NTNU – Trondheim
Norwegian University of
Science and Technology

Aerodynamic Response of Slender Suspension Bridges

Jens Einar Aaland

Civil and Environmental Engineering (2 year)
Submission date: June 2014
Supervisor: Einar Norleif Strømmen, KT
Co-supervisor: Bjørn Isaksen, Vegdirektoratet

Norwegian University of Science and Technology
Department of Structural Engineering

MASTEROPPGAVE 2014

Konstruksjonsteknikk

for

Jens Einar Aaland

DYNAMISK RESPONS AV LANGE SLANKE HENGEBRUER

Aerodynamic response of slender suspension bridges

I Norge er det for tiden under planlegging og bygging en rekke meget slanke brukonstruksjoner, for eksempel Hardangerbroen som er en klassisk hengebro og Hålogalandsbroen. Begge har hovedspenn på betydelig mer enn 1000 m. De er svært utsatt for den dynamiske lastvirkningen fra vind. Det er også under utredning en kryssing av Sognefjorden som innebærer en bro med spenn på opp til tre kilometer, og i dette tilfellet er det usikkert i hvilken grad man vil være i stand til å oppnå en konstruktiv utførelse med tilfredsstillende aerodynamisk egenskaper. Det har i den forbindelse blitt foreslått å undersøke muligheten for addere dempning til systemet ved hjelp av en eller flere massedempere (Tuned Mass Dampers). Hensikten med denne oppgaven er nettopp å se på mulige utførelser av fjordkryssinger i denne spennvidden, hvor det legges spesiell vekt på å undersøke i hvilken grad massedempere (TMD) kan bidra til å redusere faren for virvelavløsningssvingninger. Arbeidet foreslås lagt opp etter følgende plan:

1. Studenten setter seg inn i teorien for hengebroens virkemåte (se *Strømmen: Structural dynamics, Springer 2013, kapittel 3.4 og 3.5*).
2. Studenten setter seg inn i teorien for virvelavløsningsinduserte svingninger (se *Strømmen: Theory of bridge aerodynamics, Springer 2006, kapittel 6.4*).
3. For en eller flere aktuelle utførelser utførelser (avtales med veileder og Sivilingeniør K. Berntsen i Vegdirektoratet) skal det foretas en utredning med sikte på å kvantifisere faren for virvelavløsningssvingninger. (I den grad tiden tillater det skal disse beregningene utføres i form av et parameterstudium.)
4. For tilfellene som er behandlet under punkt 3 skal det foretas en undersøkelse i hvilken grad en eller flere massedempere kan bidra til å redusere eller helt fjerne problemet med virvelavløsningsinduserte svingninger. Beregningen skal baseres på en mest mulig generell teori (se *Strømmen: Structural dynamics, Springer 2013, kapittel 9.4*) og en løsning i Matlab (eller tilsvarende type program).

Studenten kan selv velge hvilke problemstillinger han ønsker å legge vekt på. Oppgaven skal gjennomføres i samarbeid med Siv.ing. Kristian Berntsen og Dr.ing. Bjørn Isaksen i Vegdirektoratet.

NTNU, 2014-01-14

Einar Strømmen

Preface

This report is written as a result of the work done in the master-thesis Aerodynamic Response of Slender Suspension Bridges, at Department of Structural Engineering at NTNU spring 2014. I have chosen to focus on the 3 last problems given in the problem description. The purpose of the thesis has been to study and learn theory of how to calculate the response from vortex induced vibrations, and the response after the installation of mass dampers. For better understanding, and to be able to use the theory, a several Matlab scripts has been developed or used along the learning path. When getting results, and especially when having problem getting results, the theory is better understood. A final Matlab script is developed, based on what is learned from theory, and using and developing other Matlab scripts. The Matlab script is used on data from the Hardanger Bridge, from which the results are obtained. Thus the purpose of the thesis is not as much the results itself, as the way the results are obtained.

In chapter 1, a detailed summary from the theory study of the tuned mass damper, is given. In chapter 2, some comments regarding the development of the Matlab script is done. Chapter 3 contain the results, among other the results from a parametric study, and analysis of the results. Some comments are made in chapter 4.

This thesis has received founding from Norwegian Public Roads Administration (Statens Vegvesen).

Acknowledgements

Before and during the work of the thesis, I have had the pleasure of having many inspiring and competent people around me. Especially I would like to thank my family who always is there for me, my co- students who have inspired me and helped me, the workers and engineers at the Dalsfjord Bridge who made me more interested in bridges, Norwegian Public Roads Administration which fund this thesis, Dr. engineer Bjørn Isaksen for answering my questions before and during the work of the thesis and for showing sympathy when things did not work as expected, and at last but certainly not least Professor Dr. engineer Einar Strømmen for helping and supporting me through the work and for spreading his good temper to everyone around. Thank you all!

Summary

During the design of long, slender suspension bridges, a lot of aerodynamics has to be taken into account. This thesis focus on one phenomenon; vortex induced vibrations. The purpose of this thesis has been to get insight in the theoretical background of how the response due to vortex induced vibrations is obtained, and how tuned mass dampers can suppress these vibrations. The theory is adopted for a real case, The Hardanger Bridge, to study the effect of tuned mass dampers by the use of Matlab. The focus has been on the vertical modes, knowing that vortex induced vibrations could happen for torsional and sometimes for horizontal modes as well. Since eventually mistakes in the Matlab script is not easily found, and since the results is not compared to the real case or earlier findings for the Hardanger Bridge, the results must be used with care.

The analysis of the Hardanger bridge in this thesis indicate that the maximum displacement response, without the use of tuned mass dampers, is in the order in the order of 0.1 meters. The analysis indicate that one or several tuned mass dampers placed at locations where the eigenmode they are supposed to damp out has its maximum, or is close to its maximum, is an efficient way to reduce the response. Different properties of the tuned mass dampers have been studied, but in general frequency according to Den Hartog, and damping ratios according to Den Hartog or slightly above, seems to give good effect of the tuned mass dampers.

A recommendation for further work is to investigate the consequences of neglecting the response term in the expression of aerodynamic damping, for the cases with tuned mass dampers installed. Another recommendation is to compare the results with other findings, or measurements from similar cases.

Table of Contents

List of Tables

List of Figures

Chapter 1

Theory

1.1 Wind field - mean and fluctuating part

Measurement of wind speed in real life are done in time domain. At a certain point in space, over a certain time window, measurement of the wind speed in one, two or three directions are done. Typically the time window is 10 minutes. The recorded time series in the main flow along- wind direction, could be divided into a constant mean wind part, and a fluctuating zero mean turbulence part, according to equation 1.1, where the mean part are defined in equation 1.2. The mean part is created by the weather system itself, while the fluctuating turbulence part is created by friction from the terrain. In the across-wind directions there are no flow other than the turbulence components in horizontal and vertical direction, v and w respectively. The turbulence components have zero mean value in every directions.

$$U = V + u \tag{1.1}$$

where

$$V = \frac{1}{T}\int_0^T U dt \tag{1.2}$$

1.2 Statistical properties

1.2.1 Stochastic process

A process is called *stochastic* when the outcome at any time or space is random, and each simulation or recording of the process, represent just one of infinitely many possible realizations [Strømmen, 2010]. The fluctuating part of the wind could be considered stochastic, with known or unknown statistical properties. The mean value of the fluctuating part is assumed to always be zero, but the distribution and standard deviation may vary. In most cases the distribution is assumed to be Gaussian.

1.2.2 Sample size

A sample of a stochastic process, may have different statistical properties than the process itself. Often the sample gives a better representation of the process as the sample size increases. The sample size are dependent on the logging frequency and the sample duration. A sample with the same duration, but a very high logging frequency may not represent the process better, as the sample points then get dependent on each other. For wind measurements the process itself, or at least its statistical properties, changes with changing weather conditions. The time window for collecting measurements for one process could not be too big, as the weather conditions could change, but it could not be too little either, as a small sample could give a bad representation of the process. The usual sampling duration is 10 minutes.

1.2.3 Time domain and ensemble statistics

Statistics preformed on a single time series, of e.g. the fluctuating part of the wind in the along wind direction, are called *time domain statistics*. While statistics preformed on extract from many time series, e.g. the mean value of many different recordings, are called *ensemble statistics*. The ensemble statistics could be preformed both on different recordings at the same place at different time, or at different places at the same time.

1.2.4 Stationary process

If the statistical properties, like the mean value and the standard deviation, of the process does not change with time, the process is said to be *stationary* [Strømmen, 2010]. In wind engineering the wind are often assumed stationary, as it is a part of a bigger weather system that does not change much in time. The weather system can last for quite a long time, typically a weather system in Norway lasts for about 3 days ref[strmmen muntlig]. To measure every minute of every weather system in every stations for wind measurements would be extremely demanding, and not either is it necessary. The statistical properties of the weather system are found from the 10 minutes measurement, often recorded in the middle of the weather system passing, when the wind has ”settled”.

1.2.5 Homogenous process

Analogous to a stationary process where the statistical properties does not change with time, the statistical properties of a *homogeneous* process does not change in space [George, 2013]. In wind engineering the wind are often assumed homogeneous, as it is a part of a weather system, by far bigger than the bridge or construction itself. Local terrain may obstruct the wind field, such that its statistical properties change along the span of a bridge. Prior to or during planning of large bridges, wind measurements are often done at several places along where the span are planned. However, the wind measurements are often done only at land-side, because of the difficulties in mounting measure equipment at seaside. To get a good understanding of the local wind field, the terrain may be modeled and wind tests preformed in wind tunnels. This is done for the Hardangerfjord Bridge, where measure equipment also are mounted along the span of the bridge after it is finished. When wind tests are done along the span of a finished bridge, it is possible to see if the wind fields statistical properties is as predicted, and to learn about differences between model tests and the reality.

1.2.6 Phase

The phase of the measured wind are not of interest, since it is only dependent on when the recording are done. If the exact wind-recordings are to be reproduced numerically, the phase for each frequency is needed. Otherwise several different numerical simulation are done with different random phase, to represent the randomness of the wind. Statistics may be preformed on a set of simulation, to find for instance a representative peak value.
How the statistical properties of the weather system are extracted from the recorded time series are shown in section 1.3, and how to produce simulations from the statistical properties are shown in section 1.4.

1.3 From time domain to frequency domain - Auto Spectral Density

Wind measurements are done in time domain. To be able to do a modal response analysis of the construction, the wind load need to be transformed into frequency domain. The eigen-frequencies and eigen-modes of the bridge are assumed easily found by for instances final element methods. The transformation of the wind load are done with Fourier Transformation. The procedure are according to Strømmen [Strømmen, 2010] as follows. A zero mean variable $x(t)$ with length T may be approximated by a sum of harmonic components $X_k(\omega_k, t)$, as shown in equation 1.3.

$$x(t) = \lim_{N\to\infty} \sum_{k=1}^{N} X_k(\omega_k, t) \qquad \text{where} \quad \begin{cases} \omega_k = k \cdot \Delta\omega \\ \Delta\omega = \frac{2\pi}{T} \end{cases} \tag{1.3}$$

where $X_k(\omega_k, t)$ is given by:

$$X_k(\omega_k, t) = c_k \cdot \cos(\omega_k t + \varphi_k) \qquad \text{where} \quad \begin{cases} c_k = \sqrt{a_k^2 + b_k^2} \\ \varphi_k = \arctan \frac{b_k}{a_k} \end{cases} \tag{1.4}$$

Where a_k and b_k is given by

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \frac{2}{T} \int_0^T x(t) \begin{bmatrix} \cos(\omega_k t) \\ \sin(\omega_k t) \end{bmatrix} dt \tag{1.5}$$

The single sided auto spectral density is defined by

$$S_x(\omega_k) = \frac{E[X_k^2]}{\Delta\omega} = \lim_{T\to\infty} \frac{1}{\Delta\omega} \cdot \frac{1}{T} \int_0^T \left[c_k \cos(\omega_k t + \varphi_k)\right]^2 dt \tag{1.6}$$

By introducing $T_k = \frac{2\pi}{\omega_k}$, the period of the harmonic component, and replacing T with $n \cdot T_k$ the following is obtained

$$S_x(\omega_k) \quad \xrightarrow{T\to n\cdot T_k} \quad = \lim_{n\to\infty} \frac{1}{\Delta\omega} \cdot \frac{1}{n \cdot T_k} \cdot n \cdot \int_0^{T_k} \left[c_k \cos(\frac{2\pi}{T_k} \cdot t + \varphi_k)\right]^2 dt = \frac{c_k^2}{2\Delta\omega} \tag{1.7}$$

From which the amplitude c_k can be obtained

$$c_k = \sqrt{2S_x(\omega_k)\Delta\omega} \tag{1.8}$$

Complex format:

$$X_k = d_k \cdot [\cos(\omega_k t + \varphi_k) + i \cdot \sin(\omega_k t + \varphi_k)] \qquad \text{where} \quad \begin{cases} \sqrt{\text{Re}[d_k]^2 + \text{Im}[d_k]^2} = \frac{1}{2}c_k \\ \arctan\left(\frac{\text{Im}[d_k]}{\text{Re}[d_k]}\right) = \varphi_k \end{cases} \tag{1.9}$$

Thus the complex Fourier amplitude d_k, that satisfy equation 1.9, must satisfy $\text{Im}[d_k] = b_k$, $\text{Re}[d_k] = a_k$, and thus

$$d_k = \frac{1}{2}(a_k - i \cdot b_k) \tag{1.10}$$

and its complex conjugate satisfy

$$\begin{bmatrix} e^{i\omega t} \\ e^{-i\omega t} \end{bmatrix} = \begin{bmatrix} \cos(\omega t) + i\sin(\omega t) \\ \cos(\omega t) - i\sin(\omega t) \end{bmatrix} \tag{1.11}$$

For the complex case the displacement $x(t)$ is obtained summing over the entire, both positive and negative, frequency range

$$x(t) = \sum_{-\infty}^{\infty} X_k(\omega_k, t) = \sum_{-\infty}^{\infty} d_k(\omega_k) \cdot e^{i \cdot \omega_k t} \qquad \text{where} \quad \{ \; d_k = \tfrac{1}{2}(a_k - i \cdot b_k) \tag{1.12}$$

Thus

$$X_k(\omega_k, t) = d_k(\omega_k) \cdot e^{i \cdot \omega_k t} = \frac{1}{2}(a_k - i \cdot b_k)[\cos\omega_k t + i \cdot \sin\omega_k t] \tag{1.13}$$

The double sided auto spectra is the variance of the Fourier components divided by $\Delta\omega$:

$$\begin{aligned} S_x(\pm\omega_k) &= \frac{E[X_k^* \cdot X_k]}{\Delta\omega} = \frac{1}{T}\int_0^T \frac{(d_k^* e^{-i\omega_k t})\,(d_k e^{i\omega t})}{\Delta\omega} dt = \frac{d_k^* d_k}{\Delta\omega} \\ &= \frac{1}{4}\frac{(a_k + i \cdot b_k) \cdot (a_k - +i \cdot b_k)}{\Delta\omega} = \frac{c_k^2}{4\Delta\omega} \end{aligned} \tag{1.14}$$

Which is the half of the single-sided spectra. Thus $S_x(\omega) = 2 \cdot S_x(\pm\omega)$.

1.4 From frequency domain to time domain - Time Domain Simulations

If the spectral density and the wind field constants of a process is known, it is possible to simulate the process in time domain. This is done by reversing the process of auto spectral density development. Now the auto spectral density is known, while the time series is unknown. The displacement response x at time instance t is given by equation 1.3, the harmonic components

X_k by equation 1.4 and the amplitude c_k by equation 1.8. Underneath they are repeated, for easier to understand the concept, assuming $N \to \infty$:

$$x(t) = \sum_{k=1}^{N} X_k$$
$$X_k = c_k \cdot \cos(\omega_k t + \varphi_k)$$
$$c_k = \sqrt{2S_x(\omega_k)\Delta\omega}$$

Combining these equations gives

$$x(t) = \sum_{k=1}^{N} \sqrt{2S_x(\omega_k)\Delta\omega} \cdot \cos(\omega_k t + \varphi_k) \tag{1.15}$$

Where the spectral density $S_x(\omega)$ is known from wind recordings Usually the phase angle φ is unknown since it is, as explained in subsection 1.2.6, it is only dependent on when the time series are recorded, and do not give any interesting information about the wind field, and thus it is not usual to store this information. By a time domain simulation, time series with the statistical properties of a given weather system and different random phases could be produced. Thus ensemble statistics could be preformed on the variety of time series inside a single weather system, or at several weather systems relevant for the site which is of interest.

1.5 Stochastic Dynamic Response Calculations

The response spectrum for the displacement response at a given position x_r is given in equation 1.16.

$$S_{r_i}(x_r, \omega) = \frac{\phi_i^2(x_r)}{\tilde{K}_i^2} \cdot \mid \hat{H}_i(\omega) \mid^2 \cdot S_{\tilde{Q}_i}(\omega) \tag{1.16}$$

The variance, which is the integral of the response spectrum over the entire positive frequency range, could be simplified and split into a sum of two integrals, shown in equation 1.17.

$$\begin{aligned} \sigma_{r_i}^2(x_r) &= \frac{\phi_i^2(x_r)}{\tilde{K}_i^2} \cdot \int_0^\infty \mid \hat{H}_i(\omega) \mid^2 \cdot S_{\tilde{Q}_i}(\omega)\mathrm{d}\omega \\ &\approx \frac{\phi_i^2(x_r)}{\tilde{K}_i^2} \cdot \left[\mid \hat{H}_i(0) \mid^2 \cdot \int_0^\infty S_{\tilde{Q}_i}(\omega)\mathrm{d}\omega + S_{\tilde{Q}_i}(\omega_i) \cdot \int_0^\infty \mid \hat{H}_i(\omega) \mid^2 \mathrm{d}\omega \right] \end{aligned} \tag{1.17}$$

Since the structure is at rest for $\omega = 0$, the value of the non-dimensional modal frequency-response-function at $\omega = 0$ can be shown to be equal unity. The first expression inside the brackets in equation 1.17 is therefore the variance of the loading. The second expression is the value of the load response spectrum at the eigenfrequency ω_i, multiplied by the integral of the frequency response function over the entire positive frequency range. Its value is given in [Strømmen, 2010], and shown in equation 1.18.

$$\int_0^\infty \mid \hat{H}_i(\omega)^2 \mid \mathrm{d}\omega = \frac{\pi\omega_i}{4(1-\kappa_{ae_i})\xi_{tot_i}} \tag{1.18}$$

Thus the simplified expression of the variance of the displacement response is given by

$$\sigma_{r_i}^2(x_r) \approx \sigma_{B_i}^2(x_r) + \sigma_{R_i}^2(x_r) = \frac{\phi_i^2(x_r)}{\tilde{K}_i^2} \cdot \left[\sigma_{\tilde{Q}_i}^2(x_r) + \frac{\pi\omega_i S_{\tilde{Q}_i}(\omega_i)}{4(1-\kappa_{ae_i})\xi_{tot_i}}\right] \tag{1.19}$$

The first term in the bracket in equation 1.19 is the variance of the loading, the second is the value of the integrated frequency-response-function according to equation 1.18. Weighted with the expression with the i-th modeshape and modal stiffness, it gives respectively the background, σ_{B_i}, and the resonant part, σ_{R_i}, of the displacement variance.

The development of the displacement response spectra, as well as the variance which is the area under the response spectra curve, is illustrated in figure 1.1, where $\acute{S}_{r_i}(\omega)$, $\acute{\sigma}_{B_i}$ and $\acute{\sigma}_{R_i}$, are the unweighted versions of $S_{r_i}(\omega)$, σ_{B_i} and σ_{R_i} respectively.

As can be seen from equation 1.17 and figure 1.1, the amplitude of the resonant part as compared to the background part depends on the value of the load response spectrum at the eigenfrequency considered, $S_{\tilde{Q}_i}(\omega_i)$. If $S_{\tilde{Q}_i}(\omega_i)$ happens to be equal unity, there will be no amplification of the squared frequency response function. If $S_{\tilde{Q}_i}(\omega_i)$ is not equal unity, the amplification will displace the resonant curve parallel, without change the shape of it. Thus, the resonant curve is mainly determined of the frequency response function which has a distinct peak close to and at the eigenfrequency ω_i, i.e. it is so called narrow banded. In contrast the background part of the variance are more widely spread in the frequency domain, and is therefore not narrow banded.

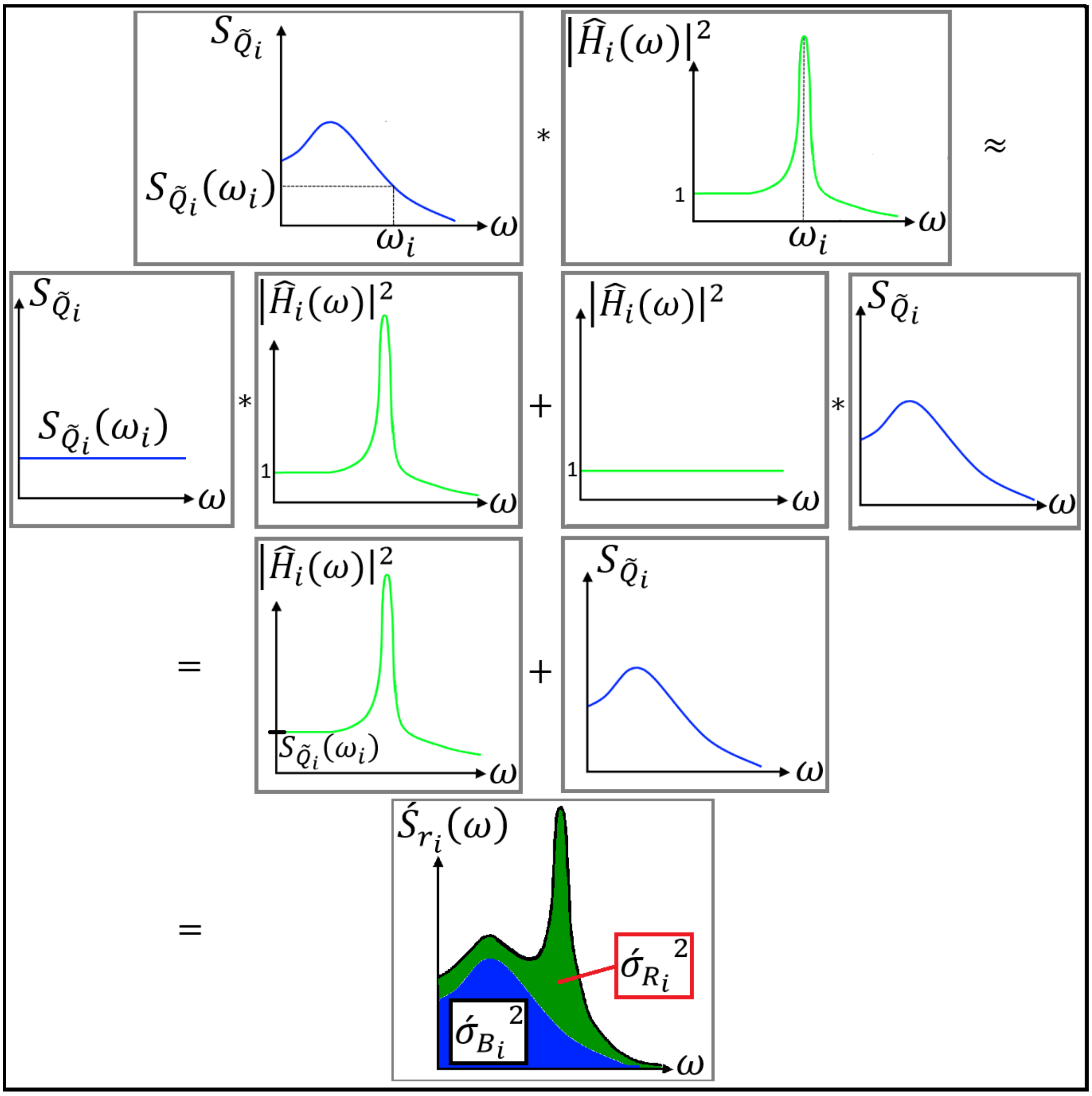

Figure 1.1: Resonant and background part of the variance

If a process of displacement fluctuations in time domain are close to harmonic it contain only a single frequency or a few very close frequencies, and is therefore narrow banded. Vortex shedding response are usually considered narrow banded, thus it is sufficient to only consider the resonant part of the variance in equation 1.19 [Strømmen, 2010].

1.6 Tuned Mass Damper

1.6.1 Single degree of freedom

The development of the tuned mass damper equations used here are done by Einar Strømmen [Strømmen, 2013]. Since learning and using the theory of tuned mass damper has been a major part of this thesis, it is found useful to be as complete as possible in summarizing this theory.

A tuned mass damper is shortly called TMD. A principle sketch of a TMD attached to the bridge girder is shown to the right in figure 1.2. To understand the theory of finding the response of the bridge girder and the tuned mass damper or dampers, it is useful to first find the response of a single degree of freedom system with a single TMD, as sketched to the left in figure 1.2.

