in each sensor pair are different. If no damage is present in the components considered, we once again would expect the curves of each respective sensor pair to follow each other closely. As the plots show, this is neither the case for sensor pair A01 and A03 or sensor pair A04 and A06. The observation is made that the curve of sensor A01, i.e. the sensor furthest away from impact location X7, looks very similar to the curve of sensor A06. Since sensor A06 is located on a damaged structural component and A01 on a undamaged structural component this implies that for impact locations further away the response of a undamaged structural component looks similar to the one of a damaged structural component.

Considering the peak value of the curves of sensors located on undamaged structural components only for impact location X7, i.e. A04, A03 and A01, it is clearly observed that the magnitude decreases as the distance between impact location and sensor increases. This means that the energy of the vibrations arriving at the different sensors decreases with distance travelled, a phenomenon covered by the term damping. This observation probably explains why moving standard deviation is able to provide the indication of damage; as there is weak/no contact between stringer and cross girder at the damaged location, the vibrations arriving at sensor A06 have to travel a further distance, resulting in a loss of energy.

All the observations discussed above lead to the conclusion that the moving standard deviation works as a technique for damage identification when the sensors considered are in the vicinity of the impact location. Further it may also be required that the distance between impact location and the sensors compared are equal, but more data with a different set up is required to fully confirm this statement.

As mentioned in section 2.4.1.1.1, the choice of window length in figure 6.3 is set to 21 data points, i.e. the standard deviation value for each point calculated includes 21 elements. The choice of window length was based of plotting the correlation between the moving standard deviation peak values of the two accelerometers being compared for different numbers of sampling points. The sample points range from 3-31 with a step length of 2. Figure 6.3 displays how the correlation between the moving standard deviation peak values only decreases by approximately 10% in the given sampling range for accelerometers A01 and A03 (impact location X3), both placed on undamaged structural components. Similarly, for accelerometer A06 and A04 (impact location X7), where A06 is located on the damaged structural component and A04 on a undamaged structural component, the correlation only goes up with 15% in the sampling range. This proves that the moving standard deviation method is robust with respect to the choice of window length, i.e. a wide specter of window lengths can be chosen without compromising its ability to indicate damage.

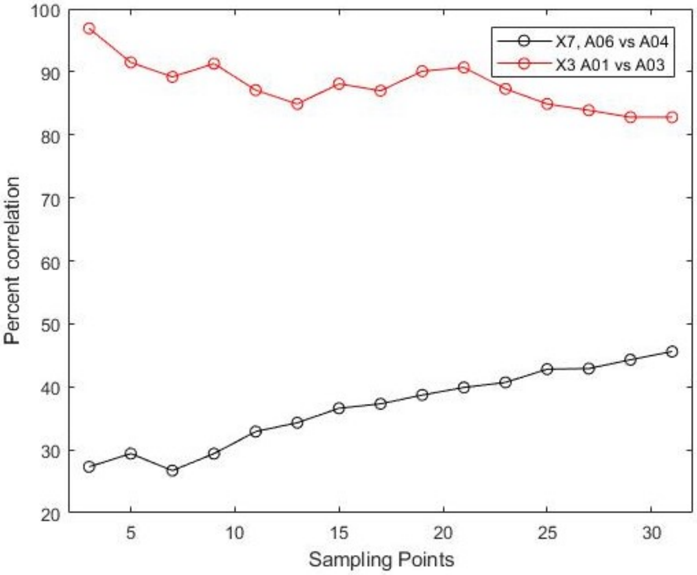


Figure 6.3: Shows correlation between MSD peak values for different window lengths

A higher number of sampling points results in a smoother plot. The choice of sampling points should be high enough to display the trends in the data sets clearly, but not so high that the smoothness compromises the ability to indicate damage.

6.1.2 Cross correlation

Observations made from evaluating the moving standard deviation of the acceleration response are also made by analyzing the cross correlation. Figure 6.4 shows the normalized cross correlation of accelerometer A06, A04, A03 and A01 to accelerometer A05 and A02 for impact location X7 and X3, respectively. A05 is given as reference sensor for impact location X7 and A02 is given as reference sensor for impact location X3. Generally, if no damage is present in the structural system, the cross correlation is expected to decrease with increasing distance between sensor and reference sensor. Therefore it should be mentioned that distance between reference sensor and sensor considered is similar for the two first rows of plots in figure 6.4, while it increases from row 2 to 3 and row 3 to 4, see figure 6.1 if unclear.

Because of the symmetry in the set up used when collecting the experimental data, cross correlation gives the opportunity to compare the results for impact location X3 and X7 to each other. If no damage is present in the structural system, one would expect the top 4 plots in figure 6.4 to show a similar trend. Further, one would expect the two plots in the third row to show a similar trend, and finally the two plots in the fourth row to show a similar trend. All plots are normalized in such a way that the autocorrelations at zero lag are equal to 1. Hence, a correlation of 1 in the plots in figure 6.4 represents a complete match, or 100% correlation.

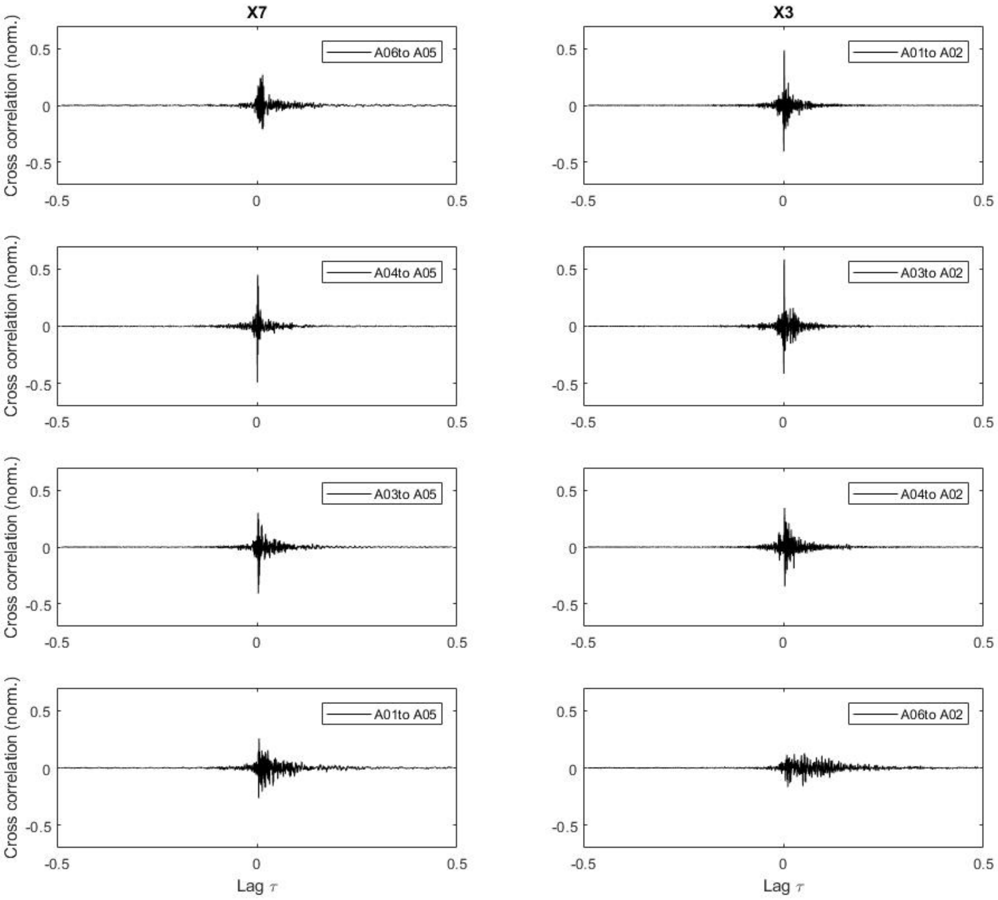


Figure 6.4: Normalized cross correlation for relevant sensors, impact location X3 and X7.

When considering the top four plots, three out of four plots show a similar trend. The plot showing a different trend is the top left plot, giving the cross correlation between sensors A06 and A05 for impact location X7, where the peak correlation is significantly lower. As sensor A06 is the only sensor connected to a damaged structural component, this shows that the cross correlation does correctly identify the presence of damage.

Further, considering the bottom four plots, comparison of the two plots in row three does not indicate any presence of damage, while comparison of the two plots in row four once again indicates damage. The cross correlation for A06 to A02 for impact location has no distinct peak and has a different shape than the cross correlation for A01 to A05 for impact location X7.

From the 8 plots in figure 6.4, two indicate the presence of damage, both for sensor A06, but for different reference sensors and impact locations. For the top left plot the distance between reference sensor and sensor is short, while for the bottom right plot the distance is greater. This indicates that cross correlation provides identification of damage regardless of distance between reference sensor and sensor. The technique is only tested for a short and equal distance between reference sensor and impact location, so to say anything about its applicability for greater distance between the two, further research has to be done.

Once again the decay in amplitude when moving further away from reference sensor probably explains why we are able to detect damage with cross correlation; as the connection between stringer and girder for the damaged structural part is weak/ not present, the vibrations have to travel further, which again leads to a weaker cross correlation.

The exact same setup as discussed above was repeated for 2 more hits at the same impact locations, not presented here, but shown in appendix C. The results look very similar and provide the same information, hence further establishing cross correlation as a good technique for damage identification.

Figure 6.5 shows the time lag between maximum cross correlation for the four sensors equally far away from reference sensor and impact location, i.e A01 and A03 to A02 for impact location X3, and A04 and A06 to A05 for impact location X7. Three bars are plotted for each combination, one for each hit performed. All four combinations have equal distance between sensor, reference sensor and impact location.

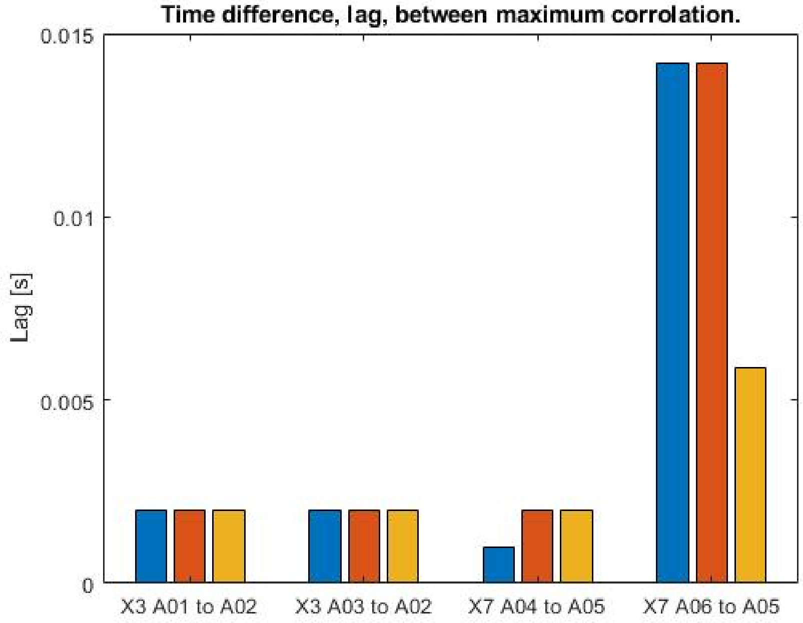


Figure 6.5: Time lag; A01 and A03 to A02 for impact locations X3 and A04 and A06 to A05 for impact location X7.

If no damage is present in the structural system, all time lags in figure 6.5 should be fairly similar. As displayed in figure 6.5, this is not the case for sensor A06 located on the damaged structural component. This observation is strongly correlated with the assumption that the vibrations have to travel a further distance for the damaged structural part.

6.1.3 Normalization

Figure 6.6 shows the normalized acceleration response of sensors A01, A03, A04 and A06 for impact locations X3 and X7. Originally it would be expected that if there is no damage present in the structural system, all curves should show a similar trend with the same decay in signal with time. From considering the 6 plots of the sensors located on undamaged structural components, i.e. all sensors but A06, the bottom right plot clearly shows that this is not the case. It is observed that as the distance between impact location and sensors increases, the damping in the signal decreases. This observation indicates that for this technique to be able to indicate damage, the sensors considered have to be equally far away from the impact location.

From the above we further expect that if there is no damage present, the four top plots should show a similar trend, the two plots in the third row should show a similar trend and finally the two plots in the fourth row should show a similar trend. If there is damage present the response of the damaged structural component should show a clear difference from the undamaged structural component.

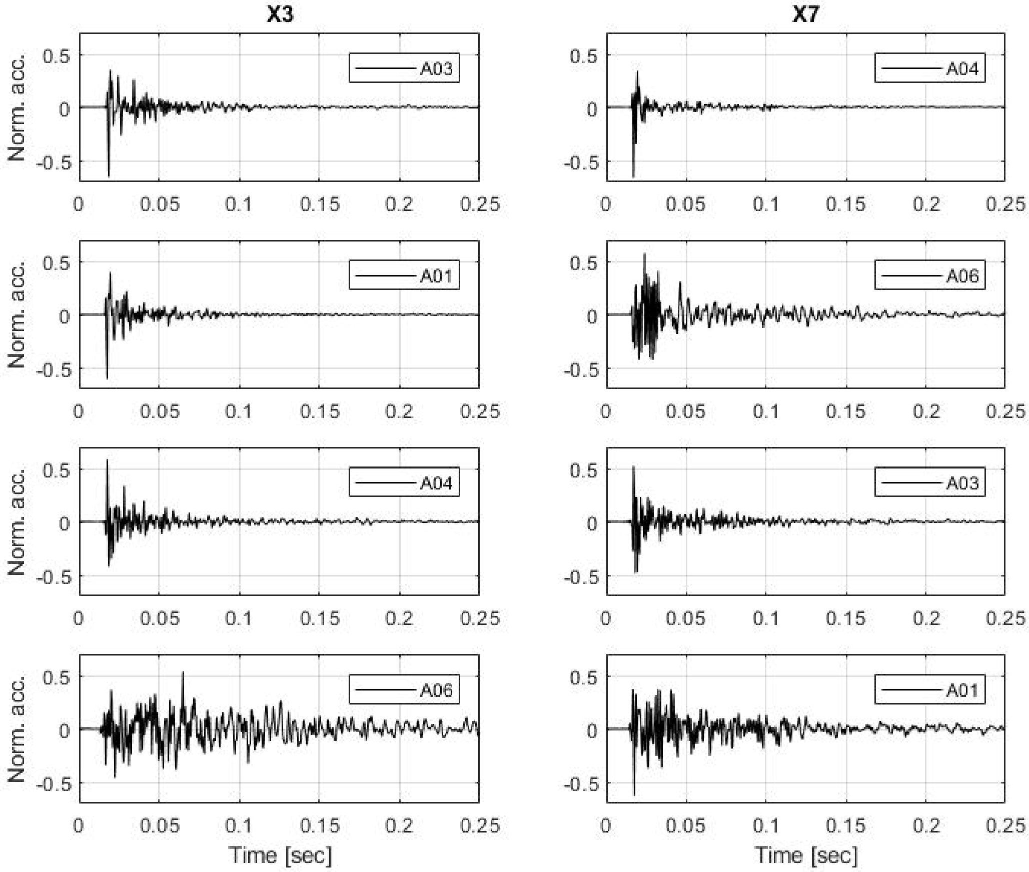


Figure 6.6: Normalized acceleration plots of relevant sensors, impact locations X3 and X7.

From doing these comparisons the following observations are made: 1) normalization works as a technique for damage identification when the sensors considered are in the vicinity of the impact location, 2) it seems to be required that the distance between impact location and the sensors compared are equal and 3) for impact locations further away from the sensor considered the response of an undamaged structural component looks similar to the one of a damaged structural component (see two bottom plots).

6.2 Frequency domain

6.2.1 Frequency response function

The FRFs established for sensors A01, A03, A04 and A06 for impact locations X3 and X7 are shown in figure 6.7. For sensors A01 and A03 at impact location X3 and X7, it is expected that the response is fairly similar since both sensors were located on undamaged structural components and close to each other. Furthermore, to be able to indicate damage similar behavior in response is not expected for sensors A04 and A06 at impact location X3 and X7, as sensor A06 is located on the damaged structural component.

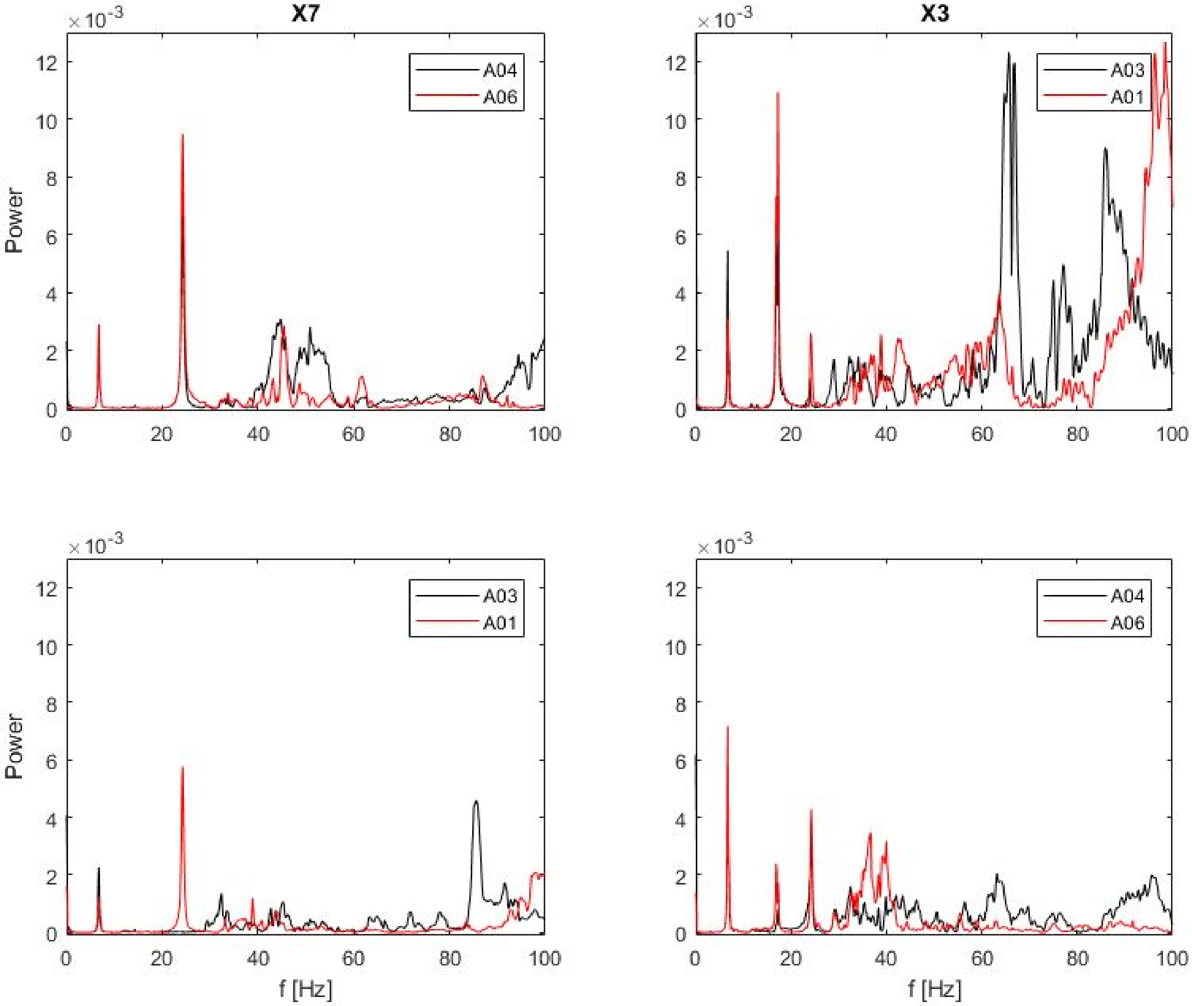
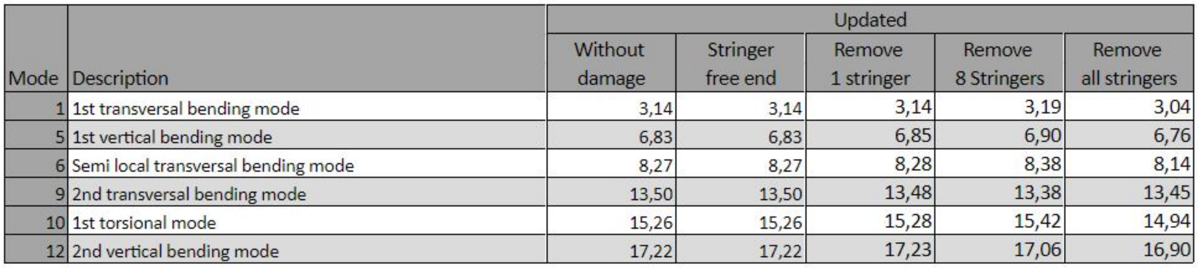


Figure 6.7: FRF of sensors A01, A03, A04 and A06 for impact locations X3 and X7.

From looking at the different plots in figure 6.7, we clearly observe the first and third vertical bending modes from the two first peaks for the two plots on the left-hand side. Similarly for the two plots on the right-hand side, the three first vertical bending modes are located. Except from these observations it's hard to find any valuable information from the figure, but it's still believed that the information about the damage should be there. To provide a better basis for evaluating the results, an effort was made to get a better understanding of the dynamic behavior of the bridge.

First, the known damage was modelled in the beam model by allowing one end of the stringer in the damaged structural component to move freely, i.e. behave like a cantilever. An eigenvalue analysis was conducted to see the effect of the damage on the global modes of the structure. Further, 3 more eigenvalue analysis were performed when removing the whole stringer, removing all eight stringers connecting at mid span, and finally completely removing all stringers and adding the weight of removed steel as nonstructural mass. The results for the first six modes with significant motion in the girder are displayed in table 6.1.



Mode	Description	Updated				
		Without damage	Stringer free end	Remove 1 stringer	Remove 8 Stringers	Remove all stringers
1	1st transversal bending mode	3,14	3,14	3,14	3,19	3,04
5	1st vertical bending mode	6,83	6,83	6,85	6,90	6,76
6	Semi local transversal bending mode	8,27	8,27	8,28	8,38	8,14
9	2nd transversal bending mode	13,50	13,50	13,48	13,38	13,45
10	1st torsional mode	15,26	15,26	15,28	15,42	14,94
12	2nd vertical bending mode	17,22	17,22	17,23	17,06	16,90

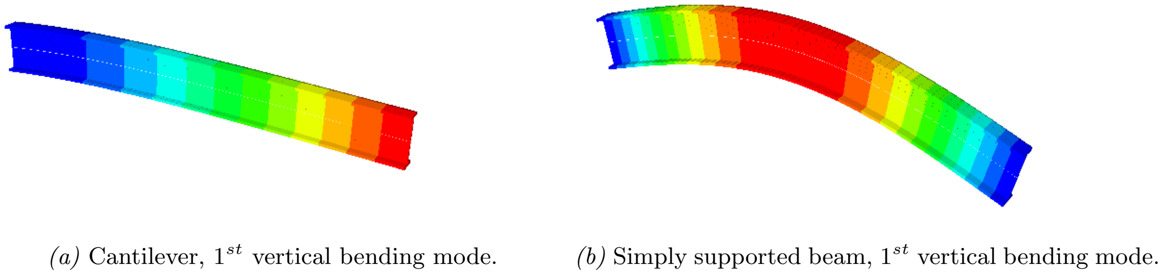
Table 6.1: Effect of different levels of damage on global modes. Frequencies are given in Hertz [Hz].

Table 6.1 shows that the stringers have little effect on the natural frequencies of the global modes. From this it's concluded that the known damage in the girder of the Hell Bridge is not big enough to impact the global response of the bridge.

To be able to detect damage one would therefore have to look further into local modes in the girder of the bridge. As will be presented in chapter 7, the connection between stringer and cross girder are made up by L-profiles, only physically connected in the webs of the respective beams. This in combination with the fact that I-profiles (such as the cross girders) generally have low torsional stiffness leads to the conclusion that if only looking at the stringer of the undamaged structural components, they are best represented by a simply supported beam. For the stringer of the damaged structural component its believed that the stringer is best represented by a cantilever.

Based on the discussion above, two simple eigenvalue analysis's were performed on beams with the same length and cross-sectional properties as the stringer; one of a simply supported beam, and one of a cantilever, i.e. beam clamped at one end and completely free to translate and rotate at the other end. The goal of this approach was to give an indication of where to look for the local vertical bending modes of the damaged and undamaged structural components.

The first three vertical bending modes of the simply supported beam were found to be 87.3 Hz, 307.5 Hz and 593.2 Hz, and similarly for the cantilever; 31.9 Hz, 174.6 Hz and 423.7 Hz. The first vertical bending mode for each case is presented in figure 6.8 (a) and (b). Not that the natural frequencies of the cantilever are significantly lower and that the difference should be possible to see in the FRFs. By the same arguments as for why the stringers of the undamaged structural components are best represented by simply supported beams, it is argued that the natural frequencies of the cantilever will probably be somewhat lower than the ones presented above.



(a) Cantilever, 1^{st} vertical bending mode. *(b)* Simply supported beam, 1^{st} vertical bending mode.

Figure 6.8: 1^{st} vertical bending modes for a cantilever beam and a simply supported beam.

With the new knowledge, the exact same plots as shown in figure 6.7 are shown in figure 6.9, only for a wider frequency range, 0-600 Hz. The magnitude of the peaks in the top right plot was so much higher than the three remaining plots, that the scaling on the y-axis is different for this plot.