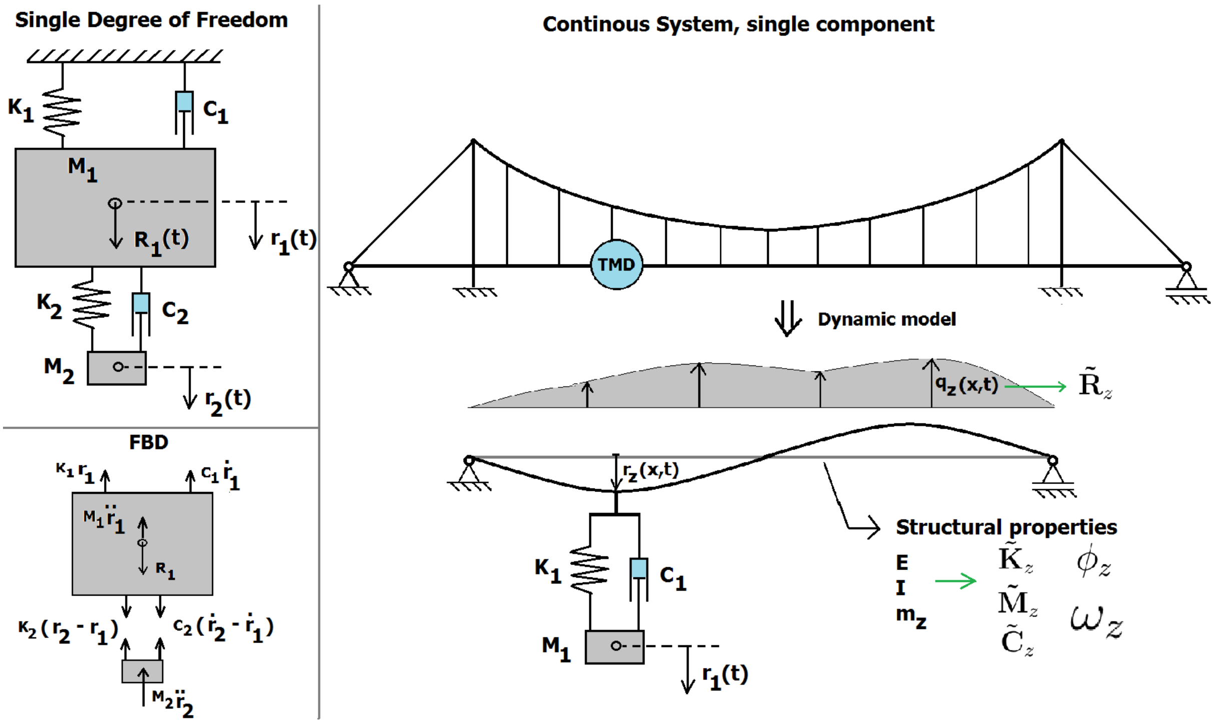

Figure 1.2: Tuned mass damper on a single degree of freedom and continuous single component system

The equilibrium condition of the two bodies illustrated to the left in figure 1.2 is:

$$\begin{aligned} M_1\ddot{r}_1 + C_1\dot{r}_1 - C_2(\dot{r}_2 - \dot{r}_1) + K_1 r_1 - K_2(r_2 - r_1) - R_1 &= 0 \\ M_2\ddot{r}_2 + C_2(\dot{r}_2 - \dot{r}_1) + K_2(r_2 - r_1) &= 0 \end{aligned}$$

where the indexes 1 refer to the main system, i.e. the bridge, while the indexes 2 refer to the mass damper. The equilibrium conditions could be written in matrix form

$$\begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \ddot{r}_2 \end{bmatrix} + \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 \end{bmatrix} \begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} + \begin{bmatrix} K_1 + K_2 & -K_2 \\ -K_2 & K_2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{1.21}$$

If we name the individual matrixes in the following way

$\begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} = \mathbf{M}_0 \quad , \quad \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 \end{bmatrix} = \mathbf{C_0} \quad and \quad \begin{bmatrix} K_1 + K_2 & -K_2 \\ -K_2 & K_2 \end{bmatrix} = \mathbf{K_0} \quad and$

$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \mathbf{r_0} \quad and \quad \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = \mathbf{R_0}$

Equation 1.21 may than be written in a compact format

$$\mathbf{M_0}\ddot{\mathbf{r}}_\mathbf{0} + \mathbf{C_0}\dot{\mathbf{r}}_\mathbf{0} + \mathbf{K_0}\mathbf{r_0} = \mathbf{R_0} \tag{1.22}$$

Since $r_2 = r_1 + \Delta r$, it is convenient to introduce $r_1 = r$ and $\Delta r = r_2 - r$, i.e.

$$\mathbf{r_0} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} r \\ r + \Delta r \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} r \\ \Delta r \end{bmatrix} = \mathbf{\Psi r} \tag{1.23}$$

where $\mathbf{\Psi} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $\mathbf{r} = \begin{bmatrix} r \\ \Delta r \end{bmatrix}$

By introducing $\mathbf{r_0} = \mathbf{\Psi r}$ into equation 1.22 and premultiplying by $\mathbf{\Psi}^T$ a equation of motion with diagonal stiffness and damping matrices are obtained

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{R} \tag{1.24}$$

where

$$\mathbf{M} = \mathbf{\Psi}^T\mathbf{M_0}\mathbf{\Psi} = \begin{bmatrix} M_1 + M_2 & M_2 \\ M_2 & M_2 \end{bmatrix} \quad , \quad \mathbf{K} = \mathbf{\Psi}^T\mathbf{K_0}\mathbf{\Psi} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{\Psi}^T\mathbf{C_0}\mathbf{\Psi} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \quad and \quad \mathbf{R} = \mathbf{\Psi}^T\mathbf{R_0}\mathbf{\Psi} = \begin{bmatrix} R_1(t) \\ 0 \end{bmatrix}$$

A diagonal stiffness matrix may easily be inverted, and thus in the eigenvalue problem may easily be solved. The eigenvalue problem is obtained by setting $\mathbf{C} = \mathbf{0}$, $\mathbf{R} = \mathbf{0}$ and $\mathbf{r} = \mathbf{\Phi}e^{i\omega t}$, where $\mathbf{\Phi} = [\phi_1 \quad \phi_2]^T$, and premultiply by $\mathbf{K}^{-1}$, obtaining

$$\left(\mathbf{I} - \omega^2\mathbf{K}^{-1}\mathbf{M}\right)\mathbf{\Phi} = \mathbf{0} \tag{1.25}$$

Since $\mathbf{K}^{-1} = \begin{bmatrix} \frac{1}{K_1} & 0 \\ 0 & \frac{1}{K_2} \end{bmatrix}$ the eigenvalue problem is

$$\begin{bmatrix} 1 - \omega^2 \frac{M_1+M_2}{K_1} & -\omega^2 \frac{M_2}{K_1} \\ -\omega^2 \frac{M_2}{K_2} & 1 - \omega^2 \frac{M_2}{K_2} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \mathbf{0} \tag{1.26}$$

which is fulfilled if, and only if

$$det\left(\begin{bmatrix} 1 - \omega^2 \frac{M_1+M_2}{K_1} & -\omega^2 \frac{M_2}{K_1} \\ -\omega^2 \frac{M_2}{K_2} & 1 - \omega^2 \frac{M_2}{K_2} \end{bmatrix}\right) = 0 \tag{1.27}$$

i.e.

$$\left(1-\omega^2\frac{M_1+M_2}{K_1}\right)\cdot\left(1-\omega^2\frac{M_2}{K_2}\right)-\left(-\omega^2\frac{M_2}{K_2}\right)\cdot\left(-\omega^2\frac{M_2}{K_1}\right)=0$$

$$\omega^4\left[\frac{M_1+M_2}{K_1}\frac{M_2}{K_2}-\frac{M_2}{K_2}\frac{M_2}{K_1}\right]-\omega^2\left[\frac{M_1+M_2}{K_1}+\frac{M_2}{K_2}\right]+1=0 \tag{1.28}$$

$$\omega^4\left[\frac{M_1}{K_1}\frac{M_2}{K_2}\right]-\omega^2\left[\frac{M_1+M_2}{K_1}+\frac{M_2}{K_2}\right]+1=0$$

from where the eigenfrequencies may be obtained, by the use of the quadratic formula, often called abc-formula

$$\omega^2=\frac{\frac{M_1+M_2}{K_1}+\frac{M_2}{K_2}\pm\sqrt{\left(\frac{M_1+M_2}{K_1}+\frac{M_2}{K_2}\right)^2-4\frac{M_1}{K_1}\frac{M_2}{K_2}}}{2\frac{M_1}{K_1}\frac{M_2}{K_2}} \tag{1.29}$$

The mass of a tuned mass damper is small as compared to the mass of the main system, the moving part of the bridge. For this reason, approximate values of the eigenfrequencies can be obtained by neglecting the additional mass of the tuned mass damper in the total mass, in equation 1.29

$$\omega^2\approx\frac{\frac{M_1}{K_1}+\frac{M_2}{K_2}\pm\left(\frac{M_1}{K_1}-\frac{M_2}{K_2}\right)}{2\frac{M_1}{K_1}\frac{M_2}{K_2}}\Rightarrow\begin{cases}\omega_1\approx\sqrt{\frac{K_1}{M_1}}\\ \omega_2\approx\sqrt{\frac{K_2}{M_2}}\end{cases} \tag{1.30}$$

From equation 1.24 the frequency response can be obtained. The first step is to pre-multiplicate the equation by $\mathbf{K}^{-1}$, which is diagonal, and taking the Fourier transform of the entire equation. Taking the Fourier transform is to let the displacement and load vectors be a sum of products of Fourier coefficient vectors, $\mathbf{a}_r(\omega)$ and $\mathbf{a}_R(\omega)$ respectively, and a complex exponential, i.e.

$$\mathbf{r}(t)=\sum_\omega \mathbf{a}_r(\omega)\cdot e^{i\omega t}\quad and\quad \mathbf{R}=\sum_\omega \mathbf{a}_R(\omega)\cdot e^{i\omega t} \tag{1.31}$$

where

$$\begin{aligned}\boldsymbol{a}_r(\omega)&=[a_r\quad a_{\Delta r}]^T\\ \boldsymbol{a}_R(\omega)&=[a_{R_1}\quad 0]^T\end{aligned} \tag{1.32}$$

The velocity and acceleration is simply

$$\begin{aligned}\dot{\boldsymbol{r}}(t)&=\sum_\omega \mathbf{a}_r(\omega)\cdot e^{i\omega t}\cdot(i\omega)\\ \ddot{\boldsymbol{r}}(t)&=\sum_\omega \mathbf{a}_r(\omega)\cdot e^{i\omega t}\cdot(-\omega^2)\end{aligned} \tag{1.33}$$

respectively.

Thus equation 1.72 turns into

$$\left[(-\omega^2)\mathbf{K}^{-1}\mathbf{M} + (i\omega)\mathbf{K}^{-1}\mathbf{C} + \mathbf{K}^{-1}\mathbf{K}\right]\mathbf{a}_r(\omega)\cdot e^{i\omega t} = \mathbf{a}_R(\omega)\cdot e^{i\omega t} \tag{1.34}$$

The complex exponential part of equation 1.34 $\mathbf{K}^{-1}\mathbf{K}$ is equal the 2 by 2 identitymatrix $\mathbf{I}$. Thus

$$\mathbf{a}_r(\omega) = \frac{1}{-\omega^2\mathbf{K}^{-1}\mathbf{M} + i\omega\mathbf{K}^{-1} + \mathbf{I}}\begin{bmatrix} a_{R_1}/K_1 \\ 0 \end{bmatrix} = \hat{\mathbf{H}}(\omega)\cdot\begin{bmatrix} a_{R_1}/K_1 \\ 0 \end{bmatrix} \tag{1.35}$$

Where $\hat{\mathbf{H}}$ is the frequency response function.

$$\begin{aligned}
\hat{\mathbf{H}}^{-1} &= \mathbf{I} + i\omega\mathbf{K}^{-1} - \omega^2\mathbf{K}^{-1}\mathbf{M} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + i\omega\begin{bmatrix} 1/K_1 & 0 \\ 0 & 1/K_2 \end{bmatrix}\cdot\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} - \omega^2\begin{bmatrix} 1/K_1 & 0 \\ 0 & 1/K_2 \end{bmatrix}\cdot\begin{bmatrix} M_1+M_2 & M_2 \\ M_2 & M_2 \end{bmatrix} \\
&= \begin{bmatrix} 1 + i\omega\frac{C_1}{K_1} - \omega^2\frac{M_1+M_2}{K_1} & 0 + i\omega\cdot 0 - \omega^2\frac{M_2}{K_1} \\ 0 + i\omega\cdot 0 - \omega^2\frac{M_2}{K_2} & 1 + i\omega\frac{C_2}{K_2} - \omega^2\frac{M_2}{K_2} \end{bmatrix} \quad , using \quad \begin{cases} \omega_1 \approx \sqrt{\frac{K_1}{M_1}} \\ \omega_2 \approx \sqrt{\frac{K_2}{M_2}} \end{cases} \\
&= \begin{bmatrix} 1 + 2\xi_1 i\frac{\omega}{\omega_1} - (1+\frac{M_2}{M_1})(\frac{\omega}{\omega_1})^2 & -\frac{M_2}{M_1}(\frac{\omega}{\omega_1})^2 \\ -(\frac{\omega}{\omega_2})^2 & 1 + 2\xi_2 i\frac{\omega}{\omega_2} - (\frac{\omega}{\omega_2})^2 \end{bmatrix} = \begin{bmatrix} D_1 & -\mu(\frac{\omega}{\omega_1})^2 \\ -(\frac{\omega}{\omega_2})^2 & D_2 \end{bmatrix}
\end{aligned} \tag{1.36}$$

where

$$\begin{aligned}
D_1(\omega) &= 1 - (1+\mu)\left(\frac{\omega}{\omega_1}\right)^2 + 2i\xi_1\frac{\omega}{\omega_1} \\
D_2(\omega) &= 1 - \left(\frac{\omega}{\omega_2}\right)^2 + 2i\xi_2\frac{\omega}{\omega_2} \\
\mu &= \frac{M_2}{M_1} \\
\xi_1 &= \frac{C_1}{2M_1\omega_1} \\
\xi_2 &= \frac{C_2}{2M_2\omega_2}
\end{aligned} \tag{1.37}$$

Since a simply expression for the inverse of the frequency response function, $\hat{\mathbf{H}}^{-1}$, is given in equation 1.36, a simplier expression for the frequency response function, $\hat{\mathbf{H}}$ itself can be found by inverting the expression in equation 1.36.

Thus

$$\hat{\mathbf{H}} = \frac{1}{Det(\hat{\mathbf{H}}^{-1})}\begin{bmatrix} D_2 & -(-\mu(\frac{\omega}{\omega_1})^2) \\ -(-(\frac{\omega}{\omega_2})^2) & D_1 \end{bmatrix} = \frac{1}{D_1D_2 - \mu\left(\frac{\omega}{\omega_1}\right)^2\left(\frac{\omega}{\omega_2}\right)^2}\begin{bmatrix} D_2 & \mu(\frac{\omega}{\omega_1})^2 \\ (\frac{\omega}{\omega_2})^2 & D_1 \end{bmatrix} \tag{1.38}$$

Another, frequent used way to write the frequency response function, can be obtained by some renaming and rewriting of the expression in equation 1.38.

Expanding the determinant expression in equation 1.38 gives

$$
\begin{aligned}
Det(\hat{\mathbf{H}}^{-1}) &= D_1 D_2 - \mu \left(\frac{\omega}{\omega_1}\right)^2 \left(\frac{\omega}{\omega_2}\right)^2 \\
&= \left[1 - (1+\mu)\left(\frac{\omega}{\omega_1}\right)^2 + 2i\xi_1 \frac{\omega}{\omega_1}\right] \cdot \left[1 - \left(\frac{\omega}{\omega_2}\right)^2 + 2i\xi_2 \frac{\omega}{\omega_2}\right] - \mu \left(\frac{\omega}{\omega_1}\right)^2 \left(\frac{\omega}{\omega_2}\right)^2 \\
&= 1 + 2i\xi_1 \frac{\omega}{\omega_1} + 2i\xi_2 \frac{\omega}{\omega_2} - \left(\frac{\omega}{\omega_2}\right)^2 - (1+\mu)\left(\frac{\omega}{\omega_1}\right)^2 - 4\xi_1\xi_2 \frac{\omega}{\omega_1}\frac{\omega}{\omega_2} \\
&- (1+\mu)2i\xi_2 \frac{\omega}{\omega_2}\left(\frac{\omega}{\omega_1}\right)^2 - 2i\xi_1 \frac{\omega}{\omega_1}\left(\frac{\omega}{\omega_2}\right)^2 + (1+\mu)\left(\frac{\omega}{\omega_1}\right)^2\left(\frac{\omega}{\omega_2}\right)^2 - \mu\left(\frac{\omega}{\omega_1}\right)^2\left(\frac{\omega}{\omega_2}\right)^2 \\
&= 1 + 2\left[\xi_1 + \frac{\omega_1}{\omega_2}\xi_2\right]\left(i\frac{\omega}{\omega_1}\right) + \left[1 + \mu + \left(\frac{\omega_1}{\omega_2}\right)^2 + 4\frac{\omega_1}{\omega_2}\xi_1\xi_2\right]\left(i\frac{\omega}{\omega_1}\right)^2 \\
&+ 2\frac{\omega_1}{\omega_2}\left[\frac{\omega_1}{\omega_2}\xi_1 + (1+\mu)\xi_2\right]\left(i\frac{\omega}{\omega_1}\right)^3 + \left(\frac{\omega_1}{\omega_2}\right)^2\left(i\frac{\omega}{\omega_1}\right)^4
\end{aligned}
\tag{1.39}
$$

The expression could be simplified by collecting terms in new variables as follows:

$$
\begin{aligned}
\hat{\omega} &= \left(\frac{\omega}{\omega_1}\right) \\
\alpha &= \left(\frac{\omega_1}{\omega_2}\right) \\
a_1 &= 2(\xi_1 + \alpha\xi_2) \\
a_2 &= 1 + \mu + \alpha^2 + 4\alpha\xi_1\xi_2 \\
a_3 &= 2\alpha[\alpha\xi_1 + (1+\mu)\xi_2] \\
a_4 &= \alpha^2
\end{aligned}
\tag{1.40}
$$

Thus

$$
Det(\hat{\mathbf{H}}^{-1}) = 1 + a_1(i\hat{\omega}) + a_2(i\hat{\omega})^2 + a_3(i\hat{\omega})^3 + a_4(i\hat{\omega})^4 \tag{1.41}
$$

In the same way expressions for the matrix could be developed:

$$
\begin{aligned}
D_2 &= 1 - \left(\frac{\omega}{\omega_2}\right)^2 + 2i\xi_2\frac{\omega}{\omega_2} \\
&= 1 + 2\xi_1\frac{\omega_1}{\omega_2}i\frac{\omega}{\omega_1} + \left(\frac{\omega_1}{\omega_2}\right)^2\left(-\left(\frac{\omega}{\omega_1}\right)^2\right) \\
&= 1 + 2\xi_2\alpha(i\hat{\omega}) + \alpha^2(i\hat{\omega})^2 \\
D_1 &= 1 - (1+\mu)\left(\frac{\omega}{\omega_1}\right)^2 + 2i\xi_1\frac{\omega}{\omega_1} \\
&= 1 + 2\xi_1(i\hat{\omega}) + (1+\mu)(i\hat{\omega})^2 \\
\left(\frac{\omega}{\omega_2}\right)^2 &= -\left(\frac{\omega_1}{\omega_2}\right)^2(i\hat{\omega})^2 \\
\mu\left(\frac{\omega}{\omega_1}\right)^2 &= -\mu(i\hat{\omega})^2
\end{aligned}
\tag{1.42}
$$

Collecting terms in new variables as follows:

$$
\begin{aligned}
b_1 &= 2\xi_2\alpha \\
b_2 &= \alpha^2 \\
c_1 &= 2\xi_1 \\
c_2 &= (1+\mu) \\
d_2 &= -\left(\frac{\omega_1}{\omega_2}\right)^2 \\
e_2 &= -\mu
\end{aligned}
\tag{1.43}
$$

Equation 1.42 simplifies to

$$
\begin{aligned}
D_2 &= 1 + b_1(i\hat{\omega}) + b_2(i\hat{\omega})^2 \\
D_1 &= 1 + c_1(i\hat{\omega}) + c_2(i\hat{\omega})^2 \\
\left(\frac{\omega}{\omega_2}\right)^2 &= d_2(i\hat{\omega})^2 \\
\mu\left(\frac{\omega}{\omega_1}\right)^2 &= e_2(i\hat{\omega})^2
\end{aligned}
\tag{1.44}
$$

Thus

$$
\hat{\mathbf{H}}(\hat{\omega}) = \frac{1}{Det(\hat{\mathbf{H}}^{-1})}\begin{bmatrix} D_2 & \mu(\frac{\omega}{\omega_1})^2 \\ (\frac{\omega}{\omega_2})^2 & D_1 \end{bmatrix} = \begin{bmatrix} \hat{H}_{11}(\hat{\omega}) & \hat{H}_{12}(\hat{\omega}) \\ \hat{H}_{21}(\hat{\omega}) & \hat{H}_{22}(\hat{\omega}) \end{bmatrix} \tag{1.45}
$$

where

$$\hat{H}_{11}(\hat{\omega}) = \frac{1 + b_1(i\hat{\omega}) + b_2(i\hat{\omega})^2}{1 + a_1(i\hat{\omega}) + a_2(i\hat{\omega})^2 + a_3(i\hat{\omega})^3 + a_4(i\hat{\omega})^4}$$
$$\hat{H}_{22}(\hat{\omega}) = \frac{1 + c_1(i\hat{\omega}) + c_2(i\hat{\omega})^2}{1 + a_1(i\hat{\omega}) + a_2(i\hat{\omega})^2 + a_3(i\hat{\omega})^3 + a_4(i\hat{\omega})^4}$$
$$\hat{H}_{21}(\hat{\omega}) = \frac{d_2(i\hat{\omega})^2}{1 + a_1(i\hat{\omega}) + a_2(i\hat{\omega})^2 + a_3(i\hat{\omega})^3 + a_4(i\hat{\omega})^4}$$
$$\hat{H}_{12}(\hat{\omega}) = \frac{e_2(i\hat{\omega})^2}{1 + a_1(i\hat{\omega}) + a_2(i\hat{\omega})^2 + a_3(i\hat{\omega})^3 + a_4(i\hat{\omega})^4} \quad (1.46)$$

1.6.2 Continuous system

In section 1.6.1 basic expressions for frequency response function and auto spectral density for a system with a single tuned mass damper was developed for a main system of a single point mass. Since a bridge is not a single point mass, but has distributed mass, as well as distributed stiffness, damping and forces, the case might be different. Also it might be necessary to add several TMDs to provide damping of several modes.

General equations for multi mode, multi component

In section 1.6.1 the first step in developing the equations was to draw a free body diagram and write the equation of motion for each of the two bodies. A way to look at a continuous system, is that it is a collection of infinitely many point masses, and thus the equation system contains infinitely many equation. Approximate solutions of different quality can be obtained by using different finite number of point masses. Another far more elegant way to solve the continuous problem, is to first find the mode shapes from an eigenvalue solution, and than use the eigenmodes in a modal approach.

The purpose of the tuned mass dampers is to artificially damp out the motions of one or several eigenmodes. Since the shape of the eigenmodes are known, the relative displacement in original coordinates of the bridge along its span are known for each mode. By the use of d'Alembert's extended principle of virtual work the modal equation of motion can be found.

The d'Alembert's principle is to give both the bridge, which is assumed to have a beam type of behavior, and the tuned mass dampers virtual displacements, and during this displacements the total energy of the system is constant. If the total energy is constant during the virtual displacements, the internal and external work must be equal.

The external work due to the bridge is

$$W_{bridge} = \int\limits_L \Big[q_z(x,t) - m_z(x)\ddot{r}_z(x,t) - c_z(x)\dot{r}_z(x,t) \Big] \delta r_z(x) dx \quad (1.47)$$

The external work due to the forces on the bridge from the tuned mass damper is

$$
\begin{aligned}
W_{bridge_{fromTMD}} &= \sum_j \Big[F_j(t)\Big]\delta r_z(x_j) \\
&= \sum_j \Big[C_j \dot{r}_{rel}(t) + K_j r_{rel}(t)\Big]\delta r_z(x_j) \\
&= \sum_j \Big[C_j[\dot{r}_{j_z}(t) - \dot{r}_z(x_j,t)] + K_j[r_{j_z}(t) - r_z(x_j,t)]\Big]\delta r_z(x_j) \\
&= \sum_j \Big[C_j \dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j \dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_z(x_j)
\end{aligned} \tag{1.48}
$$

The external work due to the forces on the the tuned mass damper is

$$
\begin{aligned}
W_{TMD} &= \sum_j \Big[M_j \ddot{r}_j(t) + F_j(t)\Big]\delta r_{j_z} \\
&= \sum_j \Big[M_j \ddot{r}_j(t) + C_j \dot{r}_{rel}(t) + K_j r_{rel}(t)\Big]\delta r_{j_z} \\
&= \sum_j \Big[M_j \ddot{r}_j(t) + C_j(\dot{r}_{j_z}(t) - \dot{r}_z(x_j,t)) + K_j(r_{j_z}(t) - r_z(x_j,t))\Big]\delta r_{j_z} \\
&= \sum_j \Big[M_j \ddot{r}_j(t) + C_j \dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j \dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_{j_z}
\end{aligned} \tag{1.49}
$$

The entire internal work are done by the bridge

$$
W_{internal} = \int\limits_L \int\limits_A \sigma_x(x,z,t)\delta\varepsilon_x(x,z,t)dAdx \tag{1.50}
$$

By Navier's hypothesis

$$
\begin{aligned}
\sigma_x &= \frac{M_y(x,t)}{I_y}z = \frac{-EI_y r_z''(x,t)}{I_y} = -Er_z''(x,t)\cdot z \\
\delta\varepsilon_x &= -\delta r_z''(x)\cdot z
\end{aligned} \tag{1.51}
$$

Thus

$$
\begin{aligned}
W_{internal} &= \int\limits_L \int\limits_A \Big[-Er_z''(x,t)\cdot z\Big]\Big[-\delta r_z''(x)\cdot z\Big]dAdx \\
&= \int\limits_L Er_z''(x,t)\cdot\delta r_z''(x)\int\limits_A z^2 dAdx \qquad ,where \quad \Big\{ \int\limits_A z^2 dA = I_y \\
&= \int\limits_L EI_y r_z''(x,t)\cdot\delta r_z''(x)dx
\end{aligned} \tag{1.52}
$$

Using d'Alembetr's principle

$$
\begin{aligned}
W_{external} &= W_{internal} \\
W_{bridge} + W_{bridge_{fromTMD}} - W_{TMD} &= W_{internal}
\end{aligned}
\tag{1.53}
$$

Inserted equations 1.47 to 1.52:

$$
\begin{aligned}
&\int_L \Big[q_z(x,t) - m_z(x)\ddot{r}_z(x,t) - c_z(x)\dot{r}_z(x,t)\Big]\delta r_z(x)dx \\
&+ \sum_j \Big[C_j\dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j\dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_z(x_j) \\
&- \sum_j \Big[M_j\ddot{r}_j(t) + C_j\dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j\dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_{j_z} \\
&= \int_L EI_y r_z''(x,t) \cdot \delta r_z''(x)dx
\end{aligned}
\tag{1.54}
$$

Rearranging

$$
\begin{aligned}
&\int_L m_z(x)\ddot{r}_z(x,t)\delta r_z(x)dx + \int_L c_z(x)\dot{r}_z(x,t)\delta r_z(x)dx + \int_L EI_y r_z''(x,t) \cdot \delta r_z''(x)dx \\
&- \sum_j \Big[C_j\dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j\dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_z(x_j) \\
&+ \sum_j \Big[M_j\ddot{r}_j(t) + C_j\dot{r}_{j_z}(t) + K_j r_{j_z}(t) - C_j\dot{r}_z(x_j,t) - K_j r_z(x_j,t)\Big]\delta r_{j_z} \\
&= \int_L q_z(x,t)\delta r_z(x)dx
\end{aligned}
\tag{1.55}
$$

Equations for single mode, single component

Equation 1.55 is the general equation that is applicable both for multi mode, multi component, and single mode, single component systems. It is more compact and easier to understand for single mode, but the concept are the same. Therefore it is best to understand the single mode, single component, before moving on to multi mode, multi component.