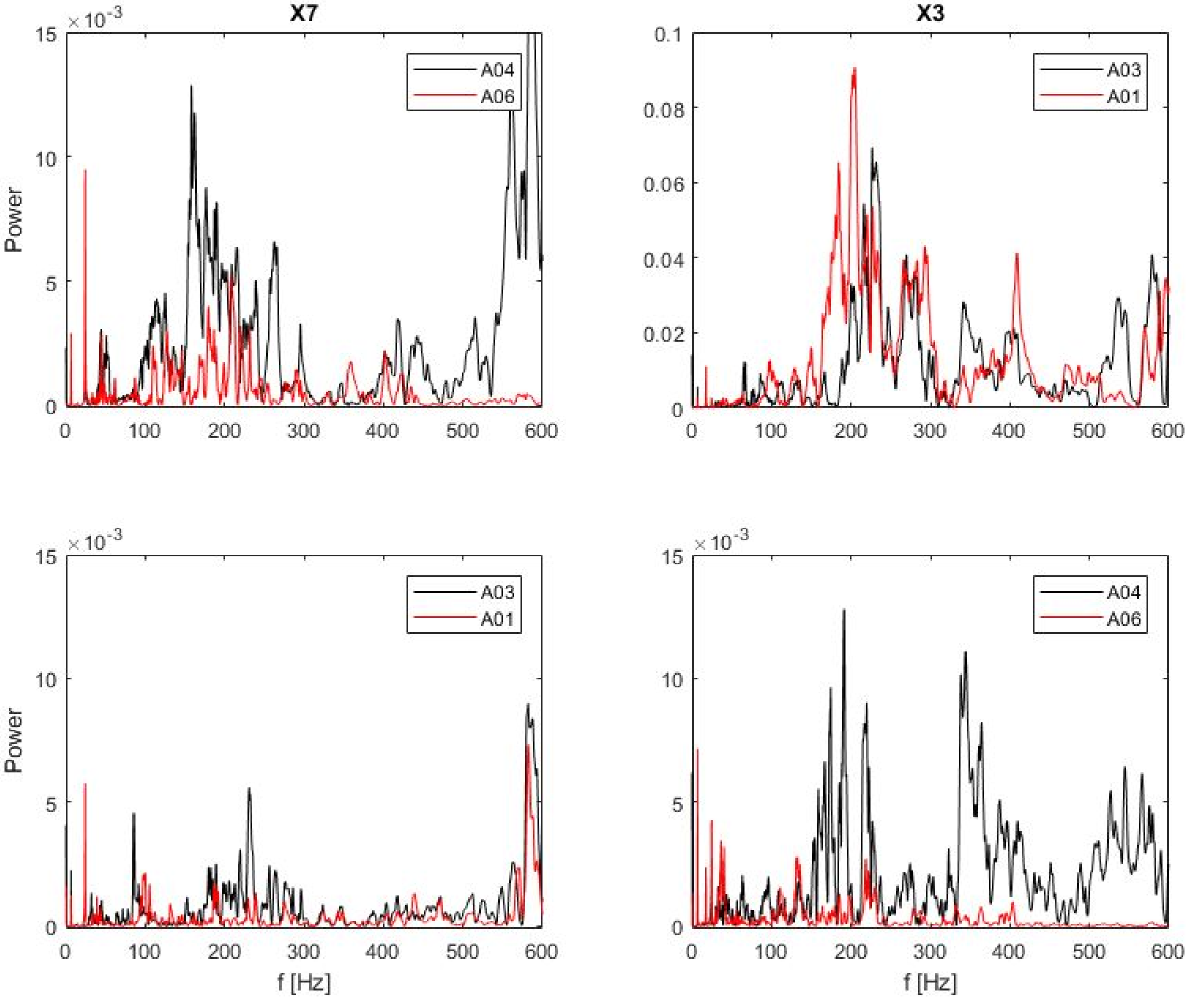


Figure 6.9: FRF of sensors A01, A03, A04 and A06 for impact locations X3 and X7

As figure 6.9 displays, none of the curves of the sensor pairs in each respective plot follow each other closely, and there is generally too much going on in the plots to see any clear similarities or differences. This makes it close to impossible to use the plot for damage identification.

A possible explanation for why the FRFs are so messy and why FRFs expected to look similar don't is that when performing the hits with the modal hammer, they were performed on top of the bridge deck while most accelerometers were attached to the underside of the stringers. This in combination with the wooden sleepers resting on top of the stringers is believed to introduce noise effects in the stored signal that degrades the FRFs. Also keep in mind that the analysis of the cantilever and simply supported beams are simplifications of how they act in reality, therefore only giving a possible indication of where to find the local natural frequencies.

The discussion above shows that even with a good understanding of both local and global modes of a structure, it's still hard to indicate the presence of damage by using the FRFs. Filtering the signal or applying more advanced methods to extract the modal parameters would be possible approaches that could be subjected to further work.

6.3 Discussion and summary

This chapter presented a method for performing bridge inspection from the bridge deck by investigating modal characteristics obtained from imposed vibration. The method uses analysis techniques in both time and frequency domain. The transmission of the signal as seen in the time domain mainly describes local phenomena, i.e. local structural response. The FRFs on the other hand generally describes both local and global structural response. Changes in the global frequencies should be possible to detect if the damage present is big enough to change the global structural stiffness. As shown by a numerical model of the Hell Bridge, serious damages may be present without it affecting the global natural frequencies. Damage detection by looking at global response alone is therefore not advised.

The trends established from the time domain techniques are not seen in the frequency domain technique. Three possible explanations for this are; 1) the damaged structural component causes a change in the transmission of the signal that is seen primarily in time domain, 2) the change in transmission of the signal should be represented somewhere in the frequency range of 20-500 Hz, but is not seen because of noise in the stored signal, and 3) the results for indication of damage by the use of FRFs is highly dependent on the damping in the structure, making it difficult to detect this type of damage. Using more advanced analysis methods for indicating this type of damage in frequency domain should be subjected to further work.

In all damage identification techniques presented, the procedures are based of comparing the response of different sensor pairs located on damaged and undamaged structural components. In all applications where the different techniques prove to work, the distance between impact location and sensors compared is identical. Since the reason for them being able to provide the identification of damage is believed to be because the vibrations have to travel a further distance for the damaged structural component, this strengthens the belief that an equal distance between impact location and sensors compared is required. For the cross correlation technique it will also be required that the distance between reference sensor and sensor is equal.

In the presentation of the cross correlation technique it is stated that the results indicate that it works regardless of distance between sensor and reference sensor. From the discussed in the previous paragraph this will only be within reasonable measures. Since the cross correlation decreases with increased distance between sensor and reference sensor there will be a point where the distance further travelled for the vibrations to the sensor of the damaged structural component becomes negligible with respect to the total distance travelled. At this point there will be no visual difference between the cross correlation of the damaged structural component and the undamaged structural component and the technique will no longer be able to detect damage.

Based on the results obtained, the following conclusions are drawn:

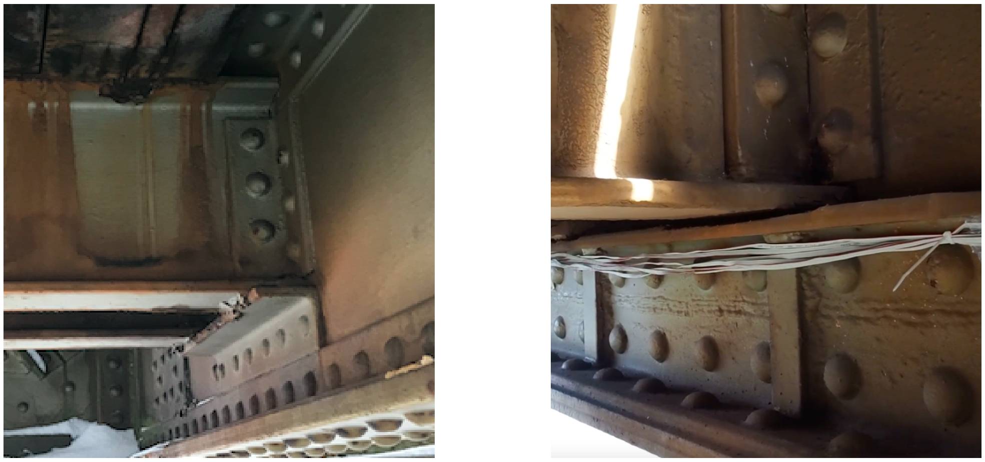
- Indication of damage is found when analyzed in time domain and not in frequency domain. The indication of damage is seen from comparison between undamaged and damaged structural components.
- Analysis techniques such as moving standard deviation and normalization provide indication of damage when the impact location is in the vicinity of the sensors being compared. Cross correlation is only tested for a short distance between impact location and reference sensor, but provides the indication of damage for both short and greater distance between sensor and reference sensor.
- The type of damage investigated is difficult to observe in the frequency domain. This is believed to be due to 1) the many global and local natural frequencies indicated by all the sensors which makes distinction hard to find and 2) noise in the signal, which probably has a greater impact on the results in frequency domain than time domain.
- The time domain analysis techniques presented are considered feasible for indicating damage from the bridge deck. Application of the method to indicate damage may be used to confirm the presence of damage and strengthen the decision for selecting areas that should be subjected to further investigation.

7 Implementing damage in numerical models

With the different techniques for damage identification from the bridge deck established in chapter 6, the known damage was implemented in the numerical models in different ways, with the goal to obtain the same results. This chapter thoroughly presents the damaged structural component and the implementation of the damage in the numerical models. Two possible ways to represent the damage numerically are presented. The first way was done with a spring formulation, both linear and nonlinear, and was only implemented in the beam model. The second way was to define a friction contact pair formulation provided by Abaqus, only implemented in the shell model.

7.1 Damage

Figure 7.1 (a) and (b), show pictures of an undamaged structural component and the damaged structural component investigated, respectively.



(a) Undamaged structural component

(b) Damaged structural component

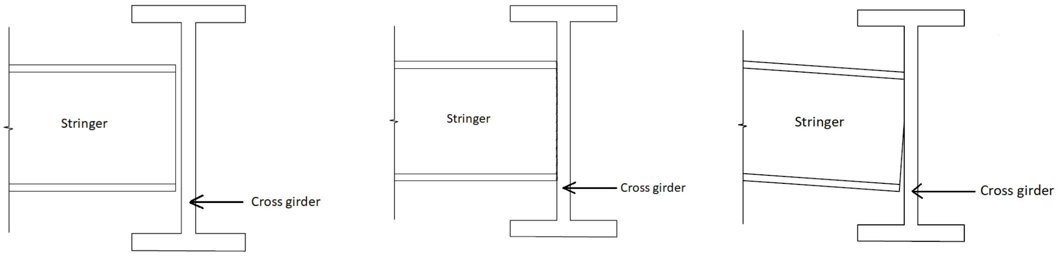
Figure 7.1: a) shows one of the undamaged connection joints between stringer and cross girder. (b) shows the damage joint connection between stringer and cross girder further investigated in this paper.

As displayed in figure 7.1, the connection consists of three L-profiles, two connecting the webs of the stringer and cross girder together, only one shown in the pictures above, and the last one lays underneath the stringer, only connected to the web of the cross girder. Figure 7.1 (a) show that the bottom flange of the stringer in the undamaged structural components rests on top of the L-profile lying underneath and that the L-profile connecting the webs is fully intact. Figure 7.1 (b) on the other hand shows that there is a considerable glitch between the bottom flange of the stringer and the L-profile lying underneath for the damaged structural component and that the L-profile connecting the webs is cracked in the angle. Further inspection of the bridge under train load shows that there is considerable vertical motion of stringer relative to the cross girder, nearly completely closing the gap shown in figure 7.1 (b).

These observations indicate that the stringer behaves somewhat like a cantilever for displacements smaller than the glitch between the L-profile and bottom flange of the stringer. If there is no contact between stringer and cross girder the stringer will act completely as a cantilever, and if there is some sort of contact between stringer and cross girder, direct contact or contact because of different particles stuck in between, the stringer will behave somewhere between a cantilever and a beam clamped at one end and simply supported at the other, probably closer to the cantilever. Possible scenarios are listed below:

- For displacements of stringer not closing the glitch:
  - No contact between stringer and cross girder. Stringer behaves completely like a cantilever.
  - Contact between stringer and girder. Normal force on stringer and contact surface remain constant.
  - Contact between stringer and cross girder. Distortion leads to bigger normal force on stringer, but contact surface decreases.
- For displacement of stringer closing glitch:
  - Same three as above until glitch is closed. Clamped at one end and simply supported at the other when glitch is closed.

The three different cases without the connecting L-profiles are physically presented in the drawings in figure 7.2



(a) No contact *(b)* Contact *(c)* Distorted contact

Figure 7.2: Physical representation of the three different cases discussed above. (a) shows no contact, (b) shows contact and (c) very exaggerated shows contact for a distorted stringer

Although inspection of the bridge during train load shows that the glitch nearly, or completely, closes, the question arises if this case is represented by data available in this thesis. For the experimental data collected, the forces from the hammer impacts are significantly smaller than the forces the bridge is subjected to during train load. It could therefore be argued that the case where the glitch closes is not relevant for this thesis as the data collected won't represent this case. This is further discussed in section 7.2

For small displacements of the stringer relative to the cross girder, i.e. before the glitch closes, we are probably faced with a linear problem, possibly with some smaller nonlinear effects from uneven friction forces. The completely linear problem can be visualized by the free harmonic oscillation of a cantilever. The smaller nonlinear effects may arise due to dirt particles stuck between stringer and L-profile and by the effect of distortion discussed above. For displacements closing the gaps we are faced with a highly nonlinear problem as the stringer now is vertically supported at both ends. This can be visualized by an oscillating cantilever where when the displacement reaches a specific value, the free oscillation is interrupted by an impact in the opposite direction of movement. The impact being contact between bottom flange of stinger and the L-profile underneath, not allowing the stringer to move any further.

In all attempts to implement the damage numerically the vertical translation degree of freedom (DOF), i.e. y-direction in figure 7.3, and the rotational DOF about the transversal direction, i.e. z direction in figure 7.3, of connection between stringer and cross girder for the damaged structural component were opened. Keep in mind that for both numerical models, the cross girder is set as master node/surface, stringer as slave node/surface and that for all other connections between stringer and cross girder, all 6 DOFS of the stringer are constrained to follow the cross girder. Further, springs with varying stiffness and contact formulations were introduced, presented in section 7.3 and 7.4.

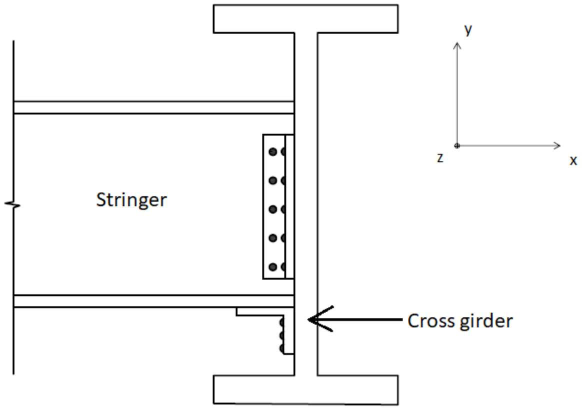


Figure 7.3: Shows a drawing of the connection between stringer and cross girder for an undamaged structural component and its reference coordinate system.

7.2 Data setup and input force

To evaluate how successful the different ways of implementing the damage numerical were, the same techniques, as presented in section 2.4.1 (theoretical) and chapter 6 (results), were used for the numerically simulated acceleration response and compared to the analysis of the full-scale measurements. The exact same accelerometer set up, impact locations and sampling frequency (2048 Hz) as for the full-scale measured acceleration data was also used for the numerical simulations. Due to run time of the numerical analysis, the durations of all simulations are only 2 seconds, compared to 30 seconds for the experimental data. This is no problem for the time domain techniques, as all results presented in chapter 6 only use 0.5 seconds of the acceleration records or less. For the frequency domain technique on the other hand, more data points would preferably have to be available to establish good and reliable FRFs. The effects of a shorter record are discussed in section 7.4.1.2.1.

When plotting the measured impulse for impact locations X3 and X7 it was found that all impacts exceeded the sensitivity of the sensor on the hammer tip. Without the full information of the impulse it´s impossible to replicate it, so a simplification was used. As it was found that the impulse was mainly introduced over 5 increments in the experimental data, the acceleration in the numerical models was therefore also imposed by introducing a triangular impulse over five increments with a peak magnitude of 22.3 kN, the sensitivity limit of the sensor on the hammer tip.

Although we don't have full knowledge about the magnitude of the input force, we know by the shape of it and linear regression that the magnitude of most impacts is between 22.3 and 25 kN. To gain some knowledge of how much the stringer of the damaged structural component moves with respect to the cross girder for an impulse of magnitude 22.3 kN, the displacement of the node at the end of the stringer of the damaged structural component and the displacement of the node at where the stringer would attach to the cross girder if no damage present was extracted. It was found that max displacement between stringer and cross girder was 0.4 mm for the numerical simulations when the damage was modelled with no contact between stringer and cross girder. As the glitch for the damaged structural component discussed earlier in this chapter is at least 5 mm, this shows that the glitch is probably not even close to closing for the experimental data and that we are not faced with this part of the problem in this thesis.

7.3 Spring formulation

As mentioned earlier, the damage was implemented in the beam model using three different spring formulations; 1) linear spring with zero stiffness, i.e. same as no spring, 2) linear spring with considerable stiffness and 3) nonlinear spring. The idea of the first approach was to represent the case where there is no contact between stringer and cross girder, presented in figure 7.2 (a). The idea of approach 2 and 3 was to represent the case where there is contact between stringer and cross girder, presented in figure 7.2 (b) and (c).

For the first formulation with zero spring stiffness, no further work than setting the spring stiffness to zero and allowing the damaged end of the structural component to rotate and move free vertically was done before running the analysis.

For the second formulation with a linear spring, an appropriate stiffness had to be found before running the analysis. To find an indication of the spring stiffness, the stiffness contribution of the shear force in figure 7.4 was calculated from equation (7.1), believing that the stiffness should be somewhere between zero, representing no contact, and the calculated value, representing full contact.

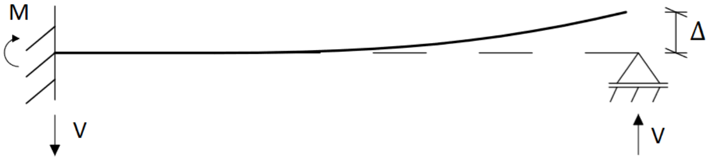


Figure 7.4: Stiffness relation for cantilever.

$$V = \frac{3EI}{L^3}\Delta \tag{7.1}$$

With $E = 210GPa$, $I = 1.961e^{-4}m^4$ and $L = 1.75m$, the stiffness contribution of the shear force was found to be approximately 23 MN/m. Although the stringers are 3.5m long, L=1.75 m was used since all stringers are connected with a u-profile at mid span. After attempting different spring stiffness's between 0 and 23 MN/m, the choice landed on a stiffness of 10 MN/m. From the reasoning above, this value should be high enough to represent something different than no contact, but still well below the stiffness representing full contact.

Finally, the third spring formulation was performed with different non-linear springs. Because these formulations gave few meaningful results, they are only presented in appendix D. The theory used in the non-linear spring formulations is therefore not included in this paper, but is represented in [24] and [25].

7.4 Friction contact pair

As previously described, the cross girders serves as master surface in the tie constraint between stringer and cross girder in the shell model. In the case of a contact formulation, the smallest surface, i.e. the penetrating surface, should ideally be the master surface[19]. In the contact formulation between stringer and cross girder, the stringer should therefore be assigned master surface. Unfortunately, it's only possible to have one master surface in each tie constraint. Since all connection joints between stringer and cross girder, except the ones at the ends, have a stringer attaching to the web of the cross girder from two sides, the stringers can not be assigned master surface. This proved to introduce a challenge as when attempting to run the contact formulation with the web of the cross girder as master surface and the end of the stringer as slave surface the simulation was interrupted by two error messages. The first message stated that a general shell surface can not be used in both a tie constraint and a contact formulation simultaneously. The second error stated that the end of the stringer, defined as an edge, can not be used as slave surface in contact formulation. To overcome these two error messages the following was introduced; first a deformable slave surface of shell elements was attached to the end of the stringer of the damaged structural component and secondly a analytical rigid master surface was attached to the web of the cross girder, see figure 7.5.

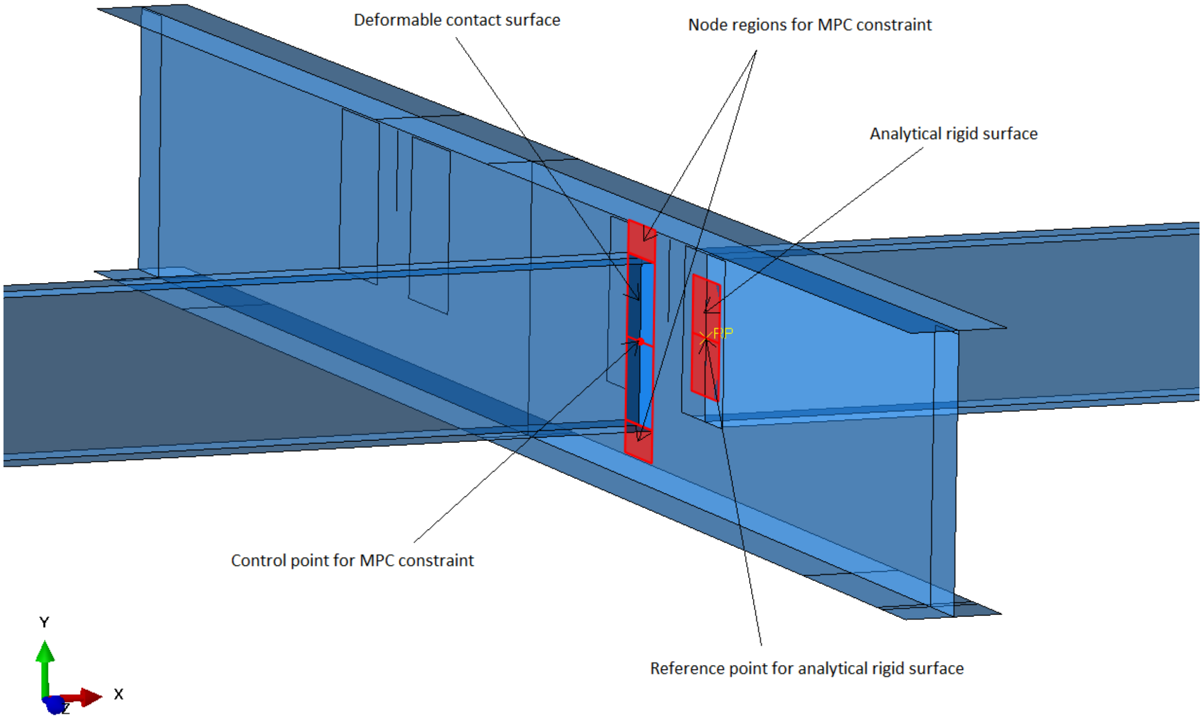

Deformable contact surface
Node regions for MPC constraint
Analytical rigid surface
RP
Control point for MPC constraint
Reference point for analytical rigid surface
Y
X
Z

Figure 7.5: Shows the small model where the friction contact pair was successfully implemented.

Further the friction contact formulation with the deformable slave surface and the analytical rigid master surface was defined based on the penalty method described in [**?**]. Finally, a boundary condition was introduced pushing the surfaces together, generating the friction force. With the contact formulation established there are still three parameters affecting the results of the simulation that are easy modified:

- Friction coefficient between the surfaces.
- Boundary condition displacement (defining level of contact between stringer and cross gider).
- Material stiffness of the deformable surface. Most likely close to 2.1 GPa.

The problem converged in a small model local model of two stringers and a cross girder, see figure 7.5, but when expanded to the global model, penetration errors occured and the convergence criterion was not met. Attempts were made to fix this, but with no luck. It's hard to say why the penetration errors occured as the field of contact mechanics requires extensive knowledge when applied to numerical models. Because there was a need to limit the scope of the work in this thesis and as the other methods for representing the damage showed promising results, no further work was done for the friction contact pair formulation.

7.5 Simulation results

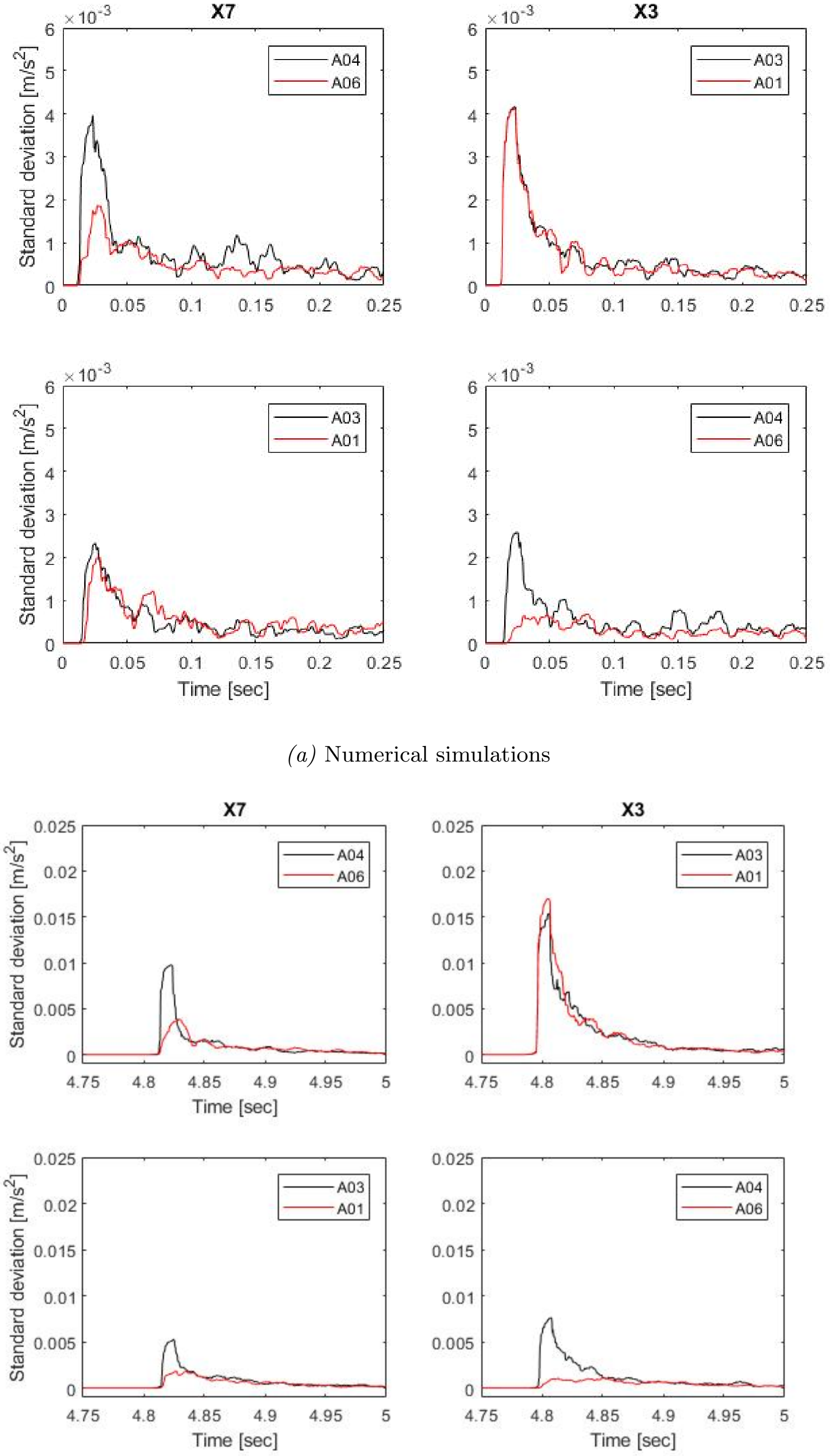
This section presents the results of the different damage identification techniques when applied on the numerically simulated acceleration data. These results are compared to the ones of the full-scale measurements presented in chapter 6 and the results are further discussed. When making the comparison it is not expected that the results of the simulated data look identical to the ones of the full-scale measurements, but that they show a similar trend and provide the same information. Expecting the results to be identical is unreasonable as the impulses are not identical and the damping in the numerical model is seriously simplified. For a thorough presentation of what to expect from the different plots, see chapter 6.

7.5.1 No spring

7.5.1.1 Time domain

7.5.1.1.1 Moving standard deviation

Figure 7.6 (a) and (b) displays the moving standard deviation plots of sensors A01, A03, A04 and A06 for impact location X7 and X3 with a window length of 21. The plots in (a) are based of the numerical simulations, and (b) of the full-scale measurements.