Displacement main system (bridge)	$\mathrm{r}_z(x,t) = \phi_z(x) \cdot \eta_z(t)$
Displacement mass damper	$\mathrm{r}_1(t) = 1 \cdot \eta_1(t)$
Virtual displacement bridge	$\delta r_z(x,t) = \phi_z(x) \cdot \delta\eta_z(t)$
Virtual displacement mass damper	$\delta r_1(t) = 1 \cdot \delta\eta_1(t)$

Table 1.1: Displacements and virtual displacements

Inserted into equation 1.55, this gives

$$\begin{aligned}
&\int\limits_L m_z(x)\phi_z(x)\ddot{\eta}_z(t)\phi_z(x)\delta\eta_z dx + \int\limits_L c_z(x)\phi_z(x)\dot{\eta}_z(t)\phi_z(x)\delta\eta_z dx + \int\limits_L EI_y\phi_z''(x)\eta_z(t)\phi_z''(x)\delta\eta_z dx \\
&-\sum_{j=1} \phi_z(x_1)\Big[C_1\dot{\eta}_1(t) + K_1\eta_1(t) - C_1\phi_z(x_1)\dot{\eta}_z(t) - K_1\phi_z(x_1)\eta_z(t)\Big]\delta\eta_z \\
&+\sum_{j=1} \Big[M_1\ddot{\eta}_1(t) + C_1\dot{\eta}_1(t) + K_1\eta_1(t) - C_1\phi_z(x_1)\dot{\eta}_z(t) - K_1\phi_z(x_1)\eta_z(t)\Big]\delta\eta_1 \\
&= \int\limits_L q_z(x,t)\phi_z\delta\eta_z(x)dx
\end{aligned} \tag{1.56}$$

Defining

$$\begin{aligned}
\tilde{M}_z &= \int\limits_L \phi_z^2(x) \cdot m_z(x)dx \\
\tilde{C}_z &= 2\xi_n\omega_n\tilde{M}_z = \int\limits_L \phi_z^2(x) \cdot c_z(x)dx \\
\tilde{K}_z &= \omega_n^2\tilde{M}_z = \int\limits_L \phi_z''^2(x) \cdot EI_y(x)dx \\
\tilde{R}_z &= \int\limits_L \phi_z(x) \cdot q_z(x,t)dx
\end{aligned} \tag{1.57}$$

Thus equation 1.81 simplifies to

$$
\begin{aligned}
&\tilde{M}_z\ddot{\eta}_z(t)\delta\eta_z + \tilde{C}_z\dot{\eta}_z(t)\delta\eta_z + \tilde{K}_z\eta_z(t)\delta\eta_z \\
&-\Big[C_1\phi_z(x_1)\dot{\eta}_1(t) + K_1\phi_z(x_1)\eta_1(t) - C_1\phi_z^2(x_1)\dot{\eta}_z(t) - K_1\phi_z^2(x_1)\eta_z(t)\Big]\delta\eta_z \\
&+\Big[M_1\ddot{\eta}_1(t) + C_1\dot{\eta}_1(t) + K_1\eta_1(t) - C_1\phi_z(x_1)\dot{\eta}_z(t) - K_1\phi_z(x_1)\eta_z(t)\Big]\delta\eta_1 \\
&= \tilde{R}_z\delta\eta_z
\end{aligned}
\tag{1.58}
$$

Some rearranging gives

$$
\begin{aligned}
&\delta\eta_z\Big[\tilde{M}_z\ddot{\eta}_z(t) + [\tilde{C}_z + \phi_z^2(x_1)C_1]\dot{\eta}_z(t) - \phi_z(x_1)C_1\dot{\eta}_1(t) + [\tilde{K}_z + K_1\phi_z^2(x_1)]\eta_z(t) - \phi_z(x_1)K_1\eta_1(t)\Big] \\
&+\delta\eta_1\Big[M_1\ddot{\eta}_1(t) - C_1\phi_z(x_1)\dot{\eta}_z(t) + C_1\dot{\eta}_1(t) - \phi_z(x_1)K_1\eta_z(t) + K_1\eta_1(t)\Big] \\
&=\delta\eta_z\Big[\tilde{R}_z\Big] + \delta\eta_1\Big[0\Big]
\end{aligned}
\tag{1.59}
$$

Or

$$
\begin{aligned}
&\begin{bmatrix}\delta\eta_z \\ \delta\eta_1\end{bmatrix}^T \left(\begin{bmatrix}\tilde{M}_z\ddot{\eta}_z(t) \\ \tilde{M}_1\ddot{\eta}_1(t)\end{bmatrix} + \begin{bmatrix}\Big(\tilde{C}_z + \phi_z^2(x_1)C_1\Big)\dot{\eta}_z(t) - \phi_z(x_1)C_1\dot{\eta}_1(t) \\ -\phi_z(x_1)C_1\dot{\eta}_z(t) + C_1\dot{\eta}_1(t)\end{bmatrix} + \begin{bmatrix}\Big(\tilde{K}_z + \phi_z^2(x_1)K_1\Big)\eta_z(t) - \phi_z(x_1)K_1\eta_1(t) \\ -\phi_z(x_1)K_1\eta_z(t) + K_1\eta_1(t)\end{bmatrix}\right) \\
&= \begin{bmatrix}\delta\eta_z & \delta\eta_1\end{bmatrix}\begin{bmatrix}\tilde{R}_z \\ 0\end{bmatrix}
\end{aligned}
\tag{1.60}
$$

Or

$$
\begin{aligned}
&\begin{bmatrix}\delta\eta_z \\ \delta\eta_1\end{bmatrix}^T \left(\begin{bmatrix}\tilde{M}_z & 0 \\ 0 & \tilde{M}_1\end{bmatrix}\begin{bmatrix}\ddot{\eta}_z(t) \\ \ddot{\eta}_1(t)\end{bmatrix} + \begin{bmatrix}\Big(\tilde{C}_z + \phi_z^2(x_1)C_1\Big) & -\phi_z(x_1)C_1 \\ -\phi_z(x_1)C_1 & C_1\end{bmatrix}\begin{bmatrix}\dot{\eta}_z(t) \\ \dot{\eta}_1(t)\end{bmatrix} + \begin{bmatrix}\Big(\tilde{K}_z + \phi_z^2(x_1)K_1\Big) & -\phi_z(x_1)K_1 \\ -\phi_z(x_1)K_1 & K_1\end{bmatrix}\begin{bmatrix}\eta_z(t) \\ \eta_1(t)\end{bmatrix}\right) \\
&= \begin{bmatrix}\delta\eta_z(t) & \delta\eta_1(t)\end{bmatrix}\begin{bmatrix}\tilde{R}_z \\ 0\end{bmatrix}
\end{aligned}
\tag{1.61}
$$

Or

$$
\boldsymbol{\delta\eta}_z^T\Big[\tilde{\mathbf{M}}_{z0}\ddot{\boldsymbol{\eta}}_{z0}(t) + \tilde{\mathbf{C}}_{z0}\dot{\boldsymbol{\eta}}_{z0}(t) + \tilde{\mathbf{K}}_{z0}\boldsymbol{\eta}_{z0}(t)\Big] = \boldsymbol{\delta\eta}_z^T\tilde{\mathbf{R}}_{z0}(t) \tag{1.62}
$$

Thus

$$
\tilde{\mathbf{M}}_{z0}\ddot{\boldsymbol{\eta}}_{z0}(t) + \tilde{\mathbf{C}}_{z0}\dot{\boldsymbol{\eta}}_{z0}(t) + \tilde{\mathbf{K}}_{z0}\boldsymbol{\eta}_{z0}(t) = \tilde{\mathbf{R}}_{z0}(t) \tag{1.63}
$$

where

$$
\begin{aligned}
&\boldsymbol{\delta\eta}_z = \begin{bmatrix}\delta\eta_z \\ \delta\eta_1\end{bmatrix} \quad , \quad \tilde{\mathbf{M}}_{z0} = \begin{bmatrix}\tilde{M}_z & 0 \\ 0 & M_1\end{bmatrix} \quad , \quad \ddot{\boldsymbol{\eta}}_{z0} = \begin{bmatrix}\ddot{\eta}_z \\ \ddot{\eta}_1\end{bmatrix} \\
&\tilde{\mathbf{C}}_{z0} = \begin{bmatrix}\tilde{C}_z + \phi_z^2(x_1)C_1 & -\phi_z(x_1)C_1 \\ -\phi_z(x_1)C_1 & C_1\end{bmatrix} \quad , \quad \tilde{\mathbf{K}}_{z0} = \begin{bmatrix}\tilde{K}_z + \phi_z^2(x_1)K_1 & -\phi_z(x_1)K_1 \\ -\phi_z(x_1)K_1 & K_1\end{bmatrix} \quad , \quad \tilde{\mathbf{R}}_{z0} = \begin{bmatrix}\tilde{R}_z \\ 0\end{bmatrix}
\end{aligned}
\tag{1.64}
$$

For the two degrees of freedom problem equation 1.22 was simplified into equation 1.72 where the stiffness matrix is diagonal, by difining the relative displacement of the damper in equation 1.23. The same thing is done for the single mode, single component case. It is important to remember the difference in the displacement components between equation 1.23 and equation 1.100:

Displacement of	2 DOF system in equation 1.23	Single mode, single component in equation 1.100
Main system	$r_1(t)$	$r_z(x,t)$
Mass damper	$r_2(t)$	$r_1(t)$

Table 1.2: Displacements for 2 DOF system and continuous, single mode single component system

As described in equation 1.1, $r_z(x,t) = \phi_z(x) \cdot \eta_z(t)$ and $r_1(t) = 1 \cdot \eta_1(t)$, thus

$$\Delta r_1(t) = 1 \cdot \Delta\eta_1(t) = r_1(t) - r_z(x_1,t) = 1 \cdot \eta_1(t) - \phi_z(x_1) \cdot \eta_z(t) \tag{1.65}$$

From where we obtain

$$\eta_1 = \phi_z \eta_z + 1 \cdot \Delta\eta_1 \tag{1.66}$$

Thus

$$\begin{bmatrix} r_z(x,t) \\ r_1(t) \end{bmatrix} = \begin{bmatrix} \phi_z(x) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_z(t) \\ \eta_1(t) \end{bmatrix} = \begin{bmatrix} r_z(x,t) \\ r_z(x_1,t) + \Delta r_1 \end{bmatrix} = \begin{bmatrix} \phi_z(x) & 0 \\ \phi_z(x_1) & 1 \end{bmatrix} \begin{bmatrix} \eta_z(t) \\ \Delta\eta_1(t) \end{bmatrix} \tag{1.67}$$

From the first part of equation 1.100 the following is obtained $\begin{bmatrix} \eta_z(t) \\ \eta_1(t) \end{bmatrix} = \begin{bmatrix} \phi_z(x) & 0 \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} r_z(x,t) \\ r_1(t) \end{bmatrix}$

Combining this with the result from the last part of equation 1.100 gives

$$\boldsymbol{\eta}_{z0}(t) = \begin{bmatrix} \eta_z(t) \\ \eta_1(t) \end{bmatrix} = \begin{bmatrix} \phi_z(x) & 0 \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \phi_z(x) & 0 \\ \phi_z(x_1) & 1 \end{bmatrix} \begin{bmatrix} \eta_z(t) \\ \Delta\eta_1(t) \end{bmatrix} = \boldsymbol{\Psi}_z(x) \cdot \boldsymbol{\eta}_z(t) \tag{1.68}$$

where

$$\boldsymbol{\Psi}_z = \begin{bmatrix} \phi_z(x) & 0 \\ 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \phi_z(x) & 0 \\ \phi_z(x_1) & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \phi_z(x_1) & 1 \end{bmatrix} \tag{1.69}$$

and

$$\boldsymbol{\eta}_z = \begin{bmatrix} \eta_z \\ \Delta\eta_1 \end{bmatrix} \tag{1.70}$$

Introducing this into equation 1.63

$$\boldsymbol{\Psi}_z^T \tilde{\mathbf{M}}_{z0} \boldsymbol{\Psi}_z \cdot \ddot{\boldsymbol{\eta}}_z(t) + \boldsymbol{\Psi}_z^T \tilde{\mathbf{C}}_{z0} \boldsymbol{\Psi}_z \cdot \dot{\boldsymbol{\eta}}_z(t) + \boldsymbol{\Psi}_z^T \tilde{\mathbf{K}}_{z0} \boldsymbol{\Psi}_z \cdot \boldsymbol{\eta}_z(t) = \boldsymbol{\Psi}_z^T \tilde{\mathbf{R}}_{z0}(t) \tag{1.71}$$

Obtaining

$$\tilde{\mathbf{M}}_z\ddot{\boldsymbol{\eta}}_z(t)+\tilde{\mathbf{C}}_z\dot{\boldsymbol{\eta}}_z(t)+\tilde{\mathbf{K}}_z\boldsymbol{\eta}_z(t)=\tilde{\mathbf{R}}_z(t) \tag{1.72}$$

where

$$\begin{aligned}
\tilde{\mathbf{M}}_z&=\boldsymbol{\Psi}_z^T\tilde{\mathbf{M}}_{z0}\boldsymbol{\Psi}_z=\begin{bmatrix}1 & \phi_z(x_1)\\0 & 1\end{bmatrix}\cdot\begin{bmatrix}\tilde{M}_z & 0\\0 & M_1\end{bmatrix}\cdot\begin{bmatrix}1 & 0\\\phi_z(x_1) & 1\end{bmatrix}=\begin{bmatrix}\tilde{M}_z+\phi_z^2(x_1)\cdot M_1 & \phi_z(x_1)\cdot M_1\\\phi_z(x_1)\cdot M_1 & M_1\end{bmatrix}\\
\tilde{\mathbf{C}}_z&=\boldsymbol{\Psi}_z^T\tilde{\mathbf{C}}_{z0}\boldsymbol{\Psi}_z=\begin{bmatrix}1 & \phi_z(x_1)\\0 & 1\end{bmatrix}\cdot\begin{bmatrix}\tilde{C}_z+\phi_z^2(x_1)C_1 & -\phi_z(x_1)C_1\\-\phi_z(x_1)C_1 & C_1\end{bmatrix}\cdot\begin{bmatrix}1 & 0\\\phi_z(x_1) & 1\end{bmatrix}=\begin{bmatrix}\tilde{C}_z & 0\\0 & C_1\end{bmatrix}\\
\tilde{\mathbf{K}}_z&=\boldsymbol{\Psi}_z^T\tilde{\mathbf{K}}_{z0}\boldsymbol{\Psi}_z=\begin{bmatrix}1 & \phi_z(x_1)\\0 & 1\end{bmatrix}\cdot\begin{bmatrix}\tilde{K}_z+\phi_z^2(x_1)K_1 & -\phi_z(x_1)K_1\\-\phi_z(x_1)K_1 & K_1\end{bmatrix}\cdot\begin{bmatrix}1 & 0\\\phi_z(x_1) & 1\end{bmatrix}=\begin{bmatrix}\tilde{K}_z & 0\\0 & K_1\end{bmatrix}\\
\tilde{\mathbf{R}}_z&=\boldsymbol{\Psi}_z^T\tilde{\mathbf{R}}_{z0}=\begin{bmatrix}1 & \phi_z(x_1)\\0 & 1\end{bmatrix}\cdot\begin{bmatrix}\tilde{R}_z\\0\end{bmatrix}=\begin{bmatrix}\tilde{R}_z\\0\end{bmatrix}
\end{aligned} \tag{1.73}$$

Since the equations 1.72 and 1.73 looks exactly like equation 1.24, the frequency response function development will be almost the same for a continuous system with one component, as it was for a single degree of freedom system also with one tuned mass damper. The only difference is that the single degree of freedom quantities has to be changed with modal quantities. Also it is important to remember the name-differences summarized in table 1.2. Thus, by taking the Fourier transform and write the equation on a compact form as in equation 1.31 to 1.46, the following frequency response function is obtained

Thus

$$\hat{\mathbf{H}}(\hat{\omega})=\begin{bmatrix}\hat{H}_{zz}(\hat{\omega}) & \hat{H}_{z1}(\hat{\omega})\\\hat{H}_{1z}(\hat{\omega}) & \hat{H}_{11}(\hat{\omega})\end{bmatrix} \tag{1.74}$$

where

$$\begin{aligned}
\hat{H}_{zz}(\hat{\omega})&=\frac{1+\tilde{b}_1(i\hat{\omega})+\tilde{b}_2(i\hat{\omega})^2}{1+\tilde{a}_1(i\hat{\omega})+\tilde{a}_2(i\hat{\omega})^2+\tilde{a}_3(i\hat{\omega})^3+\tilde{a}_4(i\hat{\omega})^4}\\
\hat{H}_{11}(\hat{\omega})&=\frac{1+\tilde{c}_1(i\hat{\omega})+\tilde{c}_2(i\hat{\omega})^2}{1+\tilde{a}_1(i\hat{\omega})+\tilde{a}_2(i\hat{\omega})^2+\tilde{a}_3(i\hat{\omega})^3+\tilde{a}_4(i\hat{\omega})^4}\\
\hat{H}_{1z}(\hat{\omega})&=\frac{\tilde{d}_2(i\hat{\omega})^2}{1+\tilde{a}_1(i\hat{\omega})+\tilde{a}_2(i\hat{\omega})^2+\tilde{a}_3(i\hat{\omega})^3+\tilde{a}_4(i\hat{\omega})^4}\\
\hat{H}_{z1}(\hat{\omega})&=\frac{\tilde{e}_2(i\hat{\omega})^2}{1+\tilde{a}_1(i\hat{\omega})+\tilde{a}_2(i\hat{\omega})^2+\tilde{a}_3(i\hat{\omega})^3+\tilde{a}_4(i\hat{\omega})^4}
\end{aligned} \tag{1.75}$$

where the tilde symbol indicates that it now is modal quantities, and where

$$
\begin{aligned}
\hat{\omega} &= \left(\frac{\omega}{\omega_z}\right) \\
\tilde{\alpha} &= \left(\frac{\omega_z}{\omega_1}\right) \\
\tilde{\mu} &= \left(\frac{M_1}{\tilde{M}_z}\right) \\
\tilde{a}_1 &= 2(\xi_z + \tilde{\alpha}\xi_1) \\
\tilde{a}_2 &= 1 + \tilde{\mu} + \tilde{\alpha}^2 + 4\tilde{\alpha}\xi_z\xi_1 \\
\tilde{a}_3 &= 2\tilde{\alpha}[\tilde{\alpha}\xi_z + (1+\tilde{\mu})\xi_1] \\
\tilde{a}_4 &= \tilde{\alpha}^2
\end{aligned}
\tag{1.76}
$$

and

$$
\begin{aligned}
\tilde{b}_1 &= 2\xi_1\tilde{\alpha} \\
\tilde{b}_2 &= \tilde{\alpha}^2 \\
\tilde{c}_1 &= 2\xi_z \\
\tilde{c}_2 &= (1+\tilde{\mu}) \\
\tilde{d}_2 &= -\tilde{\alpha}^2 \\
\tilde{e}_2 &= -\tilde{\mu}
\end{aligned}
\tag{1.77}
$$

Equations for multi mode, multi component

System with N_j dampers, with the properties for each of the j^{th} dampers, j= 1,2,...,N_j, is collected in the matrixes

$$
\begin{aligned}
\mathbf{M}_d &= diag[M_j] \\
\mathbf{C}_d &= diag[C_j] \\
\mathbf{K}_d &= diag[K_j]
\end{aligned}
\tag{1.78}
$$

Since the procedure for single mode, single component, and multi mode multi component are quite similar, some basic steps are taken in table form.

Variable	Single mode, single component	Multi mode, multi component
$r_z(x,t) =$	$\phi_z(x) \cdot \eta_z(t)$	$\sum\limits_{n=1}^{N_{mod}} \phi_{zn}(x) \cdot \eta_{zn}(t) = \boldsymbol{\phi}_z(x) \cdot \boldsymbol{\eta}_z(t)$
Eigenmode(s)	$\phi_z(x)$	$\boldsymbol{\phi}_z(x) = [\phi_{z_1} \quad ... \quad \phi_{z_n} \quad ... \quad \phi_{zN_{mod}}]$
Modal coordinate(s)	$\eta_z(t)$	$\boldsymbol{\eta}_z(t) = [\eta_{z_1} \quad ... \quad \eta_{z_n} \quad ... \quad \eta_{zN_{mod}}]$
Damper displacement	$r_1(t) = 1 \cdot \eta_1(t)$	$\boldsymbol{r}_d(t) = 1 \cdot [\eta_1 \quad ... \quad \eta_j \quad ... \quad \eta_{N_j}] = 1 \cdot \boldsymbol{\eta}_d(t)$
Modal mass, $\tilde{\boldsymbol{M}} =$	$\begin{bmatrix} \tilde{M}_z + \phi_z^2(x_1) \cdot M_1 & \phi_z(x_1) \cdot M_1 \\ \phi_z(x_1) \cdot M_1 & M_1 \end{bmatrix}$	$\begin{bmatrix} \tilde{\boldsymbol{M}}_z + \boldsymbol{\phi}_d^T \boldsymbol{M}_d \boldsymbol{\phi}_d & \boldsymbol{\phi}_d^T \boldsymbol{M}_d \\ \boldsymbol{\phi}_d^T \boldsymbol{M}_d & \boldsymbol{M}_d \end{bmatrix}$
where	$\tilde{M}_z = \int\limits_L \phi_z^2(x) \cdot m_z(x) dx$	$\tilde{\boldsymbol{M}}_z = diag[\tilde{M}_{z_n}]$ where $\tilde{M}_{z_n} = \int\limits_L \phi_{z_n}^2(x) \cdot m_z(x) dx$
Modal damping, $\tilde{\boldsymbol{C}} =$	$\begin{bmatrix} \tilde{C}_z & 0 \\ 0 & C_1 \end{bmatrix}$	$\begin{bmatrix} \tilde{\boldsymbol{C}}_z & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_d \end{bmatrix}$
where	$\tilde{C}_z = \int\limits_L \phi_z^2(x) \cdot c_z(x) dx$	$\tilde{\boldsymbol{C}}_z = diag[\tilde{C}_{z_n}]$ where $\tilde{C}_{z_n} = \int\limits_L \phi_{z_n}^2(x) \cdot c_z(x) dx$
Modal stiffness, $\tilde{\boldsymbol{K}} =$	$\begin{bmatrix} \tilde{K}_z & 0 \\ 0 & K_1 \end{bmatrix}$	$\begin{bmatrix} \tilde{\boldsymbol{K}}_z & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_d \end{bmatrix}$
where	$\tilde{K}_z = \int\limits_L \phi_z''^2(x) \cdot EI_y(x) dx$	$\tilde{\boldsymbol{K}}_z = diag[\tilde{K}_{z_n}]$ where $\tilde{K}_{z_n} = \int\limits_L \phi_{z_n}''^2(x) \cdot EI_y(x) dx$
Modal load, $\tilde{\boldsymbol{R}} =$	$\begin{bmatrix} \tilde{R}_z \\ 0 \end{bmatrix}$	$\begin{bmatrix} \tilde{\boldsymbol{R}}_z \\ \boldsymbol{0} \end{bmatrix}$ where $\tilde{\boldsymbol{R}}_z = [\tilde{R}_{z_1} \quad ... \quad \tilde{R}_{z_n} \quad ... \quad \tilde{R}_{z_{N_{mod}}}]^T$
where	$\tilde{R}_z = \int\limits_L \phi_z(x) \cdot q_z(x,t) dx$	where $\tilde{R}_{z_n} = \int\limits_L \phi_{z_n}(x) q_z(x,t) dx$

Table 1.3: Differences and similarities between single mode, single component and multi mode, multi component

Since the mode shapes is obtained from an eigenvalue solution they have to be orthogonal [Strømmen, 2013]. Therefore it is possible to make use of the orthogonality principle and thus

$$\begin{aligned}
&\int_L m_z(x)\cdot\phi_n(x)\phi_m(x)dx = 0 \quad \text{, for} \quad n \neq m \\
&\int_L c_z(x)\cdot\phi_n(x)\phi_m(x)dx = 0 \quad \text{, for} \quad n \neq m \\
&\int_L EI_y(x)\cdot\phi_n''(x)\phi_m''(x)dx = 0 \quad \text{, for} \quad n \neq m
\end{aligned} \tag{1.79}$$

The displacement $r_z(x,t)$ used in the expression for the virtual energy contains a sum of all the mode shapes multiplied by the modal coordinate. By setting the virtual displacement equal one of the modeshape multiplied by the corresponding modal coordinate,the integrals in the virtual energy expression containing the both the displacement $r_z(x,t)$ and the virtual displacement, the orthogonality cancel out every terms except the term containing the mode shape of the virtual displacement. By setting the virtual displacement equal every mode shapes, one at a time, $N_{mod} + N_j$ equations is obtained, which makes it possible to solve for the same number of unknowns. I.e. by setting the virtual displacements equal all of the following N_{mod} choices, one at a time

$$\begin{aligned}
&1) \quad \delta r_z = \phi_{z_1}\delta\eta_{z_1} \quad \text{and} \quad \delta\boldsymbol{r}_d = [\delta r_1 \quad ... \quad \delta r_j \quad ... \quad \delta r_{N_j}]^T \\
&\vdots \\
&n) \quad \delta r_z = \phi_{z_n}\delta\eta_{z_n} \quad \text{and} \quad \delta\boldsymbol{r}_d = [\delta r_1 \quad ... \quad \delta r_j \quad ... \quad \delta r_{N_j}]^T \\
&\vdots \\
&N_{mod}) \quad \delta r_z = \phi_{z_{N_{mod}}}\delta\eta_{z_{N_{mod}}} \quad \text{and} \quad \delta\boldsymbol{r}_d = [\delta r_1 \quad ... \quad \delta r_j \quad ... \quad \delta r_{N_j}]^T
\end{aligned} \tag{1.80}$$

$N_{mod} + N_j$ virtual energy equations is obtained, where the n-th equation looks like

$$\begin{aligned}
&\delta\eta_{z_n}\left[\int_L m_z(x)\phi_{z_n}(x)\ddot{\eta}_{z_n}(t)\phi_{z_n}(x)dx + \int_L c_z(x)\phi_{z_n}(x)\dot{\eta}_{z_n}(t)\phi_{z_n}(x)dx + \int_L EI_y\phi_{z_n}''(x)\eta_{z_n}(t)\phi_{z_n}''(x)dx\right] \\
&-\delta\eta_{z_n}\left[\sum_{j=1}^{N_j}\phi_{z_n}(x_j)\Big[C_j\dot{\eta}_j(t) + K_j\eta_j(t) - C_j\phi_{z_n}(x_j)\dot{\eta}_{z_n}(t) - K_j\phi_{z_n}(x_j)\eta_{z_n}(t)\Big]\right] \\
&+\sum_{j=1}^{N_j}\delta\eta_j\Big[M_j\ddot{\eta}_j(t) + C_j\dot{\eta}_j(t) + K_j\eta_j(t) - C_j\phi_{z_n}(x_j)\dot{\eta}_{z_n}(t) - K_j\phi_{z_n}(x_j)\eta_{zn}(t)\Big] \\
&= \delta\eta_{z_n}\int_L q_z(x,t)\phi_{z_n}dx
\end{aligned} \tag{1.81}$$

Using the definitions in table 1.3
$\int_L \phi_{z_n}^2(x)\cdot m_z(x)dx = \tilde{M}_{z_n}$, $\int_L \phi_{z_n}^2(x)\cdot c_z(x)dx = \tilde{C}_{z_n}$, $\int_L \phi_{z_n}''^2(x)\cdot EI_y(x)dx = \tilde{K}_{z_n}$ and $\int_L \phi_{z_n}(x)q_z(x,t)dx = \tilde{R}_{z_n}$

equation 1.81 turns into

$$
\begin{aligned}
&\delta\eta_{z_n}\left[\tilde{M}_{z_n}\ddot{\eta}_{z_n}(t)+\tilde{C}_{z_n}\dot{\eta}_{z_n}(t)+\tilde{K}_{z_n}\eta_{z_n}(t)\right]\\
&-\delta\eta_{z_n}\left[\sum_{j=1}^{N_j}\phi_{z_n}(x_j)\Big[C_j\dot{\eta}_j(t)+K_j\eta_j(t)-C_j\phi_{z_n}(x_j)\dot{\eta}_{z_n}(t)-K_j\phi_{z_n}(x_j)\eta_{z_n}(t)\Big]\right]\\
&+\sum_{j=1}^{N_j}\delta\eta_j\Big[M_j\ddot{\eta}_j(t)+C_j\dot{\eta}_j(t)+K_j\eta_j(t)-C_j\phi_{z_n}(x_j)\dot{\eta}_{z_n}(t)-K_j\phi_{z_n}(x_j)\eta_{z_n}(t)\Big]\\
&=\delta\eta_{z_n}\tilde{R}_{z_n}(t)
\end{aligned}
\tag{1.82}
$$

Defining

$$
\begin{aligned}
&\sum_{j=1}^{N_j}\phi_{z_n}(x_j)\cdot C_j\dot{\eta}_j(t)=\phi_{z_n}(x_1)\cdot C_1\dot{\eta}_1+...+\phi_{z_n}(x_j)\cdot C_j\dot{\eta}_j+...+\phi_{z_n}(x_{N_j})\dot{\eta}_{N_j}\\
&=\begin{bmatrix}\phi_{z_n}(x_1) & ... & \phi_{z_n}(x_j) & ... & \phi_{z_n}(x_{N_j})\end{bmatrix}\cdot\begin{bmatrix}C_1 & ... & 0 & ... & 0\\ & \ddots & & & \\ 0 & ... & C_j & ... & 0\\ & & & \ddots & \\ 0 & ... & 0 & ... & C_{N_j}\end{bmatrix}\cdot\begin{bmatrix}\dot{\eta}_1\\ \vdots\\ \dot{\eta}_j\\ \vdots\\ \dot{\eta}_{N_j}\end{bmatrix}=\boldsymbol{\Phi}_{d_n}^T\boldsymbol{C}_d\dot{\boldsymbol{\eta}}_d
\end{aligned}
\tag{1.83}
$$

and

$$
\begin{aligned}
&\sum_{j=1}^{N_j}\phi_{z_n}^2(x_j)\cdot C_j\dot{\eta}_{z_n}(t)=\left[\phi_{z_n}^2(x_1)\cdot C_1+...+\phi_{z_n}^2(x_j)\cdot C_j+...+\phi_{z_n}^2(x_{N_j})\right]\dot{\eta}_{z_n}\\
&=\begin{bmatrix}\phi_{z_n}(x_1) & ... & \phi_{z_n}(x_j) & ... & \phi_{z_n}(x_{N_j})\end{bmatrix}\cdot\begin{bmatrix}C_1 & ... & 0 & ... & 0\\ & \ddots & & & \\ 0 & ... & C_j & ... & 0\\ & & & \ddots & \\ 0 & ... & 0 & ... & C_{N_j}\end{bmatrix}\cdot\begin{bmatrix}\phi_{z_n}(x_1)\\ \phi_{z_n}(x_j)\\ \phi_{z_n}(x_{N_j})\end{bmatrix}\cdot\dot{\eta}_{z_n}\\
&=\boldsymbol{\Phi}_{d_n}^T\boldsymbol{C}_d\boldsymbol{\Phi}_{d_n}\dot{\eta}_{z_n}
\end{aligned}
\tag{1.84}
$$

and

$$\sum_{j=1}^{N_j} \delta\eta_j M_j \ddot{\eta}_j(t) = \delta\eta_1 \cdot M_1 \cdot \ddot{\eta}_1 + ... + \delta\eta_j \cdot M_j \cdot \ddot{\eta}_j + ... + \delta\eta_{N_j} \cdot M_{N_j} \cdot \ddot{\eta}_{N_j}$$

$$= \begin{bmatrix} \delta\eta_1 & ... & \delta\eta_j & ... & \delta\eta_{N_j} \end{bmatrix} \cdot \begin{bmatrix} M_1 & ... & 0 & ... & 0 \\ & \ddots & & & \\ 0 & ... & M_j & ... & 0 \\ & & & \ddots & \\ 0 & ... & 0 & ... & M_{N_j} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\eta}_1(t) \\ \vdots \\ \ddot{\eta}_j(t) \\ \vdots \\ \ddot{\eta}_{N_j}(t) \end{bmatrix} = \boldsymbol{\delta\eta}_d^T \boldsymbol{M}_d \ddot{\boldsymbol{\eta}}_d \tag{1.85}$$

and

$$\sum_{j=1}^{N_j} \delta\eta_j \phi_{z_n}(x_j) \cdot C_j \dot{\eta}_{z_n}(t) = \delta\eta_1 \cdot C_1 \cdot \phi_{z_n}(x_1) + ... + \delta\eta_j \cdot C_j \cdot \phi_{z_n}(x_j) + ... + \delta\eta_{N_j} \cdot C_{N_j} \cdot \phi_{z_n}(x_{N_j})$$

$$= \begin{bmatrix} \delta\eta_1 & ... & \delta\eta_j & ... & \delta\eta_{N_j} \end{bmatrix} \cdot \begin{bmatrix} C_1 & ... & 0 & ... & 0 \\ & \ddots & & & \\ 0 & ... & C_j & ... & 0 \\ & & & \ddots & \\ 0 & ... & 0 & ... & C_{N_j} \end{bmatrix} \cdot \begin{bmatrix} \phi_{z_n}(x_1) \\ \vdots \\ \phi_{z_n}(x_j) \\ \vdots \\ \phi_{z_n}(x_{N_j}) \end{bmatrix} \cdot \dot{\eta}_{z_n}(t) = \boldsymbol{\delta\eta}_d^T \boldsymbol{C}_d \boldsymbol{\Phi}_{d_n} \dot{\eta}_{z_n}(t) \tag{1.86}$$

Thus

$$\begin{bmatrix} \delta\eta_{z_n} & \boldsymbol{\delta\eta}_d^T \end{bmatrix} \left(\begin{bmatrix} \tilde{M}_{z_n} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M_d} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\eta}_{z_n}(t) \\ \ddot{\boldsymbol{\eta}}_d(t) \end{bmatrix} + \begin{bmatrix} [\tilde{C}_{z_n} + \boldsymbol{\Phi}_{d_n}^T \boldsymbol{C}_d \boldsymbol{\Phi}_{d_n}] & \boldsymbol{\Phi}_{d_n}^T \boldsymbol{C}_d \\ \boldsymbol{C}_d \boldsymbol{\Phi}_{d_n} & \boldsymbol{C}_d \end{bmatrix} \cdot \begin{bmatrix} \dot{\eta}_{z_n}(t) \\ \dot{\boldsymbol{\eta}}_d(t) \end{bmatrix} \right)$$
$$+ \begin{bmatrix} \delta\eta_{z_n} & \boldsymbol{\delta\eta}_d^T \end{bmatrix} \left(\begin{bmatrix} [\tilde{K}_{z_n} + \boldsymbol{\Phi}_{d_n}^T \boldsymbol{K}_d \boldsymbol{\Phi}_{d_n}] & \boldsymbol{\Phi}_{d_n}^T \boldsymbol{K}_d \\ \boldsymbol{K}_d \boldsymbol{\Phi}_{d_n} & \boldsymbol{K}_d \end{bmatrix} \cdot \begin{bmatrix} \eta_{z_n}(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} \right) = \begin{bmatrix} \delta\eta_{z_n} & \boldsymbol{\delta\eta}_d^T \end{bmatrix} \begin{bmatrix} \tilde{R}_{z_n} \\ \mathbf{0} \end{bmatrix} \tag{1.87}$$

is the n^{th} virtual energy equation on matrix form. As can be seen, the n^{th} equation contains the information connected to the n^{th} mode shape and virtual displacement, and the n^{th} modal displacement can be solved from this equation. Also it contains the modal degrees of freedom to the N_j mass dampers. Since the n^{th} equation only involves the n^{th} mode shape, the modal displacement of the mass dampers is only based on this n^{th} mode shape. With one dominating mode, the answer obtained from a single n^{th} equation can be quite good, but in reality and in most practical cases the mass damper displacement is based on two or more mode shapes.

The differences between the n^{th} equation and the reality with N_{mod} equations, regarding the modal solution and which mode shape the modal mass damper displacement is based on, is illustrated in table 1.4.

	Modal DOF solution	$\boldsymbol{\eta}_d$ obtained from mode shape(s)
n^{th} equation	$\begin{bmatrix} \eta_{z_n} \\ \boldsymbol{\eta}_d \end{bmatrix} = \begin{bmatrix} \eta_{z_n} & \eta_1 & ... & \eta_j & ... & \eta_{N_j} \end{bmatrix}^T$	ϕ_{z_n}
Reality	$\begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\eta}_d \end{bmatrix} = \begin{bmatrix} \eta_{z_1} & ... & \eta_{z_n} & ... & \eta_{z_{N_{mod}}} & \eta_1 & ... & \eta_j & ... & \eta_{N_j} \end{bmatrix}^T$	$\boldsymbol{\Phi}_z = [\phi_{z_1} \quad ... \quad \phi_{z_n} \quad ... \quad \phi_{z_{N_{mod}}}]^T$

Table 1.4: Differences between the n^{th} equation and the reality with N_{mod} equations

To include all N_{mod} mode shapes it is useful to make some definitions

$$\boldsymbol{\delta\eta}_z = \begin{bmatrix} \delta\eta_{z_1} & ... & \delta\eta_{z_n} & .. & \delta\eta_{z_{N_{mod}}} \end{bmatrix}^T \quad \text{and} \quad \boldsymbol{\eta}_z = \begin{bmatrix} \eta_{z_1} & ... & \eta_{z_n} & .. & \eta_{z_{N_{mod}}} \end{bmatrix}^T$$

$$\tilde{\boldsymbol{M}}_z = diag[\tilde{M}_{z_n}] \quad , \quad \tilde{\boldsymbol{C}}_z = diag[\tilde{C}_{z_n}] \quad \text{and} \quad \tilde{\boldsymbol{K}}_z = diag[\tilde{K}_{z_n}]$$

$$\boldsymbol{\Phi}_d = [\boldsymbol{\Phi}_{d_1} \quad ... \quad \boldsymbol{\Phi}_{d_n} \quad ... \quad \boldsymbol{\Phi}_{d_{N_{mod}}}] = \begin{bmatrix} \boldsymbol{\Phi}_z(x_1) \\ \vdots \\ \boldsymbol{\Phi}_z(x_j) \\ \vdots \\ \boldsymbol{\Phi}_z(x_{N_j}) \end{bmatrix} = \begin{bmatrix} \phi_{z_1}(x_1) & ... & \phi_{z_n}(x_1) & ... & \phi_{z_{N_{mod}}}(x_1) \\ \vdots & & \vdots & & \vdots \\ \phi_{z_1}(x_j) & ... & \phi_{z_n}(x_j) & ... & \phi_{z_{N_{mod}}}(x_j) \\ \vdots & & \vdots & & \vdots \\ \phi_{z_1}(x_{N_j}) & ... & \phi_{z_n}(x_{N_j}) & ... & \phi_{z_{N_{mod}}}(x_{N_j}) \end{bmatrix} \tag{1.88}$$

To illustrate the difference between the n^{th} equation case with scalar $\delta\eta_z^T$ and ϕ_{z_n}, and the case that includes all N_{mod} mode shapes, relevant matrix-expressions is written out underneath

The n^{th} equation case:

$$\delta\eta_z \boldsymbol{\Phi}_{d_n}^T \boldsymbol{C}_d \boldsymbol{\eta}_d =$$

$$\delta\eta_{z_n} \begin{bmatrix} \phi_{z_n}(x_1) \cdot C_1 & ... & \phi_{z_n}(x_j) \cdot C_j & ... & \phi_{z_n}(x_{N_j}) \cdot C_{N_j} \end{bmatrix} \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_j \\ \vdots \\ \dot{\eta}_{N_j} \end{bmatrix} \tag{1.89}$$

$$= \delta\eta_{z_n} \cdot \left(\phi_{z_n}(x_1) \cdot C_1 \cdot \dot{\eta}_1 + ... + \phi_{z_n}(x_j) \cdot C_j \cdot \dot{\eta}_j + ... + \phi_{z_n}(x_{N_j}) \cdot C_{N_j} \cdot \dot{\eta}_{N_j} \right)$$

The case that includes all N_{mod} mode shapes:

$$\boldsymbol{\delta\eta}_z^T \boldsymbol{\Phi}_d^T \boldsymbol{C}_d \boldsymbol{\eta}_d =$$

$$\begin{bmatrix} \delta\eta_{z_1} & ... & \delta\eta_{z_n} & ... & \delta\eta_{z_{N_{mod}}} \end{bmatrix} \begin{bmatrix} \phi_{z_1}(x_1)\cdot C_1 & ... & \phi_{z_1}(x_j)\cdot C_j & ... & \phi_{z_1}(x_{N_j})\cdot C_{N_j} \\ \vdots & & \vdots & & \vdots \\ \phi_{z_n}(x_1)\cdot C_1 & ... & \phi_{z_n}(x_j)\cdot C_j & ... & \phi_{z_n}(x_{N_j})\cdot C_{N_j} \\ \vdots & & \vdots & & \vdots \\ \phi_{z_{N_{mod}}}(x_1)\cdot C_1 & ... & \phi_{z_{N_{mod}}}(x_j)\cdot C_j & ... & \phi_{z_{N_{mod}}}(x_{N_j})\cdot C_{N_j} \end{bmatrix} \begin{bmatrix} \dot\eta_1 \\ \vdots \\ \dot\eta_j \\ \vdots \\ \dot\eta_{N_j} \end{bmatrix}$$

$$= \begin{bmatrix} \delta\eta_{z_1}\cdot\left(\phi_{z_1}(x_1)\cdot C_1\cdot\dot\eta_1 + ... + \phi_{z_1}(x_j)\cdot C_j\cdot\dot\eta_j + ... + \phi_{z_1}(x_{N_j})\cdot C_{N_j}\cdot\dot\eta_{N_j}\right) \\ \vdots \\ \delta\eta_{z_n}\cdot\left(\phi_{z_n}(x_1)\cdot C_1\cdot\dot\eta_1 + ... + \phi_{z_n}(x_j)\cdot C_j\cdot\dot\eta_j + ... + \phi_{z_n}(x_{N_j})\cdot C_{N_j}\cdot\dot\eta_{N_j}\right) \\ \vdots \\ \delta\eta_{z_{N_{mod}}}\cdot\left(\phi_{z_{N_{mod}}}(x_1)\cdot C_1\cdot\dot\eta_1 + ... + \phi_{z_{N_{mod}}}(x_j)\cdot C_j\cdot\dot\eta_j + ... + \phi_{z_{N_{mod}}}(x_{N_j})\cdot C_{N_j}\cdot\dot\eta_{N_j}\right) \end{bmatrix} \tag{1.90}$$

where the n-th row can be found in equation 1.82, as $\delta\eta_{z_n}\left[\sum_{j=1}^{N_j}\phi_{z_n}(x_j)C_j\dot\eta_j(t)\right]$

Using the definitions in equation 1.88, an equation similar to 1.87 is obtained, with all the N_{mod} unknowns modal degree of freedom, η_{z_n}'s. Thus all the N_j unknown mass damper modal degree of freedom is based on all N_{mod} mode shapes. The number N_{mod} is set by the user, since a continuous system in theory have a infinite number of mode shapes, and since it often is the first, lowest frequency modes that is most relevant for vortex shedding calculations.

Thus

$$\begin{aligned} &[\boldsymbol{\delta\eta}_z^T \quad \boldsymbol{\delta\eta}_d^T]\left(\begin{bmatrix} \tilde{\boldsymbol{M}}_z & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M_d} \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{\eta}}_z(t) \\ \ddot{\boldsymbol{\eta}}_d(t) \end{bmatrix} + \begin{bmatrix} [\tilde{\boldsymbol{C}}_z + \boldsymbol{\Phi}_d^T\boldsymbol{C}_d\boldsymbol{\Phi}_d] & \boldsymbol{\Phi}_d^T\boldsymbol{C}_d \\ \boldsymbol{C}_d\boldsymbol{\Phi}_d & \boldsymbol{C}_d \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{\eta}}_z(t) \\ \dot{\boldsymbol{\eta}}_d(t) \end{bmatrix}\right) \\ &+ [\boldsymbol{\delta\eta}_z^T \quad \boldsymbol{\delta\eta}_d^T]\left(\begin{bmatrix} [\tilde{\boldsymbol{K}}_z + \boldsymbol{\Phi}_d^T\boldsymbol{K}_d\boldsymbol{\Phi}_d] & \boldsymbol{\Phi}_d^T\boldsymbol{K}_d \\ \boldsymbol{K}_d\boldsymbol{\Phi}_d & \boldsymbol{K}_d \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix}\right) = [\boldsymbol{\delta\eta}_z^T \quad \boldsymbol{\delta\eta}_d^T]\begin{bmatrix} \tilde{\boldsymbol{R}}_z \\ \boldsymbol{0} \end{bmatrix} \end{aligned} \tag{1.91}$$

Omitting $[\boldsymbol{\delta\eta}_z^T \quad \boldsymbol{\delta\eta}_d^T]$ gives

$$\begin{bmatrix} \tilde{\boldsymbol{M}}_z & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M_d} \end{bmatrix}\begin{bmatrix} \ddot{\boldsymbol{\eta}}_z(t) \\ \ddot{\boldsymbol{\eta}}_d(t) \end{bmatrix} + \begin{bmatrix} [\tilde{\boldsymbol{C}}_z + \boldsymbol{\Phi}_d^T\boldsymbol{C}_d\boldsymbol{\Phi}_d] & \boldsymbol{\Phi}_d^T\boldsymbol{C}_d \\ \boldsymbol{C}_d\boldsymbol{\Phi}_d & \boldsymbol{C}_d \end{bmatrix}\begin{bmatrix} \dot{\boldsymbol{\eta}}_z(t) \\ \dot{\boldsymbol{\eta}}_d(t) \end{bmatrix} + \begin{bmatrix} [\tilde{\boldsymbol{K}}_z + \boldsymbol{\Phi}_d^T\boldsymbol{K}_d\boldsymbol{\Phi}_d] & \boldsymbol{\Phi}_d^T\boldsymbol{K}_d \\ \boldsymbol{K}_d\boldsymbol{\Phi}_d & \boldsymbol{K}_d \end{bmatrix}\begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{R}}_z \\ \boldsymbol{0} \end{bmatrix} \tag{1.92}$$

In short form

$$\tilde{\boldsymbol{M}}_{z0}\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{C}}_{z0}\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{K}}_{z0}\boldsymbol{\eta} = \tilde{\boldsymbol{R}}_{z0} \tag{1.93}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} \tag{1.94}$$

As described in table 1.3, $r_z(x,t) = \boldsymbol{\Phi}_z(x) \cdot \boldsymbol{\eta}_z(t)$ and $\boldsymbol{r}_d(t) = 1 \cdot \boldsymbol{\eta}_d(t)$, thus

$$\Delta r_j(t) = 1 \cdot \Delta\eta_j(t) = r_j(t) - r_z(x_j,t) = 1 \cdot \eta_j(t) - \phi_z(x_j) \cdot \eta_z(t) \tag{1.95}$$

and

$$\boldsymbol{\Delta r}_d(t) = 1 \cdot \boldsymbol{\Delta\eta}_d(t) = \boldsymbol{r}_d(t) - \boldsymbol{r}_z(x_j,t) = 1 \cdot \boldsymbol{\eta}_d(t) - \boldsymbol{\Phi}_z(x) \cdot \boldsymbol{\eta}_z(t) \tag{1.96}$$

From where we obtain

$$\eta_j = \phi_z \eta_z + 1 \cdot \Delta\eta_j \tag{1.97}$$

and

$$\boldsymbol{\eta}_d = \boldsymbol{\Phi}_z \boldsymbol{\eta}_z + 1 \cdot \boldsymbol{\Delta\eta}_d \tag{1.98}$$

Letting

$$\begin{aligned} &\boldsymbol{x}_d = [x_1 \quad ... \quad x_j \quad ... \quad x_{N_j}] \qquad \text{and} \\ &\boldsymbol{r}_z(\boldsymbol{x}_d,t) = [r_z(x_1,t) \quad ... \quad r_z(x_j,t) \quad ... \quad r_z(x_{N_j},t)]^T = \boldsymbol{\Phi}_d(x) \cdot \boldsymbol{\eta}_z(t) \end{aligned} \tag{1.99}$$

The following is obtained

$$\begin{bmatrix} r_z(x,t) \\ \boldsymbol{r}_d(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} = \begin{bmatrix} r_z(x,t) \\ \boldsymbol{r}_z(\boldsymbol{x}_d,t) + \boldsymbol{\Delta r}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{\Phi}_d & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\Delta\eta}_d(t) \end{bmatrix} \tag{1.100}$$

From the first part of equation 1.100 the following is obtained $\begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \cdot \begin{bmatrix} r_z(x,t) \\ \boldsymbol{r}_d(t) \end{bmatrix}$

Combining this with the result from the last part of equation 1.100 gives

$$\boldsymbol{\eta}(t) = \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\eta}_d(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{\Phi}_d(x) & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\Delta\eta}_d(t) \end{bmatrix} = \boldsymbol{\Psi}(x) \cdot \boldsymbol{\eta}_{rel}(t) \tag{1.101}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \boldsymbol{\Phi}_z(x) & \boldsymbol{0} \\ \boldsymbol{\Phi}_d(x) & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\Phi}_d & \boldsymbol{I} \end{bmatrix} \tag{1.102}$$

and

$$\boldsymbol{\eta}_{rel} = \begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\Delta\eta}_d \end{bmatrix} \tag{1.103}$$

Introducing $\boldsymbol{\eta}(t) = \boldsymbol{\Psi}(x) \cdot \boldsymbol{\eta}_{rel}(t)$ and pre-multiply by $\boldsymbol{\Psi}^T$ in equation 1.92, gives the equilibrium condition in relative degrees of freedom, shown in equation 1.104. The mass,damping and stiffness matrixes and the load vector in equation 1.104, is given in table 1.5, where $\tilde{\boldsymbol{M}}_0$, $\tilde{\boldsymbol{C}}_0$, $\tilde{\boldsymbol{K}}_0$ and $\tilde{\boldsymbol{R}}_0$ is extracted from equations 1.92 and 1.93.

$\tilde{\boldsymbol{M}}$	$= \boldsymbol{\Psi}^T \tilde{\boldsymbol{M}}_0 \boldsymbol{\Psi}$	$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_d \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{M}}_z & 0 \\ 0 & \boldsymbol{M_d} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{\Phi}_d & \boldsymbol{I} \end{bmatrix}$	$= \begin{bmatrix} \tilde{\boldsymbol{M}}_z + \boldsymbol{\Phi}_d^T \boldsymbol{M}_d \boldsymbol{\Phi}_d & \boldsymbol{\Phi}_d^T \boldsymbol{M}_d \\ \boldsymbol{M}_d \boldsymbol{\Phi}_d & \boldsymbol{M}_d \end{bmatrix}$
$\tilde{\boldsymbol{C}}$	$= \boldsymbol{\Psi}^T \tilde{\boldsymbol{C}}_0 \boldsymbol{\Psi}$	$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_d \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{C}}_z + \boldsymbol{\Phi}_d^T \boldsymbol{C}_d \boldsymbol{\Phi}_d & -\boldsymbol{\Phi}_d^T \boldsymbol{C}_d \\ -\boldsymbol{C}_d \boldsymbol{\Phi}_d & \boldsymbol{C}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{\Phi}_d & \boldsymbol{I} \end{bmatrix}$	$= \begin{bmatrix} \tilde{\boldsymbol{C}}_z & 0 \\ 0 & \boldsymbol{C}_d \end{bmatrix}$
$\tilde{\boldsymbol{K}}$	$= \boldsymbol{\Psi}^T \tilde{\boldsymbol{K}}_0 \boldsymbol{\Psi}$	$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_d \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{K}}_z + \boldsymbol{\Phi}_d^T \boldsymbol{K}_d \boldsymbol{\Phi}_d & -\boldsymbol{\Phi}_d^T \boldsymbol{K}_d \\ -\boldsymbol{K}_d \boldsymbol{\Phi}_d & \boldsymbol{K}_d \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{\Phi}_d & \boldsymbol{I} \end{bmatrix}$	$= \begin{bmatrix} \tilde{\boldsymbol{K}}_z & 0 \\ 0 & \boldsymbol{K}_d \end{bmatrix}$
$\tilde{\boldsymbol{R}}$	$= \boldsymbol{\Psi}^T \tilde{\boldsymbol{R}}_0$	$= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{\Phi}_d \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{R}}_z \\ 0 \end{bmatrix}$	$= \begin{bmatrix} \tilde{\boldsymbol{R}}_z \\ 0 \end{bmatrix}$

Table 1.5: Table of mass, damping, stiffness and load matrices

Thus obtaining

$$\tilde{\boldsymbol{M}} \ddot{\boldsymbol{\eta}}_{rel}(t) + \tilde{\boldsymbol{C}} \dot{\boldsymbol{\eta}}_{rel}(t) + \tilde{\boldsymbol{K}} \boldsymbol{\eta}_{rel}(t) = \tilde{\boldsymbol{R}}(t) \tag{1.104}$$

By splitting the modal mass matrix

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} \tilde{\boldsymbol{M}}_z & 0 \\ 0 & \boldsymbol{M}_d \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{M}}_z \hat{\boldsymbol{D}} \boldsymbol{\Phi}_d & \tilde{\boldsymbol{M}}_z \hat{\boldsymbol{D}} \\ \boldsymbol{M}_d \boldsymbol{\Phi}_d & 0 \end{bmatrix} \tag{1.105}$$

where

$$\hat{\boldsymbol{D}} = \tilde{\boldsymbol{M}}_z^{-1} \boldsymbol{\Phi}_d^T \boldsymbol{M}_d \tag{1.106}$$

After pre-multiplying the entire equation by $\tilde{\boldsymbol{K}}^{-1}$, the Fourier transform is taken

$$\begin{bmatrix} \boldsymbol{\eta}_z \\ \boldsymbol{\Delta\eta}_d \end{bmatrix} = \sum_{\omega} \begin{bmatrix} \boldsymbol{a}_{\eta_z}(\omega) \\ \boldsymbol{a}_{\Delta\eta_d} \end{bmatrix} e^{i\omega t} \tag{1.107}$$

and

$$\begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \tilde{\boldsymbol{R}}_z \\ 0 \end{bmatrix} = \sum_{\omega} \begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \boldsymbol{a}_{\tilde{R}_z}(\omega) \\ 0 \end{bmatrix} e^{i\omega t} \tag{1.108}$$

Which gives

$$\begin{bmatrix} \boldsymbol{a}_{\eta_z}(\omega) \\ \boldsymbol{a}_{\Delta\eta_d}(\omega) \end{bmatrix} = \hat{\boldsymbol{H}}(\omega) \cdot \begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \boldsymbol{a}_{\tilde{R}_z}(\omega) \\ 0 \end{bmatrix} \tag{1.109}$$

where

$$\hat{\boldsymbol{H}}^{-1}(\omega) = \begin{bmatrix} (\boldsymbol{I} - \hat{\boldsymbol{\omega}}_z^2(\boldsymbol{I} + \hat{\boldsymbol{D}} \boldsymbol{\Phi}_d) + 2i\hat{\boldsymbol{\omega}}_z \boldsymbol{\xi}_d) & -\hat{\boldsymbol{\omega}}_z^2 \hat{\boldsymbol{D}} \\ -\hat{\boldsymbol{\omega}}_d^2 \boldsymbol{\Phi}_d & (\boldsymbol{I} - \hat{\boldsymbol{\omega}}_d^2 + 2i\hat{\boldsymbol{\omega}}_d \boldsymbol{\xi}_d) \end{bmatrix} \tag{1.110}$$

where $\hat{\boldsymbol{\omega}}_z = diag[\omega/\omega_{z_n}]$, $\hat{\boldsymbol{\omega}}_d = diag[\omega/\omega_j]$, $\boldsymbol{\xi}_z = diag[\xi_{z_n}]$ and $\boldsymbol{\xi}_d = diag[\xi_j]$, $\boldsymbol{I}$ is the identity-matrix with size according to the size of $\hat{\boldsymbol{H}}^{-1}$

$$\text{Size of} \quad \hat{\boldsymbol{H}}^{-1} = \begin{bmatrix} N_{mod}\text{x}N_{mod} & N_{mod}\text{x}N_j \\ N_j\text{x}N_{mod} & N_j\text{x}N_j \end{bmatrix} \tag{1.111}$$

To find the displacement response the modal coordinates must be multiplied by a $\boldsymbol{\Psi}$- matrix

$$\begin{bmatrix} r_z(x_r,t) \\ \boldsymbol{\Delta r}_d(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_z(x_r) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\Delta\eta}_d(t) \end{bmatrix} = \boldsymbol{\Psi}_r(x_r) \begin{bmatrix} \boldsymbol{\eta}_z(t) \\ \boldsymbol{\Delta\eta}_d(t) \end{bmatrix} \tag{1.112}$$

Thus

$$\begin{bmatrix} a_{r_z}(x_r,\omega) \\ \boldsymbol{a}_{\Delta r}(\omega) \end{bmatrix} = \boldsymbol{\Psi}_r(x_r) \begin{bmatrix} \boldsymbol{a}_{\eta_z}(\omega) \\ \boldsymbol{a}_{\Delta\eta_d}(\omega) \end{bmatrix} \tag{1.113}$$

1.6.3 From Frequency Response Function and Fourier amplitude to Spectral Density

The approach to go from frequency response function and Fourier amplitude to Spectral density is similar for all 3 cases; single degree of freedom system with a TMD, continuous system with a TMD (single component), and a continuous system with multi mode, multi components. In general

$$\boldsymbol{S}_{variable} = \lim_{T\to\infty} \frac{1}{\pi T} \boldsymbol{a}^*_{variable} \boldsymbol{a}^T_{variable} \tag{1.114}$$

where the $\boldsymbol{a}$ is the Fourier amplitude from the Fourier transform. The subscript $variable$ can be for instance displacement r_z or modal coordinate η. The asterisk (*) means that it is the complex conjugated. The T in the limit when T goes to infinity is the total length of the time series, while it means transposed when it is in the power of T.

Single degree of freedom

$$\begin{aligned} \boldsymbol{S}_r &= \lim_{T\to\infty} \frac{1}{\pi T} \boldsymbol{a}^*_r \boldsymbol{a}^T_r = \lim_{T\to\infty} \frac{1}{\pi T} \begin{bmatrix} a^*_r a_r & a^*_r a_{\Delta r} \\ a^*_{\Delta r} a_r & a^*_{\Delta r} a_{\Delta r} \end{bmatrix} \\ &= \lim_{T\to\infty} \frac{1}{\pi T} \left(\hat{\boldsymbol{H}}(\omega) \begin{bmatrix} a_{R_1}(\omega)/K_1 \\ 0 \end{bmatrix} \right)^* \left(\hat{\boldsymbol{H}}(\omega) \begin{bmatrix} a_{R_1}(\omega)/K_1 \\ 0 \end{bmatrix} \right)^T \\ &= \frac{\lim_{T\to\infty} \frac{1}{\pi T} a^*_{R_1}(\omega) a_{R_1}(\omega)}{K_1^2} \cdot \begin{bmatrix} |\hat{H}_{11}(\hat{\omega})|^2 & \hat{H}^*_{11}(\hat{\omega})\hat{H}_{21}(\hat{\omega}) \\ \hat{H}^*_{21}(\hat{\omega})\hat{H}_{11}(\hat{\omega}) & |\hat{H}_{21}(\hat{\omega})|^2 \end{bmatrix} \\ &= \frac{S_{R_1}(\omega)}{K_1^2} \cdot \begin{bmatrix} |\hat{H}_{11}(\hat{\omega})|^2 & \hat{H}^*_{11}(\hat{\omega})\hat{H}_{21}(\hat{\omega}) \\ \hat{H}^*_{21}(\hat{\omega})\hat{H}_{11}(\hat{\omega}) & |\hat{H}_{21}(\hat{\omega})|^2 \end{bmatrix} \\ &= \begin{bmatrix} S_{rr} & S_{r\Delta r} \\ S_{\Delta rr} & S_{\Delta r\Delta r} \end{bmatrix} \end{aligned} \tag{1.115}$$

where $S_{R_1}(\omega) = \lim\limits_{T\to\infty} \frac{1}{\pi T} a^*_{R_1}(\omega) a_{R_1}(\omega)$. is the spectral density of the load acting on the main system. The $\hat{\boldsymbol{H}}$ matrix is for a single degree of freedom system with a TMD defined in equations 1.45 and 1.46.