(a) Numerical simulations

(b) Full-scale measurements

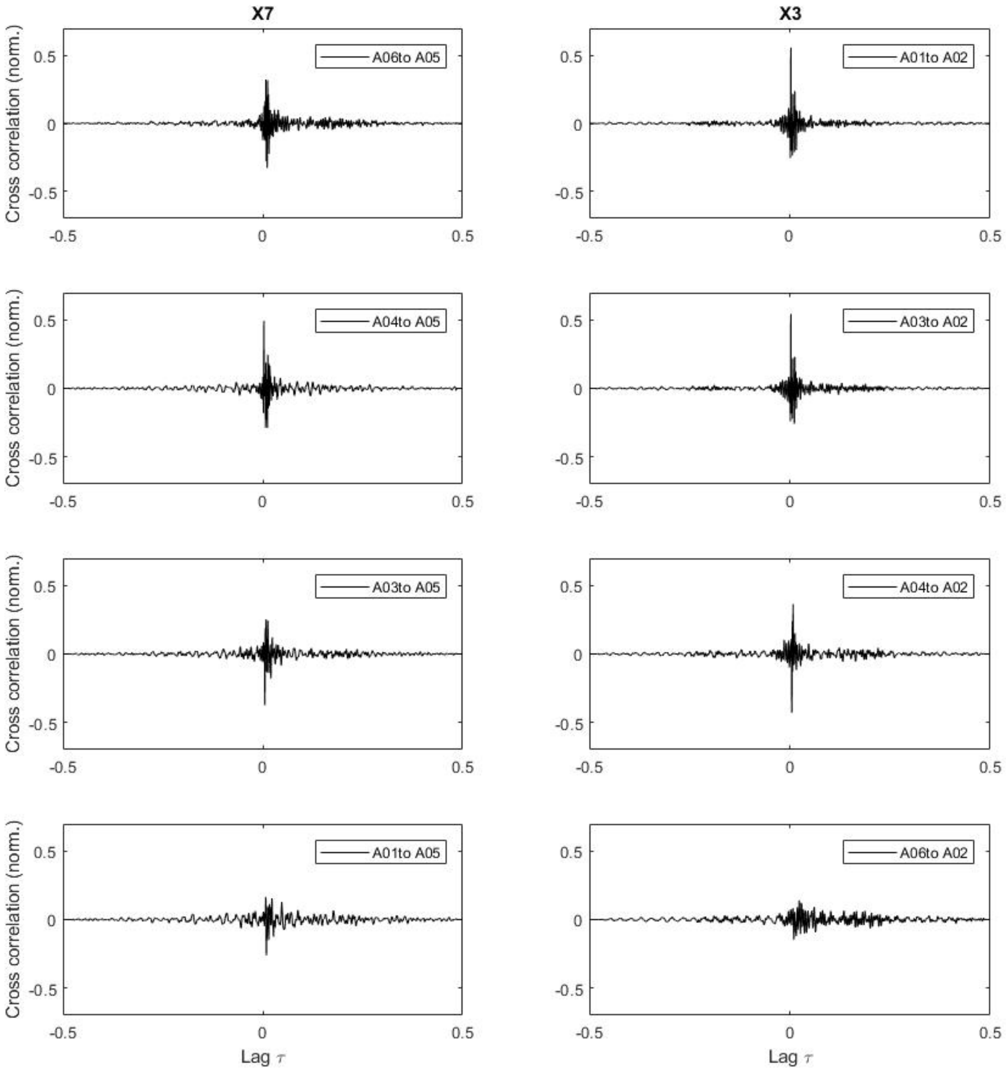
Figure 7.6: Shows moving standard deviation for relevant sensors, impact locations X3 and X7, (a) based on the simulated acceleration data and (b) based on the full-scale measurements.

If we first consider the two top plots in both figures, they provide the exact same information, i.e. identification of damage for the structural component A06 is attached to. Further if we consider the two bottom plots in both figures, the bottom right plots look very similar, but the bottom left deviate from each other. Whereas for the full-scale measurements the technique only provides identification of damage for the components considered in the vicinity of the impact location, it also provides identification of damage for a greater distance between impact location and components considered for the numerical simulations.

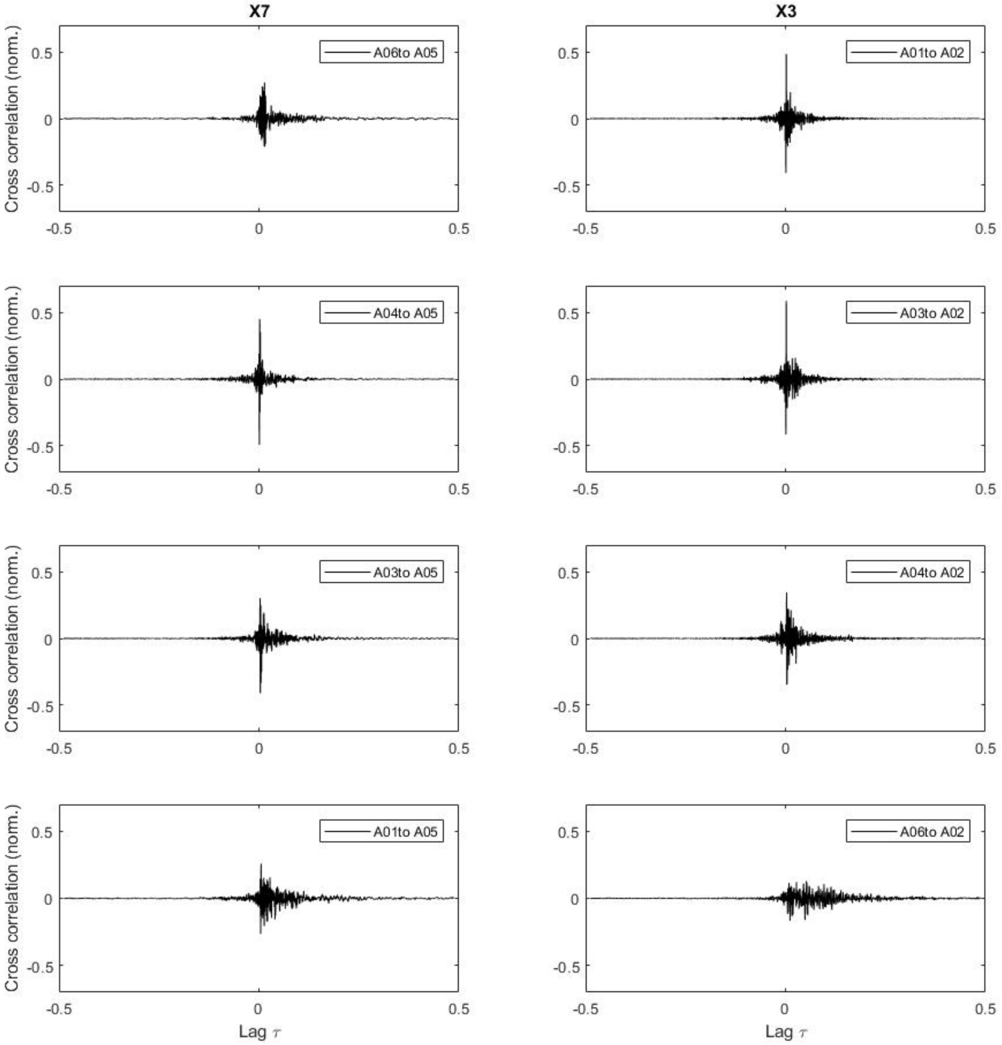
Note that the standard deviation in both figures is approximately halved going from the two top plots to the two bottom plots, i.e. for impact locations close to accelerometers compared vs further away. This might be an indication that the overall damping in the numerical model is reasonable.

7.5.1.1.2 Cross correlation

Figure 7.7 shows the normalized cross correlation for accelerometer A06, A04, A03 and A01 to accelerometer A05 and A02 for impact location X7 and X3, respectively. The plots in (a) are based of the numerical simulations and (b) of the full-scale measurements.



(a) Numerical simulations



(b) Full-scale measurements

Figure 7.7: Shows normalized cross correlation for relevant sensors, impact location X3 and X7, (a) based on the simulated acceleration data and (b) based on the full-scale measurements.

From comparison, both figures provide the exact same information, i.e. the indication of damage for the structural component accelerometer A06 is attached to. Although the peak correlation is significantly higher for the undamaged structural component, it could be argued that for the sensor far away from reference sensor, the identification of damage is somewhat less clear for the normalized cross correlation of the numerical simulations, see two bottom plots in both figure 7.7 (a) and (b).

Similarly as in section 6.1.2, the time lag between maximum cross correlation for the four sensors equally far away from reference sensor and impact location, i.e A01 and A03 to A02 for impact location X3, and A04 and A06 to A05 for impact location X7, were plotted. Figure 7.8 shows that as for the full-scale measurements, the results display a significant increase in time lag for the damaged structural component for the numerical simulations.

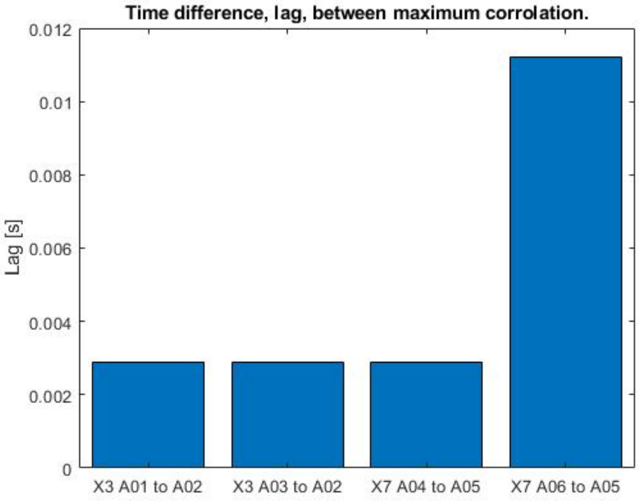
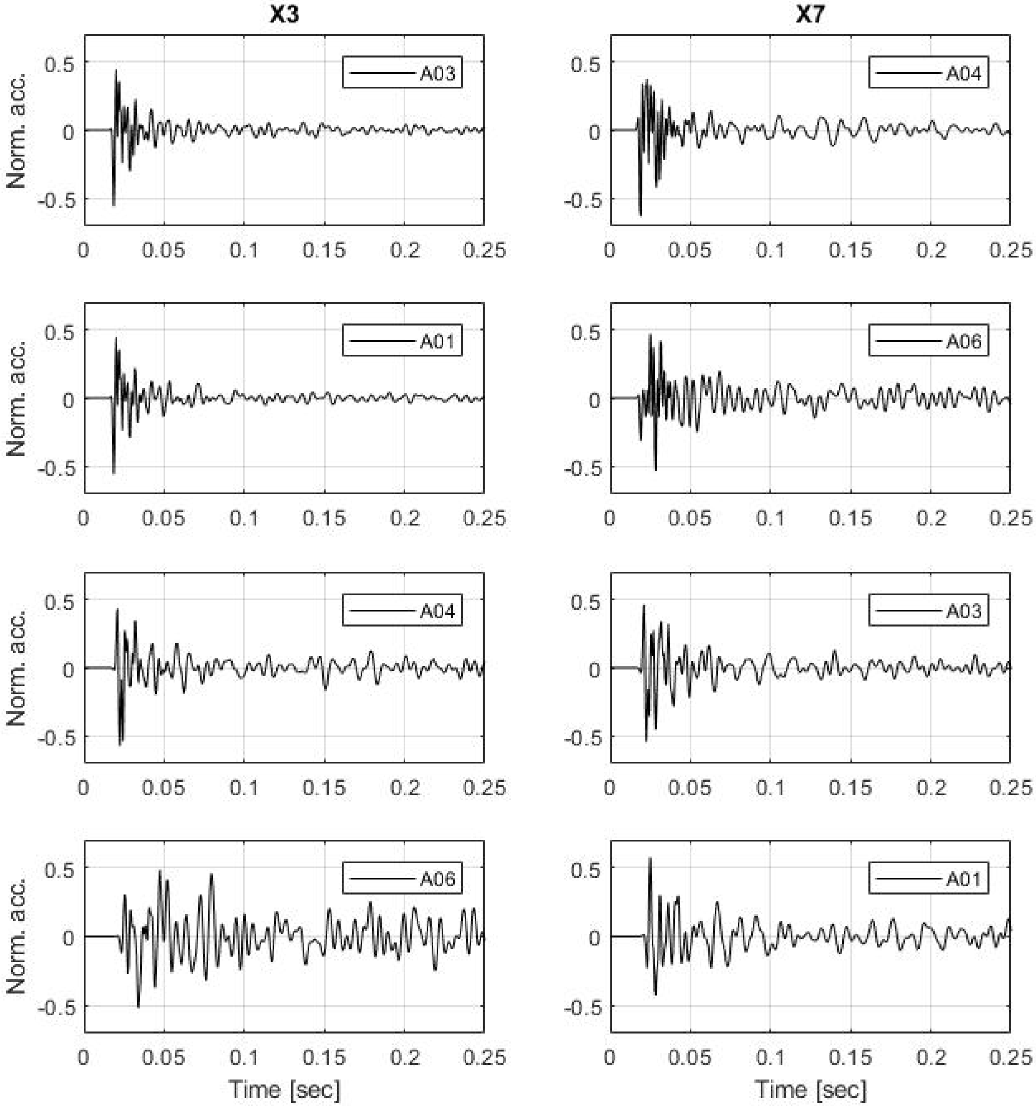


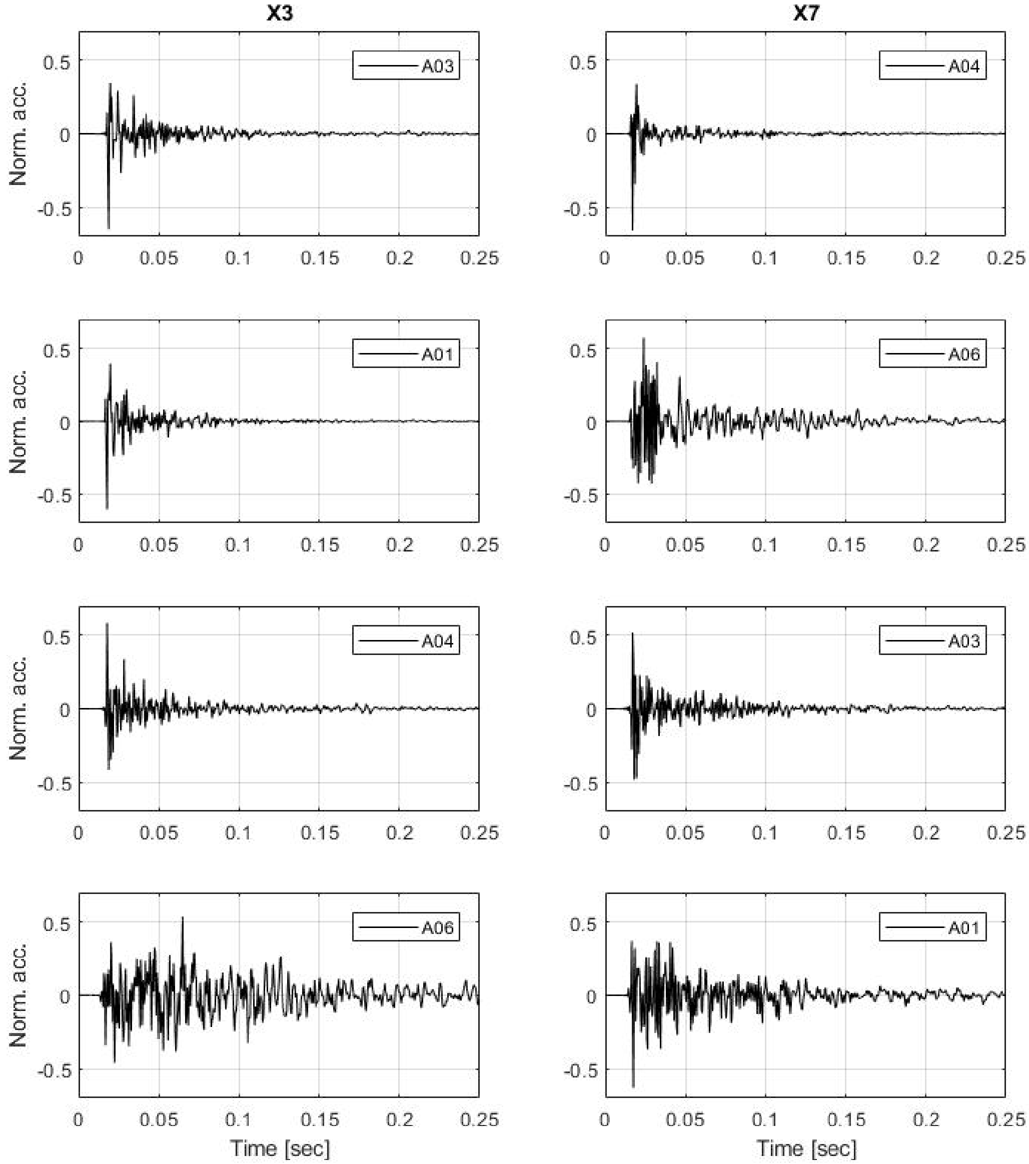
Figure 7.8: Time lag; A01 and A03 to A02 for impact locations X3 and A04 and A06 to A05 for impact location X7.

7.5.1.1.3 Normalization

Figure 7.9 shows the normalized acceleration response of sensors A01, A03, A04 and A06 for impact locations X3 and X7. Once again the plots in (a) is the acceleration response from the numerical simulations and (b) the acceleration response from full-scale measurements.



(a) Numerical simulations



(b) Full-scale measurements

Figure 7.9: Shows the normalized acceleration response of relevant sensors with impact locations X3 and X7. (a) is the acceleration response from numerical simulations and (b) is the acceleration response from full-scale measurements.

Although comparison of matching impact locations and accelerometers for the figures don't look identical (partially because the acceleration response of the full-scale measurements generally are more noisy), comparison of the four top plots in 7.9 (a) show a similar trend to the four top plots in 7.9 (b) and clearly indicate the presence of damage for accelerometer A06. Note that for both numerical simulations and full-scale measurements, the peak acceleration comes in the first wave for all undamaged structural components whereas it comes after a few oscillations for the damage structural component.

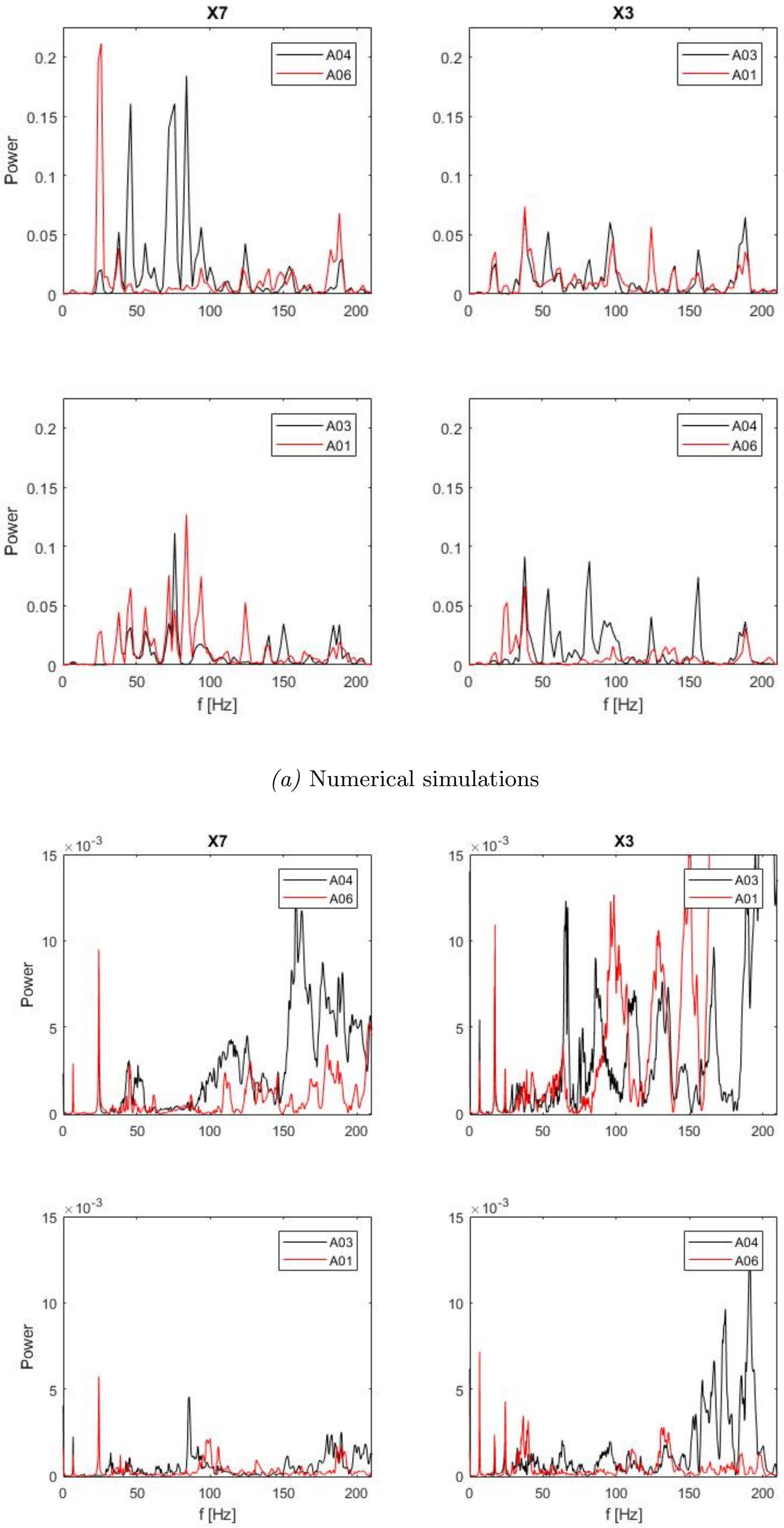
If we further consider the bottom four plots in both figures, the results of the numerical simulations devaite from the ones of the full-scale measurements. Whereas the bottom four plots of figure 7.9 (b) indicate what one might interpret as the presence of damage in both A06 and A01, the bottom four plots of figure 7.9 (a) only does so for accelerometer A06. The same observations as for moving standard deviation are therefore made; the damage identification techniques provide the same information when performed on numerical simulations and full-scale measurements only when distance between impact location and the accelerometers considered are short. When the distance increases, the results from the numerical simulations provide the information we originally would expect from the full-scale measurements, i.e. only giving indication of damage for the component A06 is attached to.

Also note that although comparison of A04 and A06 for impact location X7 for the simulated data provides the indication of damage, comparison of the numerical simulations and the full-scale measurements of A04 for impact location X7 show a significant difference. This might be an effect of the simplifications in the numerical model resulting in the structural components close to the damaged structural component being overly affected by the change.

7.5.1.2 Frequency domain

7.5.1.2.1 Frequency response function

The FRFs established from the numerical simulations for sensors A01, A03, A04 and A06 for impact locations X3 and X7 are shown in figure 7.10 (a), and similarly from the full-scale measurements in 7.10 (b).



(a) Numerical simulations

(b) Full-scale measurements

Figure 7.10: Shows FRFs for relevant sensors with impact locations X3 and X7, (a) based of numerical simulations and (b) of full-scale measurements.

The first thing noticed is that the global modes in the range 0-25 Hz are far from as dominant in figure 7.10 (a) as for 7.10 (b), some of them not even found in 7.10 (a). This is believed to be a result of the fact that the FRFs of the simulated data are only based of 2 second long records vs 30 seconds for the full-scale measured acceleration records. The result of this seems to be that the response is dominated by the local response for the shorter record.

With the fact that the use of FRFs proved to not work as a technique to provide indication of damage for the full-scale measurements, further comparison of figure 7.10 (a) and (b) is not of interest. What we can take from figure 7.10 (a) is that for impact locations in the vicinity of the accelerometers being compared, establishment of the FRFs does work as technique for damage identification for an idealized structure (numerical model). This is seen from comparison between the two top plots in figure 7.10 (a), where in the right plot, the two curves follow each other fairly close and in the left plot the curves are very different. Although the comparison of the two top plots in figure 7.10 (a) indicate damage, it should be noted that if we study the top right plot closely we see that there is one peak at around 50 Hz that is only represented by the curve of sensor A03, and similarly at about 125 Hz only for the curve of sensor A01. For a case where the difference is not as clear as in the two top plots, this would therefore possibly provide identification of damage in the top right plot even if not present, making the reliability of this technique questionable.

Other information taken away from this is indication of where to find the local modes of the damaged and undamaged structural component, which was used when applying damping in the numerical models.

7.5.2 Linear spring

The results for the different damage identification methods when modelling the contact between stringer and cross girder of the damaged structural component with a linear spring of stiffness k=10 MN/m looks very similar to the results obtained with no spring, thus provides the same information. The only plot where a difference between the two is clearly seen is for the normalized acceleration response for accelerometer A06. In this subsection, only a comparison of the normalized acceleration response of accelerometer A06 for different spring stiffness's is therefore shown, whereas the remaining results can be found in appendix D.

The results are displayed in figure 7.11, where the first row presents the normalized acceleration response of the full-scale measurements, and second and third row presents the normalized acceleration response for the numerical simulations with spring stiffness k=0 and k = 10 MN/m, respectively.

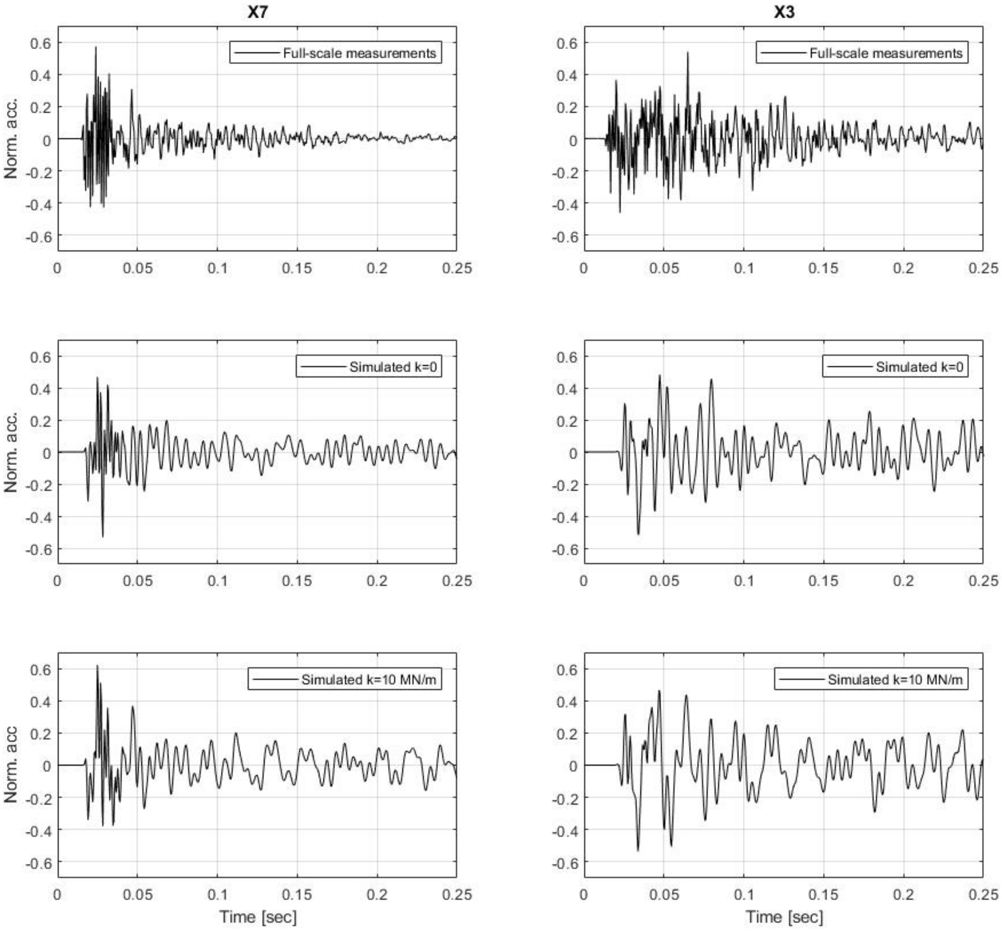


Figure 7.11: Normalized acceleration response of accelerometer A06 for impact location X3 and X7.