Continuous system with a TMD (single component)

$$\begin{aligned}
\boldsymbol{S}_{r_z}(x,\omega) &= \lim_{T\to\infty} \frac{1}{\pi T} \left(\begin{bmatrix} a_{r_z}(x,\omega) \\ a_{\Delta r_1}(\omega) \end{bmatrix}^* \begin{bmatrix} a_{r_z}(x,\omega) \\ a_{\Delta r_1}(\omega) \end{bmatrix}^T \right) \\
&= \lim_{T\to\infty} \frac{1}{\pi T} \left(\boldsymbol{\Psi}_0 \begin{bmatrix} a_{\eta_z}(\omega) \\ a_{\Delta\eta_1}(\omega) \end{bmatrix} \right)^* \left(\boldsymbol{\Psi}_0 \begin{bmatrix} a_{\eta_z}(\omega) \\ a_{\Delta\eta_1}(\omega) \end{bmatrix} \right)^T \\
&= \boldsymbol{\Psi}_0 \boldsymbol{S}_{\eta_z}(\omega) \boldsymbol{\Psi}_0^T
\end{aligned} \tag{1.116}$$

where

$$\boldsymbol{\Psi}_0 = \begin{bmatrix} \phi_z(x) & 0 \\ 0 & 1 \end{bmatrix} \tag{1.117}$$

and

$$\begin{aligned}
\boldsymbol{S}_{\eta_z}(\omega) &= \lim_{T\to\infty} \frac{1}{\pi T} \boldsymbol{a}^*_{eta} \boldsymbol{a}^T_{eta} = \lim_{T\to\infty} \frac{1}{\pi T} \begin{bmatrix} a^*_\eta a_\eta & a^*_\eta a_{\Delta\eta} \\ a^*_{\Delta\eta} a_\eta & a^*_{\Delta\eta} a_{\Delta\eta} \end{bmatrix} \\
&= \lim_{T\to\infty} \frac{1}{\pi T} \left(\hat{\boldsymbol{H}}(\omega) \begin{bmatrix} a_{\tilde{R}_z}(\omega)/\tilde{K}_z \\ 0 \end{bmatrix} \right)^* \left(\hat{\boldsymbol{H}}(\omega) \begin{bmatrix} a_{\tilde{R}_z}(\omega)/\tilde{K}_z \\ 0 \end{bmatrix} \right)^T \\
&= \frac{S_{\tilde{R}_z}(\omega)}{\tilde{K}_z^2} \begin{bmatrix} |\hat{H}_{zz}(\hat{\omega})|^2 & \hat{H}^*_{zz}(\hat{\omega})\hat{H}_{1z}(\hat{\omega}) \\ \hat{H}^*_{1z}(\hat{\omega})\hat{H}_{zz}(\hat{\omega}) & |\hat{H}_{1z}(\hat{\omega})|^2 \end{bmatrix} \\
&= \begin{bmatrix} S_{\eta_z\eta_z} & S_{\eta_z\Delta\eta} \\ S_{\Delta\eta\eta_z} & S_{\Delta\eta\Delta\eta} \end{bmatrix}
\end{aligned} \tag{1.118}$$

Where the $\hat{\boldsymbol{H}}$ for a continuous system with a single component is given in equations 1.74 and 1.75.
Thus $\boldsymbol{S}_{r_z}$ is given by

$$\boldsymbol{S}_{r_z} = \frac{S_{\tilde{R}_z}(\omega)}{\tilde{K}_z^2} \begin{bmatrix} \phi_z^2(x)|\hat{H}_{zz}(\hat{\omega})|^2 & \phi_z(x)\hat{H}^*_{zz}(\hat{\omega})\hat{H}_{1z}(\hat{\omega}) \\ \phi_z(x)\hat{H}^*_{1z}(\hat{\omega})\hat{H}_{zz}(\hat{\omega}) & |\hat{H}_{1z}(\hat{\omega})|^2 \end{bmatrix} = \begin{bmatrix} S_{r_z r_z} & S_{r_z\Delta r_1} \\ S_{\Delta r_1 r_z} & S_{\Delta r_1\Delta r_1} \end{bmatrix} \tag{1.119}$$

where $S_{\tilde{R}_z}$ is the spectral density of the modal load on the bridge girder.

$$S_{\tilde{R}_z}(\omega) = \lim_{T\to\infty} \frac{1}{\pi T} a^*_{\tilde{R}_z}(\omega) a^T_{\tilde{R}_z}(\omega) = \int_L \int_L \phi_z(x_a) \cdot \phi_z(x_b) \cdot S_{q_z}(\omega, \Delta x) dx_a dx_b \tag{1.120}$$

where $\Delta x = |x_a - x_b|$, where x_a and x_b is arbitrary positions.

Strømmen has assumed that the coherence length λD of the vortexes is small compared to the length of the bridge, and that the modal load can by sufficient accuracy be defined as

$$S_{\tilde{R}_z}(\omega) \approx 2\lambda D S_{q_z}(\omega) \int_0^L \phi_z^2(x)dx = \frac{\sigma_{\tilde{R}_z}^2}{\sqrt{\pi} b_z \omega_s} \exp\left[-\left(\frac{1-\omega/\omega_s}{b_z} \right)^2 \right] \tag{1.121}$$

where the cross spectral density is

$$\begin{aligned} S_{q_z}(\omega) &= \frac{\sigma_{q_z}^2}{\sqrt{\pi}\omega_s b_z} \exp\left[-\left(\frac{1-\omega/\omega_s}{b_z} \right)^2 \right] \\ \sigma_{\tilde{R}_z}^2 &= 2\lambda D \sigma_{q_z}^2 \int_0^L \phi_{z_2}^2(x)dx \\ \sigma_{q_z} &= \frac{1}{2}\rho V^2 B \hat{\sigma}_{q_z} \end{aligned} \tag{1.122}$$

and where $\sigma_{q_z}^2$ is the variance of the cross sectional vortex shedding load,$\sigma_{\hat{q}_z}^2$ is the normalized version of $\sigma_{q_z}^2$, b_z is the band width, ω_s is the vortex shedding frequency ($\omega_s = 2\pi V St/D$), V is the mean wind velocity, St the relevant Strouhal number, λ is the non-dimensional coherence length scale of vortices, D is the cross sectional depth, and L the length of the bridge.

Continuous system with multi mode, multi components

$$\begin{aligned} \boldsymbol{S}_r(x_r,\omega) &= \lim_{T\to\infty} \frac{1}{\pi T} \begin{bmatrix} a_{r_z} \\ \boldsymbol{a}_{\Delta r} \end{bmatrix}^* \begin{bmatrix} a_{r_z} \\ \boldsymbol{a}_{\Delta r} \end{bmatrix}^T \\ &= \lim_{T\to\infty} \frac{1}{\pi T} \left(\boldsymbol{\Psi}_r(x_r)\hat{\boldsymbol{H}}(\omega) \cdot \begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \boldsymbol{a}_{\tilde{R}_z}(\omega) \\ \boldsymbol{0} \end{bmatrix} \right)^* \left(\boldsymbol{\Psi}_r(x_r)\hat{\boldsymbol{H}}(\omega) \cdot \begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \boldsymbol{a}_{\tilde{R}_z}(\omega) \\ \boldsymbol{0} \end{bmatrix} \right)^T \\ &= \boldsymbol{\Psi}_r(x_r) \cdot \hat{\boldsymbol{H}}^*(\omega) \cdot \hat{\boldsymbol{S}}_{\tilde{R}}(\omega) \cdot \hat{\boldsymbol{H}}^T(\omega) \cdot \boldsymbol{\Psi}_r^T(x_r) \end{aligned} \tag{1.123}$$

where $\hat{\boldsymbol{H}}$ for the multi mode, multi component case is defined in equation 1.110. and where

$$\boldsymbol{\Psi}_r = \begin{bmatrix} \boldsymbol{\Phi}_z(x_r) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{1.124}$$

where $\boldsymbol{I}$ is the N_j by N_j identity matrix, where N_j is the number of tuned mass dampers. $\boldsymbol{\Phi}_z(x_r)$ is a vector of the mode shapes at the position of the tuned mass dampers.

$$\boldsymbol{\Phi}_z(x_r) = \begin{bmatrix} \phi_{z1}(x_r) & ... & \phi_{zN_{mod}}(x_r) \end{bmatrix} \tag{1.125}$$

The spectral density of the modal load is defined as

$$\hat{\boldsymbol{S}}_{\tilde{R}}(\omega) = \begin{bmatrix} \tilde{\boldsymbol{K}}_z^{-1} \boldsymbol{S}_{\tilde{R}_z}(\omega) (\tilde{\boldsymbol{K}}_z^{-1})^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{1.126}$$

where

$$\boldsymbol{S}_{\tilde{R}_z}(\omega) = \lim_{T\to\infty} \frac{1}{\pi T}\boldsymbol{a}^*_{\tilde{R}_z}(\omega)\cdot \boldsymbol{a}^T_{\tilde{R}_z}(\omega) = \lim_{T\to\infty} \frac{1}{\pi T} \begin{bmatrix} a_{\tilde{R}_{z1}} \\ \vdots \\ a_{\tilde{R}_{zN_{mod}}} \end{bmatrix}^* \begin{bmatrix} a_{\tilde{R}_{z1}} \\ \vdots \\ a_{\tilde{R}_{zN_{mod}}} \end{bmatrix}^T$$
$$= \begin{bmatrix} S_{\tilde{R}_{z1}\tilde{R}_{z1}} & \cdots & S_{\tilde{R}_{z1}\tilde{R}_{zN_{mod}}} \\ \vdots & \ddots & \vdots \\ S_{\tilde{R}_{zN_{mod}}\tilde{R}_{z1}} & \cdots & S_{\tilde{R}_{zN_{mod}}\tilde{R}_{zN_{mod}}} \end{bmatrix} \tag{1.127}$$

where the on-diagonal terms would, when integrated, give the variance of the vortex shedding forces at resonance of the given mode. The off-diagonal terms would give the covariance of the vortex shedding forces at different resonance velocities. The covariance is often assumed negligible, and the on-diagonal terms is given by

$$\begin{bmatrix} S_{\tilde{R}_{z1}\tilde{R}_{z1}} \\ \vdots \\ S_{\tilde{R}_{zN_{mod}}\tilde{R}_{zN_{mod}}} \end{bmatrix} = 2\lambda D S_{q_z} \int_0^L \begin{bmatrix} \phi^2_{z1} \\ \vdots \\ \phi^2_{zN_{mod}} \end{bmatrix} dx \tag{1.128}$$

where the cross spectral density S_{q_z} is defined in equation 1.122.

Written out, for a system with 3 TMDs, (TMD A, TMD B and TMD C), the spectral density is

$$\boldsymbol{S}_r(x_r,\omega) = \begin{bmatrix} S_{r_z} & S_{r_z\Delta r_A} & S_{r_z\Delta r_B} & S_{r_z\Delta r_C} \\ & S_{\Delta r_A} & S_{\Delta r_A\Delta r_B} & S_{\Delta r_A\Delta r_C} \\ & & S_{\Delta r_B} & S_{\Delta r_B\Delta r_C} \\ Sym. & & & S_{\Delta r_C} \end{bmatrix} \tag{1.129}$$

1.7 Tuning of the TMD

In chapter 1 so far the tuned mass dampers (TMD's) have had general characteristics; mass-ratio μ_d, damping $\xi_d$and frequency ω_d. When tuning a mass damper, these characteristics is chosen such that the damper is effective to damp displacement from a certain eigen-mode. The principle of the TMD is Newtons second law, and the transformation between forces and acceleration. The bridge girder and the TMD are interacting, as the bridge girder is accelerating the mass of the TMD, and the TMD's acceleration transfer forces to the bridge girder. The TMD therefore need to be set in motion by the bridge girder as the girder start to move with the frequency of the mode it should damp out. If the eigen-frequency to the TMD is equal or close to the eigen-frequency of the mode under consideration, it will be excited when the bridge get excited, how large forces to get transferred depends on the TMD's mass-ratio and damping. The damping must not be too big, such that the TMD's acceleration disappear, neither to small such that small forces is transferred. The effectiveness of the TMD is in general better, the higher mass- ratio. It must however be inside a reasonable value regarding the practical placement on the bridge, the costs and the assumption made in the theory that the eigenvalue analysis is done without considering the mass of the TMDs ($M_z >> M_{TMD}$). Usually μ is in the range 0.005 to 0.05 [Strømmen, 2013].

Since the dampers are to be placed at or inside the bridge, and often have limited space available, the motion of the dampers must not exceed the given space limitations. It is therefore also important not only to estimate the effectiveness of the tuning of the damper on the bridge girder motion, but also estimate the motion of the TMD itself.

Two models for optimum choice of damping and frequency of the TMD, after the mass-ratio is decided is the models by Den Hartog and R. Luft [Strømmen, 2013].

Den Hartog recommends:

$$\begin{aligned} \omega_d &= \frac{\omega_z}{1+\mu} \\ \xi_d &= \sqrt{\frac{3\mu}{8(1+\mu)^3}} \end{aligned} \tag{1.130}$$

R. Luft recommends:

$$\begin{aligned} \omega_d &= \frac{\omega_z}{\sqrt{1+\frac{3\mu}{2}}} \\ \xi_d &= \sqrt{\frac{\mu}{4}\left(1-\frac{3\mu}{4}\right)} \end{aligned} \tag{1.131}$$

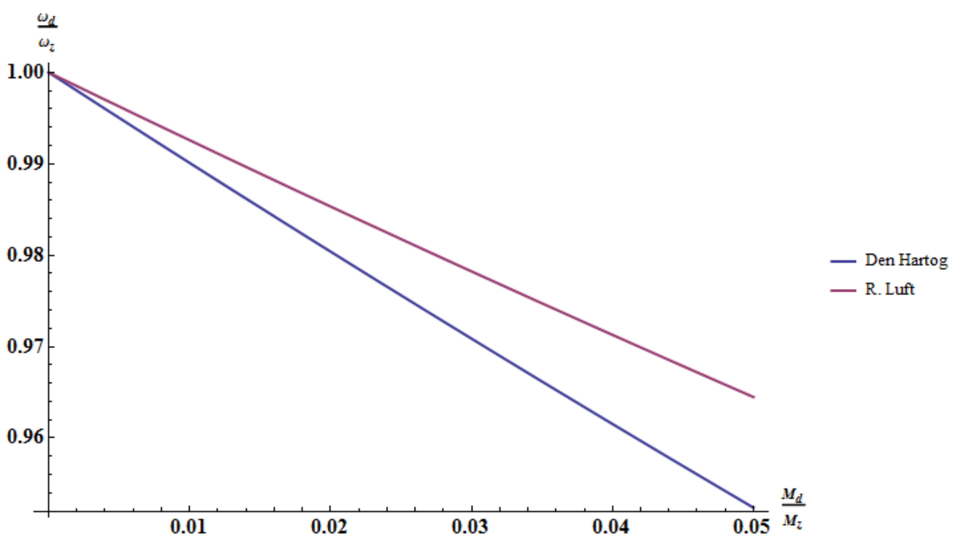

Figure 1.3: Frequency ratio plotted against mass ratio for Den Hartog and R. Luft recommendations

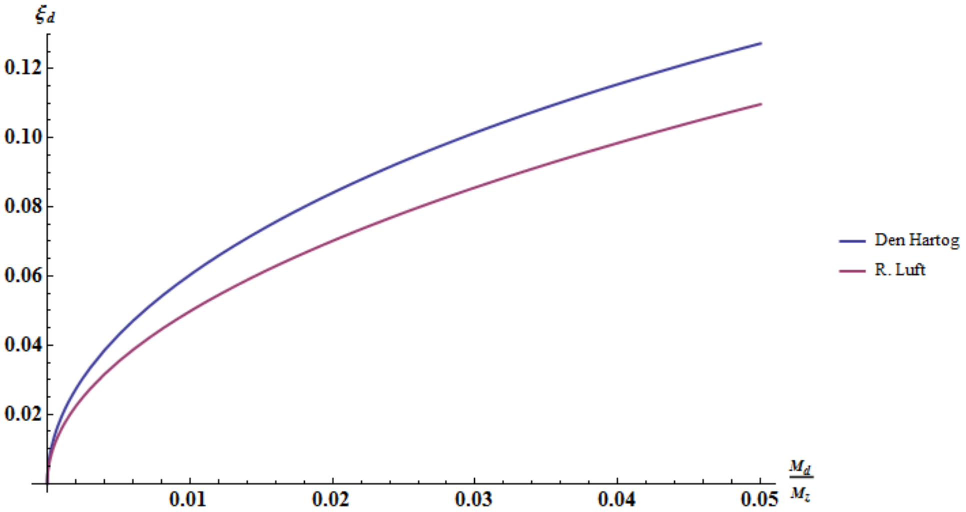

Figure 1.4: Damping plotted against mass ratio for Den Hartog and R. Luft recommendations

Chapter 2

Experiment

2.1 Developing a Matlab script

The theory presented in chapter 1 is used in developing a Matlab script for determine the response of the main girder due to vertical mode shape induced vortex shedding of a suspension bridge. The purpose of making the Matlab script has been to better understand the theory, by using the theory and by giving some results that has practical and physical meaning. The purpose has not been to make a perfect Matlab script for later use, nor to be able to give precise recommendations regarding vortex induced vibrations in a particular case.

The principle in the Matlab script is the theory presented in chapter 1. However, the theory is presented with use of continuous variables like the frequency ω, while the Matlab script operates in a numerical format. When making the numerical ω- axis it is important to not only make it fine enough, but to place the ω-s wisely. The way this is done in the Matlab script is, as illustrated in figure 2.1, to let every eigen-frequencies be represented in the numerical ω-vector, and divide the interval between the eigen-frequencies into narrow-spaced close to the eigen-frequencies and more scattered frequencies elsewhere. If the eigen-frequencies is close together, the narrow-spaced ω-s will dominate.

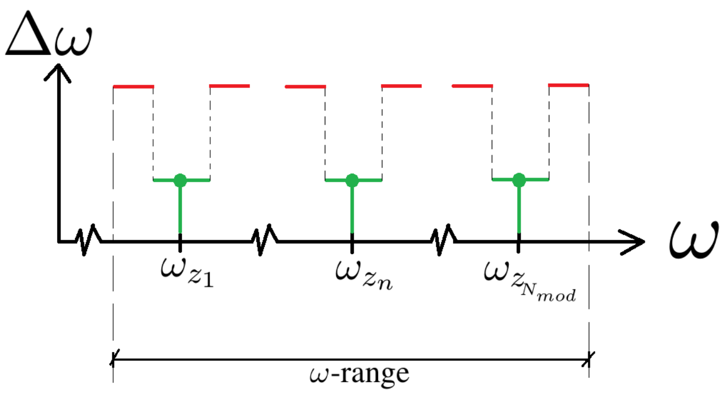

Figure 2.1: Prinsiple of omega-axis

When making a time simulation a phase ($0 \leq \phi \leq 2\pi$) is added for each frequency ω. Thus one phase angle is needed for each ω value. If the phase vector or diagram is known, time series could be reproduced exactly similar to the original. Often the phase is unknown and of little interest since it is only dependent on when wind recordings are done, see section 1.2.6, and when doing time series simulations random phase angles is often used. Time series with different random phase angles is different, but their statistical properties is still the same. Since the phase angles in principle make the response from one time series simulation different from another, several time series simulations must be done to get an idea of both the maximum value over several simulations and spreading of the response. The standard deviation of the process is constant and could be found exactly from integrating the spectral density, however the standard deviation from each simulation of the process does vary slightly. It is important to distinguish between a process and the simulation of a process.

In the Matlab script the built in function ”rand” is used to make a random number between 0 and 1, and the random number is multiplied by 2π to give a random phase. A limitation by the built in function is that it is not completely random, it is just a large list of ”random” numbers that repeat itself when it is at the end. However for the purpose of making 10 minutes time series it is in this case considered sufficient random to be used. To get the same load at the bridge girder without TMD, and the girder with TMD the Matlab script make a random matrix (number of simulations x number of frequencies), that is used for all time domain simulations. The random vector plotted against ω for simulation nr 1, is shown in figure 2.2. For low values of ω, there are few values of ”rand” because $\Delta\omega$ is bigger.

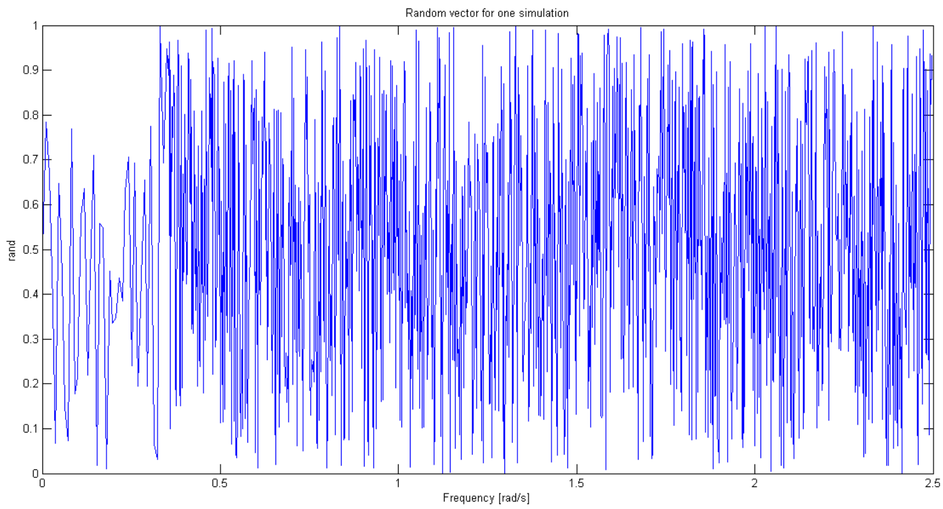

Figure 2.2: The random vector from one simulation

2.1.1 Developing a simplified expression for the bridge girder response

Usually when the response of the bridge girder is found by calculations based on the theory presented in chapter 1, it is done where the maximum response is, i.e. at the mode-shapes maximum. However, in the mission of understanding the theory, it has here been considered useful to be able to visualize the bridge girder response for the entire span. Th single point response could in principle be done everywhere along the bridge span, and it is done for the original system without TMD in the Matlab script developed. The single point response at each point is correct, but when the single point response in several points is combined it could give wrong response along the span. That is because single point response at two points where the response is equal in magnitude, but has opposite sign, could be equal. The sign problem could be easily avoided if only a single mode is considered, since the sign of the response at a point relatively to the response at another point then is known. When several modes, with different frequencies, is combined it is not an easy task to decide the sign of the combination of single point response. Therefore the response along the span is found by first finding the response in modal coordinates, and then multiply it by the mode shapes. That is:

$$r_z(x,t) = \mathbf{\Phi}(x)\boldsymbol{\eta}(t) = \sum_{n=1}^{N_{mod}} \phi_n \eta_n \tag{2.1}$$

To see if the η_n is statistical independent processes that could be found individual and than added together, the expression for the variance of the response is developed using an example with 2 modes:

$$\begin{aligned} \sigma_{r_z}^2 &= E[r_z^2] = E[(\phi_1\eta_1 + \phi_2\eta_2)^2] \\ &= E[(\phi_1\eta_1)^2 + 2(\phi_1\eta_1)(\phi_2\eta_2) + (\phi_2\eta_2)^2] \\ &= \phi_1^2 E[\eta_1^2] + 2\phi_1\phi_2 E[\eta_1\eta_2] + \phi_2^2 E[\eta_2^2] \\ &\approx \phi_1^2\sigma_{\eta_1}^2 + \phi_2^2\sigma_{\eta_2}^2 \end{aligned} \tag{2.2}$$

Thus if $E[\eta_1\eta_2] = \sigma_{\eta_1\eta_2}$ is negligible, all the η_n, $n = 1, 2, ..., N_{mod}$ could be treated independently. The condition for $E[\eta_1\eta_2]$ to be negligible is when the spectral densities of η_1 and η_2 does not overlap, i.e. when the eigen-frequencies is not too close. For the Hardanger bridge some of the eigen-frequencies is quite close, especially for mode 3 and 4 where ω_{z_3} is 1.27 rad/s, while ω_{z_4} 1.36 rad/s. Also mode 1 and 2 is quite close in frequency, with ω_{z_1} and ω_{z_2} equal 0.71 and 0.9 rad/s respectively.

However, it is for now assumed that the frequencies are spread enough that an approximate solution of the bridge girder displacement along the span can be obtained. η_n is found by the same principle as r_z, by a time domain simulation:

$$\eta_n(t) = \sum_{k=1}^{N_\omega} \left[\sqrt{2S_{\eta_n}(\omega_k)\Delta\omega_j} \cos(\omega_j t + \psi_j) \right] \tag{2.3}$$

where N_ω is the length of the ω- vector, and where ψ is the random-vector. The response r_z can be than be found by a summation according to equation 2.1. Since the random vector has big influence on the time series produced, and since just a single random vector is used for for the single point r_z simulation, exactly the same random vector that is used for the r_z simulation must be used for all η_n to get the same time series. In the Matlab script the same ψ is used for all η_n, such that it is possible to compare the response with the one found by a single point r_z simulation. However, in general by the assumption that each η_n is independent, it is possible to use different random vectors for the different η_n, getting a different time series than finding the r_z directly from a simulation with $S_r(\omega)$, but still a correct time series.

To make a time domain simulation, as shown in equation 2.3, S_{η_n} must be found. Recalling the expression for $\boldsymbol{S}_r$

$$\boldsymbol{S}_r(x_r,\omega) = \mathbf{\Psi}_r(x_r) \cdot \hat{\boldsymbol{H}}^*(\omega) \cdot \hat{\boldsymbol{S}}_{\hat{R}}(\omega) \cdot \hat{\boldsymbol{H}}^T(\omega) \cdot \mathbf{\Psi}_r^T(x_r) \tag{2.4}$$

The expression for $\boldsymbol{S}_\eta$ is similar, but without the $\mathbf{\Psi}_r$ matrix

$$\boldsymbol{S}_\eta(\omega) = \hat{\boldsymbol{H}}^*(\omega) \cdot \hat{\boldsymbol{S}}_{\hat{R}}(\omega) \cdot \hat{\boldsymbol{H}}^T(\omega) \tag{2.5}$$

Written out, for a system with 3 TMDs, (TMD A, TMD B and TMD C), the expression for $\boldsymbol{S}_r$ is given in equation 1.129, while the expression for $\boldsymbol{S}_\eta$ is a $N_{mod} + N_j$ by $N_{mod} + N_j$ matrix, where N_{mod} is the number of eigen-modes considered, and N_j the number of tuned mass dampers installed.

$$\begin{aligned}\boldsymbol{S}_\eta(\omega) &= \begin{bmatrix} \boldsymbol{S}_{\eta_z\eta_z} & \boldsymbol{S}_{\eta_z\Delta\eta} \\ \boldsymbol{S}_{\Delta\eta\eta_z} & \boldsymbol{S}_{\Delta\eta\Delta\eta} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\boldsymbol{H}}_{11} & \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{21} & \hat{\boldsymbol{H}}_{22} \end{bmatrix}^* \begin{bmatrix} \hat{\boldsymbol{S}}_{\hat{R}_{11}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{H}}_{11} & \hat{\boldsymbol{H}}_{12} \\ \hat{\boldsymbol{H}}_{21} & \hat{\boldsymbol{H}}_{22} \end{bmatrix}^T \\ &= \begin{bmatrix} \hat{\boldsymbol{H}}_{11}^*\hat{\boldsymbol{S}}_{\hat{R}_{11}}\hat{\boldsymbol{H}}_{11}^T & \hat{\boldsymbol{H}}_{11}^*\hat{\boldsymbol{S}}_{\hat{R}_{11}}\hat{\boldsymbol{H}}_{21}^T \\ \hat{\boldsymbol{H}}_{21}^*\hat{\boldsymbol{S}}_{\hat{R}_{11}}\hat{\boldsymbol{H}}_{11}^T & \hat{\boldsymbol{H}}_{21}^*\hat{\boldsymbol{S}}_{\hat{R}_{11}}\boldsymbol{H}_{21}^T \end{bmatrix}\end{aligned} \tag{2.6}$$

where

$$\hat{\boldsymbol{H}}_{11} = \boldsymbol{I}_{N_{mod}} - \hat{\boldsymbol{\omega}}_z^2(\boldsymbol{I}_{N_{mod}} + \hat{\boldsymbol{D}}\boldsymbol{\Phi}_d) + 2i\hat{\boldsymbol{\omega}}_z\boldsymbol{\xi}_z \tag{2.7}$$

is a N_{mod} by N_{mod} matrix, that is diagonal if $\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d = \boldsymbol{0}$.
If $\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d$ is nonzero, the $\boldsymbol{S}_{\eta_z\eta_z}$ matrix given in equation 2.8 is non-diagonal.

$$\begin{aligned}\boldsymbol{S}_{\eta_z\eta_z} &= \hat{\boldsymbol{H}}_{11}^*\hat{\boldsymbol{S}}_{\hat{R}_{11}}\hat{\boldsymbol{H}}_{11}^T \\ &= \begin{bmatrix} S_{\eta1\eta1} & \dots & S_{\eta1\eta_n} & \dots & S_{\eta1\eta_{N_{mod}}} \\ \vdots & \ddots & & & \vdots \\ S_{\eta_n\eta1} & \dots & S_{\eta_n\eta_n} & \dots & S_{\eta_n\eta_{N_{mod}}} \\ \vdots & & & \ddots & \vdots \\ S_{\eta_{N_{mod}}\eta1} & \dots & S_{\eta_{N_{mod}}\eta_n} & \dots & S_{\eta_{N_{mod}}\eta_{N_{mod}}} \end{bmatrix}\end{aligned} \tag{2.8}$$

Where the diagonal terms $S_{\eta_n\eta_n}$ will produce the η_n in a time domain simulation, while the off-diagonal terms $S_{\eta_n\eta_m}$ will produce the $\eta_n\eta_m$, that earlier is assumed zero. To be able to say something about the consequences of assuming that $\eta_n\eta_m$ is zero when the eigen-frequencies is close together, a further look at the $\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d$ is done, since it is because of this term that $\boldsymbol{S}_{\eta_z\eta_z}$ is non-diagonal.