If we once again disregard the fact that the full-scale measured acceleration response is somewhat noisier, it's clear that from studying the acceleration response alone, that between the two simulated responses, the one with zero spring stiffness is the one looking most similar to the acceleration response of the full-scale measurements. As all remaining results for a spring stiffness of 10 MN/m look close to identical to the ones displayed in section 7.5.1.1, this becomes the only way to determine which way of implementing the damage is best. It is therefore concluded that zero spring stiffness gives a better representation of the damage structural component, which if there is no contact between stringer and cross girder is obvious.

7.6 Discussion of results

The results presented in section 7.5 show that implementing the damage in the beam model by allowing the stringer to move free vertically and rotate free about the local transversal beam axis, seems to be a good representation. Although this approach provides fairly good results, this does not mean that there is no contact between stringer and cross girder, only that the damage is better represented numerically with no contact than with different linear and non-linear springs. Different contact formulations should still be subjected to further investigation.

Further the results show that the moving standard deviation and normalization techniques provide the same information when applied on numerically simulated acceleration response and acceleration response from full-scale measurements only for a short distance between accelerometers and impact locations. Since the moving standard deviation technique indicates that the overall damping in the model is reasonable, it is further believed that the reason for the above is an effect of local damping rather than an effect of the damage being modelled incorrectly. For the set up used for the damage identification techniques the accelerometers compared are only equally far away from impact location when the impact location is close. For example, accelerometer A01 and A03 are equally far away from impact location X3, but A01 is somewhat further away than A03 for impact location X7. For the full scale measurements, this distance that the signal has to travel further damps the signal to an extent that the moving standard deviation indicates damage, although not present. For the simulated data on the other hand, the small distance further traveled does not affect the signal. This is an interesting observation indicating that the damping, or more precisely the local damping, applied in the numerical model does not represent the damping in the real bridge sufficiently accurate. From the discussion in section 3.1.6 this can be explained by the choice of damping model, Rayleigh damping. As presented in section 2.2, damping is a very complex problem, only partially covered by Rayleigh damping. Rayleigh damping will only be able to catch some of many damping effects of the real structure,and expecting the study of local vibrations to be identical to full-scale measurements is probably unreasonable.

Recall from section 5.3 that the Rayleigh coefficients, a_0 and a_1, were calculated based on a damping of 0.5% in $\omega_1 = 19.67\frac{rad}{s}$ and $\omega_2 = 250\frac{rad}{s}$. Attempts were made to run the analysis with both higher and lower damping, applied to both higher and lower ω_2, to see if we were able to find a better representation of the damping, but without any luck. The only valuable information taken from this was that the acceleration response obtained from the numerical simulations is sensitive to the choice of damping. The conclusion is made that damping should be subjected to further investigation and that more advanced methods of numerical damping should be introduced before spending a considerate amount of time on representing the damage in other ways.

The results of the FRFs established from the simulated data indicates that this technique does work for an idealized structure, but it should be noted that the results are obtained from a short acceleration record and therefore may not be reliable. This means that in theory, establishment of the FRFs is a possible valid technique for damage identification, but this does not mean that it will work in practice. It's still believed that noise and local damping effects from both structural and non-structural components, such as wooden sleepers, will seriously affect the FRFs, especially when looking at local vibrations. It would therefore be advised to implement more advanced frequency domain techniques to be able to find indication of damage in the frequency domain.

When establishing whether or not the different ways of implementing the damage numerically are good by comparing the results obtained from analysis of full-scale measurements and numerical simulations, the moving standard deviation and cross correlation techniques indicate that the damage can be modelled in many different ways. To establish which way of implementing the damage is closest to reality, other techniques should therefore be used. Looking at the normalized acceleration response and making adjustments until the simulated acceleration of the damaged structural component is perfectly similar to the full-scale measured acceleration response would of course ideally be the best way. The problem with this is that the response is dependent on so many other things than the connection of the stringer and cross girder at the damage location, making it practically impossible to make them look perfectly similar. To establish which approach of implementing the damage is best, apart from the obvious, other techniques should be implemented.

From the results in section 5 it's concluded that the simplifications made in the numerical model are fair simplifications for the global dynamic response. Regardless of this, when looking a local response, it is possible that the simplifications made have a serious effect on the results. For instance, all undamaged stringers are modelled with all DOFs constrained to follow the ones of the cross girder where they coincide. As described in section 7, these connections are made up by L-profiles connected to the web of stringer and girder only, meaning that only constraining the translational DOFS of the stringer would possibly be a better representation of the joint. Knowing that a beam clamped at both ends is considerably stiffer than a simply supported beam, this will possibly lead to significantly different results when looking at local acceleration response. If the goal is to get a best possible representation of the local response of the stringers, a possible approach would be to establish a shell model with the same simplifications as the shell model presented in section 5.2, only modelling the connection between stringer and girder at a much higher detail level, i.e. including all the components of the joint connection in the model (plates, L-profiles, etc).

When studying the problem of the damaged structural component both for small and big relative displacement between stringer and cross girder, i.e. displacements smaller than the glitch and displacements closing the glitch, it's without doubt a highly nonlinear problem. In this section only small relative displacements are thoroughly discussed as the data available is believed to only represents this part of the problem. The problem with forces big enough to close the glitch between stringer and cross girder should be subjected to further investigation as this is the problem faced with when the bridge is subjected to train loads.

8 Concluding remarks

The main goal of this thesis has been to assess the global dynamic behavior of the Hell Bridge Test Arena and further investigation of a known damage to find new, simple damage identification techniques. The dynamic behavior has been assessed by two numerical models, one beam and one shell model, and from modal analysis of full-scale measurements. The known damage in the connection between stringer and cross girder has been investigated by local measurements of imposed vibrations which further has been used as a foundation to implement the damage in the numerical models.

The results show that for the purpose of extracting the natural frequencies and mode shape estimates of the Hell Bridge Test Arena, the use of the simple frequency domain techniques Peak Picking and Frequency Domain Decomposition is sufficient. Further the results show that mode shapes and natural frequencies found from eigenvalue analysis of both numerical models fit the ones obtained from modal analysis good. As neither of the numerical models stick out as more accurate than the other and establishing the beam model was found much less troublesome than the shell model, it is concluded that for the purpose of this paper, a simple beam model is the better choice of the two.

In the study of damage identification techniques from the bridge deck it was found that all three time domain techniques introduced, moving standard deviation, cross correlation and normalization are feasible for indicating damage. Application of them can be used to confirm the presence of damage and strengthen the decision for selecting areas that should be subjected to further investigation. The frequency domain technique, establishing the FRF, on the other hand does not work even with a good understanding of both global and local dynamic behavior of the bridge. Its concluded that to be able to find indication of damage in the frequency domain, more advanced techniques should be applied.

Numerical tests to evaluate the effect of the known damage on the global dynamic response showed that the observed damage is too small to have any effect on the 14 first natural frequencies presented in this paper. As the stringers are a vital structural part in the load distribution process, this does not mean that the observed damage is not serious, it only implies that from studying global natural frequencies alone, indication of this type of damage will not be found.

Attempts were made to implement the know damage in both numerical models, but was only successfully done so in the beam model. The damaged structural component was represented by removing constraints, allowing the stringer to act like a cantilever, and further three different connections between stringer and cross girder were modelled; no spring, linear spring and non-linear spring. The results obtained proved to be highly dependent on the choice of numerical damping, providing a source of error that should be subjected to further investigation. By comparing results obtained from the different damage identification techniques of the numerical simulations and the full-scale measurements, the method with no spring, i.e. no contact between stringer and cross girder, provided the best results, giving a good representation.

Finally, the experimental data induced by a modal hammer introduces much lower forces than the ones from a train passage. The result of this is believed to introduce a completely different problem when it comes to representing the damage numerically. For smaller forces it is shown that the stringer of the damaged structural component behaves similar to a cantilever, but for bigger forces, the glitch between stringer and L-profile will close, giving the stringer vertical support at both ends. These two problems are both relevant and very different, so attempts to replicate the latter should be subjected to further work.

9 Further work

This thesis has been a challenging and very interesting study. Through the process, many obstacles requiring serious brainstorming have occurred. Some of the ideas have been implemented, but many of them had to be laid aside because of the need to limit the extent and scope of this thesis, and also by the lack of time. This has led to a number of suggestions for possible further work:

- In this thesis, we have only looked at vertical response for the damaged structural component. Expand this to look at the response in all directions to see if it's possible to see the damage in the other directions also.
- Investigate the known damage for loads big enough to close the glitch between stringer an L-profile connected to the cross girder. This could be possible with a shaker, which is ordered and should arrive at the Hell Bridge Test Arena before 2019..
- Further investigate of the techniques for damage identification presented in this paper to further established them as reliable techniques. This can be done by introducing a different set up for accelerometers and impact locations.
- Investigate other methods for performing bridge inspection. This can be done by introducing other excitation sources such as the shaker.
- Implement more advanced techniques for modal analysis to establish good damping estimates for the structure. Use these results in combination with more advanced damping models in the numerical models too assign more appropriate damping ratios.
- Implement more advanced frequency domain techniques for damage identification.
- Investigate the effects of the simplifications made in the numerical models on local vibrations by constructing a shell model with high detail level of connection joints between stringer and cross girder.
- Further investigate the friction contact pair formulation and other methods to represent the damage numerically. A possible approach would be to model the cross girder where the damaged structural component attaches/ should attach with 3D solid elements.

References

[1] NTNU, Structural dynamics group. Hell Bridge test arena [internet]. Trondheim: NTNU; 2017 [cited 2018 Jun 07]. Available from: https://www.ntnu.edu/kt/research/dynamics/research/hell-arena

[2] Srinivasan Gopalakrishnan, Massimo Ruzzene and Sathyanaraya Hanagud. Computational techniques for structural health monitoring. London: Springer-Verlag London; 2014.

[3] K. M. Mathisen. TKT4197 Nonlinear Finite Element Analysis, Lecture 6 - The Finite Element Method applied to Structural Dynamics and Vibrations. Trondheim: Department of Structural Engineering, Norwegian University of Science and Technology; 2015.

[4] Anil K. Chopra. Dynamics of Structures: Theory and Applications to Earthquake Engineering. Upper Saddle River, New Jersey: Prentice Hall; 2012.

[5] Roy R. Craig and Andrew J. Kurdila. Fundamentals of structural dynamics. New Jersey: John Wiley & Sons; 2006.

[6] O. A. Øiseth. TKT4201 Structural Dynamics, Lecture 14/15 - Free undamped and damped vibration. Trondheim: Department of Structural Engineering, Norwegian University of Science and Technology; 2015.

[7] Ladislav Frýba. DYNAMICS OF RAILWAY BRIDGES. London, England: Thomas Telford Publishing; 1996.

[8] Ray W Clough and Joseph Penzien. Dynamics of structures. McGraw-Hill, New York, 1993.

[9] Carlo Rainieri and Giovanni Fabbrocino. Operational modal analysis of civil engineering structures. New York, USA: Springer; 2014.

[10] Wei-Xin Ren, Zhou-Hong Zong. Output-only modal parameter identification of civil engineering structures. Structural Engineering and Mechanics. 2004;(17).

[11] Filipe Magalhães, Elsa Caetano and Alvaro Cunha. Challenges in the Application of Stochastic Modal Identification Methods to a Cable-Stayed Bridge. Journal of Bridge Engineering - J BRIDGE ENG. 2007;(12).

[12] Svend Gade, Nils B. Møller, Henrik Herlufsen and Hans Konstantin-Hansen. Frequency domain techniques for operational modal analysis. 1st IOMAC Conference. Copenhagen: Brüel Kjær Sound and Vibration Measurements A/S. 2005.

[13] Paul H. Wirsching, Thomas L Paez and Keith Ortiz. Random vibrations: theory and practice. North Chelmsford, Chelmsford, Massachusetts: Courier Corporation; 2006.

[14] Zhi-Fang Fu and Jimin He. Modal analysis. Oxford, England: Butterworth-Heinemann; 2001.

[15] NTNU, Structural dynamics group. Hell Test Arena, instrumentation setup for modal hammer testing - Results. 2017.

[16] MATLAB user's guide (2017a). Natick, Massachusetts: The Mathworks Inc.; 2017.

[17] Mohammad Farshchin. Frequency Domain Decomposition (FDD) [Internet]. Natick, Massachusetts, USA: The Mathworks Inc; 2015. [cited 2018 Jun 07]. Available from:

[18] Martin H. Weik. Nyquist sampling rate. Computer Science and Communications Dictionary. 2004. https://se.mathworks.com/matlabcentral/fileexchange/50988-frequency-domain-decomposition–fdd-.

[19] Abaqus User's manual (6.14). Pawtucket, RI: Dassault Systèmes Simulia Corp.; 2014.

[20] NS-EN 13145:2001 Railway applications - Track - Wood sleepers and bearers, January 2012. 2003.

[21] Kolbein Bell. An engineering approach to finite element analysis of linear structural mechanics problems. Trondheim: Akademika Publ; 2013.

[22] EN 1991-2:2003 Actions on structures, part 2: Traffic loads on bridges, September.

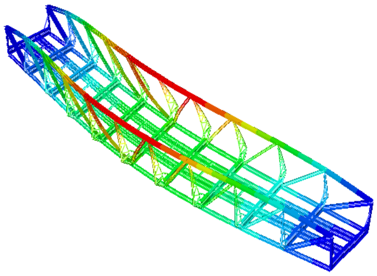
[23] B.T. Svendsen, G.T. Frøseth and A. Rønnquist. Inspection of steel bridges by modal hammer from bridge deck only. 2018.

[24] Florian Pichler, Wolfgang Witteveen and Peter Fischer. A complete strategy for efficient and accurate multibody dynamics of flexible structures with large lap joints considering contact and friction. 2017. Multibody System Dynamics;40(4):407-436.

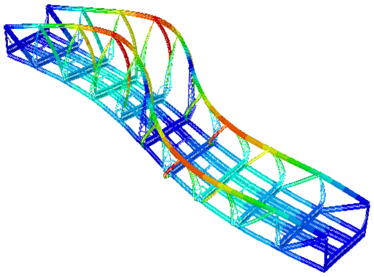
[25] Carlos Garcia Garino and J-P Ponthot. A quasi-coulomb model for frictional contact interfaces: Application to metal forming simulations. Latin American applied research. 2008;38(2);95-104.

A Appendix A - Modal analysis

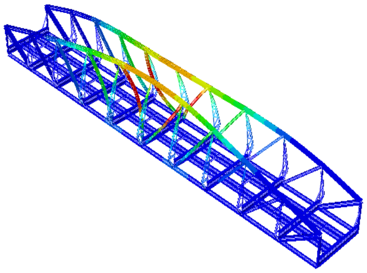
A.1 Modes beam model



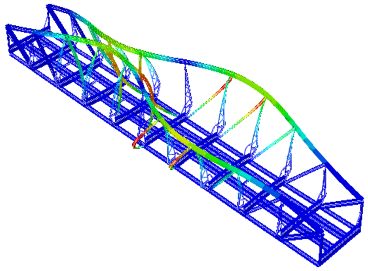
(1) Mode 1, 3.14 Hz



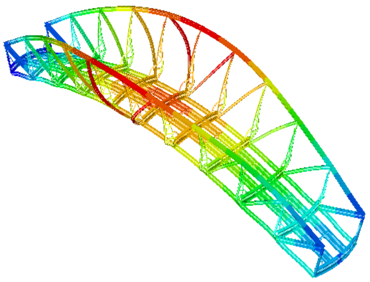
(2) Mode 2, 6.11 Hz



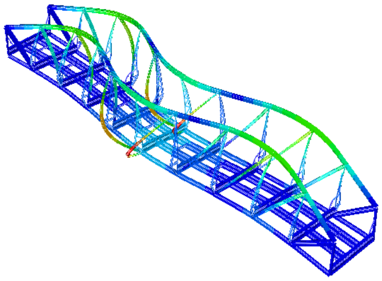
(3) Mode 3, 7.19 Hz



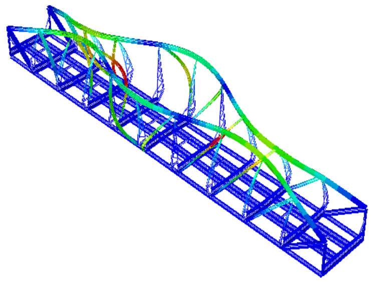
(4) Mode 4, 7.66 Hz



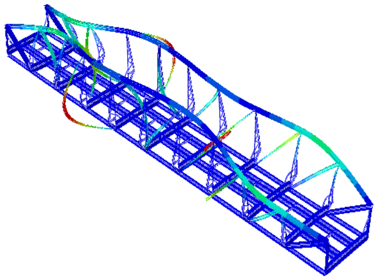
(5) Mode 5, 6.83 Hz



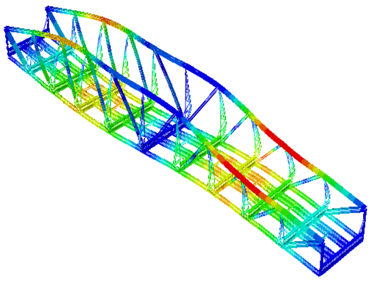
(6) Mode 6, 8.27 Hz



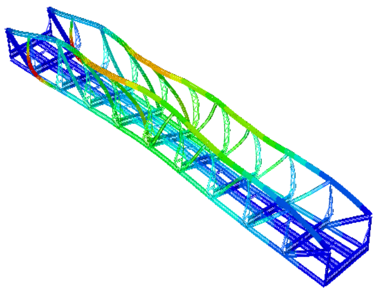
(7) Mode 7, 8.84 Hz



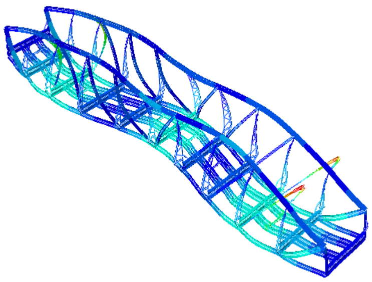
(8) Mode 8, 10.28 Hz



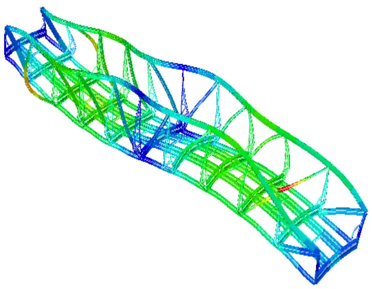
(9) Mode 9, 13.50 Hz



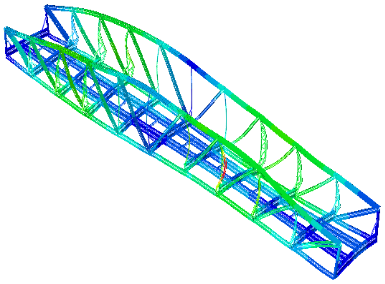
(10) Mode 10, 15.26 Hz



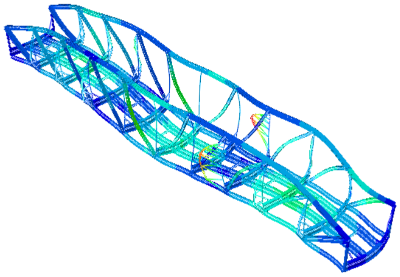
(11) Mode 11, 17.45 Hz



(12) Mode 12, 17.22 Hz

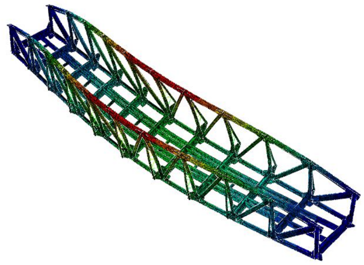


(13) Mode 13, 23.85 Hz

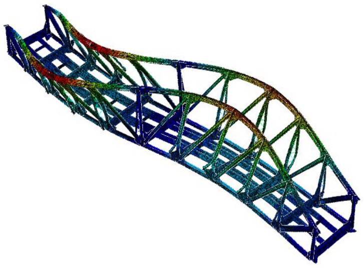


(14) Mode 14, 25.34 Hz

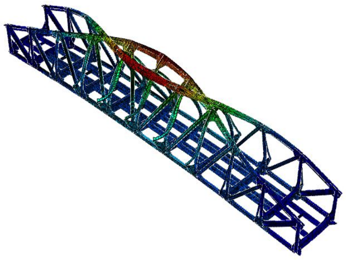
A.2 Modes shell model



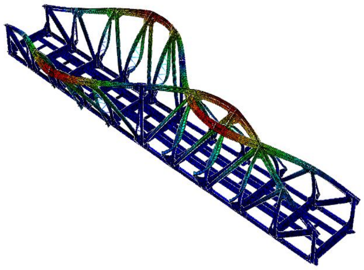
(1) Mode 1, 3.20 Hz



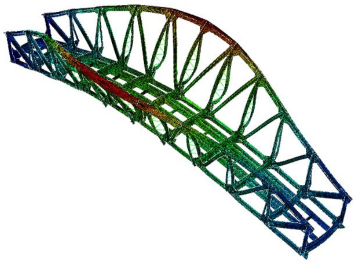
(2) Mode 2, 6.35 Hz



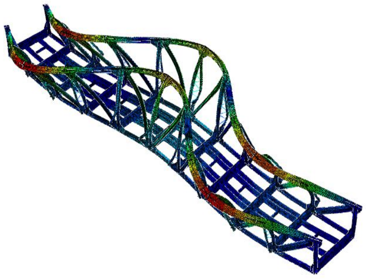
(3) Mode 3, 7.07 Hz



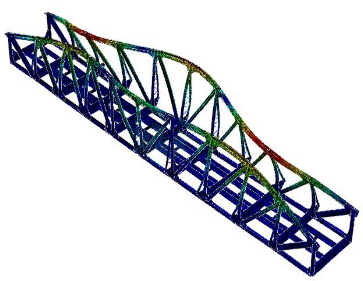
(4) Mode 4, 7.69 Hz



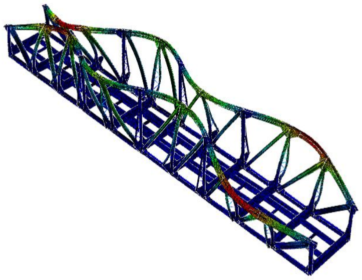
(5) Mode 5, 6.75 Hz



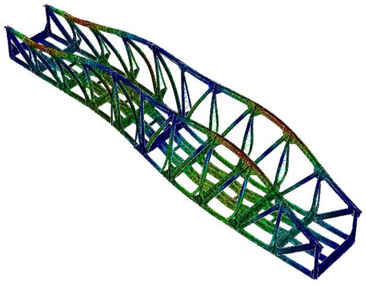
(6) Mode 6, 8.50 Hz



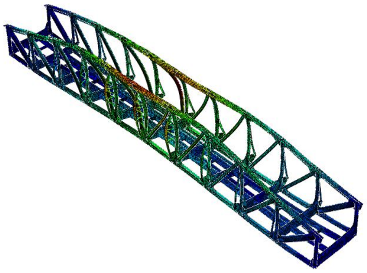
(7) Mode 7, 9.15 Hz



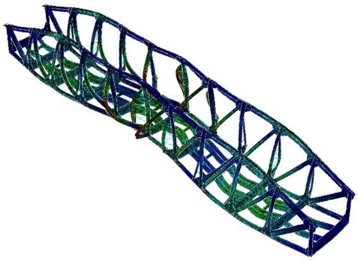
(8) Mode 8, 10.66 Hz



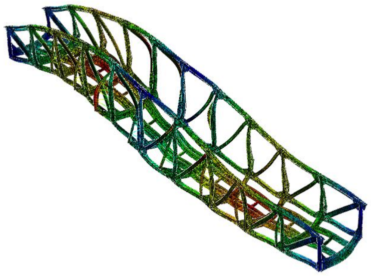
(9) Mode 9, 12.45 Hz



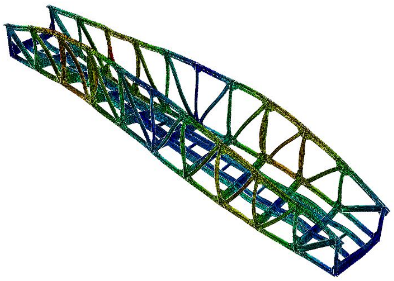
(10) Mode 10, 14.04 Hz



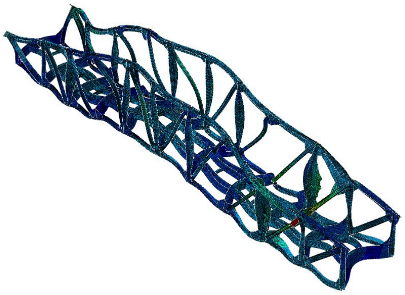
(11) Mode 11, 17.76 Hz



(12) Mode 12, 17.71 Hz



(13) Mode 13, 24.25 Hz



(14) Mode 14, 24.53 Hz

B Appendix B - Matlab scripts

B.1 Peak picking

```
clc
clear all
close all
% 1. Bruke hele tidsserien
% 2. Starter tidsserien noen sekunder for toget kommer til broa og lar den
% gaa en del sekunder etter toget har passert.
% 3. Starter tidsserien i det toget kommer til broa og avslutter saa fort
% akselerometeret er tilbake til normale verdier.
% 4. Starter tidsserien rett etter toget har passert broa og avslutter en
% liten stund etter aksellerometerene viser nromale verdier igjen.
% 5. Tidsserie fra mellom togpasseringer, dvs naar akselerometerene viser
% "normale verider.
% 6. Tar ut alle i alternativ 3 og legger de sammen til en ny tidsserie


%---------------------------------------------------------------%
%%%%% Change these paramaters %%%%%%

Record     = 1;          %Record, see below.
PassingNr  = 1;          %Train passage nr. wanted
Nwelch     = 0;          %number of divisions, set to zero to use autovalues
alt        = 5;          %Alternative 1 if whole acceleration file should be
                         %used, se top of script for more info
AccStart   = 5;          %Start accelerometer to look at
AccStop    = 5;          %End accelerometer
AccStep    = 1;          %Step accelerometer

FreqPlotStart=0.1;         %Defines start of x axis in plot
FreqPlotEnd=30;          %Defines end of x axis in plot
PlotFreq=0;             %End of frequency value to mark peaks
MinPeakDist = 3;      %Minimum distance between peaks
%---------------------------------------------------------------%

if Record == 1
    recname='2016-09-07-09-15-54.mat';
elseif Record == 2
    recname='2016-09-07-11-04-18.mat';
elseif Record == 3
    recname='2016-09-08-10-05-42.mat';
elseif Record == 4
    recname='2016-09-08-12-13-06.mat';
end
```

```
load(recname);
nt=length(ConvertedData.Data.MeasuredData(2).Data); %number of samples in
%the measurement series
dt=1/400; %sample rate is 400 Hz
t=0:dt:dt*(nt-1); %create the corresponding time vector
T=t(end);
%Directions as in Abaqus model!
accelerometers =[33 32 34; 36 35 37; 39 38 40; 42 41 43; 45 44 46;...
    48 47 49; 1 0 2;
    4 3 5; 7 6 8; 10 9 11; 13 12 14; 51 50 52; 54 53 55; 57 56 58;...
    60 59 61;
    16 15 17; 19 18 20; 22 21 23; 25 24 26; 28 27 29];
accelerometers = accelerometers + 2;
A = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19];


AccStart = AccStart + 1;
AccStop = AccStop + 1;
if alt == 1
else
    if recname == '2016-09-07-09-15-54.mat'
        if alt == 2
            plotTime =[1540 1700;2450 2650; 4100 4400; 5275 5400];    %2.
        elseif alt == 3
            plotTime = [1576 1592; 2527 2543; 4210 4270; 5316 5332]; %3.
        elseif alt == 4
            plotTime =[1582 1700;2532 2650; 4250 4400; 5321 5420];    %4.
        elseif alt == 5
            plotTime =[1 100;1900 2200; 2800 4000; 4600 5000];        %5.
        elseif alt == 6
            plotTime = [1576 1592; 2527 2543; 4210 4270; 5316 5332]; %6.
        end
    elseif recname == '2016-09-07-11-04-18.mat'
        if alt == 2
            plotTime =[460 660; 2420 2620; 3120 3320; 5320 5520;...
                5900 6100; 6650 6850];  %2.
        elseif alt == 3
            plotTime = [530 550; 2476 2492; 3178 3192; 5391 5410;...
                5955 5972; 6694 6710]; %3.
        elseif alt == 4

        elseif alt == 5
            plotTime =[1 350; 800 2200; 2800 3000; 3500 5200;...
                5650 5800; 6200 6500];      %5.
        elseif alt == 6
            plotTime = [530 550; 2476 2492; 3178 3192; 5391 5410;...
```

```
                5955 5972; 6694 6710]; %6.
        end
    elseif recname == '2016-09-08-10-05-42.mat'

    elseif recname == '2016-09-08-12-13-06.mat'
        if alt == 2
            plotTime =[1750 1950; 2480 2680; 5200 5400; 6020 6220;...
                6750 6950]; %2.
        elseif alt == 3
            plotTime =[1802 1818; 2536 2552; 5272 5290; 6075 6095;...
                6815 6835]; %3.
        elseif alt == 4

        elseif alt == 5
            plotTime =[1 1600; 2100 2300; 2800 5000; 5550 5900;...
                6400 6600];     %5.
        elseif alt == 6
            plotTime =[1802 1818; 2536 2552; 5272 5290; 6075 6095;...
                6815 6835]; %6.
        end
    end
end

if alt == 1
    %nothing
else
    tTemp = plotTime(PassingNr,1):dt:plotTime(PassingNr,2);
end

if Nwelch == 0
    %nothing
elseif alt == 6
    %nothing
else
    if alt == 1
        Nwindow=round(length(t)/Nwelch); %length of window
        nfft=2^nextpow2(Nwindow); %number of FFT points
    else
        Nwindow=round(length(tTemp)/Nwelch); %length of window
        nfft=2^nextpow2(Nwindow); %number of FFT points
    end
end

for i = AccStart:AccStep:AccStop

    x=ConvertedData.Data.MeasuredData(accelerometers(i,1)).Data;
    y=ConvertedData.Data.MeasuredData(accelerometers(i,2)).Data;
    z=ConvertedData.Data.MeasuredData(accelerometers(i,3)).Data;
    x=x-mean(x);
    y=y-mean(y);
    z=z-mean(z);
    if alt == 1
        if Nwelch == 0
            [Pxx,fx] = pwelch(x,[],[],[],400);
            [Pyy,fy] = pwelch(y,[],[],[],400);
            [Pzz,fz] = pwelch(z,[],[],[],400);
%             figure()
%             spectrogram(z,[],[],[],400,'yaxis')
        else
            [Pxx,fx] = pwelch(x,hanning(Nwindow),[],nfft,400);
            [Pyy,fy] = pwelch(y,hanning(Nwindow),[],nfft,400);
            [Pzz,fz] = pwelch(z,hanning(Nwindow),[],nfft,400);
%             figure()
%             spectrogram(z,hanning(Nwindow),[],nfft,400,'yaxis')
        end

        %Find peaks x
        [pk_x,lk_x] = findpeaks(Pxx,fx,'MinPeakDistance',MinPeakDist);
        freq_x=zeros(length(lk_x),1);
        val_x = lk_x(1,1);
        count_x = 0 ;
        while val_x<PlotFreq
            count_x = count_x+1;
            freq_x(count_x,1) = lk_x(count_x,1);
            val_x = lk_x(count_x+1,1);
        end
        %Plot natural frequencies in plot
        freq_x(count_x+1:length(lk_x)) = [];
        pk_x(count_x+1:length(lk_x)) = [];
        lk_x(count_x+1:length(lk_x)) = [];
        freq1_x =freq_x;
        freq_x = num2str(freq_x);

        %Find peaks y
        [pk_y,lk_y] = findpeaks(Pyy,fy,'MinPeakDistance',MinPeakDist);
        freq_y=zeros(length(lk_y),1);
        val_y = lk_y(1,1);
        count_y = 0 ;
        while val_y<PlotFreq
            count_y = count_y+1;
            freq_y(count_y,1) = lk_y(count_y,1);
            val_y = lk_y(count_y+1,1);
        end
        %Plot natural frequencies in plot
```

```
freq_y(count_y+1:length(lk_y)) = [];
pk_y(count_y+1:length(lk_y)) = [];
lk_y(count_y+1:length(lk_y)) = [];
freq1_y =freq_y;
freq_y = num2str(freq_y);

%Find peaks z
[pk_z,lk_z] = findpeaks(Pzz,fz,'MinPeakDistance',MinPeakDist);
freq_z=zeros(length(lk_z),1);
val_z = lk_z(1,1);
count_z = 0 ;
while val_z<PlotFreq
    count_z = count_z+1;
    freq_z(count_z,1) = lk_z(count_z,1);
    val_z = lk_z(count_z+1,1);
end
%Plot natural frequencies in plot
freq_z(count_z+1:length(lk_z)) = [];
pk_z(count_z+1:length(lk_z)) = [];
lk_z(count_z+1:length(lk_z)) = [];
freq1_z =freq_z;
freq_z = num2str(freq_z);

% Plot spectrum
figure(); grid on; hold on;
set(gcf,'color','w');
box on
plot(fx,Pxx);
xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
title([num2str(recname) '   A' num2str(A(1,i)) '-x']);
xlim([FreqPlotStart FreqPlotEnd])
plot(lk_x,pk_x,'o')
text(lk_x,pk_x,freq_x,'HorizontalAlignment','left');

figure(); grid on; hold on;
set(gcf,'color','w');
box on
plot(fy,Pyy);
xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
title([num2str(recname) '   A' num2str(A(1,i)) '-y']);
xlim([FreqPlotStart FreqPlotEnd])
plot(lk_y,pk_y,'o')
text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

figure(); grid on; hold on;
set(gcf,'color','w');
box on
```

```
        plot(fz,Pzz);
        xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
        title([num2str(recname) '  A' num2str(A(1,i)) '-z']);
        xlim([FreqPlotStart FreqPlotEnd])
        plot(lk_z,pk_z,'o')
        text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');


    elseif alt == 6

        xTemp = x(plotTime(1,1)*400:plotTime(1,2)*400);
        yTemp = y(plotTime(1,1)*400:plotTime(1,2)*400);
        zTemp = z(plotTime(1,1)*400:plotTime(1,2)*400);

        for k = 2:length(plotTime)
            xT = x(plotTime(k,1)*400:plotTime(k,2)*400);
            yT = y(plotTime(k,1)*400:plotTime(k,2)*400);
            zT = z(plotTime(k,1)*400:plotTime(k,2)*400);
            xTemp = [xTemp; xT];
            yTemp = [yTemp; yT];
            zTemp = [zTemp; zT];
        end

        if Nwelch == 0
            %nothing
        else
            Nwindow=round(length(xTemp)/Nwelch); %length of window
            nfft=2^nextpow2(Nwindow); %number of FFT points
        end

        if Nwelch == 0
            [Pxxtemp,fxtemp] = pwelch(xTemp,[],[],[],400);
            [Pyytemp,fytemp] = pwelch(yTemp,[],[],[],400);
            [Pzztemp,fztemp] = pwelch(zTemp,[],[],[],400);
        else
            [Pxxtemp,fxtemp] = pwelch(xTemp,hanning(Nwindow),[],nfft,400);
            [Pyytemp,fytemp] = pwelch(yTemp,hanning(Nwindow),[],nfft,400);
            [Pzztemp,fztemp] = pwelch(zTemp,hanning(Nwindow),[],nfft,400);
        end
        %Find peaks xTemp
        [pk_x,lk_x] = findpeaks(Pxxtemp,fxtemp,'MinPeakDistance',...
            MinPeakDist);
        freq_x=zeros(length(lk_x),1);
        val_x = lk_x(1,1);
        count_x = 0 ;
        while val_x<PlotFreq
            count_x = count_x+1;
    freq_x(count_x,1) = lk_x(count_x,1);
    val_x = lk_x(count_x+1,1);
end
%Plot natural frequencies in plot
freq_x(count_x+1:length(lk_x)) = [];
pk_x(count_x+1:length(lk_x)) = [];
lk_x(count_x+1:length(lk_x)) = [];
freq1_x =freq_x;
freq_x = num2str(freq_x);

%Find peaks yTemp
[pk_y,lk_y] = findpeaks(Pyytemp,fytemp,'MinPeakDistance',...
    MinPeakDist);
freq_y=zeros(length(lk_y),1);
val_y = lk_y(1,1);
count_y = 0 ;
while val_y<PlotFreq
    count_y = count_y+1;
    freq_y(count_y,1) = lk_y(count_y,1);
    val_y = lk_y(count_y+1,1);
end
%Plot natural frequencies in plot
freq_y(count_y+1:length(lk_y)) = [];
pk_y(count_y+1:length(lk_y)) = [];
lk_y(count_y+1:length(lk_y)) = [];
freq1_y =freq_y;
freq_y = num2str(freq_y);

%Find peaks zTemp
[pk_z,lk_z] = findpeaks(Pzztemp,fztemp,'MinPeakDistance',...
    MinPeakDist);
freq_z=zeros(length(lk_z),1);
val_z = lk_z(1,1);
count_z = 0 ;
while val_z<PlotFreq
    count_z = count_z+1;
    freq_z(count_z,1) = lk_z(count_z,1);
    val_z = lk_z(count_z+1,1);
end
%Plot natural frequencies in plot
freq_z(count_z+1:length(lk_z)) = [];
pk_z(count_z+1:length(lk_z)) = [];
lk_z(count_z+1:length(lk_z)) = [];
freq1_z =freq_z;
freq_z = num2str(freq_z);

if Nwelch == 0
```

```
    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fxtemp,Pxxtemp);
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-x.']);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_x,pk_x,'o')
    text(lk_x,pk_x,freq_x,'HorizontalAlignment','left');

    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fytemp,Pyytemp);
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-y.']);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_y,pk_y,'o')
    text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fztemp,Pzztemp);
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-z.']);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_z,pk_z,'o')
    text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');
else
    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fxtemp,Pxxtemp);
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-x.'...
        ' Nwelch=' num2str(Nwelch)]);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_x,pk_x,'o')
    text(lk_x,pk_x,freq_x,'HorizontalAlignment','left');

    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fytemp,Pyytemp);
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-y.' ...
                '  Nwelch=' num2str(Nwelch)]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_y,pk_y,'o')
            text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

            figure(); grid on; hold on;
            set(gcf,'color','w');
            box on
            plot(fztemp,Pzztemp);
            xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
            title([num2str(recname) '.  A' num2str(A(1,i)) '-z.'...
                '  Nwelch=' num2str(Nwelch)]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_z,pk_z,'o')
            text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');
        end
%         tTemp = 0:dt:dt*(length(xTemp)-1);
%         figure()
%         plot(tTemp,zTemp,'b');
%         xlabel('Time [s]'); ylabel('Acc. [g]');
%         title([num2str(recname) '.  A' num2str(A(1,i)) '-z.   ']);

    else
        xTemp = x(plotTime(PassingNr,1)*400:plotTime(PassingNr,2)*400);
        yTemp = y(plotTime(PassingNr,1)*400:plotTime(PassingNr,2)*400);
        zTemp = z(plotTime(PassingNr,1)*400:plotTime(PassingNr,2)*400);

        if Nwelch == 0
            [Pxxtemp,fxtemp] = pwelch(xTemp,[],[],[],400);
            [Pyytemp,fytemp] = pwelch(yTemp,[],[],[],400);
            [Pzztemp,fztemp] = pwelch(zTemp,[],[],[],400);
%             figure()
%             spectrogram(zTemp,[],[],[],400,'yaxis')
        else
            [Pxxtemp,fxtemp] = pwelch(xTemp,hanning(Nwindow),[],nfft,400);
            [Pyytemp,fytemp] = pwelch(yTemp,hanning(Nwindow),[],nfft,400);
            [Pzztemp,fztemp] = pwelch(zTemp,hanning(Nwindow),[],nfft,400);
%             figure()
%             spectrogram(zTemp,hanning(Nwindow),[],nfft,400,'yaxis')
        end
        %Find peaks xTemp
        [pk_x,lk_x] = findpeaks(Pxxtemp,fxtemp,'MinPeakDistance',...
            MinPeakDist);
        freq_x=zeros(length(lk_x),1);
        val_x = lk_x(1,1);
        count_x = 0 ;
        while val_x<PlotFreq
    count_x = count_x+1;
    freq_x(count_x,1) = lk_x(count_x,1);
    val_x = lk_x(count_x+1,1);
end
%Plot natural frequencies in plot
freq_x(count_x+1:length(lk_x)) = [];
pk_x(count_x+1:length(lk_x)) = [];
lk_x(count_x+1:length(lk_x)) = [];
freq1_x =freq_x;
freq_x = num2str(freq_x);

%Find peaks yTemp
[pk_y,lk_y] = findpeaks((Pyytemp),fytemp,'MinPeakDistance',...
    MinPeakDist);
freq_y=zeros(length(lk_y),1);
val_y = lk_y(1,1);
count_y = 0 ;
while val_y<PlotFreq
    count_y = count_y+1;
    freq_y(count_y,1) = lk_y(count_y,1);
    val_y = lk_y(count_y+1,1);
end
%Plot natural frequencies in plot
freq_y(count_y+1:length(lk_y)) = [];
pk_y(count_y+1:length(lk_y)) = [];
lk_y(count_y+1:length(lk_y)) = [];
freq1_y =freq_y;
freq_y = num2str(freq_y);

%Find peaks zTemp
[pk_z,lk_z] = findpeaks((Pzztemp),fztemp,'MinPeakDistance',...
    MinPeakDist);
freq_z=zeros(length(lk_z),1);
val_z = lk_z(1,1);
count_z = 0 ;
while val_z<PlotFreq
    count_z = count_z+1;
    freq_z(count_z,1) = lk_z(count_z,1);
    val_z = lk_z(count_z+1,1);
end
%Plot natural frequencies in plot
freq_z(count_z+1:length(lk_z)) = [];
pk_z(count_z+1:length(lk_z)) = [];
lk_z(count_z+1:length(lk_z)) = [];
freq1_z =freq_z;
freq_z = num2str(freq_z);
```

```
if Nwelch == 0


    figure(); grid on; hold on;
    plot(fxtemp,Pxxtemp);
    set(gcf,'color','w');
    box on
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-x.'...
        ' Passing nr:' num2str(PassingNr)]);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_x,pk_x,'o')
    text(lk_x,pk_x,freq_x,'HorizontalAlignment','left');

    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fytemp,(Pyytemp));
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-y.'...
        ' Passing nr:' num2str(PassingNr)]);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_y,pk_y,'o')
    text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

    figure(); grid on; hold on;
    set(gcf,'color','w');
    box on
    plot(fztemp,(Pzztemp));
    xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
    title([num2str(recname) '. A' num2str(A(1,i)) '-z.' ...
        ' Passing nr:' num2str(PassingNr)]);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_z,pk_z,'o')
    text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');

    figure(); hold on;
    set(gcf,'color','w');
    box on
    subplot(1,2,1)
    plot(fztemp,(Pzztemp),'k');
    xlabel('f [Hz]'); ylabel('Spectral density');
    title(['Transversal accelerometer A0' num2str(A(1,i))]);
    xlim([FreqPlotStart FreqPlotEnd])
    plot(lk_z,pk_z,'o')
    text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');
```

```
            subplot(1,2,2)
            plot(fytemp,(Pyytemp),'k');
            xlabel('f [Hz]'); ylabel('Spectral density');
            title(['Vertical accelerometer A0' num2str(A(1,i))]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_y,pk_y,'o')
            text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

        else
            figure(); grid on; hold on;
            set(gcf,'color','w');
            box on
            plot(fxtemp,Pxxtemp);
            xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
            title([num2str(recname) '. A' num2str(A(1,i)) '-x.'...
                ' Passing nr:' num2str(PassingNr)...
                '. Nwelch=' num2str(Nwelch)]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_x,pk_x,'o')
            text(lk_x,pk_x,freq_x,'HorizontalAlignment','left');

            figure(); grid on; hold on;
            set(gcf,'color','w');
            box on
            plot(fytemp,Pyytemp);
            xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
            title([num2str(recname) '. A' num2str(A(1,i)) '-y.'...
                ' Passing nr:' num2str(PassingNr)...
                '. Nwelch=' num2str(Nwelch)]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_y,pk_y,'o')
            text(lk_y,pk_y,freq_y,'HorizontalAlignment','left');

            figure(); grid on; hold on;
            set(gcf,'color','w');
            box on
            plot(fztemp,Pzztemp);
            xlabel('f [Hz]'); ylabel('Spectral density [(m/s^2)^2/(1/s)]');
            title([num2str(recname) '. A' num2str(A(1,i)) '-z.'...
                ' Passing nr:' num2str(PassingNr)...
                '. Nwelch=' num2str(Nwelch)]);
            xlim([FreqPlotStart FreqPlotEnd])
            plot(lk_z,pk_z,'o')
            text(lk_z,pk_z,freq_z,'HorizontalAlignment','left');
        end
    end
end
```

B.2 Frequency domain decomposition

```
function [Frq,phi]=FDD(Input,Fs)

% Frequency Domain Decomposition (FDD) algorithm for modal analysis
% This code allows you to manually select the peaks by simply drawing a
% rectangle around the peaks.
% Programmer: Mohammad Farshchin, Ph.D candidate at The UofM
% Email: Mohammad.Farshchin@gmail.com
% Last modified: 4/6/2015

% Input: the name of input file that contains time history data
% Fs: sampling frequency
% Frq: identified frequencies
% phi: identified mode shapes
% Example:

% For detailed information about this method see: Brincker R, Zhang LM,
%Andersen P. Modal identification from ambient responses using Frequency
%Domain Decomposition. In: Proceedings of the 18th International Modal
%Analysis Conf., USA: San Antonio, 2000.

% ----------------------------------------------------------------------
% Initialization
close all
% ----------------------------------------------------------------------
% Import time history data: Processed accelereation data must be
% arranged in a columnwise format (one column for each measurement channel)
% Note that the acceleration data must be preprocessed
%(detrend, filtered etc.).
% Read acceleration data from the excel file
Acc=xlsread(Input);
display('FDD is in progress, please wait ...')
% ----------------------------------------------------------------------
% Compute Power Spectral Density (PSD) matrix.
% CPSD function, with default settings, is used to compute the cross power
% spectral density matrix. More sophisticated methods can also be
% applied for more accuracy.
for I=1:size(Acc,2)
    for J=1:size(Acc,2)
        [PSD(I,J,:),F(I,J,:)]=cpsd(Acc(:,I),Acc(:,J),[],[],[],Fs);
    end
end
Frequencies(:,1)=F(1,1,:);
% ----------------------------------------------------------------------
% Perform Modal Analysis (Use the Identifier function, below)
[Frq,phi,Fp,s1] = Identifier(PSD,Frequencies);
```

```
% Save results
save('IdResults.mat','phi','Fp','s1','Frequencies')
% ——————————————————————————————————————————————
% Print results
display('——————————————————————————————————————————')
display('               Identification Results             ')
display('——————————————————————————————————————————')
% Print frequencies
display('Identified frequencies')
for I=1:size(Frq,1)
    fprintf('Mode: %d; Modal Frequency: %6.4g (Hz)\n',I,Frq(I))
end
% Print Mode shapes
display('Related mode shapes')
for I=1:size(Frq,1)
    fprintf('Mode shape # %d:\n\n',I)
    disp(phi(:,I))
end

end

%% ————————————————— subfunctions —————————————————
function [Frq,phi,Fp,s1] = Identifier(PSD,F)

% Compute SVD of the PSD at each frequency
for I=1:size(PSD,3)
    [u,s,~] = svd(PSD(:,:,I));
    s1(I) = s(1);                           % First eigen values
    s2(I) = s(2,2);                         % Second eigen values
    ms(:,I)=u(:,1);                         % Mode shape
end

% Plot first singular values of the PSD matrix
figure
set(gcf,'color','w');
hold on
plot(F,mag2db(s1),'k')
box on
xlabel('Frequency (Hz)')
ylabel('1st Singular values of the PSD matrix (db)')
xlim([0 50])

% ——————————————————————————————————————————————
% Peak selection
% a: Draw rectangles around peaks while holding left click
% b: Press "Space" key to continue peak selection
% c: Press "any other key" if you have selected a peak by mistake and want
% to ignore it
% ----------------------------------------------------------------------------

Fp=[];% Frequencies related to selected peaks
NumPeaks=input('Enter the number of desired peaks:');
display('------------------------------------------------------------')
display('Peak selection procedure')
display('a: Draw rectangles around peaks while holding left click')
display('b: Press "Space" key to continue the peak selection')
display('c: Press "any other key" if you have selected a peak by'...
    'mistake and want to ignore it')
k=0;
while k~=NumPeaks
    A=getrect;                          % Draw a rectangle around the peak
    [~,P1]=min(abs(F-A(1)));
    [~,P2]=min(abs(F-(A(1)+A(3))));
    [~,B]=max(s1(P1:P2));
    Max=B+P1-1;                         % Frequency at the selected peak
    scatter(F(Max),mag2db(s1(Max)),...
        'MarkerEdgeColor','b','MarkerFaceColor','b')      % Mark this peak
    pause;key=get(gcf,'CurrentKey');
    Fp(end+1,:)=[Max,F(Max)];
    if strcmp(key,'space')
        % Press space to continue peak selection
        k=k+1;
        scatter(F(Max),mag2db(s1(Max)),...
  'MarkerEdgeColor','g','MarkerFaceColor','g') % Mark this peak as green
    else
        % Press any other key to ignore this peak
        Fp(end,:)=[];
        scatter(F(Max),mag2db(s1(Max)),'MarkerEdgeColor','r',...
            'MarkerFaceColor','r')      % Mark this peak as red
    end
end
% Number selected peaks, respectively
[~,Sr]=sort(Fp(:,2));
Fp=Fp(Sr,:);
clf
plot(F,mag2db(s1),'k')
set(gcf,'color','w');
box on
hold on
xlabel('Frequency (Hz)')
ylabel('1st Singular values of the PSD matrix (db)')
for I=1:size(Fp,1)
    scatter(Fp(I,2),mag2db(s1(Fp(I,1))),'MarkerEdgeColor','g',...
        'MarkerFaceColor','g')
    text(Fp(I,2), mag2db(s1(Fp(I,1)))*1.05, mat2str(I))
end
% Identified modal frequencies
Frq=Fp(:,2);
% Compute mode shapes for each selected peak
for J=1:size(Fp,1)
    [ug, ~, ~] = svd(PSD(:,:,Fp(J,1)));
    phi(:,J) = ug(:,1);
end

%% Added by Leander
dim = size(phi);
if dim(1) == 5
    figure()
    set(gcf,'color','w');
    grid on; hold on
    xlim([-.1 dim(1)+1.1]);
    ylim([-1.1 1.1]);
    xvalP = 0:1:dim(1)+1;
    phiPlot = zeros(dim(1)+2,dim(2));
    phiPlot(2:dim(1)+1,1:dim(2)) = phi;
elseif dim(1) == 6
    figure()
    set(gcf,'color','w');
    grid on; hold on
    xlim([-.1 dim(1)-.9]);
    ylim([-1.1 1.1]);
    xvalP = 0:1:dim(1)-1;
    phiPlot = phi;
end
    for i = 1:size(Fp,1)
        plot(xvalP,real(phiPlot(:,i))/(max(abs(real(phiPlot(:,i))))));
        box on
    end
        if size(Fp,1) == 1
            legend(['Mode: 1 ' num2str(Frq(1))]);
        elseif size(Fp,1) == 2
            legend(['Mode: 1, Frq=' num2str(Frq(1))], ...
                ['Mode: 2, Frq=' num2str(Frq(2))]);
        elseif size(Fp,1) == 3
            legend(['Mode: 1, Frq=' num2str(Frq(1))], ...
['Mode: 2, Frq=' num2str(Frq(2))], ...['Mode: 3, Frq=' num2str(Frq(3))]);
        elseif size(Fp,1) == 4
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq = '...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))],...
['Mode: 4, Frq=' num2str(Frq(4))]);
        elseif size(Fp,1) == 5
```

```
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))]);
         elseif size(Fp,1) == 6
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))],...
['Mode: 6, Frq=' num2str(Frq(6))]);
         elseif size(Fp,1) == 7
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))]);
         elseif size(Fp,1) == 8
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))]);
         elseif size(Fp,1) == 9
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))]);
         elseif size(Fp,1) == 10
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))]);
         elseif size(Fp,1) == 11
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))], ['Mode: 11, Frq=' num2str(Frq(11))]);
         elseif size(Fp,1) == 12
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))], ['Mode: 11, Frq=' num2str(Frq(11))],...
['Mode: 12, Frq=' num2str(Frq(12))]);
```

```
        elseif size(Fp,1) == 13
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))], ['Mode: 11, Frq=' num2str(Frq(11))],...
['Mode: 12, Frq=' num2str(Frq(12))], ['Mode: 13, Frq=' num2str(Frq(13))]);
        elseif size(Fp,1) == 14
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))], ['Mode: 11, Frq=' num2str(Frq(11))],...
['Mode: 12, Frq=' num2str(Frq(12))], ['Mode: 13, Frq='...
num2str(Frq(13))], ['Mode: 14, Frq=' num2str(Frq(14))]);
        elseif size(Fp,1) == 15
legend(['Mode: 1, Frq=' num2str(Frq(1))], ['Mode: 2, Frq='...
num2str(Frq(2))], ['Mode: 3, Frq=' num2str(Frq(3))], ['Mode: 4, Frq='...
num2str(Frq(4))], ['Mode: 5, Frq=' num2str(Frq(5))], ['Mode: 6, Frq='...
num2str(Frq(6))], ['Mode: 7, Frq=' num2str(Frq(7))], ['Mode: 8, Frq='...
num2str(Frq(8))], ['Mode: 9, Frq=' num2str(Frq(9))], ['Mode: 10, Frq='...
num2str(Frq(10))], ['Mode: 11, Frq=' num2str(Frq(11))],...
['Mode: 12, Frq=' num2str(Frq(12))], ['Mode: 13, Frq='...
num2str(Frq(13))], ['Mode: 14, Frq=' num2str(Frq(14))],...
['Mode: 15, Frq=' num2str(Frq(15))]);
        end
end
```