In the case with 3 TMDs tuned to mode 1, 2 and 1 respectively (2 modes considered), the $\boldsymbol{S}_{\eta_z\eta_z}$ looks like

$$\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d = \tilde{\boldsymbol{M}}_z^{-1}\boldsymbol{\Phi}_d^T\boldsymbol{M}_d\boldsymbol{\Phi}_d$$
$$= \begin{bmatrix} \tilde{M}_{z1}^{-1} & 0 \\ 0 & \tilde{M}_{z2}^{-1} \end{bmatrix} \begin{bmatrix} \phi_{z1}(x_A) & \phi_{z1}(x_B) & \phi_{z1}(x_C) \\ \phi_{z2}(x_A) & \phi_{z2}(x_B) & \phi_{z2}(x_C) \end{bmatrix} \cdot$$
$$\begin{bmatrix} \mu_A\tilde{M}_{z1} & 0 & 0 \\ 0 & \mu_B\tilde{M}_{z2} & 0 \\ 0 & 0 & \mu_C\tilde{M}_{z1} \end{bmatrix} \begin{bmatrix} \phi_{z1}(x_A) & \phi_{z2}(x_A) \\ \phi_{z1}(x_B) & \phi_{z2}(x_B) \\ \phi_{z1}(x_C) & \phi_{z2}(x_C) \end{bmatrix} \tag{2.9}$$
$$= \begin{bmatrix} \tilde{M}_{z1}^{-1}\phi_{z1}(x_A) & \tilde{M}_{z1}^{-1}\phi_{z1}(x_B) & \tilde{M}_{z1}^{-1}\phi_{z1}(x_C) \\ \tilde{M}_{z2}^{-1}\phi_{z2}(x_A) & \tilde{M}_{z2}^{-1}\phi_{z2}(x_B) & \tilde{M}_{z2}^{-1}\phi_{z2}(x_C) \end{bmatrix} \begin{bmatrix} \mu_A\tilde{M}_{z1}\phi_{z1}(x_A) & \mu_A\tilde{M}_{z1}\phi_{z2}(x_A) \\ \mu_B\tilde{M}_{z2}\phi_{z1}(x_B) & \mu_B\tilde{M}_{z2}\phi_{z2}(x_B) \\ \mu_C\tilde{M}_{z4}\phi_{z1}(x_C) & \mu_C\tilde{M}_{z4}\phi_{z2}(x_C) \end{bmatrix}$$

If $\mu_A = \mu_B = \mu_C = \mu$ and $\tilde{M}_{z1} = \tilde{M}_{z2} = \tilde{M}_{z3} = \tilde{M}_{z4} = \tilde{M}_z$ the equation simplifies to:

$$\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d = \mu \cdot \begin{bmatrix} \phi_{z1}^2(x_A) + \phi_{z1}^2(x_B) + \phi_{z1}^2(x_C) & \phi_{z1}(x_A)\phi_{z2}(x_A) + \phi_{z1}(x_B)\phi_{z2}(x_B) + \phi_{z1}(x_C)\phi_{z2}(x_C) \\ \phi_{z1}(x_A)\phi_{z2}(x_A) + \phi_{z1}(x_B)\phi_{z2}(x_B) + \phi_{z1}(x_C)\phi_{z2}(x_C) & \phi_{z2}^2(x_A) + \phi_{z2}^2(x_B) + \phi_{z2}^2(x_C) \end{bmatrix} \tag{2.10}$$

The normalized mode shapes ϕ_{zn} has maximum value of 1. The off- diagonal terms has multiplication of mode 1 and 2, and its reasonable that not both of them has maximum value in all TMD positions. However, if a conservative assumption is done that $\phi_{zn}(x_m) = 1$, the following is obtained:

$$\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d = \mu \cdot \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} \quad \text{or more general} \quad \hat{\boldsymbol{D}}\boldsymbol{\Phi}_d = \mu \cdot \Big[\mathbf{1}\Big]_{N_{mod}} \tag{2.11}$$

where $[\mathbf{1}]_{N_mod}$ is a N_{mod} by N_{mod} matrix with 1 in each cell.

The mass ratio μ is usually expected in the range 0.005 to 0.05 **??**. By setting these values for μ in the expression, gives the order of magnitude (assuming $\phi_{zn}(x_m) = 1$, n = 1,2,..., N_{mod}, m = 1,2,...,N_j) that is expected on the off-diagonal terms in $\hat{\boldsymbol{D}}\boldsymbol{\Phi}_d$. It gives roughly off-diagonal terms in the range 0.015 to 0.15, which is 1.5 to 15 % of the diagonal N_{mod} by N_{mod} identity matrix in equation 2.7. Thus if the eigen-frequencies of the main girder are close together the consequences of neglecting the off-diagonal terms will be bigger for bigger mass ratios μ. Since all modes is not equal 1 at every TMD position in reality, it is here assumed that the consequences of neglecting the off-diagonal terms is not to big to get a reasonable estimate of the bridge girder movement.

Chapter 3

Results and Analysis

3.1 Spectral density used as a measure of response

In the following, plot of spectral density is used to show the response of the bridge girder and TMD's. The phrase "response" is used both for the physical displacement and the frequency domain spectral density distribution. Another use of the phrase response, which is more frequent used in literature as a measure of the response, is the standard deviation. The standard deviation is the square root of the integral over the entire frequency range of the spectral density. Despite that the standard deviation and spectral density seems like two sides of the same issue, they still gives different information regarding the response. The standard deviation is somewhat more physical, as it gives information about a metric value of the deviation of the displacement response from the mean value, while the spectral density does not give any directly value of the displacement. On the other hand, the standard deviation is just a number, while the spectral density is a two dimensional function, thus information regarding the distribution of the response in frequency domain is lost, if just the standard deviation is given as the response. When comparing plots of different spectral densities it is also possible to roughly estimate which of them who has biggest standard deviation, since the variance is the area under the spectral density curve.

Since the purpose of the analysis in this case is mostly educational and an attempt to understand the theory behind the results, both spectral density plot and some plot of the change in standard deviation by changing the TMD's properties is given as the response.

3.2 Testing the Matlab script on the Hardanger Bridge

The Matlab script is tested on data from a real suspension bridge, the Hardanger bridge. The Hardanger bridge is 1310 m long and the data is given by Hjorth-Hansen and Strømmen [Hjorth-Hansen and Strømmen, 2001] and Vegdirektoratet. Only vortex induced vibrations on the vertical modes is considered.

As can be seen from figure 3.1, at position $x = L/4$ vertical mode 1 is at its maximum, while vertical mode 2 is zero. Vertical modes 3, 5 and 6 are also close to its maximum at $x = L/4$. In figure 3.2 the frequency domain response of the bridge girder at $x = L/4$ at different mean wind velocities is shown. The velocities V_{cr1} to $V_{crN_{mod}}$ is the velocities at resonance for each of the N_{mod} eigen-modes.

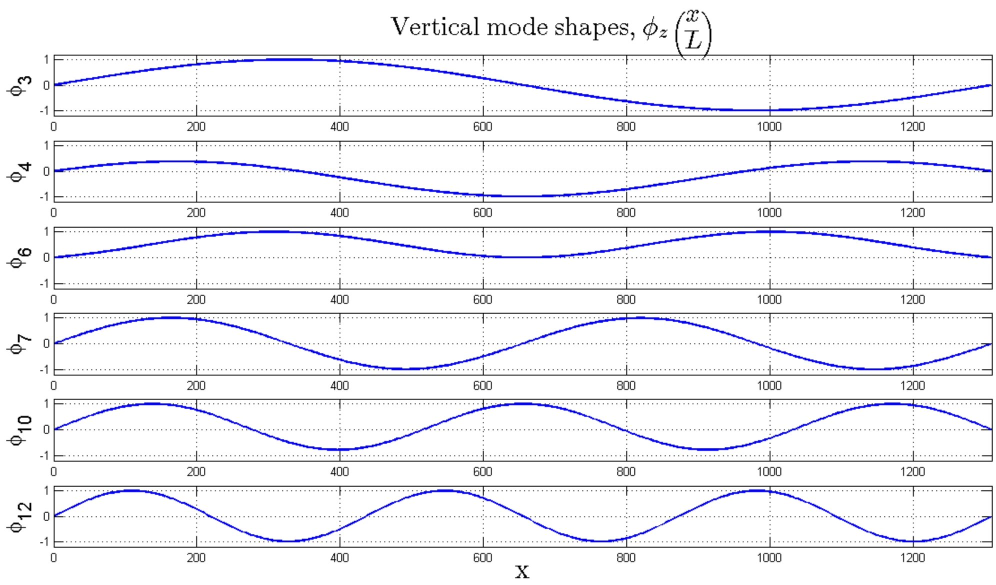

Figure 3.1: Vertical modes (normalized) of the Hardanger Bridge

$V = V_{cr1}$ gives the biggest response in terms of spectral density, S_{r_z}. The biggest response is for frequency equal the eigen-frequency of the first vertical mode, ω_{z1}, called the first resonance frequency. The other peak in figure 3.2 worth mention is for the third mode, ω_{z3}, which also is close to its maximum at the current x- position. The velocity which make biggest response is as expected the critical velocity for the first vertical mode V_{cr1}. Since ω_{z1} and ω_{z2} is quite close, the corresponding V_{cr1} and V_{cr2} is close enough that V_{cr2} also can activate the vertical mode 1. For the same reason, both V_{cr3} and V_{cr4} activates vertical mode 3.

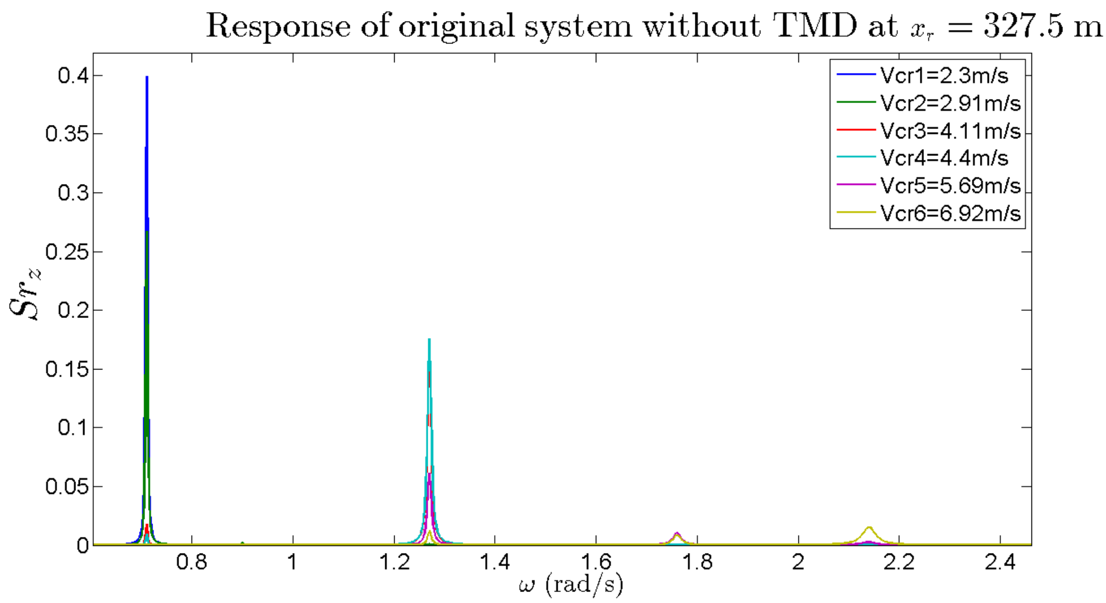

Figure 3.2: Original response of bridge girder at $x = 327.5$ m

From figure 3.2 it seems like V_{cr4} gives more resonance of mode 3, than V_{cr3}, which is not as expected. It is assumed that there is several reasons for this. Reasons which are to be discussed in next chapter. If the response of the bridge girder is considered to large, a TMD might make the response manageable. The first vertical mode contributes most to the spectral density, according to figure 3.2, and the first mode has its maximum value at $x = \frac{L}{4}$ (and $x = \frac{3L}{4}$), according to figure 3.1. Thus it would be reasonable to place a TMD tuned to damp out the first vertical mode at $x = \frac{L}{4} = 327.5m$ (or/and $x = \frac{3L}{4}$), as sketched in figure 3.3.

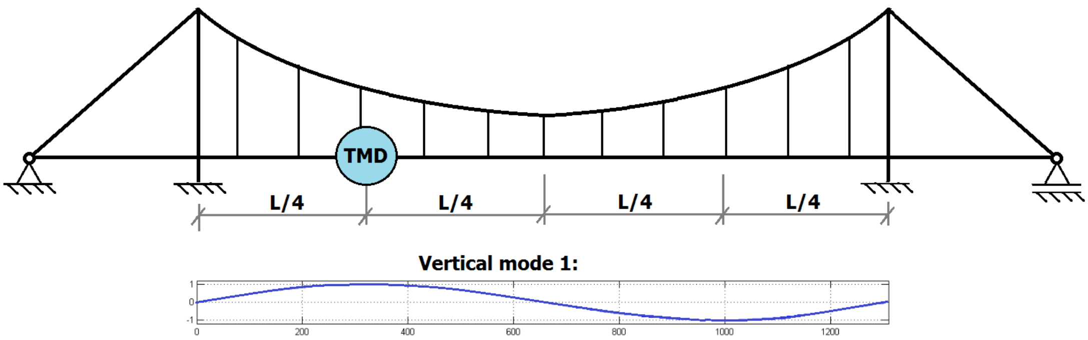

Figure 3.3: Sketch of the location of the TMD. (Not in scale)

The TMD is tuned to a frequency close to, but not equal the eigen-frequency of the first mode. In this case the first eigen-frequency is 0.710 rad/s, and 5 cases of the TMD's frequency is tested, while keeping the damping ratio constant, $\xi_d = 0.0430$. The frequency response function, FRF, when only the first mode is taken into consideration, is shown in figure 3.4. Keeping in mind that the vertical axis is logarithmic, there is no doubt that the TMD reduce the frequency response function $\hat{H}_{rz}$. Thus the response in terms of spectral density and displacement also will be reduced.

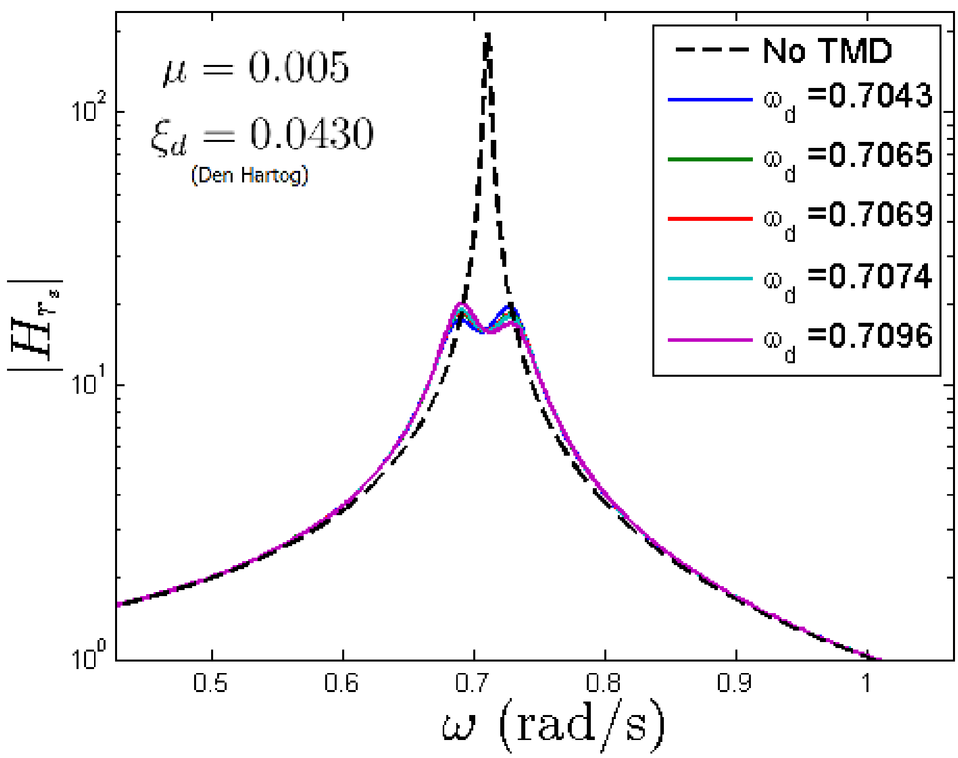

Figure 3.4: Frequency response function of main system with TMD at different values of frequency

To be able to see the difference between the different cases of frequency, a closer look at the FRF is shown in figure 3.5. Since the damping ratio is kept constant and equal the Den Hartog optimization, the R. Luft case shown is not truly an R. Luft optimization, but a combination of Den Hartog and R. Luft. However, it could be seen that the frequency optimization of Den Hartog and R.Luft is quite close, but Den Hartogs optimization give peaks that is more at the same level than R. Luft. It is not a surprise that Den Hartogs optimization gives peaks at roughly the same level, since this was the idea behind the Den Hartogs optimization [Strømmen, 2010]. In general it looks like the area under the different curves is roughly the same, and that they all cross each other at roughly the same point for frequency just below $omega_{z1}$. It looks like the frequency of the TMD decide the skewness or the relative level of the two peaks in the frequency response function. Assuming that the modal load is at its maximum at $\omega = \omega_z$ and that it is symmetric about ω_z, the optimal choice of frequency is the one that gives peaks mostly at level. In this case, when only vertical mode 1 is considered, it is Den Hartogs choice of damping that is closest to give leveled peaks. However a slightly lower frequency than Den Hartog could give completely leveled peaks, as lowest case of frequency gives the highest peak on the opposite side than the other cases.

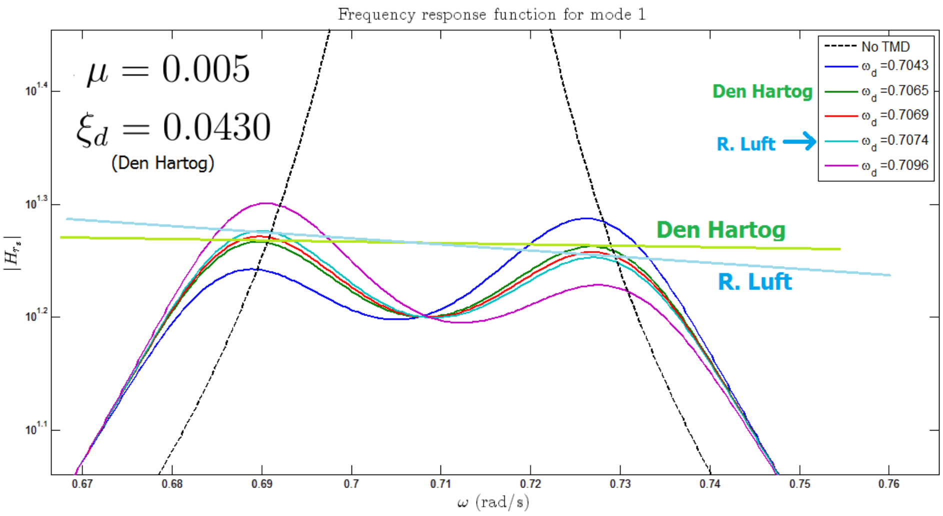

Figure 3.5: A closer view of the frequency response function of main system with TMD at different values of frequency

To see the effect different damping ratios has on the response when a single mode model is used, five different cases of damping is considered, while the frequency is kept constant according to the Den Hartog optimization. The response get reduced in all cases, roughly by the same as in the case of different frequencies earlier shown in figure 3.4. A closer look at the results, to see the differences between the different values of damping are shown in figure 3.6. While all the curves of different frequencies were crossing in a single point, the curves of different damping ratios are crossing in two separate points; one at a lower frequecy than ω_{z1}, the other at a higher frequency. That the point at a higher frequency than ω_{z1} is slightly lower in response, is assumed to come from the frequency ω_d used, i.e. the Den Hartog optimization of frequency. Thus it seems like the damping ratio has no affect on the skewness, i.e. relative height of the peaks, but rather the height of the peaks and how much the response is reduced at the eigen-frequency ω_{z1}. The tendency is that lower damping ratios gives most reduction of the frequency response function at and close around ω_{z1}, while it gives higher response for lower and higher frequencies. For higher damping ratios the frequency response function is more smooth with higher response at and around the eigen-frequency, but lower peaks than the more light damped cases.

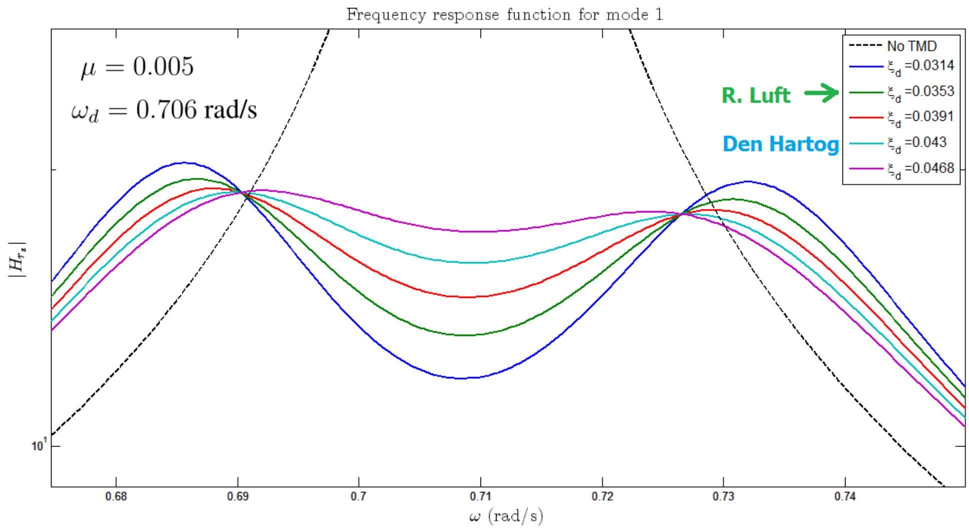

Figure 3.6: A closer view of the frequency response function of main system with TMD at different values of damping

A TMD is placed at $x = \frac{L}{4}$ with mass ratio μ = 0.01 and damping ξ_d = 0.0603, tuned to damp out vertical mode 1 by setting ω_d = 0.7030. The TMD will turn the original response of the bridge shown in figure 3.2 into the response shown in figure 3.7, when 6 vertical modes is considered. As can be seen the response of the first vertical mode is, after the installation of the TMD, almost zero as compared to the original response. Mode 3 is also slightly damped, but not significant. The "Case 1" reference in the title, is used in the parametric tests not yet introduced, and have no meaning by now.

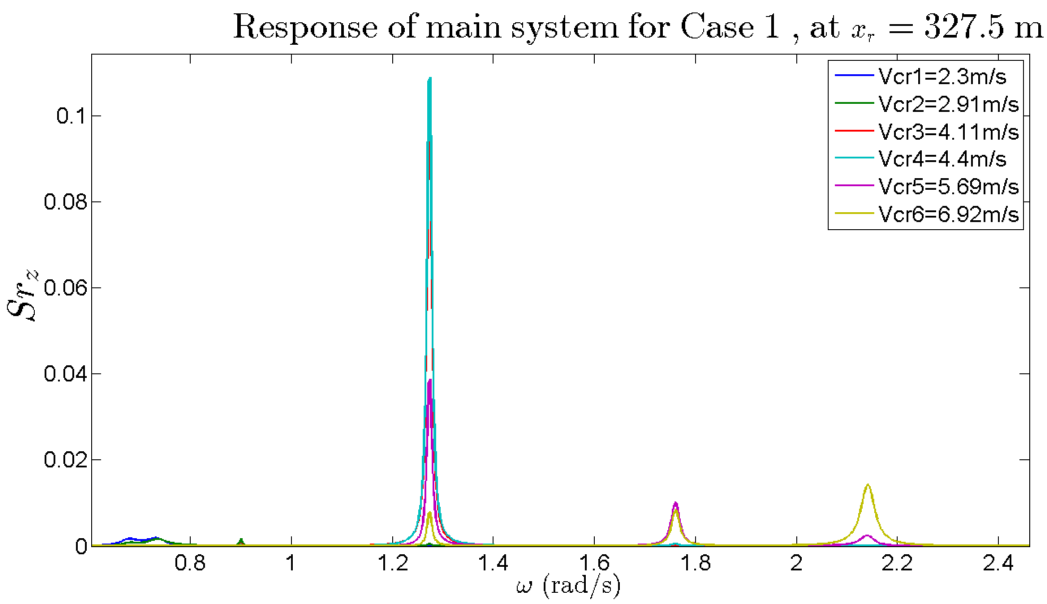

Figure 3.7: Response of main system with TMD

When the tuned mass damper damp out the response of the first mode, it means that the damper itself are in relative motion compared to the bridge at this frequency. The motion of the TMD is dependent not only on its eigen-frequency, but also on its damping- and mass-ratio and on the movement of the bridge because of the sum of all eigen-modes. It is not correct that the TMD's motion is always in the opposite direction of the bridge girder, but at many point in time it is. The TMD spectral density response for different modes critical wind velocities is shown in figure 3.8.

The figure illustrates several concepts. Firstly the response of the TMD is as expected biggest for the mode it is tuned to damp out. Because the first vertical mode also get excited by the critical wind velocity of the second mode, the TMD have relatively large response for V_{cr2}. The bridge also have some response for the third vertical mode, and the TMD does get some response despite that its eigen-frequency is relatively far away.

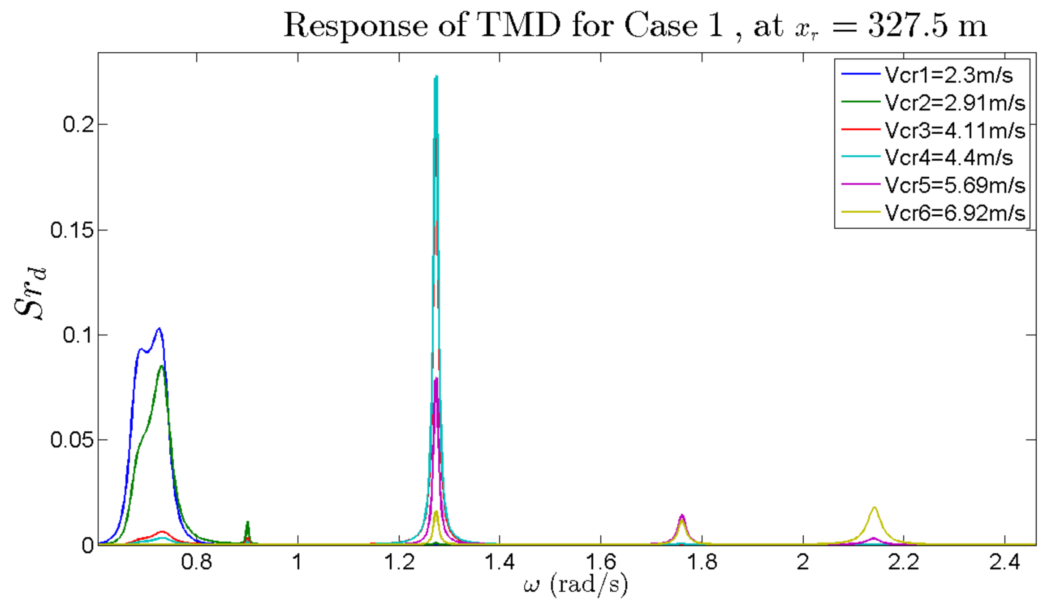

Figure 3.8: Response of TMD at x = 327 m for Case 1

3.2.1 Parametric study

To find out where to place the TMDs and roughly which μ, ξ_d and ω_d values to be used, for an effective response reduction, a parametric study is done. The purpose of the study is not to decide optimal values of the parameters, and certainly not to the accurate optimum. However, by testing several cases it is possible to discuss within which range each parameter should be for best effect of damping out vortex shedding induced vibrations.

Firstly 10 cases are done with one TMD at different locations. In the first five of them the TMD is tuned to damp out the first vertical mode, while it in the next five is tuned to damp out the second vertical mode. The parameters used in the first 10 cases is summarized in table 3.1.