B.3 Moving standard deviation

Moving standard deviation plot

```
function MovingStandardDevationPlot(Hit,alt,Acc1,Acc2,SamplePoints,...
    xlimLow,xlimHigh,ylimLow,ylimHigh,funcCount)
%----------------------------------------------------%
%%%%% Change these paramaters %%%%%%%

% Hit          = 7;      %Impact number, 1 to 10 representing x1 to x10
% alt          = 1;      %Three different alternatives for every hit
% Acc1         = 4;
% Acc2         = 6;
% SamplePoints = 9;
% xlimLow      = 4.5;
% xlimHigh     = 5.4;
%----------------------------------------------------%

if Hit == 1
    if alt == 1
        recname='2017-08-25-10-02-41 - X1_2.mat';
    elseif alt == 2
        recname='2017-08-25-10-03-11 - X1_3.mat';
    elseif alt == 3
        recname='2017-08-25-10-04-09 - X1_5.mat';
    end
elseif Hit == 2
    if alt == 1
        recname='2017-08-25-10-24-22 - X2_5.mat';
    elseif alt == 2
        recname='2017-08-25-10-25-04 - X2_6.mat';
    elseif alt == 3
        recname='2017-08-25-10-25-36 - X2_7.mat';
    end
elseif Hit == 3
    if alt == 1
        recname='2017-08-25-10-39-45 - X3_3.mat';
    elseif alt == 2
        recname='2017-08-25-10-41-30 - X3_6.mat';
    elseif alt == 3
        recname='2017-08-25-10-42-02 - X3_7.mat';
    end
elseif Hit == 4
    if alt == 1
        recname='2017-08-25-10-50-54 - X4_1.mat';
    elseif alt == 2
        recname='2017-08-25-10-51-57 - X4_3.mat';
    elseif alt == 3
        recname='2017-08-25-10-52-27 - X4_4.mat';
    end
elseif Hit == 5
    if alt == 1
        recname='2017-08-25-10-54-21 - X5_2.mat';
    elseif alt == 2
        recname='2017-08-25-10-55-23 - X5_4.mat';
    elseif alt == 3
        recname='2017-08-25-10-55-54 - X5_5.mat';
    end
elseif Hit == 6
    if alt == 1
        recname='2017-08-25-10-56-38 - X6_1.mat';
    elseif alt == 2
        recname='2017-08-25-10-57-08 - X6_2.mat';
    elseif alt == 3
        recname='2017-08-25-10-58-13 - X6_4.mat';
    end
elseif Hit == 7
    if alt == 1
        recname='2017-08-25-11-16-58 - X7_3.mat';
    elseif alt == 2
        recname='2017-08-25-11-17-59 - X7_5.mat';
    elseif alt == 3
        recname='2017-08-25-11-20-01 - X7_9.mat';
    end
elseif Hit == 8
    if alt == 1
        recname='2017-08-25-11-26-38 - X8_1.mat';
    elseif alt == 2
        recname='2017-08-25-11-27-40 - X8_3.mat';
    elseif alt == 3
        recname='2017-08-25-11-29-12 - X8_6.mat';
    end
elseif Hit == 9
    if alt == 1
        recname='2017-08-25-11-29-58 - X9_1.mat';
    elseif alt == 2
        recname='2017-08-25-11-31-59 - X9_5.mat';
    elseif alt == 3
        recname='2017-08-25-11-33-35 - X9_8.mat';
    end
elseif Hit == 10
    if alt == 1
        recname='2017-08-25-11-35-25 - X10_3.mat';
    elseif alt == 2
```

```
        recname='2017-08-25-11-35-57 - X10_4.mat';
    elseif alt == 3
        recname='2017-08-25-11-37-32 - X10_7.mat';
    end
end

load(recname);
Fs = 2048; % Sampling frequency
%number of samples in the measurement series
nt=length(ConvertedData.Data.MeasuredData(2).Data);
dt=1/Fs; %sample rate is 400 Hz
t=0:dt:dt*(nt-1); %create the corresponding time vector
T=t(end);
%Directions as in Abaqus model!
accelerometers =[4 6 5; 8 10 9; 12 14 13; 16 18 17; 20 22 21; 24 26 25];
accelerometers = accelerometers + 2;
A = [1 2 3 4 5 6];
AllAcc = [Acc1 Acc2];
if Hit < 3 && min(AllAcc) == 1
    disp('Data from A01 for hit X1 and X2 was garbage')
else

if Acc1 == 2 || Acc1 == 5
   y1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
   %*19.62 to go from V to m/s^2 (read desciption dodument.
else
   y1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
   %*19.62 to go from V to m/s^2 (read desciption dodument.
end

if Acc2 == 2 || Acc2 == 5
   y2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
   %*19.62 to go from V to m/s^2 (read desciption dodument.
else
    y2=-ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
end

    My1 = movstd(y1,SamplePoints)/1000;
    My2 = movstd(y2,SamplePoints)/1000;


    if funcCount == 1
        figure()
        subplot(2,2,1);hold on;
        set(gcf,'color','w');
        plot(t,My1,'k');
        box on
        xlabel('Time [sec]'); ylabel('Standard deviation [m/s^2]');
        title(['X' num2str(Hit)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        hold on
        plot(t,My2,'r');
        legend(['A0' num2str(Acc1)],['A0' num2str(Acc2)])
    elseif funcCount == 2
        subplot(2,2,2);hold on;
        plot(t,My1,'k');
        box on
%         xlabel('Time [sec]'); ylabel('Standard deviation [m/s^2]');
        title(['X' num2str(Hit)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        hold on
        plot(t,My2,'r');
        legend(['A0' num2str(Acc1)],['A0' num2str(Acc2)])
    elseif funcCount == 3
        subplot(2,2,3);hold on;
        plot(t,My1,'k');
        box on
        xlabel('Time [sec]'); ylabel('Standard deviation [m/s^2]');
%         title(['X' num2str(Hit)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        hold on
        plot(t,My2,'r');
        legend(['A0' num2str(Acc1)],['A0' num2str(Acc2)])
    elseif funcCount == 4
        subplot(2,2,4);hold on;
        plot(t,My1,'k');
        box on
        xlabel('Time [sec]'); %ylabel('Standard deviation [m/s^2]');
%         title(['X' num2str(Hit)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        hold on
        plot(t,My2,'r');
        legend(['A0' num2str(Acc1)],['A0' num2str(Acc2)])
    end

end
end
```

Moving standard deviation subplot

```
%% Cross Corolation Subplot
clear all
clc
close all
%% Parameter same for all plots
alt = 1;
SamplePoints = 21;
xlimLow   = 4.75;
xlimHigh  = 5;
ylimLow   = -.001;
ylimHigh  = .025;
%% First Plot
funcCount = 1;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 4;
Acc2      = 6;
MovingStandardDevationPlot(Hit,alt ,Acc1,Acc2,SamplePoints,xlimLow,...
    xlimHigh,ylimLow,ylimHigh,funcCount)
%% Second Plot
funcCount = 2;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 3;
Acc2      = 1;
MovingStandardDevationPlot(Hit,alt ,Acc1,Acc2,SamplePoints,xlimLow,...
    xlimHigh,ylimLow,ylimHigh,funcCount)
%% Third Plot
funcCount = 3;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 3;
Acc2      = 1;
MovingStandardDevationPlot(Hit,alt ,Acc1,Acc2,SamplePoints,xlimLow,...
    xlimHigh,ylimLow,ylimHigh,funcCount)
%% Fourth Plot
funcCount = 4;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 4;
Acc2      = 6;
MovingStandardDevationPlot(Hit,alt ,Acc1,Acc2,SamplePoints,xlimLow,...
    xlimHigh,ylimLow,ylimHigh,funcCount)
```

B.4 Cross correlation

Cross correlation function

```
function CrossCorrFunc(Hit,alt ,Reference,Acc2,funcCount)
%-----------------------------------------------------------%
%%%% Change these paramaters %%%%%
```

```
xlimLow    = -.5;
xlimHigh   = .5;
ylimLow    = -.7;
ylimHigh   = .7;
%--------------------------------------------------%

if Hit == 1
    if alt == 1
        recname='2017-08-25-10-02-41 - X1_2.mat';
    elseif alt == 2
        recname='2017-08-25-10-03-11 - X1_3.mat';
    elseif alt == 3
        recname='2017-08-25-10-04-09 - X1_5.mat';
    end
elseif Hit == 2
    if alt == 1
        recname='2017-08-25-10-24-22 - X2_5.mat';
    elseif alt == 2
        recname='2017-08-25-10-25-04 - X2_6.mat';
    elseif alt == 3
        recname='2017-08-25-10-25-36 - X2_7.mat';
    end
elseif Hit == 3
    if alt == 1
        recname='2017-08-25-10-39-45 - X3_3.mat';
    elseif alt == 2
        recname='2017-08-25-10-41-30 - X3_6.mat';
    elseif alt == 3
        recname='2017-08-25-10-42-02 - X3_7.mat';
    end
elseif Hit == 4
    if alt == 1
        recname='2017-08-25-10-50-54 - X4_1.mat';
    elseif alt == 2
        recname='2017-08-25-10-51-57 - X4_3.mat';
    elseif alt == 3
        recname='2017-08-25-10-52-27 - X4_4.mat';
    end
elseif Hit == 5
    if alt == 1
        recname='2017-08-25-10-54-21 - X5_2.mat';
    elseif alt == 2
        recname='2017-08-25-10-55-23 - X5_4.mat';
    elseif alt == 3
        recname='2017-08-25-10-55-54 - X5_5.mat';
    end
elseif Hit == 6
```

```
    if alt == 1
        recname='2017-08-25-10-56-38 - X6_1.mat';
    elseif alt == 2
        recname='2017-08-25-10-57-08 - X6_2.mat';
    elseif alt == 3
        recname='2017-08-25-10-58-13 - X6_4.mat';
    end
elseif Hit == 7
    if alt == 1
        recname='2017-08-25-11-16-58 - X7_3.mat';
    elseif alt == 2
        recname='2017-08-25-11-17-59 - X7_5.mat';
    elseif alt == 3
        recname='2017-08-25-11-20-01 - X7_9.mat';
    end
elseif Hit == 8
    if alt == 1
        recname='2017-08-25-11-26-38 - X8_1.mat';
    elseif alt == 2
        recname='2017-08-25-11-27-40 - X8_3.mat';
    elseif alt == 3
        recname='2017-08-25-11-29-12 - X8_6.mat';
    end
elseif Hit == 9
    if alt == 1
        recname='2017-08-25-11-29-58 - X9_1.mat';
    elseif alt == 2
        recname='2017-08-25-11-31-59 - X9_5.mat';
    elseif alt == 3
        recname='2017-08-25-11-33-35 - X9_8.mat';
    end
elseif Hit == 10
    if alt == 1
        recname='2017-08-25-11-35-25 - X10_3.mat';
    elseif alt == 2
        recname='2017-08-25-11-35-57 - X10_4.mat';
    elseif alt == 3
        recname='2017-08-25-11-37-32 - X10_7.mat';
    end
end

load(recname);
Fs = 2048; % Sampling frequency
%number of samples in the measurement series
nt=length(ConvertedData.Data.MeasuredData(2).Data);
dt=1/Fs;
t=0:dt:dt*(nt-1); %create the corresponding time vector
```

```
T=t(end);
%Directions as in Abaqus model!
accelerometers =[4 6 5; 8 10 9; 12 14 13; 16 18 17; 20 22 21; 24 26 25];
accelerometers = accelerometers + 2;
A = [1 2 3 4 5 6];
AllAcc = [Reference Acc2];
if Hit < 3 && min(AllAcc) == 1
    disp('Data from A01 for hit X1 and X2 was garbage')
else

    if Reference == 2 || Reference == 5
        y1=ConvertedData.Data.MeasuredData(...
            accelerometers(Reference,2)).Data*19.62;
        %*19.62 to go from V to m/s^2 (read desciption dodument.
    else
        y1=-ConvertedData.Data.MeasuredData(...
            accelerometers(Reference,2)).Data*19.62;
        %*19.62 to go from V to m/s^2 (read desciption dodument.
    end

    if Acc2 == 2 || Acc2 == 5
        y2=ConvertedData.Data.MeasuredData(...
            accelerometers(Acc2,2)).Data*19.62;
        %*19.62 to go from V to m/s^2 (read desciption dodument.
    else
        y2=-ConvertedData.Data.MeasuredData(...
            accelerometers(Acc2,2)).Data*19.62;
        %*19.62 to go from V to m/s^2 (read desciption dodument.
    end

    [acory,lagy] = xcorr(y2,y1,1000,'coeff');
    [~,Iy] = max(abs(acory));
    lagDiffy = lagy(Iy);
    timeDiffy = lagDiffy/Fs;
    lagy = lagy/Fs;

    if funcCount == 1
        figure()
        subplot(4,2,1);hold on;
        set(gcf,'color','w');
        plot(lagy,acory,'k')
        box on
        ylabel('Cross correlation (norm.)');
        title(['X' num2str(Hit)]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
    elseif funcCount == 2
```

```
        subplot(4,2,2);hold on;
        plot(lagy,acory,'k')
        box on
        title(['X' num2str(Hit)]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
    elseif funcCount == 3
        subplot(4,2,3);hold on;
        plot(lagy,acory,'k')
        box on
        ylabel('Cross correlation (norm.)');
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
     elseif funcCount == 4
        subplot(4,2,4);hold on;
        plot(lagy,acory,'k')
        box on
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
     elseif funcCount == 5
        subplot(4,2,5);hold on;
        plot(lagy,acory,'k')
        box on
        ylabel('Cross correlation (norm.)');
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
     elseif funcCount == 6
        subplot(4,2,6);hold on;
        plot(lagy,acory,'k')
        box on
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
    elseif funcCount == 7
        subplot(4,2,7);hold on;
        plot(lagy,acory,'k')
        box on
        xlabel('Lag \tau');
        ylabel('Cross correlation (norm.)');
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
     elseif funcCount == 8
        subplot(4,2,8);hold on;
        plot(lagy,acory,'k')
        box on
        xlabel('Lag \tau');
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc2) 'to A0' num2str(Reference)])
    end

end
end
```

Cross correlation subplot

```
%% Cross Corolation Subplot
clear all
clc
close all
%% Parameter same for all plots
alt = 1;
%% First Plot
funcCount = 1;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Reference = 5;
Acc2      = 6;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Second Plot
funcCount = 2;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Reference = 2;
Acc2      = 1;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Third Plot
funcCount = 3;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Reference = 5;
Acc2      = 4;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Fourth Plot
funcCount = 4;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Reference = 2;
Acc2      = 3;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Fifth Plot
funcCount = 5;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Reference = 5;
Acc2      = 3;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Sixht Plot
funcCount = 6;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Reference = 2;
```

```
Acc2      = 4;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Seventh Plot
funcCount = 7;
Hit       = 7;       %Impact number, 1 to 10 representing x1 to x10
Reference = 5;
Acc2      = 1;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
%% Eight Plot
funcCount = 8;
Hit       = 3;       %Impact number, 1 to 10 representing x1 to x10
Reference = 2;
Acc2      = 6;
CrossCorrFunc(Hit,alt,Reference,Acc2,funcCount)
```

B.5 Normalization

Normalization plot

```
function NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow, xlimHigh,funcCount)
%---------------------------------------------------------------------%
%%%%% Change these paramaters %%%%%%

% Hit1      = 3;      %Impact number, 1 to 10 representing x1 to x10
% Hit2      = 3;      %Impact number, 1 to 10 representing x1 to x10
% Acc1      = 1;
% Acc2      = 3;
% alt       = 1;      %Three different alternatives for every hit
% normalize = 1;      %If normalize = 0 the plot is not normalized.
                      %Normalized if normalize = 1.
% detect    = .5;     %The plot will not start when the absoulute value
                      %is lower than this value
% minAcc    = 10;     %The plot will always start if the absoulute value
                      %of the acceleration is bigger minAcc
% xlimLow   = 0;
% xlimHigh  = .2;
% ylimLow   = -20;
% ylimHigh  = 20;
%---------------------------------------------------------------------%

ylimLow   = -.7;
ylimHigh  = .7;

if Hit1 == 1
    if alt == 1
        recname1='2017-08-25-10-02-41 - X1_2.mat';
    elseif alt == 2
        recname1='2017-08-25-10-03-11 - X1_3.mat';
    elseif alt == 3
        recname1='2017-08-25-10-04-09 - X1_5.mat';
    end
elseif Hit1 == 2
    if alt == 1
        recname1='2017-08-25-10-24-22 - X2_5.mat';
    elseif alt == 2
        recname1='2017-08-25-10-25-04 - X2_6.mat';
    elseif alt == 3
        recname1='2017-08-25-10-25-36 - X2_7.mat';
    end
elseif Hit1 == 3
    if alt == 1
```

```
        recname1='2017-08-25-10-39-45 - X3_3.mat';
    elseif alt == 2
        recname1='2017-08-25-10-41-30 - X3_6.mat';
    elseif alt == 3
        recname1='2017-08-25-10-42-02 - X3_7.mat';
    end
elseif Hit1 == 4
    if alt == 1
        recname1='2017-08-25-10-50-54 - X4_1.mat';
    elseif alt == 2
        recname1='2017-08-25-10-51-57 - X4_3.mat';
    elseif alt == 3
        recname1='2017-08-25-10-52-27 - X4_4.mat';
    end
elseif Hit1 == 5
    if alt == 1
        recname1='2017-08-25-10-54-21 - X5_2.mat';
    elseif alt == 2
        recname1='2017-08-25-10-55-23 - X5_4.mat';
    elseif alt == 3
        recname1='2017-08-25-10-55-54 - X5_5.mat';
    end
elseif Hit1 == 6
    if alt == 1
        recname1='2017-08-25-10-56-38 - X6_1.mat';
    elseif alt == 2
        recname1='2017-08-25-10-57-08 - X6_2.mat';
    elseif alt == 3
        recname1='2017-08-25-10-58-13 - X6_4.mat';
    end
elseif Hit1 == 7
    if alt == 1
        recname1='2017-08-25-11-16-58 - X7_3.mat';
    elseif alt == 2
        recname1='2017-08-25-11-17-59 - X7_5.mat';
    elseif alt == 3
        recname1='2017-08-25-11-20-01 - X7_9.mat';
    end
elseif Hit1 == 8
    if alt == 1
        recname1='2017-08-25-11-26-38 - X8_1.mat';
    elseif alt == 2
        recname1='2017-08-25-11-27-40 - X8_3.mat';
    elseif alt == 3
        recname1='2017-08-25-11-29-12 - X8_6.mat';
    end
elseif Hit1 == 9
```

```
    if alt == 1
        recname1='2017-08-25-11-29-58 - X9_1.mat';
    elseif alt == 2
        recname1='2017-08-25-11-31-59 - X9_5.mat';
    elseif alt == 3
        recname1='2017-08-25-11-33-35 - X9_8.mat';
    end
elseif Hit1 == 10
    if alt == 1
        recname1='2017-08-25-11-35-25 - X10_3.mat';
    elseif alt == 2
        recname1='2017-08-25-11-35-57 - X10_4.mat';
    elseif alt == 3
        recname1='2017-08-25-11-37-32 - X10_7.mat';
    end
end

load(recname1);
Fs = 2048; % Sampling frequency
%number of samples in the measurement series
nt=length(ConvertedData.Data.MeasuredData(2).Data);
dt=1/Fs; %sample rate is 400 Hz
t=0:dt:dt*(nt-1); %create the corresponding time vector
T=t(end);
%Directions as in Abaqus model!
accelerometers =[4 6 5; 8 10 9; 12 14 13; 16 18 17; 20 22 21; 24 26 25];
accelerometers = accelerometers + 2;
A = [1 2 3 4 5 6];
if Hit1 < 3 && Acc1 == 1
    disp('Data from A01 for hit X1 and X2 was garbage')
else

    if Acc1 == 2 || Acc1 == 5
%    x1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,1)).Data*19.62;

%*19.62 to go from V to m/s^2 (read desciption dodument.
    y1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
%     z1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,3)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
    else
%    x1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,1)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
   y1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
   %*19.62 to go from V to m/s^2 (read desciption dodument.
%  z1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,3)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
    end

    startPlot = 0;
    for i = Fs/8+1:length(y1)
        if abs(y1(i,1)) > 10*abs(y1(i-Fs/8,1)) && abs(y1(i,1))...
                > detect || abs(y1(i,1)) > minAcc
            startPlot = i;
            break
        end
    end

    yplot1 = y1(startPlot-Fs/64:length(t));
    xplot1 = 0:dt:(length(yplot1)-1)*dt;
    normVal1 = max(y1)-min(y1);

    if funcCount == 1
        figure();
        subplot(4,2,funcCount);grid on; hold on;
        set(gcf,'color','w');
        plot(xplot1,yplot1/normVal1,'k');
        box on
%         xlabel('Time [sec]');
        ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
        title(['X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 2
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
%         xlabel('Time [sec]'); ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
        title(['X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 3
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
%         xlabel('Time [sec]');
        ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%         title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 4
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
%           xlabel('Time [sec]'); ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%           title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 5
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
%           xlabel('Time [sec]');
        ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%           title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 6
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
%           xlabel('Time [sec]'); ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%           title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 7
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
        xlabel('Time [sec]'); ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%           title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 8
        subplot(4,2,funcCount);grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        box on
        xlabel('Time [sec]'); %ylabel('Norm. acc.');
        %title([num2str(recname) '. y-Direction']);
%          title(['Impact X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    elseif funcCount == 9
        figure();grid on; hold on;
        plot(xplot1,yplot1/normVal1,'k');
        set(gcf,'color','w');
        box on
        xlabel('Time [sec]');
        ylabel('Norm. acc.');
        title(['X' num2str(Hit1)]);
        xlim([xlimLow xlimHigh]);
        ylim([ylimLow ylimHigh]);
        legend(['A0' num2str(Acc1)])
    end


end

end
```

Normalization subplot

```
%% SUBPLOTS
clear all
clc
close all
%% Same for all plots
alt       = 1;        %Three different alternatives for every hit
xlimLow   = 0;
xlimHigh  = .25;
%% First Plot
funcCount = 1;
Hit1      = 3;        %Impact number, 1 to 10 representing x1 to x10
Acc1      = 3;
detect    = .5;       %The plot will not start when the absoulute value is
                      %lower than this value
minAcc    = 10;       %The plot will always start if the absoulute value of
                      %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Second Plot
funcCount = 2;
Hit1      = 7;        %Impact number, 1 to 10 representing x1 to x10
Acc1      = 4;
```

```
detect    = .5;      %The plot will not start when the absoulute value is
                     %lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Third Plot
funcCount = 3;
Hit1      = 3;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 1;
detect    = .5;      %The plot will not start when the absoulute value is
                     %lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Fourth Plot
funcCount = 4;
Hit1      = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 6;
detect    = .5;      %The plot will not start when the absoulute value is
                     %lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Fifth Plot
funcCount = 5;
Hit1      = 3;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 4;
detect    = .5;      %The plot will not start when the absoulute value is
                     %lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Sixth Plot
funcCount = 6;
Hit1      = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 3;
detect    = .5;      %The plot will not start when the absoulute value
                     %is lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value
                     %of the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Seventh Plot
```

```
funcCount = 7;
Hit1      = 3;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 6;
detect    = .5;      %The plot will not start when the absoulute value
                     %is lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
%% Eigth Plot
funcCount = 8;
Hit1      = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 1;
detect    = .5;      %The plot will not start when the absoulute value is
                     %lower than this value
minAcc    = 10;      %The plot will always start if the absoulute value of
                     %the acceleration is bigger minAcc
NormalizationPlot(Hit1, Acc1, alt, detect, minAcc, xlimLow,...
    xlimHigh,funcCount)
```

B.6 Frequency response function

```
clc
clear all
close all
%————————————————————————————————————————————%
%%%%% Change these paramaters %%%%%%%

Hit1      = 7;       %Impact number, 1 to 10 representing x1 to x10
Acc1      = 1;
Acc2      = 3;
alt       = 3;       %Three different alternatives for every hit
Nwelch    = 0;

FreqPlotStart=.5;        %Defines start of x axis in plot
FreqPlotEnd=200;       %Defines end of x axis in plot
PlotFreq=200;            %End of frequency value to mark peaks
MinPeakDist = 0.5;     %Minimum distance between peaks
ylimLow = -.0001;
ylimHigh = .015;

%————————————————————————————————————————————%
%Directions as in Abaqus model!
accelerometers =[4 6 5; 8 10 9; 12 14 13; 16 18 17; 20 22 21; 24 26 25];
accelerometers = accelerometers + 2;
A = [1 2 3 4 5 6];
Fs = 2048; % Sampling frequency
```

```
dt=1/Fs;

if Hit1 == 1
    if alt == 1
        recname1='2017-08-25-10-02-41 - X1_2.mat';
    elseif alt == 2
        recname1='2017-08-25-10-03-11 - X1_3.mat';
    elseif alt == 3
        recname1='2017-08-25-10-04-09 - X1_5.mat';
    end
elseif Hit1 == 2
    if alt == 1
        recname1='2017-08-25-10-24-22 - X2_5.mat';
    elseif alt == 2
        recname1='2017-08-25-10-25-04 - X2_6.mat';
    elseif alt == 3
        recname1='2017-08-25-10-25-36 - X2_7.mat';
    end
elseif Hit1 == 3
    if alt == 1
        recname1='2017-08-25-10-39-45 - X3_3.mat';
    elseif alt == 2
        recname1='2017-08-25-10-41-30 - X3_6.mat';
    elseif alt == 3
        recname1='2017-08-25-10-42-02 - X3_7.mat';
    end
elseif Hit1 == 4
    if alt == 1
        recname1='2017-08-25-10-50-54 - X4_1.mat';
    elseif alt == 2
        recname1='2017-08-25-10-51-57 - X4_3.mat';
    elseif alt == 3
        recname1='2017-08-25-10-52-27 - X4_4.mat';
    end
elseif Hit1 == 5
    if alt == 1
        recname1='2017-08-25-10-54-21 - X5_2.mat';
    elseif alt == 2
        recname1='2017-08-25-10-55-23 - X5_4.mat';
    elseif alt == 3
        recname1='2017-08-25-10-55-54 - X5_5.mat';
    end
elseif Hit1 == 6
    if alt == 1
        recname1='2017-08-25-10-56-38 - X6_1.mat';
    elseif alt == 2
        recname1='2017-08-25-10-57-08 - X6_2.mat';
    elseif alt == 3
        recname1='2017-08-25-10-58-13 - X6_4.mat';
    end
elseif Hit1 == 7
    if alt == 1
        recname1='2017-08-25-11-16-58 - X7_3.mat';
    elseif alt == 2
        recname1='2017-08-25-11-17-59 - X7_5.mat';
    elseif alt == 3
        recname1='2017-08-25-11-20-01 - X7_9.mat';
    end
elseif Hit1 == 8
    if alt == 1
        recname1='2017-08-25-11-26-38 - X8_1.mat';
    elseif alt == 2
        recname1='2017-08-25-11-27-40 - X8_3.mat';
    elseif alt == 3
        recname1='2017-08-25-11-29-12 - X8_6.mat';
    end
elseif Hit1 == 9
    if alt == 1
        recname1='2017-08-25-11-29-58 - X9_1.mat';
    elseif alt == 2
        recname1='2017-08-25-11-31-59 - X9_5.mat';
    elseif alt == 3
        recname1='2017-08-25-11-33-35 - X9_8.mat';
    end
elseif Hit1 == 10
    if alt == 1
        recname1='2017-08-25-11-35-25 - X10_3.mat';
    elseif alt == 2
        recname1='2017-08-25-11-35-57 - X10_4.mat';
    elseif alt == 3
        recname1='2017-08-25-11-37-32 - X10_7.mat';
    end
end

load(recname1);
%number of samples in the measurement series
nt=length(ConvertedData.Data.MeasuredData(2).Data);
t=0:dt:dt*(nt-1); %create the corresponding time vector
T=t(end);

if Hit1 < 3 && Acc1 == 1
    disp('Data from A01 for hit X1 and X2 was garbage')
else
```

```
if Acc1 == 2 || Acc1 == 5
x1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,1)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
y1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
z1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,3)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
else
x1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,1)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
y1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
z1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,3)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
end

if Acc2 == 2 || Acc2 == 5
x2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,1)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
y2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
z2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,3)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
else
x2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,1)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
y2=-ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
z2=-ConvertedData.Data.MeasuredData(accelerometers(Acc2,3)).Data*19.62;
%*19.62 to go from V to m/s^2 (read desciption dodument.
end

if Nwelch == 0
    [Pyy1,fy1] = pwelch(y1,[],[],[],Fs);
    [Pyy2,fy2] = pwelch(y2,[],[],[],Fs);
else
    Nwindow=round(length(t)/Nwelch); %length of window
    nfft=2^nextpow2(Nwindow); %number of FFT points
    [Pyy1,fy1] = pwelch(y1,hanning(Nwindow),[],nfft,Fs);
    [Pyy2,fy2] = pwelch(y2,hanning(Nwindow),[],nfft,Fs);
end

    figure();hold on;
    set(gcf,'color','w');
    plot(fy1,(Pyy1)*100,'k');
    plot(fy2,(Pyy2)*100,'r');
    box on
```

```
        xlabel('f [Hz]'); ylabel('Power');
        title(['X' num2str(Hit1) ]);
        xlim([FreqPlotStart FreqPlotEnd])
        ylim([ylimLow ylimHigh])
        legend(['A0' num2str(Acc1)],['A0' num2str(Acc2)])

end
```

FRFvsPSD

```
clc
clear all
close all
%----------------------------------------------------------------%
%%%% Change these paramaters %%%%%%

Hit1        = 3;        %Impact number, 1 to 10 representing x1 to x10
Acc1        = 1;
Acc2        = 3;
alt         = 3;        %Three different alternatives for every hit

FreqPlotStart=.5;          %Defines start of x axis in plot
FreqPlotEnd=100;           %Defines end of x axis in plot
ylimLow = -.0001;
ylimHigh = .0105;

%----------------------------------------------------------------%
%Directions as in Abaqus model!
accelerometers =[4 6 5; 8 10 9; 12 14 13; 16 18 17; 20 22 21; 24 26 25];
accelerometers = accelerometers + 2;
A = [1 2 3 4 5 6];
Fs = 2048; % Sampling frequency
dt=1/Fs;

if Hit1 == 1
    if alt == 1
        recname1='2017-08-25-10-02-41 - X1_2.mat';
    elseif alt == 2
        recname1='2017-08-25-10-03-11 - X1_3.mat';
    elseif alt == 3
        recname1='2017-08-25-10-04-09 - X1_5.mat';
    end
elseif Hit1 == 2
    if alt == 1
        recname1='2017-08-25-10-24-22 - X2_5.mat';
    elseif alt == 2
        recname1='2017-08-25-10-25-04 - X2_6.mat';
    elseif alt == 3
```

```
            recname1='2017-08-25-10-25-36 - X2_7.mat';
        end
    elseif Hit1 == 3
        if alt == 1
            recname1='2017-08-25-10-39-45 - X3_3.mat';
        elseif alt == 2
            recname1='2017-08-25-10-41-30 - X3_6.mat';
        elseif alt == 3
            recname1='2017-08-25-10-42-02 - X3_7.mat';
        end
    elseif Hit1 == 4
        if alt == 1
            recname1='2017-08-25-10-50-54 - X4_1.mat';
        elseif alt == 2
            recname1='2017-08-25-10-51-57 - X4_3.mat';
        elseif alt == 3
            recname1='2017-08-25-10-52-27 - X4_4.mat';
        end
    elseif Hit1 == 5
        if alt == 1
            recname1='2017-08-25-10-54-21 - X5_2.mat';
        elseif alt == 2
            recname1='2017-08-25-10-55-23 - X5_4.mat';
        elseif alt == 3
            recname1='2017-08-25-10-55-54 - X5_5.mat';
        end
    elseif Hit1 == 6
        if alt == 1
            recname1='2017-08-25-10-56-38 - X6_1.mat';
        elseif alt == 2
            recname1='2017-08-25-10-57-08 - X6_2.mat';
        elseif alt == 3
            recname1='2017-08-25-10-58-13 - X6_4.mat';
        end
    elseif Hit1 == 7
        if alt == 1
            recname1='2017-08-25-11-16-58 - X7_3.mat';
        elseif alt == 2
            recname1='2017-08-25-11-17-59 - X7_5.mat';
        elseif alt == 3
            recname1='2017-08-25-11-20-01 - X7_9.mat';
        end
    elseif Hit1 == 8
        if alt == 1
            recname1='2017-08-25-11-26-38 - X8_1.mat';
        elseif alt == 2
            recname1='2017-08-25-11-27-40 - X8_3.mat';
    elseif alt == 3
        recname1='2017-08-25-11-29-12 - X8_6.mat';
    end
elseif Hit1 == 9
    if alt == 1
        recname1='2017-08-25-11-29-58 - X9_1.mat';
    elseif alt == 2
        recname1='2017-08-25-11-31-59 - X9_5.mat';
    elseif alt == 3
        recname1='2017-08-25-11-33-35 - X9_8.mat';
    end
elseif Hit1 == 10
    if alt == 1
        recname1='2017-08-25-11-35-25 - X10_3.mat';
    elseif alt == 2
        recname1='2017-08-25-11-35-57 - X10_4.mat';
    elseif alt == 3
        recname1='2017-08-25-11-37-32 - X10_7.mat';
    end
end

load(recname1);
%number of samples in the measurement series
nt=length(ConvertedData.Data.MeasuredData(2).Data);
t=0:dt:dt*(nt-1); %create the corresponding time vector
T=t(end);

if Hit1 < 3 && Acc1 == 1
    disp('Data from A01 for hit X1 and X2 was garbage')
else

    if Acc1 == 2 || Acc1 == 5
    y1=ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
    else
    y1=-ConvertedData.Data.MeasuredData(accelerometers(Acc1,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
    end

    if Acc2 == 2 || Acc2 == 5
    y2=ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
    else
    y2=-ConvertedData.Data.MeasuredData(accelerometers(Acc2,2)).Data*19.62;
    %*19.62 to go from V to m/s^2 (read desciption dodument.
    end
    Hammer=ConvertedData.Data.MeasuredData(2).Data*4347.8261;
    %*4347.8261 to go from V/N to N/V (read desciption dodument.

    [Pyy1,fy1] = pwelch(y1,[],[],[],Fs);
    [Pyy2,fy2] = pwelch(y2,[],[],[],Fs);

    [Sxx,f] = pwelch(y1(:),[],[],[],Fs); % single-sided PSD
    [Sff,~] = pwelch(Hammer(:),[],[],[],Fs); % single-sided PSD
    [Sxf,~] = cpsd(y1(:),Hammer(:),[],[],[],Fs); % single-sided CSD
    [Sfx,~] = cpsd(Hammer(:),y1(:),[],[],[],Fs); % single-sided CSD

    [Sxx2,f2] = pwelch(y2(:),[],[],[],Fs); % single-sided PSD
    [Sff2,~] = pwelch(Hammer(:),[],[],[],Fs); % single-sided PSD
    [Sxf2,~] = cpsd(y2(:),Hammer(:),[],[],[],Fs); % single-sided CSD
    [Sfx2,~] = cpsd(Hammer(:),y2(:),[],[],[],Fs); % single-sided CSD

    H = (Sxf./Sff).';
    H2 = (Sxf2./Sff2).';

    figure();hold on
    set(gcf,'color','w');
    subplot(1,2,1);hold on
    plot(f,(abs(H)),'r');
    plot(fy1,(Pyy1)*100,'k');
    box on
    xlabel('f [Hz]'); ylabel('Power');
    title(['X' num2str(Hit1) ]);
    xlim([FreqPlotStart FreqPlotEnd])
    ylim([ylimLow ylimHigh])
    legend(['FRF A0' num2str(Acc1)],['PSD A0' num2str(Acc1)])

    set(gcf,'color','w');
    subplot(1,2,2); hold on
    plot(f2,(abs(H2)),'r');
    plot(fy2,(Pyy2)*100,'k');
    box on
    xlabel('f [Hz]'); ylabel('Power');
    title(['X' num2str(Hit1) ]);
    xlim([FreqPlotStart FreqPlotEnd])
    ylim([ylimLow ylimHigh])
    legend(['FRF A0' num2str(Acc2)],['PSD A0' num2str(Acc2)])
end
```

B.7 Non-linear spring formulation

```
clc
clear all
```

```
%The script produces the force-displacement relations for the
%Non-linear spring formulations
F0          = 0;      %Initial friction force
k3          = 2.25e7  %N/m Defines contact with underlying stiffener
u3          = 0.0045  %4.5mm clearing to the underlying L-profile
F3          = k3*u3;
kmax        = 2.25e7;
Fkmax       = kmax*u3
xlimLow = 0;
xlimHigh = 0.0045;
ylimLow = 0;
ylimHigh = 1e5;
%Parameters
%% Linear spring formulation
kl1                   = 1e6;
%% Multi linear penalty model
k1                   = 1e7;
k2                   = 1;
u1                   = 0.0005
u2                   = 0.004
%% Power-function-based nonlinear penalty model
m                    = 2;
i                    = 20;
ke                   = 1e9;
kebased              = 0;
mbased               = 1;
%% Combined quadratic-linear penalty model
uc0                   = -0.001;
uc1                   = 0.002;
kc1                   = 2e7;
i                    = 15;
%CALCULATIONS
%% Linear spring formulation
    uaa = [0,u3/2,u3];
    Fa = [0,(u3/2)*kl1,u3*kl1];
%% Multi linear penalty model calculations
    F1 = k1*u1
    F2 = F1+k2*u2
    ubb = [0,u1,u2,u3];
    Fb = [0,F1,F2,F3];
%% Power-function-based nonlinear penalty model calculations
    ucc = 0:u3/i:u3;
    Etest = (Fkmax)/(u3^m);
    mtest = log((Fkmax/ke))/log(0.0045);
    if kebased == 1;
    ke = Etest;
    end
    if mbased == 1;
    m = mtest;
    end
    Fc = ke*(abs(ucc)).^m;
    %% Combined quadratic-linear penalty model calculations
    uca = [uc0:uc1/i:uc1];
    %ua = sort(ua);
    ucb = u3;
    F1 = (1/(2*(uc1-uc0)))*kc1*((uca-uc0).^2);
    Ftest=F1(1)
    F2 = kc1*ucb-0.5*k1*(uc1+uc0);
    Fd  = [F1,F2];
    udd  = [uca,ucb];
%% Plotting the data
figure();hold on
plot(uaa,Fa','k')
plot(ubb,Fb,'--k')
plot(ucc,Fc,'--*k')
plot(udd,Fd,'-.k')
ylim([ylimLow ylimHigh]);
xlim([xlimLow xlimHigh]);
xlabel('u[mm]')
ylabel('F_f[N]')
legend('Linear','ML','PFB','CQL','location','northwest')
%% Output data for the FE model
AbaDataLinear=[uaa',Fa'];             %Linear spring formulation
AbaDataMultiLinear=[ubb',Fb'];        %Multi-linear spring formulation
AbaDataFBD=[ucc',Fc'];                %Power function-based spring formulation
AbaDataCQL=[udd',Fd'];                %Combined quadratic linear formulation
```

C Appendix C - Damage detection techniques performed on full scale measurements

C.1 Cross correlation

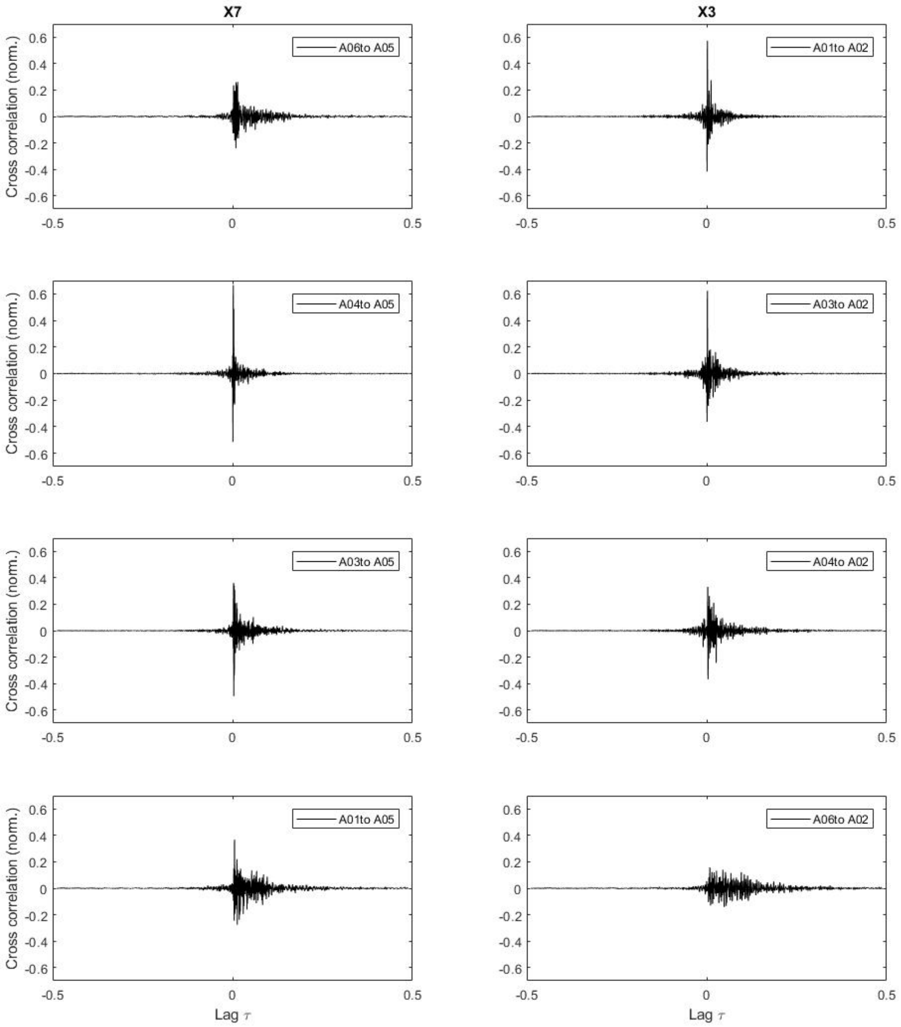


Figure C.1: Shows cross correlation for the second hit

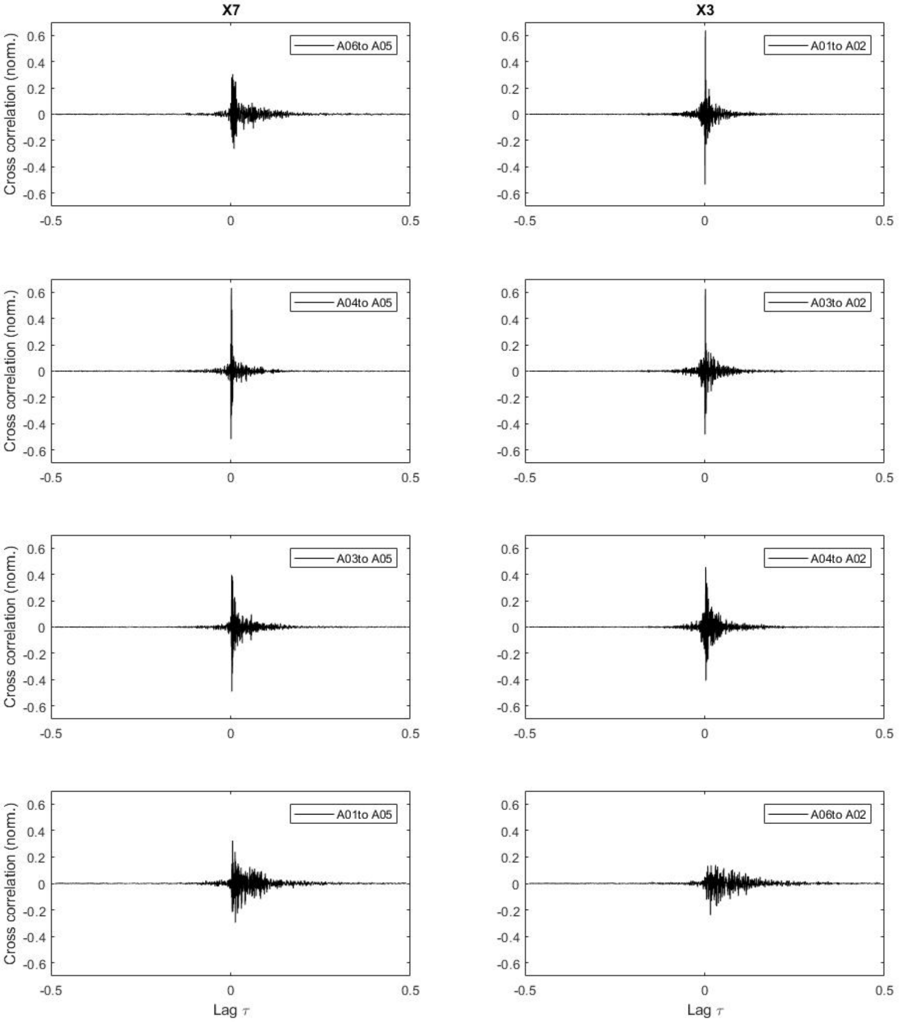


Figure C.2: Shows cross correlation for the third hit

D Appendix D - Simulation results

D.1 Spring formulations

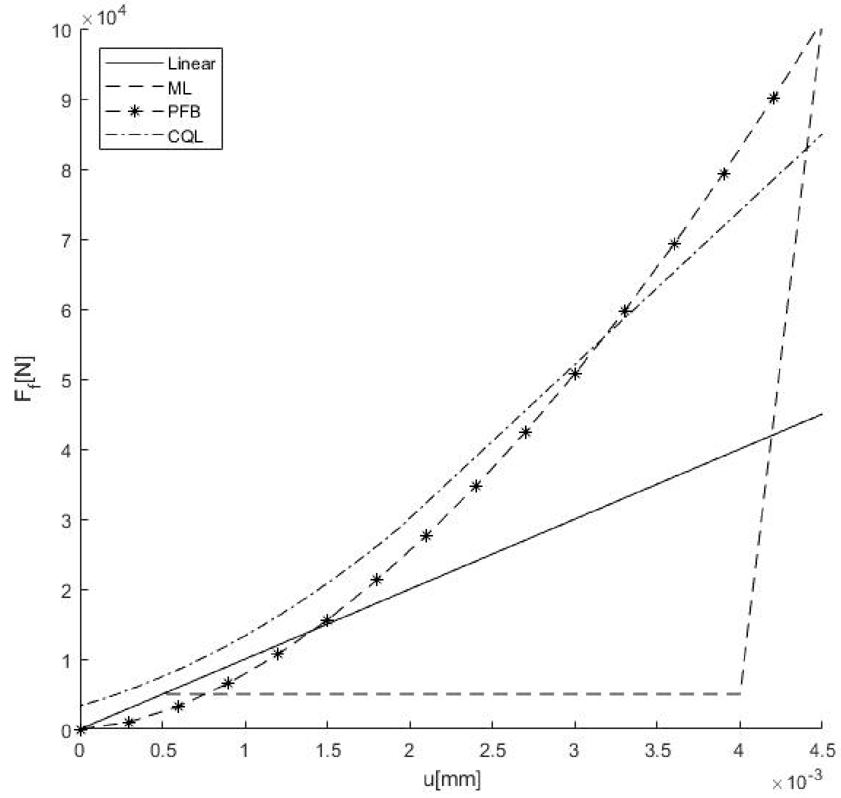


Figure D.1: Stiffness profiles of the non-linear springs.

Linear spring formulation, k=10 MN/m

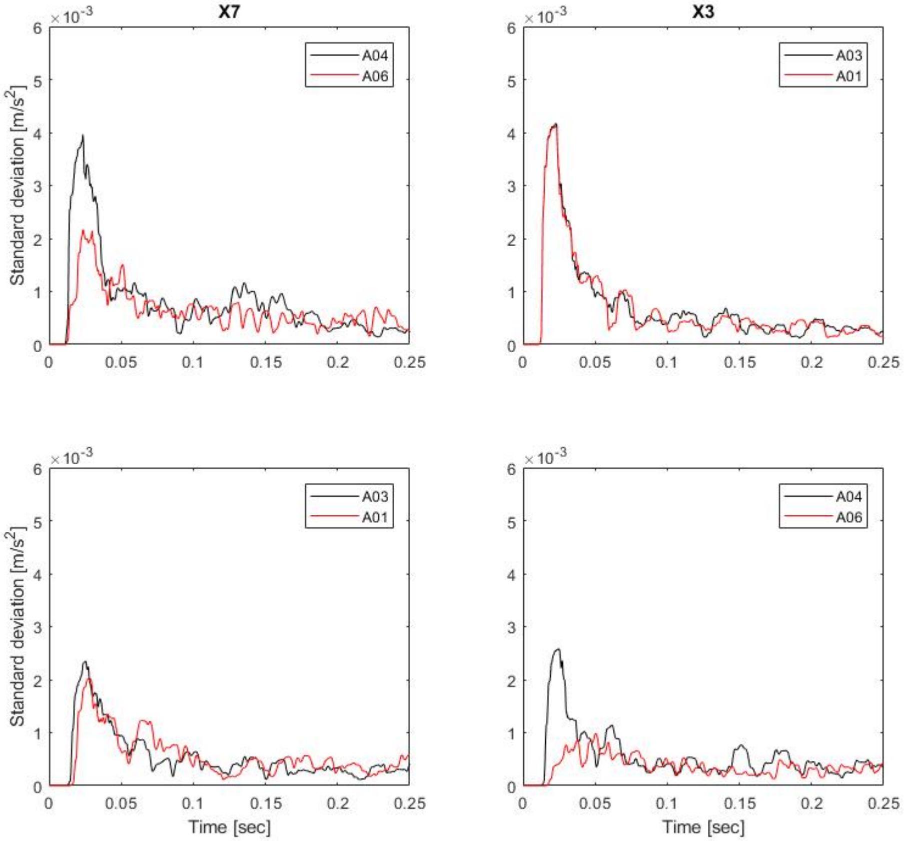


Figure D.2: Moving standard deviation for the linear spring formulation

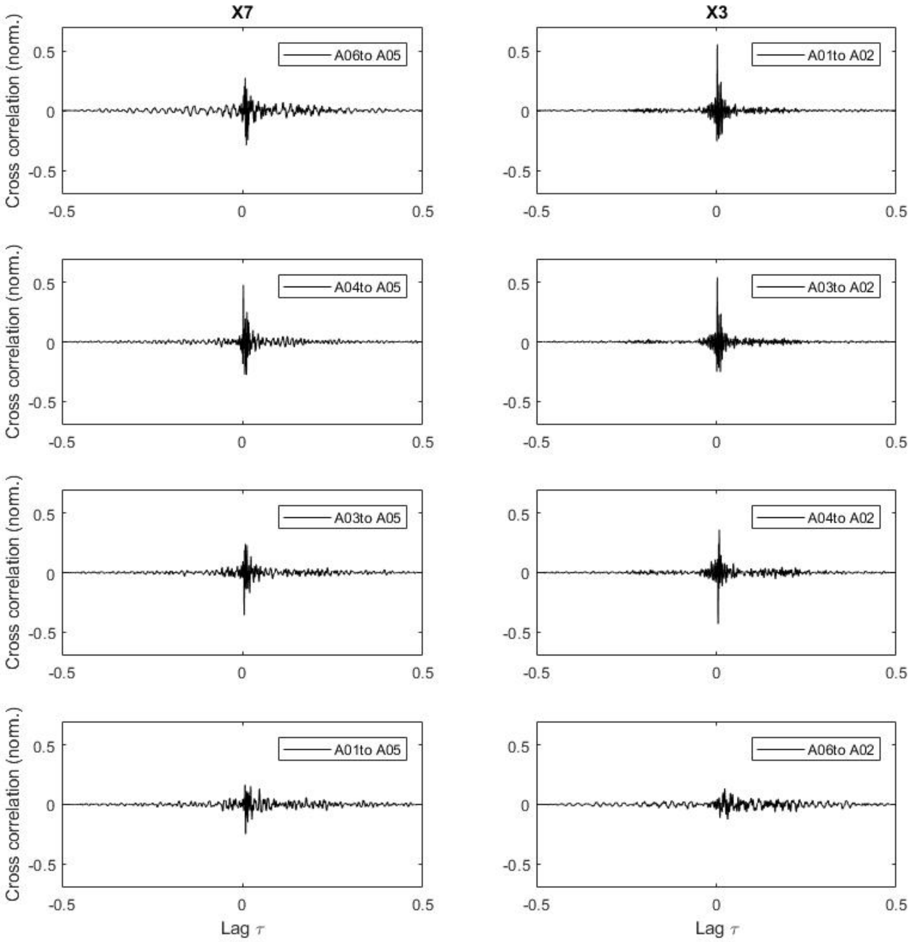


Figure D.3: Cross correlation for the linear spring formulation

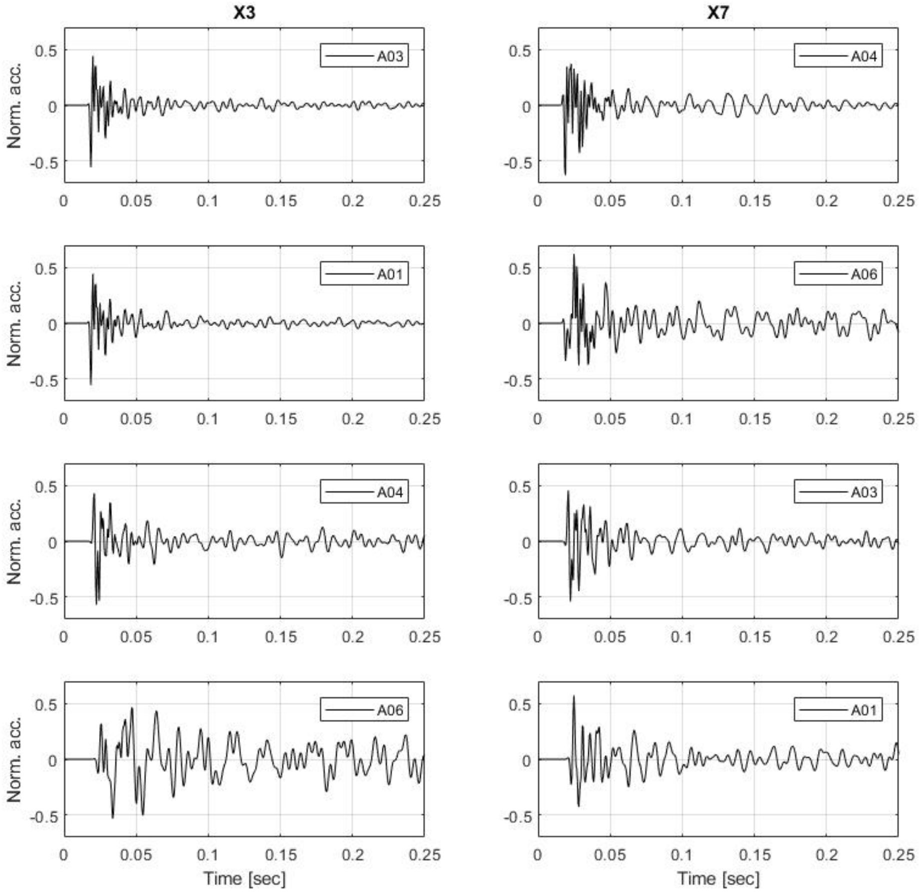


Figure D.4: Normalization for the linear spring formulation

Multi-linear spring formulation (ML)