1 Tuned mass damper			
Tuned to damp mode 1		Tuned to damp mode 2	
μ_d = 0.01		μ_d = 0.01	
ξ_d= 0.0603		ξ_d = 0.0603	
ω_d = 0.8911		ω_d = 0.7030	
Case nr.	Position of TMD x_d [m]	Case nr.	Position of TMD x_d [m]
1	327.5	6	327.5
2	400	7	400
3	500	8	500
4	600	9	600
5	655	10	655

Table 3.1: Properties and placement of TMDs in case 1 to 10

The spectral density for $x = \frac{L}{4}$ is already shown in figure 3.2, while the response with TMD and the response of the TMD itself is shown in figure 3.7 and 5.21 respectively. The original response at x = 655 m, is shown in figure 3.9. As can be seen the response of the second mode dominate totally. Also, the second mode get very little excited by the critical wind velocity for the firs mode, but it get excited by the critical wind velocities of mode 3 and 4.

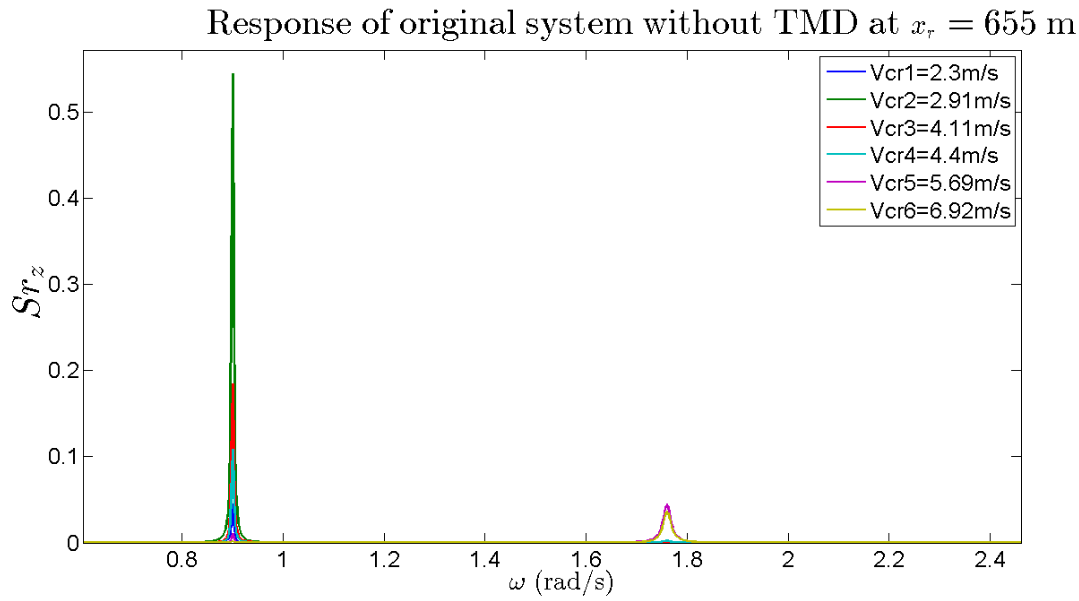

Figure 3.9: Response of original system without TMD at x = 655 m

When the same TMD as in case 1, i.e. the TMD is tuned to damp out the first vertical mode response, is moved to the mid-span at x = 655 m, the response in figure 3.10 is obtained. Since the TMD is not tuned to damp the second mode, the response from the second mode does not disappear completely. However, the peak of the spectral density is reduced from about 0.5 to about 0.06. Despite that the peak is a bit wider, the area under the curve is also reduced. The response of the second mode is biggest for resonance velocity for mode 2, the in descending order; mode 3,4,1,5. Also mode 5, has some response at x = 655, and by looking at the mode shapes in figure 3.1 this is not surprisingly, as it is the only two modes (of the first 6 vertical modes), that is nonzero at mid-span.

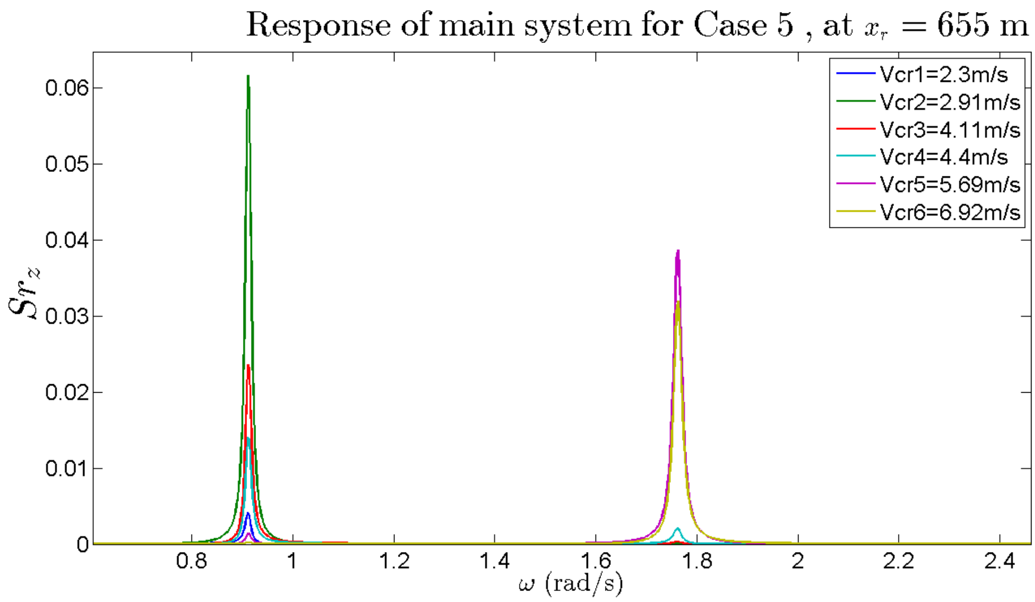

Figure 3.10: Response after TMD installation at x = 655 m for Case 5

The relative response for the TMD for case 5 is shown in figure 3.11, and looks almost identical to the response of the main system. The difference is that the response for the TMD is bigger than for the bridge girder.

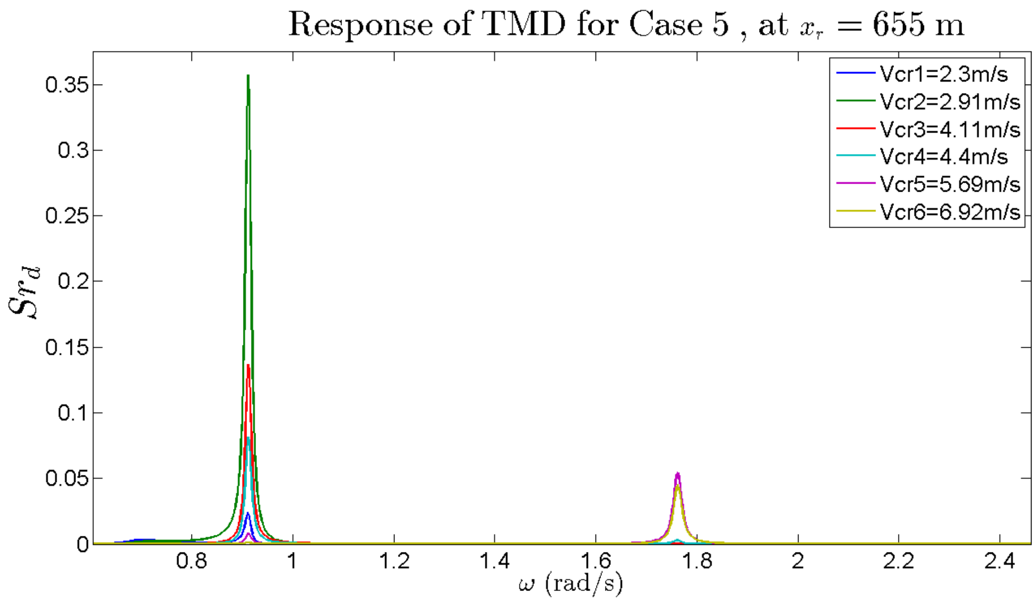

Figure 3.11: Response of TMD at x = 655 m for Case 5

For case 6 to 10 the TMD is tuned to damp out the second vertical mode. In case 6 it is installed at x = L/4, where the second vertical mode is almost zero. Therefore it is expected that the effect of the TMD is small. However the peak of the spectral density for the second eigen-frequency is reduced from about 0.4, as given in figure 3.2 to about 0.1 as given in figure 3.12. Noticing that the width of the peak has increased, could mean that the standard deviation has not changed that much. The other peak, for the third vertical mode, is also reduced, from about 0.15 to about 0.05.

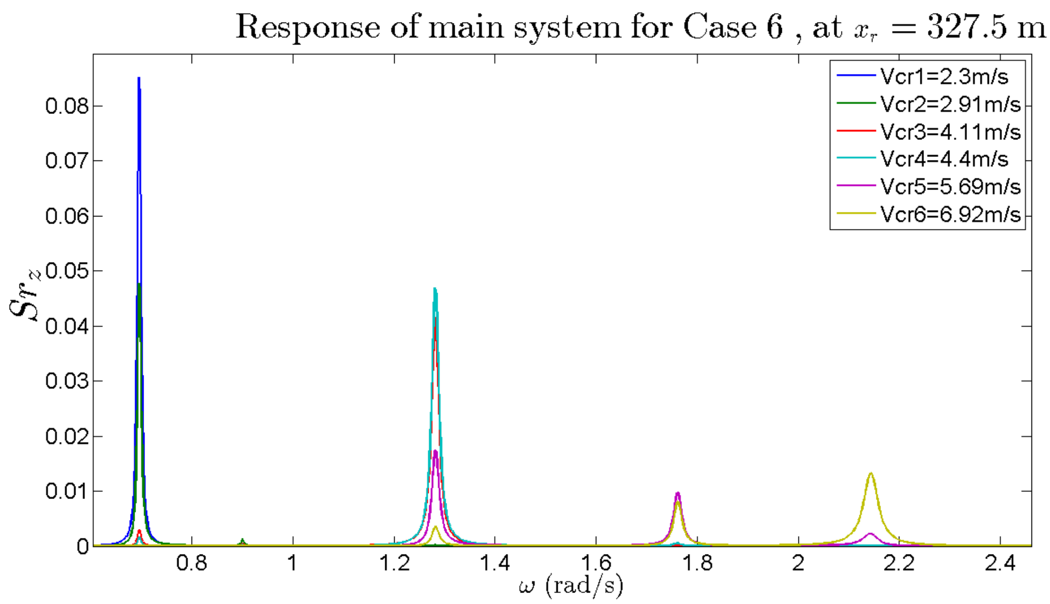

Figure 3.12: Response after TMD installation at x = 327.5 m for Case 6

In figure 3.13 the response of the TMD is shown. The TMD does get some response at the mode it is tuned to despite that the original system did not have any response for this mode. The TMD also get response at mode 1 and 3.

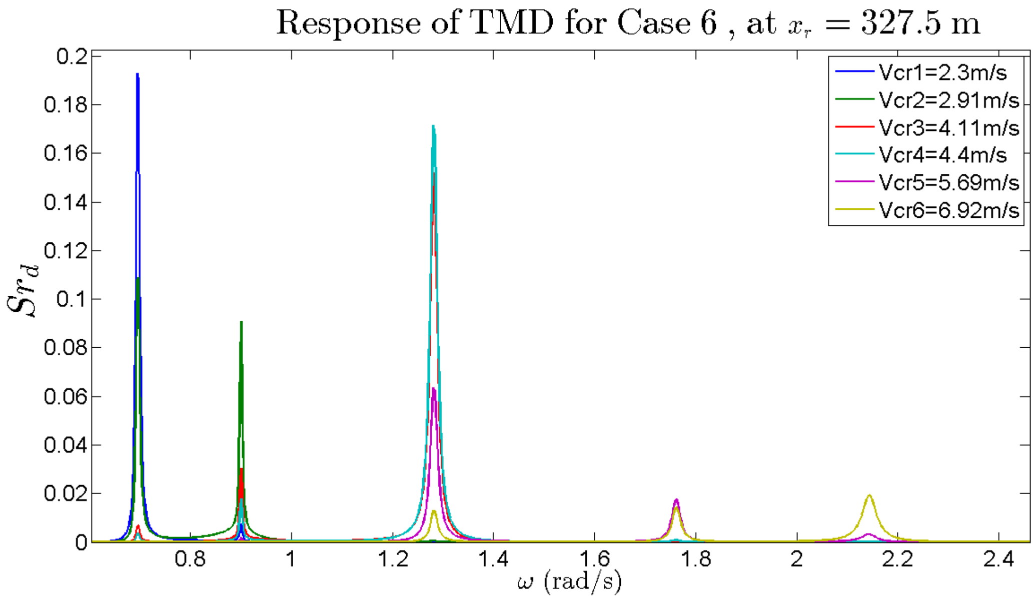

Figure 3.13: Response of TMD at x = 327.5 m for Case 6

Because the response in cases 2-4 and 7-9 does not get calculated at x = L/4 or x = L/2, it is chosen not to present the results from these cases as spectral density. The displacement response of all cases based on the simplified method presented in section 2.1.1 is given at the end of all cases.

In all the following cases, 3 TMDs is used. In the first cases the positions and which modes they are tuned to damp is tested, according to table 3.2

3 TMDs (TMD A, TMD B and TMD C)									
$\mu_d = 0.01$									
$\xi_d = 0.0603$									
	Positions of TMDs x_d [m] TMD name			Tuning to modes TMD name			Frequency, ω_d TMD name		
Case	A	B	C	A	B	C	A	B	C
11	327.5	655	982.5	1	2	1	0.7030	0.8911	0.7030
12	327.5	655	982.5	2	1	2	0.8911	0.7030	0.8911
13	327.5	655	982.5	1	2	3	0.7030	0.8911	1.2574
14	327.5	655	327.5	1	2	3	0.7030	0.8911	1.2574
15	200	400	500	1	2	4	0.7030	0.8911	1.3465

Table 3.2: Properties and placement of TMDs in case 11 to 15

The response of the main girder at mid-span for case 11 is shown in figure 3.14. It effectively damp out the second vertical mode, while it still is some response for the fifth mode.

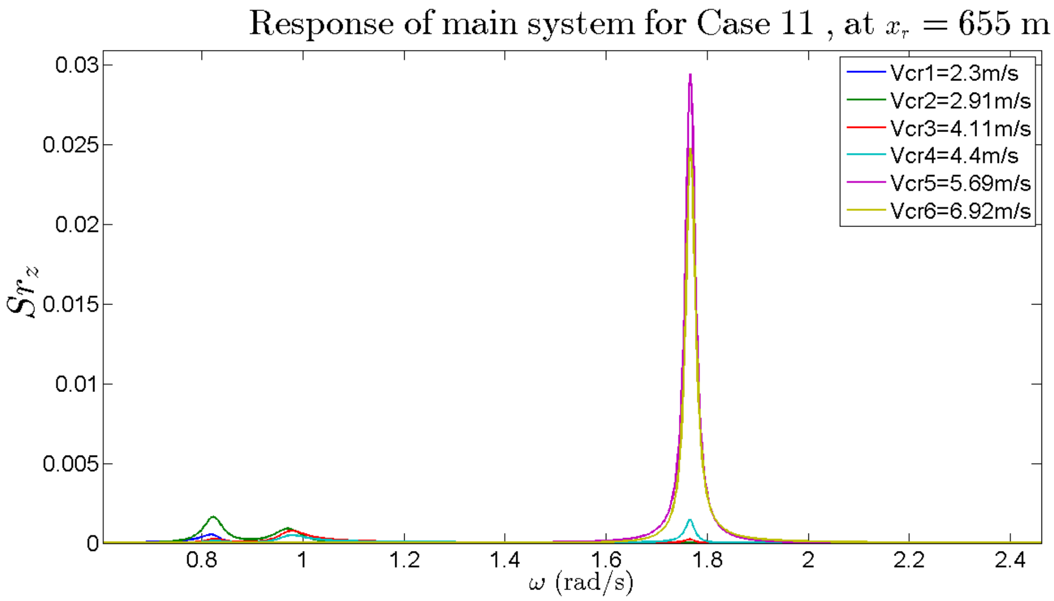

Figure 3.14: Response of bridge girder at x = 655 m for Case 11

The response of the TMD at mid-span in case 11 is shown in figure 3.15. There are some response at the fifth mode, but the biggest response in terms of area under the curve, i.e. the variance, is for the second mode. For critical wind velocities for mode two and one the peak of the response is highest to the left, i.e. for lower frequencies, while for higher resonance wind velocities the peaks to the peak to the right is highest.

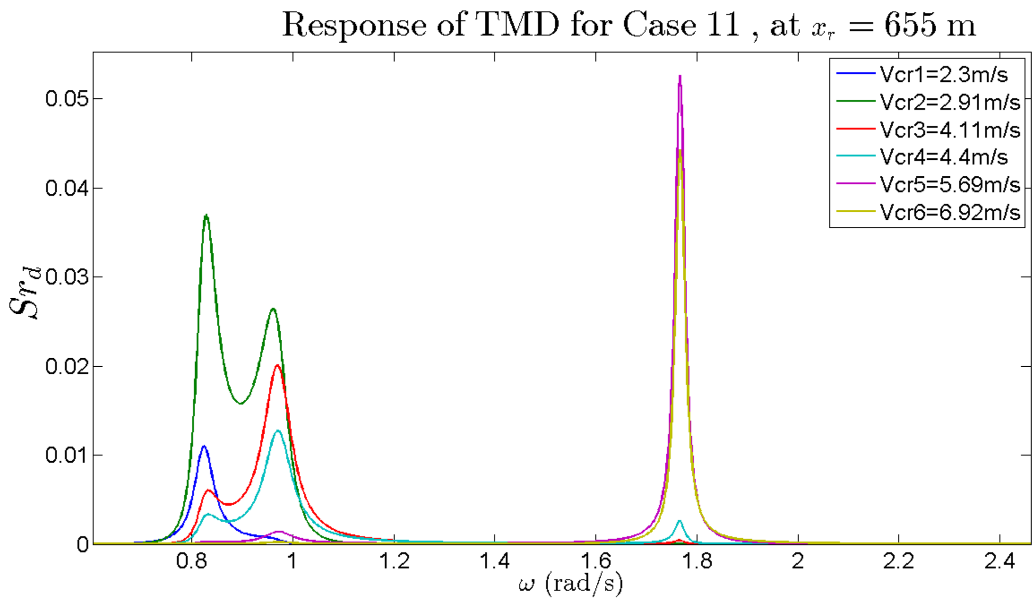

Figure 3.15: Response of TMD at x = 655 m for Case 11

For case 14 one TMD is placed at midspan to damp out the second vertical mode, while two TMDs is placed at $x = L/4$, one to damp out the first vertical mode, the other to damp out the third mode. It is assumed that there is enough place available to place two TMDs at the same x-location. Otherwise they have to be placed in front of each other, and it is assumed that a tiny dislocation in the x-direction does not affect the results in any significant way. The response of the bridge girder at mid-span for case 14 is shown in figure 3.16. It seems like the second mode still is damped out, while the fift mode also is slightly better damped than in case 11. The response of the TMD at mid-span is shown in figure 3.16, and it looks like the same tendency regarding the peaks a in case 11 is appearing in case 14 as well.

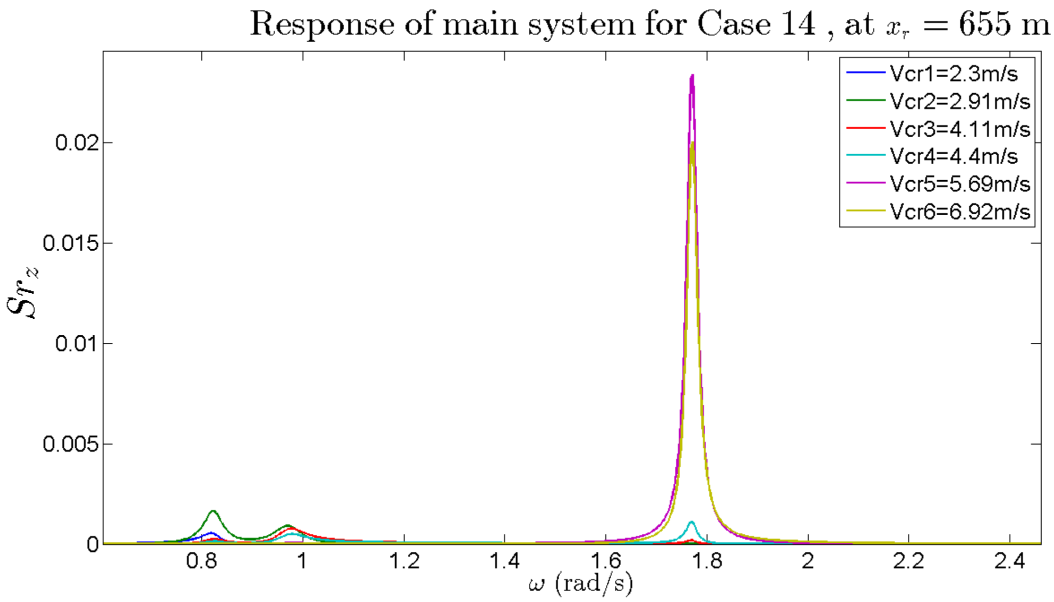

Figure 3.16: Response of bridge girder at x = 655 m for Case 14

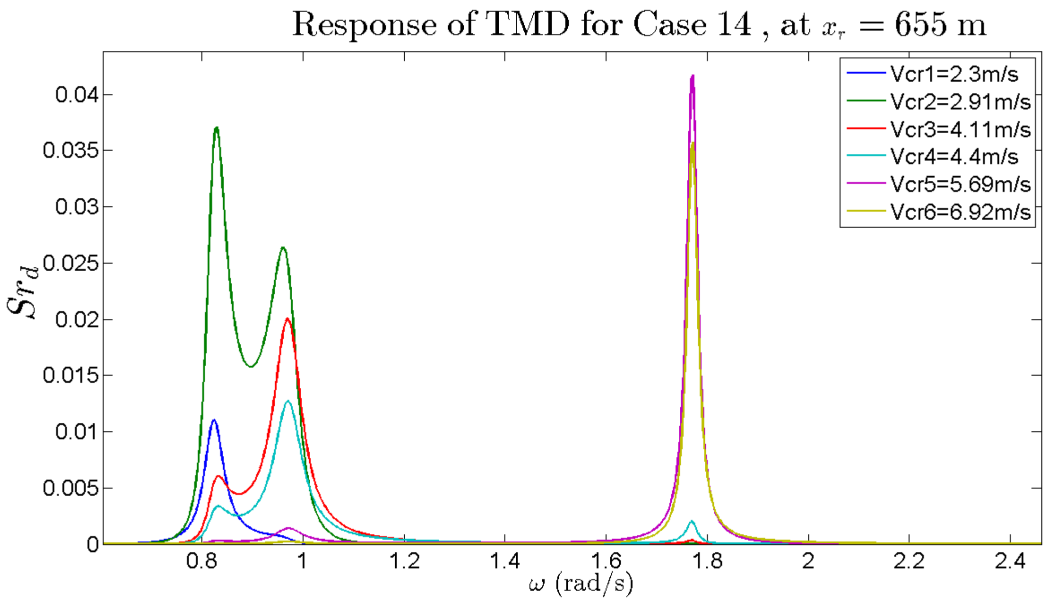

Figure 3.17: Response of TMD at x = 655 m for Case 14

Like for the single mode case, all parameters are now kept constant except mass ratio μ, to see how it affect the response. The TMDs is placed on $x = L/4$, $L/2$ and $3L/4$ respectively. The TMD at mid-span is tuned to damp out the second vertical mode, while the other two is tuned to damp the first vertical mode. The mass ratios, as well as other important data is shown in table 3.3.

3 TMDs (TMD A, TMD B and TMD C)				
$\xi_d = 0.0603$				
TMD positions [m]: TMD A: $x_d = 327.5$, TMD B: $x_d = 655$, TMD C: $x_d = 982.5$				
	ω_d [rad/s]			μ_d
	TMD name			For all TMDs
Case	A	B	C	TMD A,B and C
16	0.7030	0.8911	0.7031	0.002
17	0.7030	0.8911	0.7031	0.005
18	0.7030	0.8911	0.7031	0.010
19	0.7030	0.8911	0.7031	0.025
20	0.7030	0.8911	0.7031	0.050

Table 3.3: Properties and placement of TMDs in case 16 to 20

The respons of the bridge girder at $x = L/4$ is shown in figure 3.18. The first and second mode is damped out, while the third mode at $\omega = 1.76$ rad/s has the highest response. The TMD is effective and get large response for ω_{z1}, as shown in figure 3.19. For case 20, as shown in figure 5.6, the peak of the spectral density is about one tenth of the peak for case 16. It looks like the highest peak is either at ω_{z3} or ω_{z4} and the resonance velocities for vertical mode 4,3 and 5 lead to response of the mode, in decreasing order.

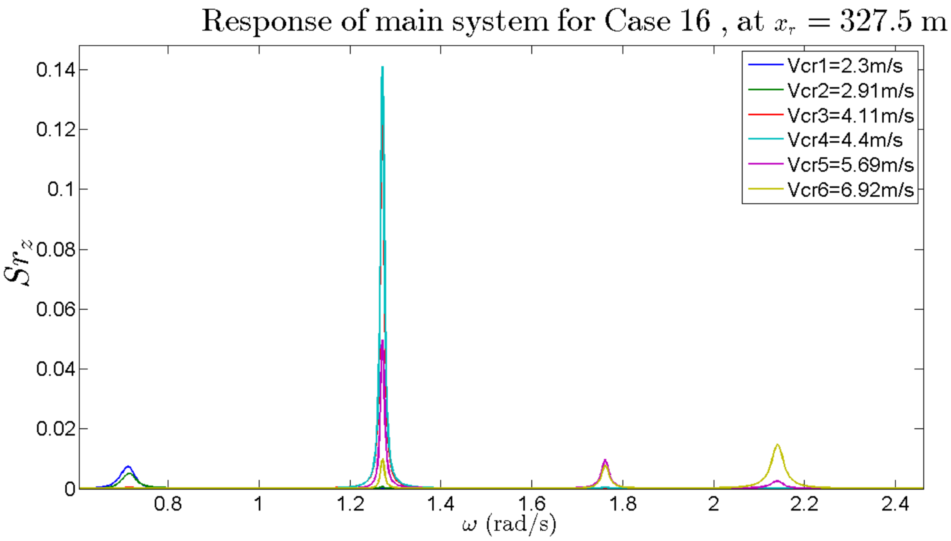

Figure 3.18: Response of bridge girder at x = 327.5 m for Case 16

The response of the bridge girder and TMD at $x = L/4$ is shown in figures 3.20 and 3.21 respectively. The peak values of the spectral density is roughly one tenth of the values in case

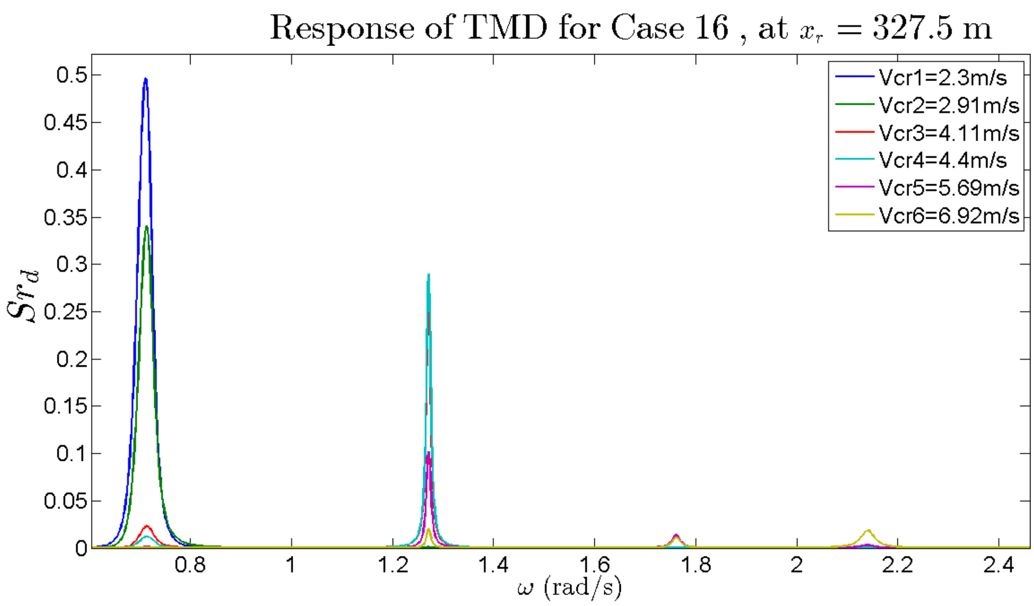

Figure 3.19: Response of TMD at x = 327.5 m for Case 16

16, while the peaks tend to be a bit wider such that the standard deviation does not have as big reduction. For the TMD the peaks of the two lowest resonant velocities is more spread and the peak of the lowest resonant velocity tend to move downwards on the frequency axis.