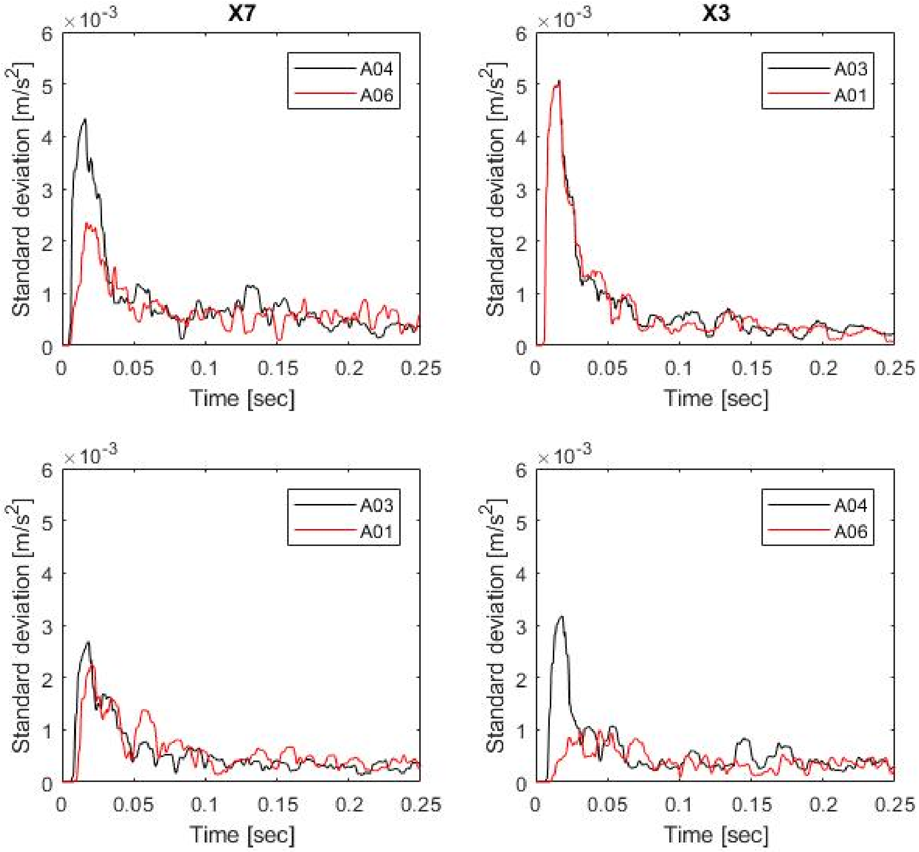


Figure D.5: Moving standard deviation for the multi-linear spring formulation

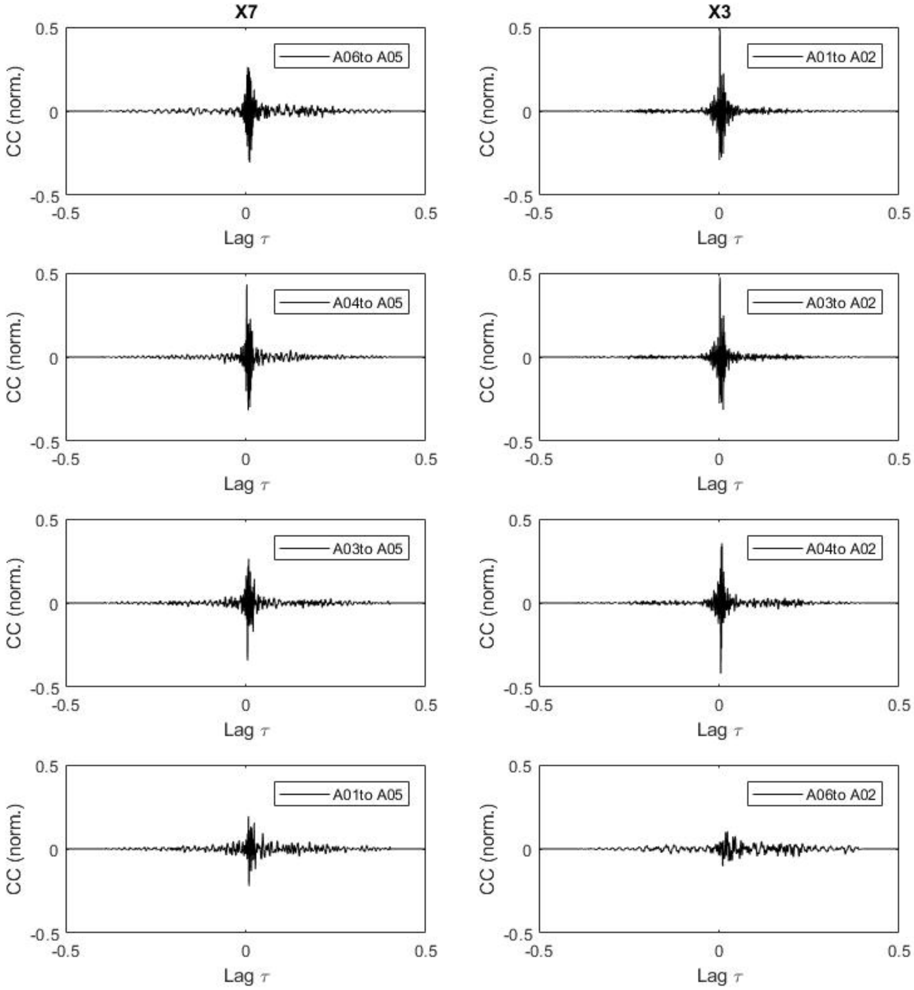


Figure D.6: Cross correlation for the multi-linear spring formulation

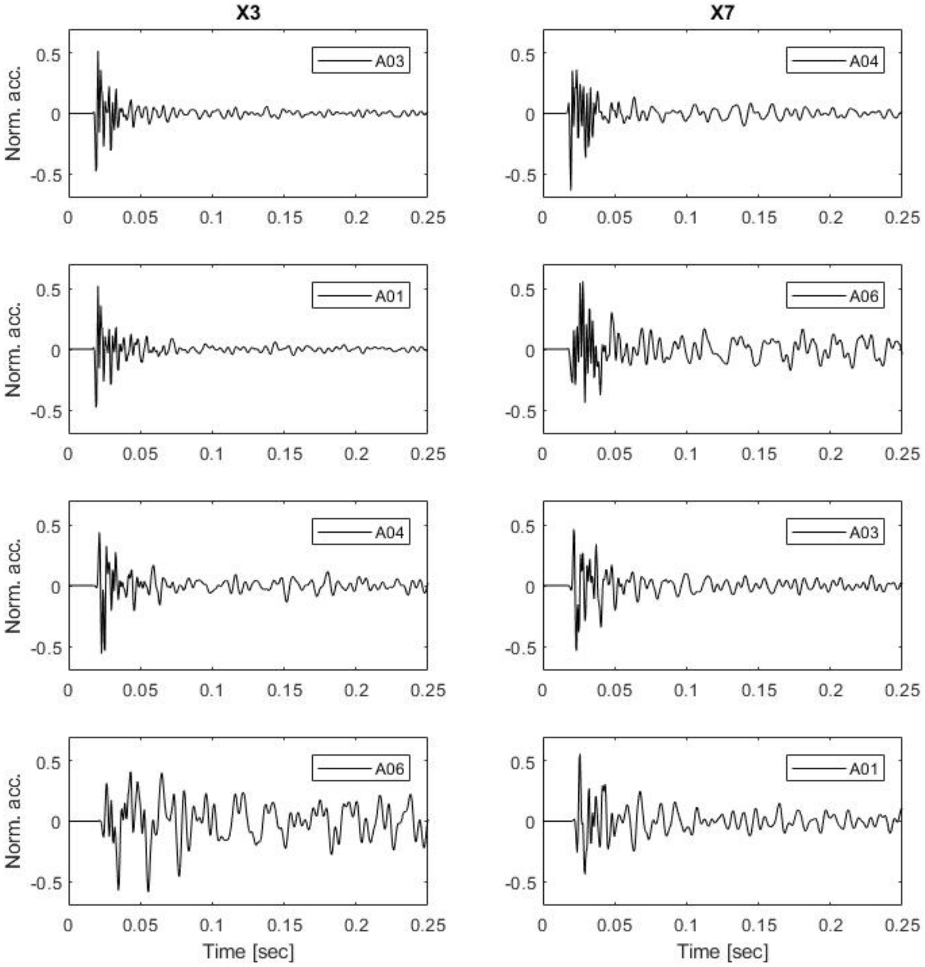


Figure D.7: Normalization for the multi-linear spring formulation

Power function based formulation (PFB)

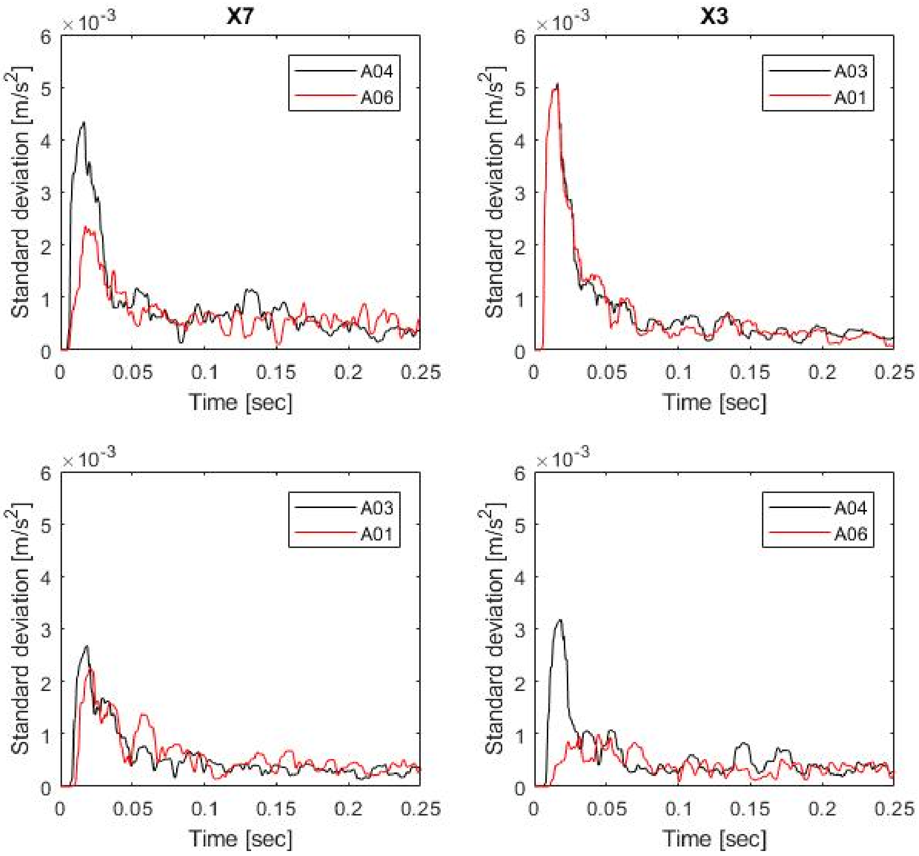


Figure D.8: Moving standard deviation for the power function-based spring formulation

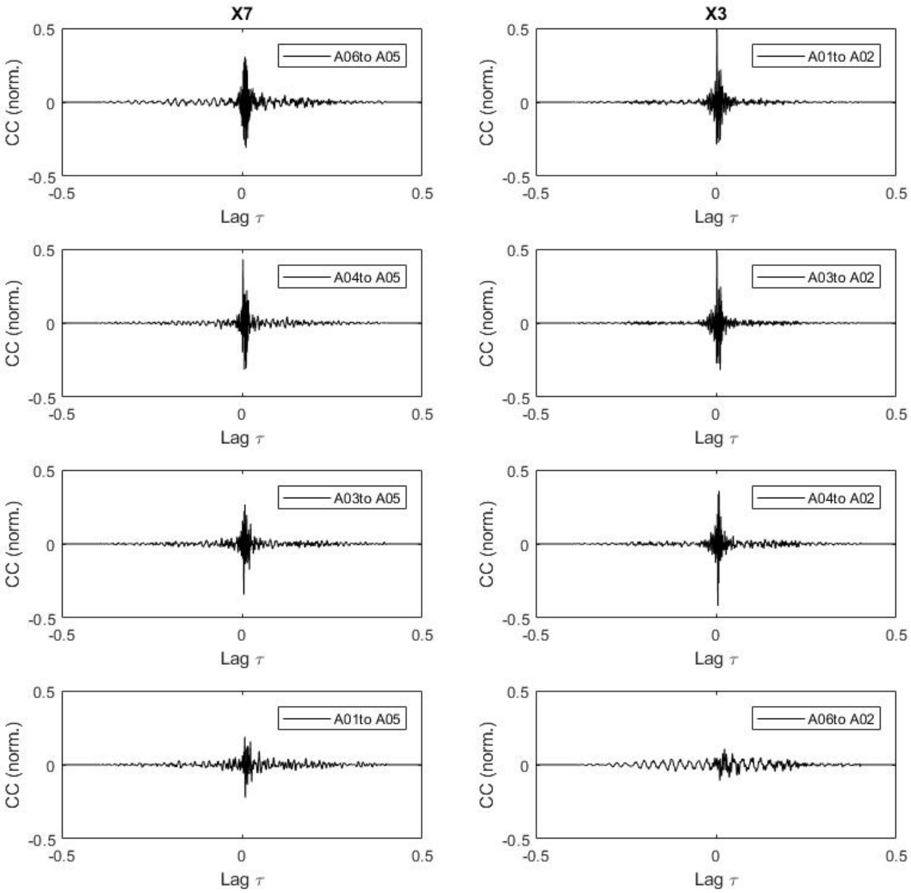


Figure D.9: Cross correlation for the power function-based spring formulation

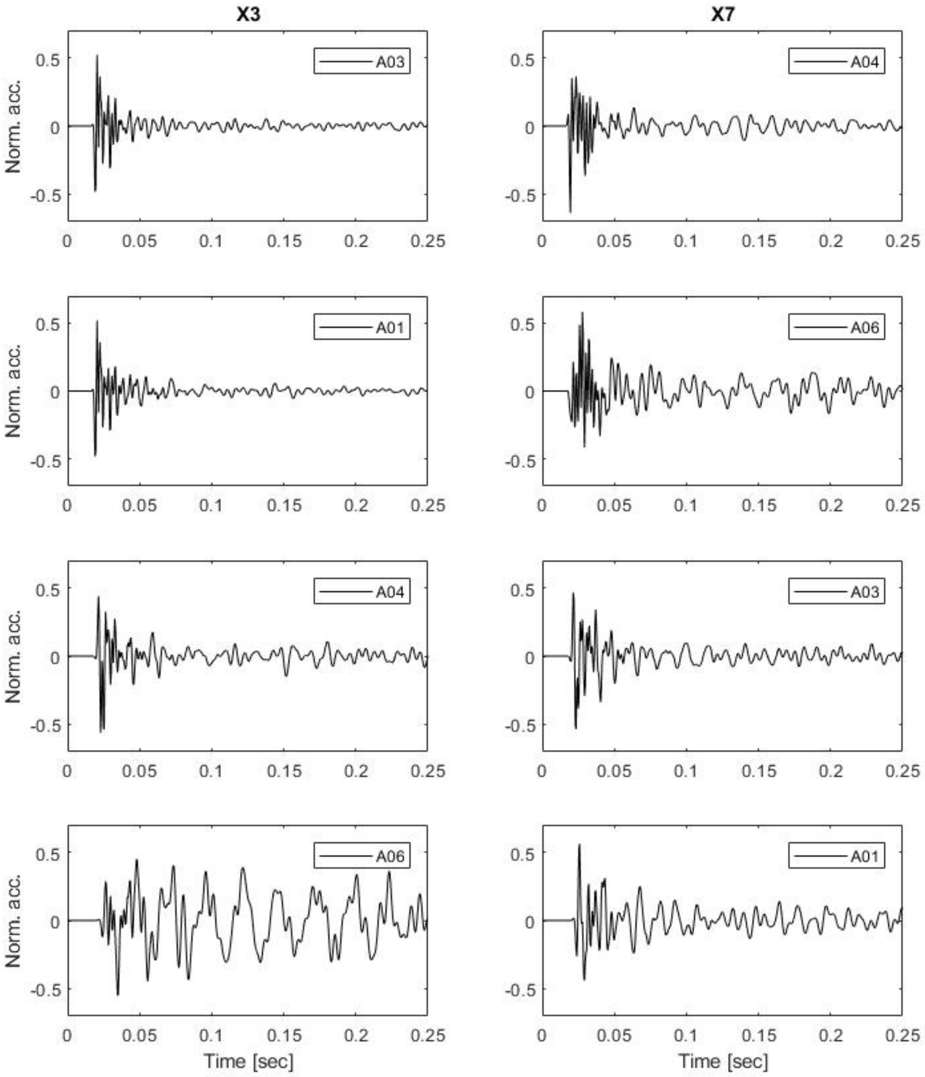


Figure D.10: Normalization for the power function-based spring formulation

Combined quadratic linear formulation (CQL)

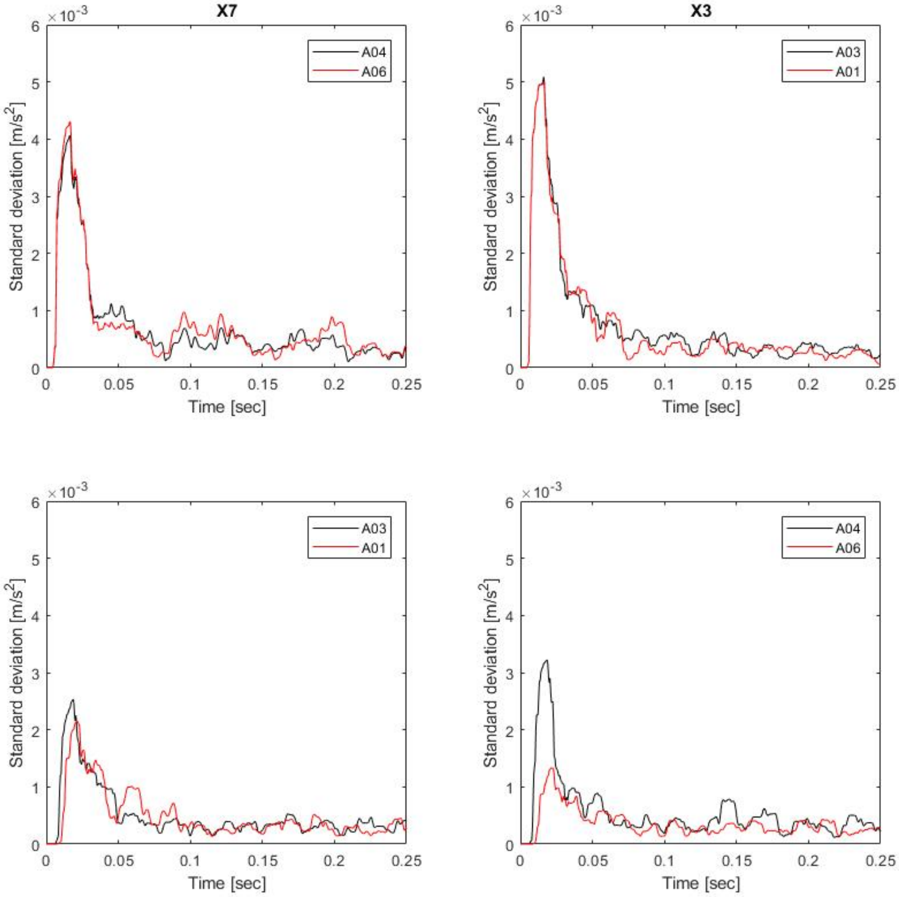


Figure D.11: Moving standard deviation for the combined quatric-linear spring formulation

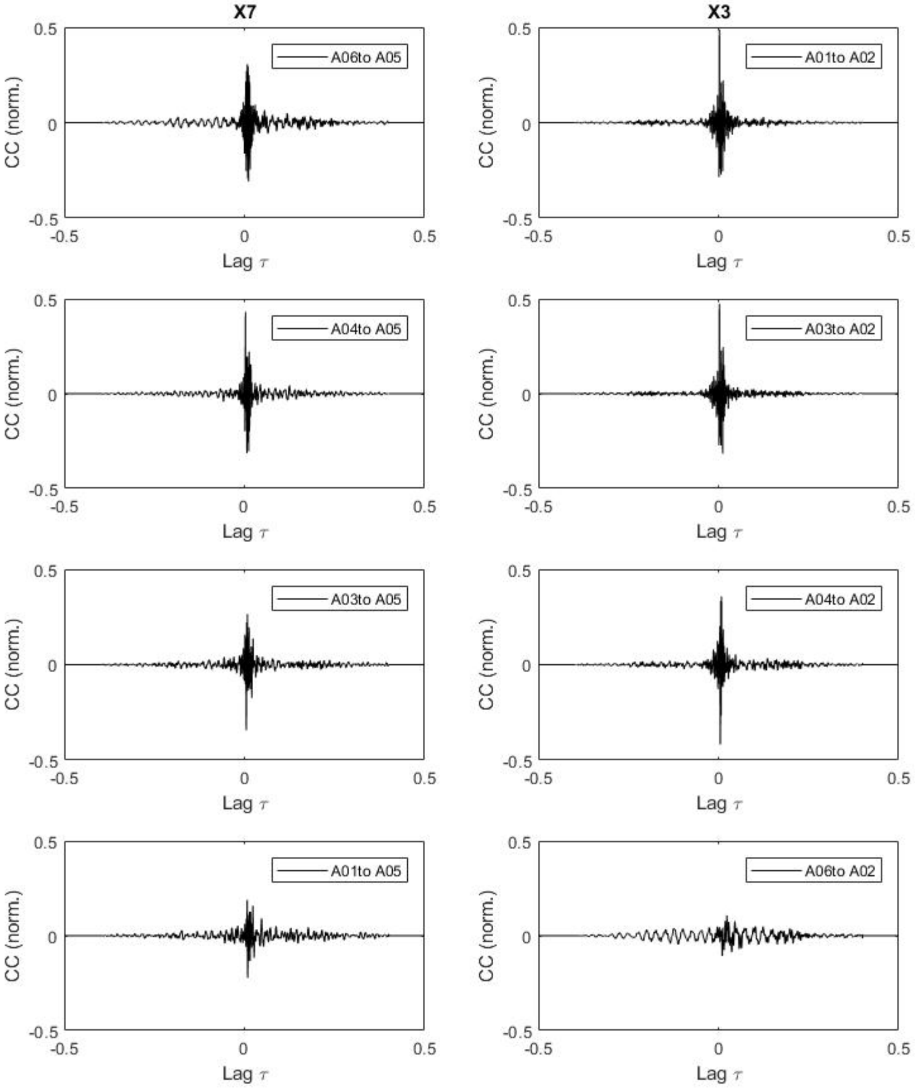


Figure D.12: Cross correlation for the combined quatric-linear spring formulation

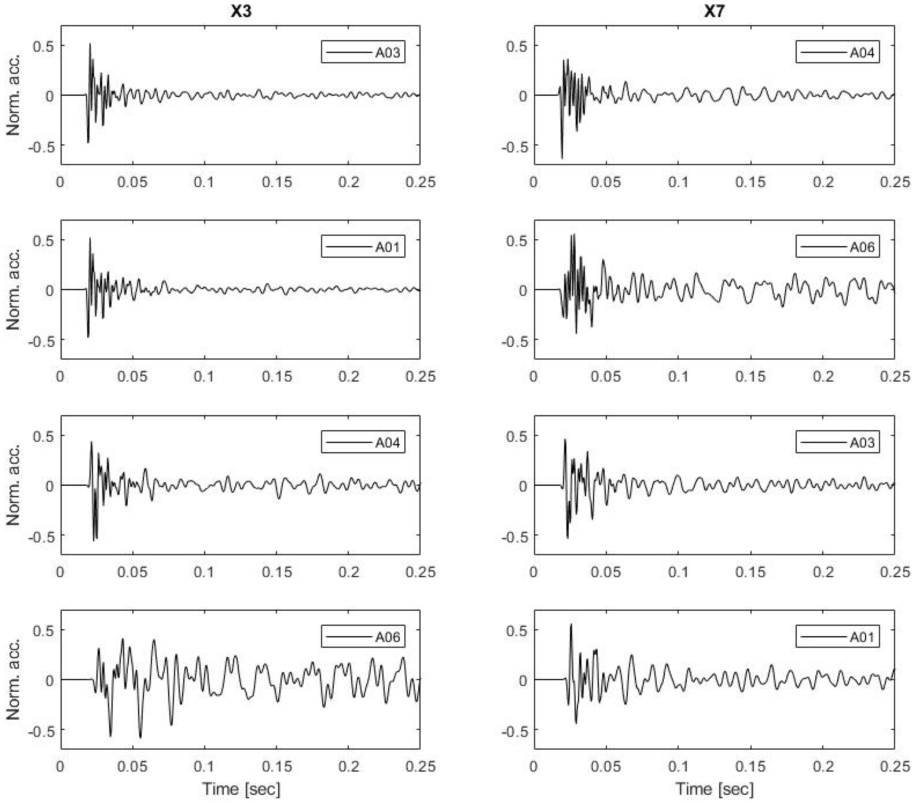


Figure D.13: Normalization for the combined quatric-linear spring formulation