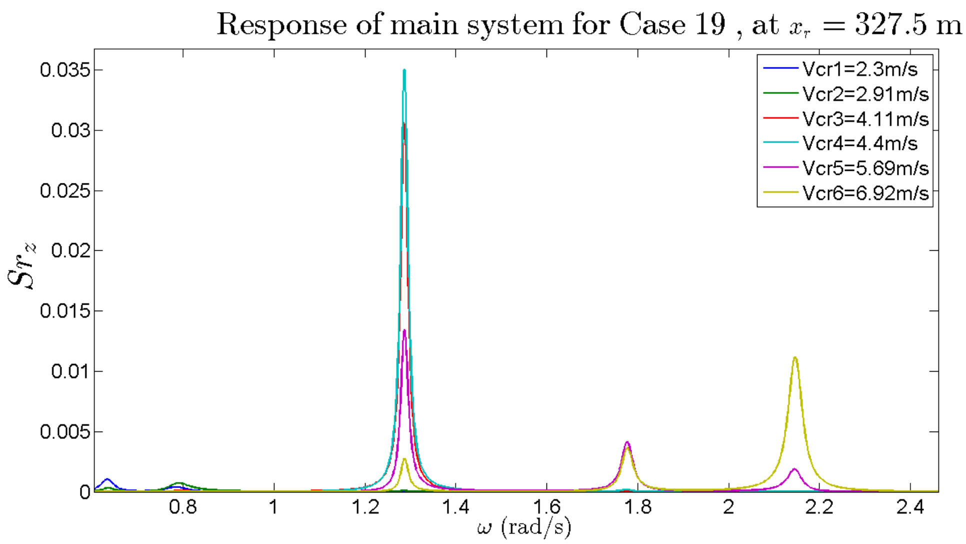

Figure 3.20: Response of bridge girder at x = 327.5 m for Case 19

A sketch of the change in computed standard deviation from the spectral density by the changing mass ratios is shown in figure 3.22. The circles is the computed values, while the lines is a manually added trend line for the purpose of easier see the trend of the data. The trend-lines does not represent the reality, and far more data points is needed to say something certain about the trend. The standard deviation of the original system without TMD is constant, and added in the purpose of comparing the magnitude of the standard deviation with and without TMD. The trend that increasing mass ratios makes the standard deviation less, does fit with the trend found by Hjorth-Hansen and Strømmen [Hjorth-Hansen and Strømmen, 2001].

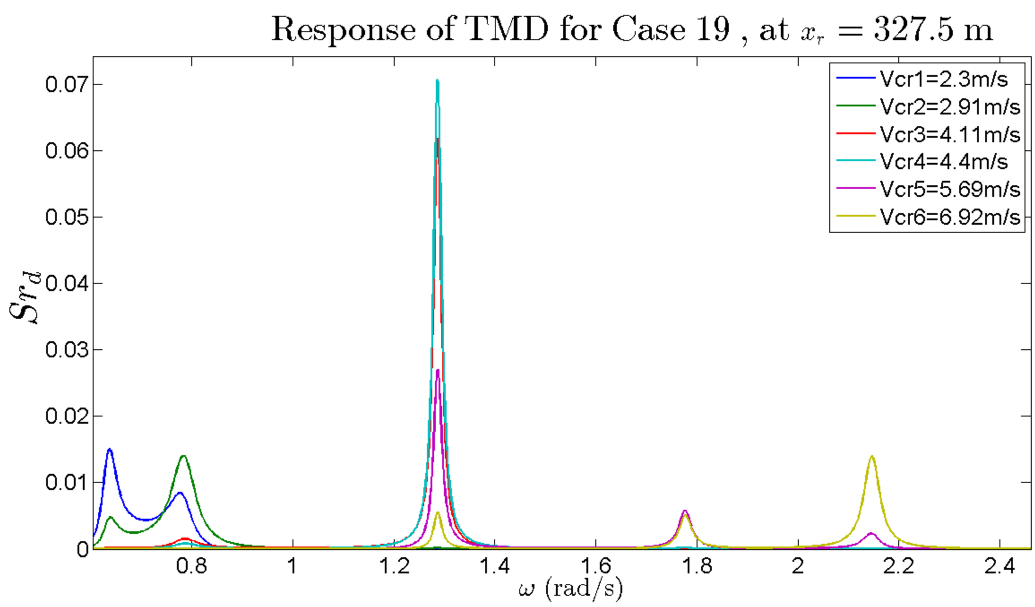

Figure 3.21: Response of TMD at x = 327.5 m for Case 19

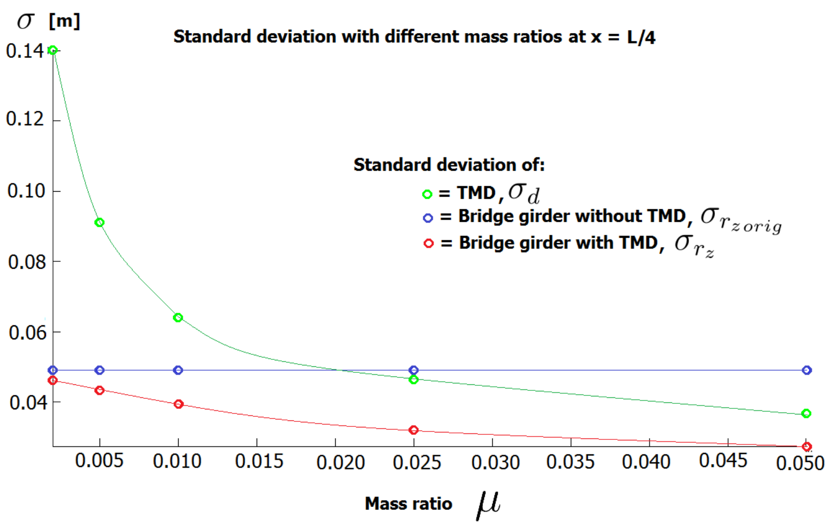

Figure 3.22: Values of standard deviation with different mass ratios

The next parameter to be tested is the damping ratio ξ_d

3 TMDs (TMD A, TMD B and TMD C)				
$\mu = 0.01$				
TMD positions x_d [m]: TMD A: $x_d = 327.5$, TMD B: $x_d = 655$, TMD C: $x_d = 982.5$				
	ω_d [rad/s]			ξ_d
	TMD name			For all TMDs
Case	A	B	C	TMD A,B and C
21	0.7030	0.8911	0.7031	0.0340
22	0.7030	0.8911	0.7031	0.0498
23	0.7030	0.8911	0.7031	0.0551
24	0.7030	0.8911	0.7031	0.0603
25	0.7030	0.8911	0.7031	0.0761

Table 3.4: Properties and placement of TMDs in case 21 to 25

The response of the bridge girder and TMD of case 21 at $x = L/2$ is shown in figure 3.23 and 3.24 respectively. It is only the fifth vertical mode that has response worth mention for the bridge girder. The rests of the second mode is small and spread, leaving two peaks, one at each side of ω_{z2}, which is assumed to come from the single peak at ω_{z2}. The TMD response show the same tendencies as in case 14, that the left peak is higher than the right for resonance velocity of the second and first vertical mode, while at other resonance velocities the right one is highest.

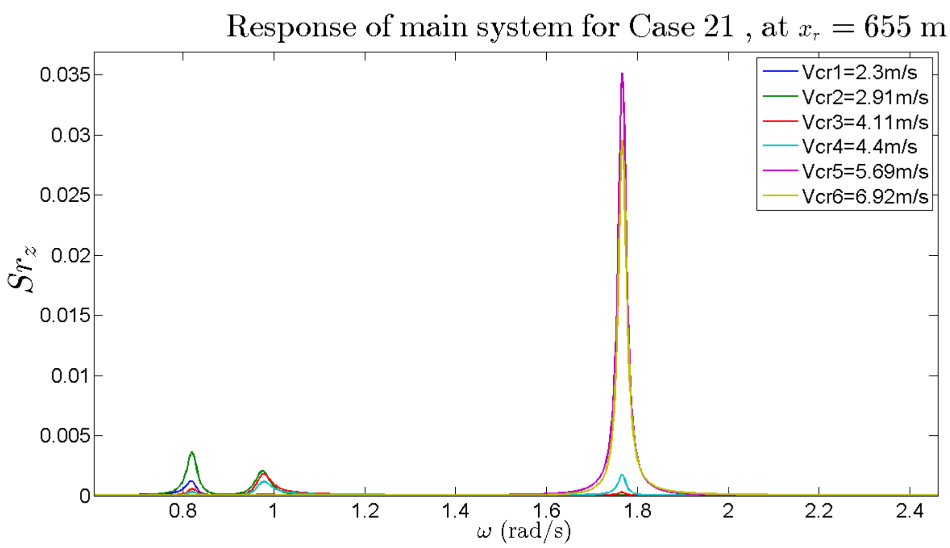

Figure 3.23: Response of bridge girder at x = 655 m for Case 21

The response from case 25 at $x = 655m$ is shown in figure 3.25 and 3.26 for the bridge girder and TMD respectively. The shape of the spectral densities for case 25, does not differ much from case 21. However, the response is lower, also for the TMD. In case 21 there were high peaks for the spectral density at the first mode for the TMD, while it in case 25 is considerably lower and also more smooth.

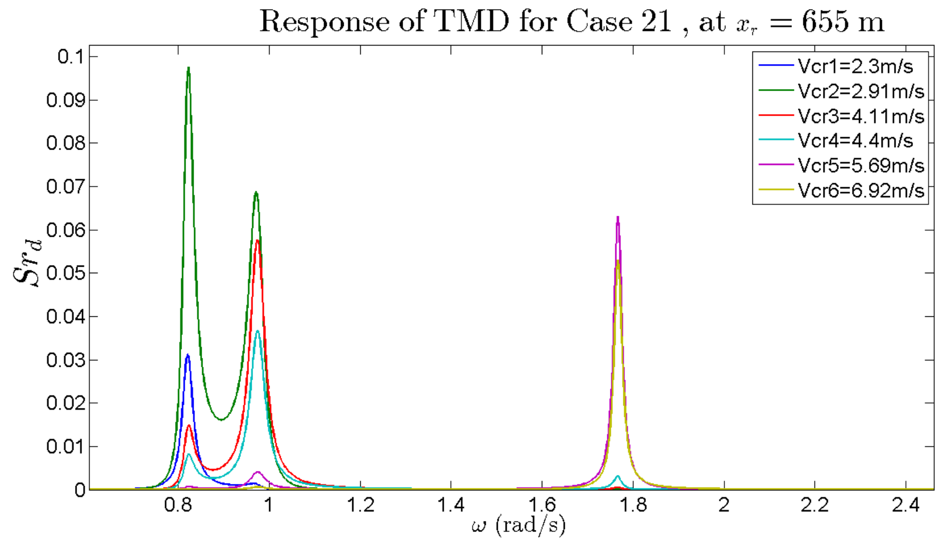

Figure 3.24: Response of TMD at x = 655 m

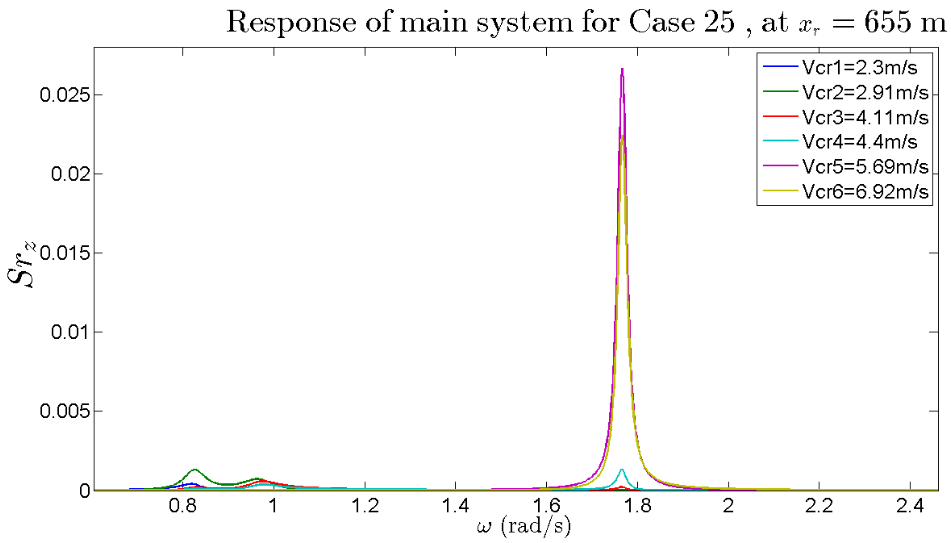

Figure 3.25: Response of bridge girder at x = 655 m for Case 25

A sketch of the values of standard deviation, obtained from the computed spectral densities, with different damping ratios is shown in figure 3.27. the circles indicates the computed values, while the lines is only provided for easier reading. The bridge girder without TMD should have constant standard deviation, and is plotted by the purpose of comparing its values with the other curves. The standard deviation of the TMD has a tendency to decrease with increasing value of damping, for the damping ratios considered. The standard deviation for the bridge girder is also decreasing, but not much. By Hjorth-Hansen and Strømmen, the standard deviation for the bridge girder should eventually increase for higher values of damping [Hjorth-Hansen and Strømmen, 2001]. A reason why it does not in this case, might be that the Matlab script does not iterate to find the response when the system has a TMD. That is for the sake of simplicity, and since an effective mass damper will make the response negligible in the expression for the aerodynamic damping.

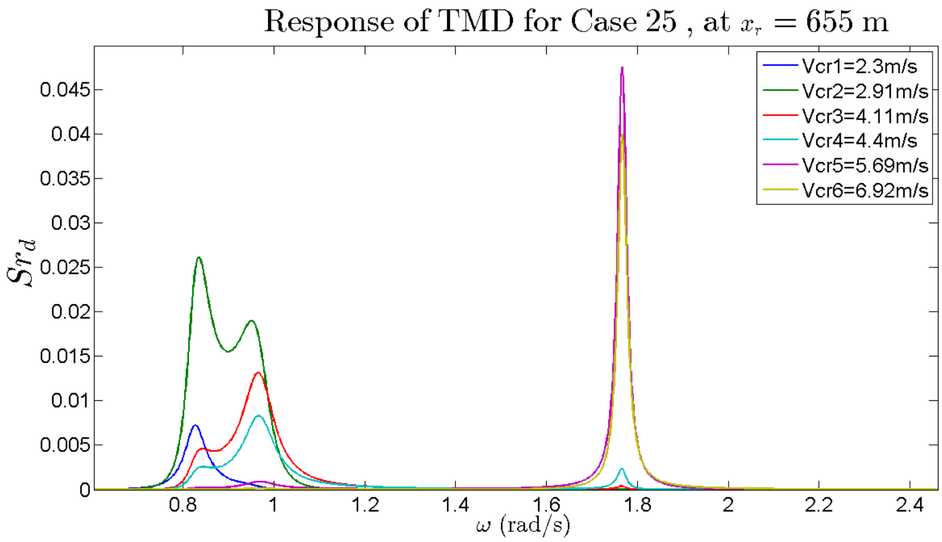

Figure 3.26: Response of TMD at x = 655 m for Case 25

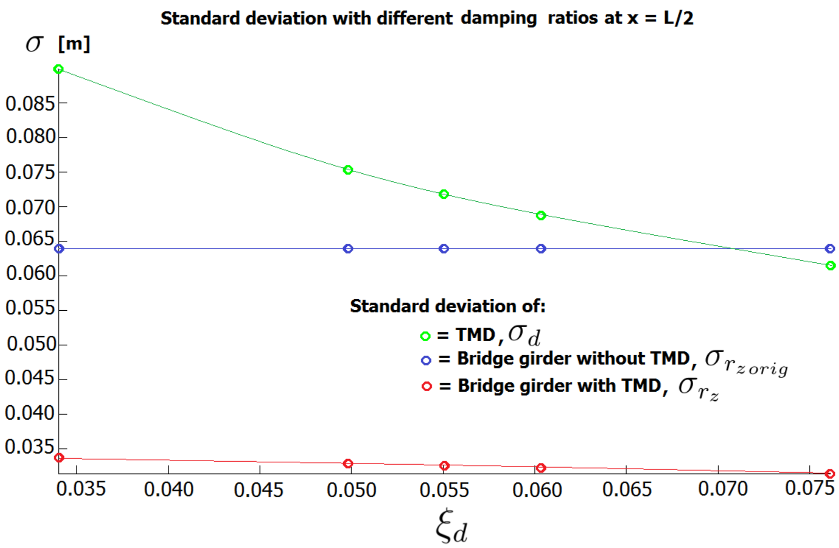

Figure 3.27: Values of standard deviation with different damping ratios

The last parameter to be tested is the frequency ω_d. The different values of ω_d used is given in table 3.5.

3 TMDs (TMD A, TMD B and TMD C)			
$\mu = 0.01$			
$\xi_d = 0.0603$			
TMD positions [m]:			
TMD A: $x_d = 327.5$			
TMD B: $x_d = 655$			
TMD C: $x_d = 982.5$			
	ω_d [rad/s]		
Case	TMD A	TMD B	TMD C
26	0.6677	0.8464	0.6677
27	0.6853	0.8687	0.6853
28	0.7030	0.8911	0.7030
29	0.7206	0.9134	0.7206
30	0.7382	0.9358	0.7382

Table 3.5: Properties and placement of TMDs in case 26 to 30

The spectral densities of the bridge girder and the TMD at $x = 655m$ is for case 26 shown in figure 3.28 and 3.29 respectively. The response of the bridge girder is damped efficiently away, other than the fifth vertical mode. The TMD at mid-span does have response for mode 5, but mostly for mode 2, which it is damping away. The relative peak high at left and right side of ω_{z2} is not following the trend from for instance case 21, where the peaks for resonant velocity of second mode was higher on the left hand side. However, the response of the rest of the resonance velocity seems to be according to the tendencies found for case 21.

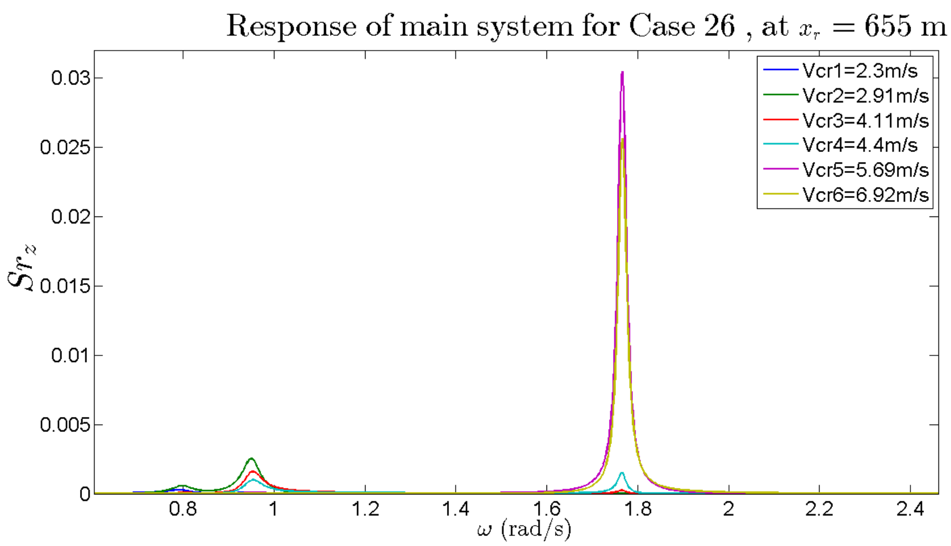

Figure 3.28: Response of bridge girder at x = 655 m for Case 26

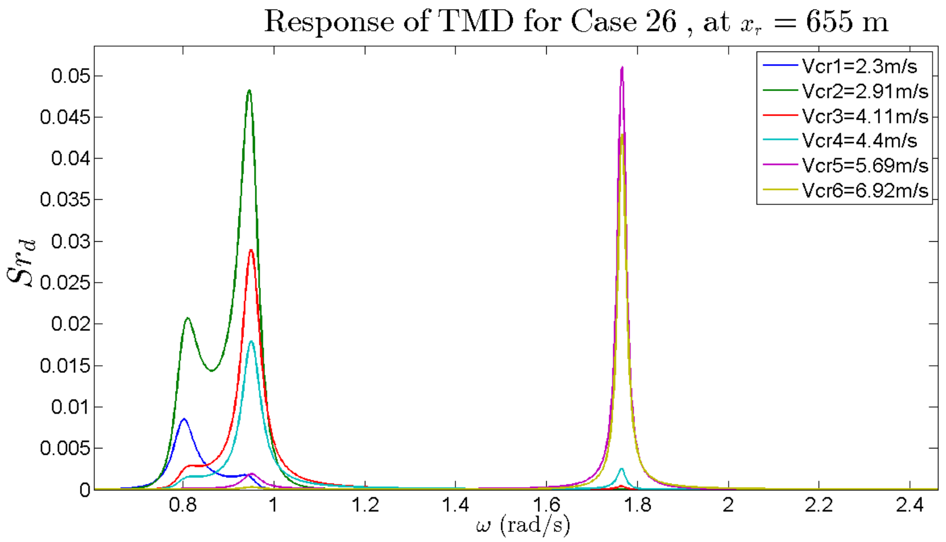

Figure 3.29: Response of TMD at x = 655 m for Case 26

The response for case 30, with higher values of ω_d, is shown in figure 3.30 and 3.31 for the bridge girder and TMD at $x = L/2$ respectively. It is almost impossible to say from the spectral density, how the standard deviation response is changed from case 26 to 30, as it looks like they almost have the same area under their graph. However, there is difference in the frequency distribution of the spectra. The spectral density peak from the resonant velocity of the second vertical mode is, for a higher value of frequency ratio, highest on the left side of ω_{z2}.

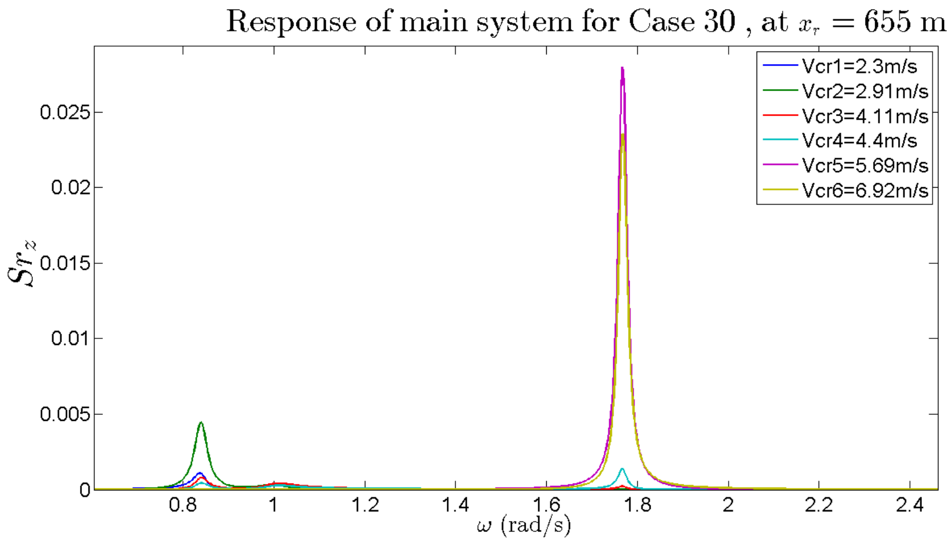

Figure 3.30: Response of bridge girder at x = 655 m for Case 30

A sketch of the values of standard deviation, obtained from the computed spectral densities, with different frequency ratios is shown in figure 3.32. the circles indicates the computed values, while the lines is only provided for easier reading. The bridge girder without TMD should have constant standard deviation, and is plotted by the purpose of comparing its values with the other curves. The standard deviation of the TMD has a tendency to decrease with increasing value of frequency ratio, for the values considered. The standard deviation for the bridge girder is

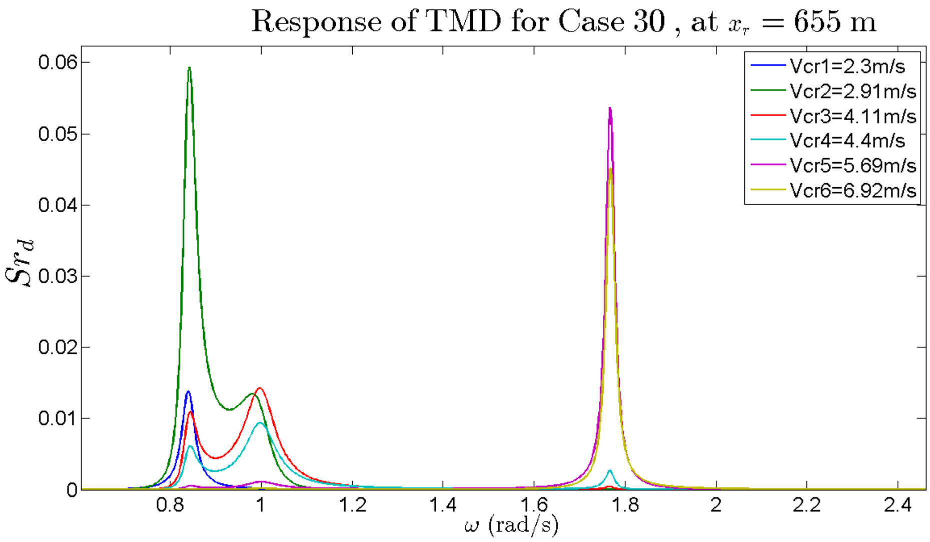

Figure 3.31: Response of TMD at x = 655 m for Case 30

also decreasing, but not much. By Hjorth-Hansen and Strømmen, the standard deviation for the bridge girder should increase for higher and lower values of damping then the optimum [Hjorth-Hansen and Strømmen, 2001].

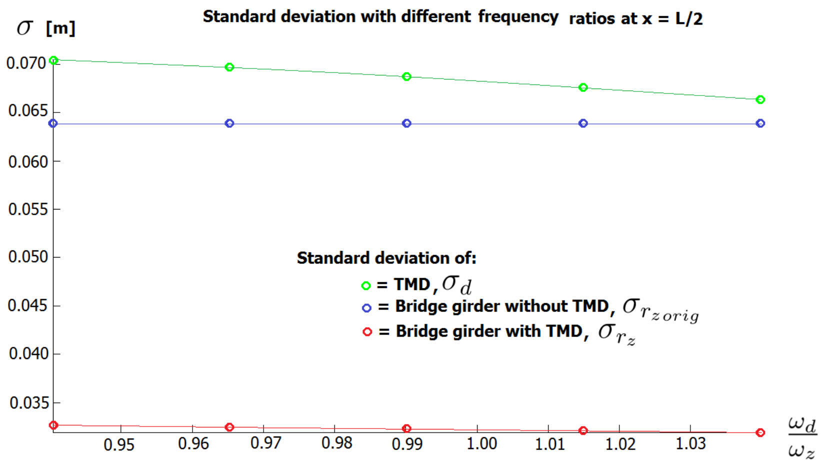

Figure 3.32: Values of standard deviation with different frequency ratios

Frequency Response Function for the different cases

The spectral density for a multi mode, multi component case is earlier given in equation 1.129. By the idea that vortex shedding response usually are considered narrow banded, the resonant part of the variance $\sigma^2_{R_i}$, as given in equation 1.19, will dominate the response. Since the resonant part mainly is determined by the frequency response function,FRF, the FRF will give a good indication of the response. To easy see the difference between the cases, the frequency response function for the cases that study the same variable is plotted together. In figure 3.33 the FRF for the first vertical mode is given.

As can be seen in the upper left plot, case 1 and 2 has the lowest response, while case 4 and 5 has nearly no reduction at all. This is not a surprise, since case 1 is to place a TMD tuned to vertical mode 1 where it is at its maximum, while case 5 is to place the TMD tuned to mode 1, where mode 1 is zero. The next plot is of cases 6 to 10, which is to place a TMD tuned to damp vertical mode 2 at different locations. Clearly the TMD tuned to mode 2, does not damp much of mode 1. Despite that it is as expected, and that the plot is not very interesting, it has been included for the purpose of understanding, and as a check of whether or not the results produced by the Matlab script is reliable. For case 11 to 15, it looks like case 11 reduce the response the most. Case 11 is the case where a TMD tuned to vertical mode 2 is placed at its maximum at mid-span, while 2 TMDs placed at $x = L/4$ and $x = 3L/4$ respectively, is tuned to dam mode 1. For case 16 to 20 it is not easy to know from the FRF which mass ratio is best suited. It looks like increasing mass ratios damp the response near the eigen-frequency most, while it make the response at two other frequencies higher. For the damping ratios in case 21 to 25 there is an opposite effect of increasing the value of the parameter. At low values of damping, the response is redused the most around the eigen-frequency, while farter away from the eigen-frequency, the peaks of the response is higher. For higher damping ratios, the curve is more smooth, which is assumed to give the best effect of the damping. For the frequency ratio, it is case 27 that has peaks at the most equal level, and thus has lowest peak. Case 27 is, not surprisingly according to the results when a single mode was considered in figure 3.5.

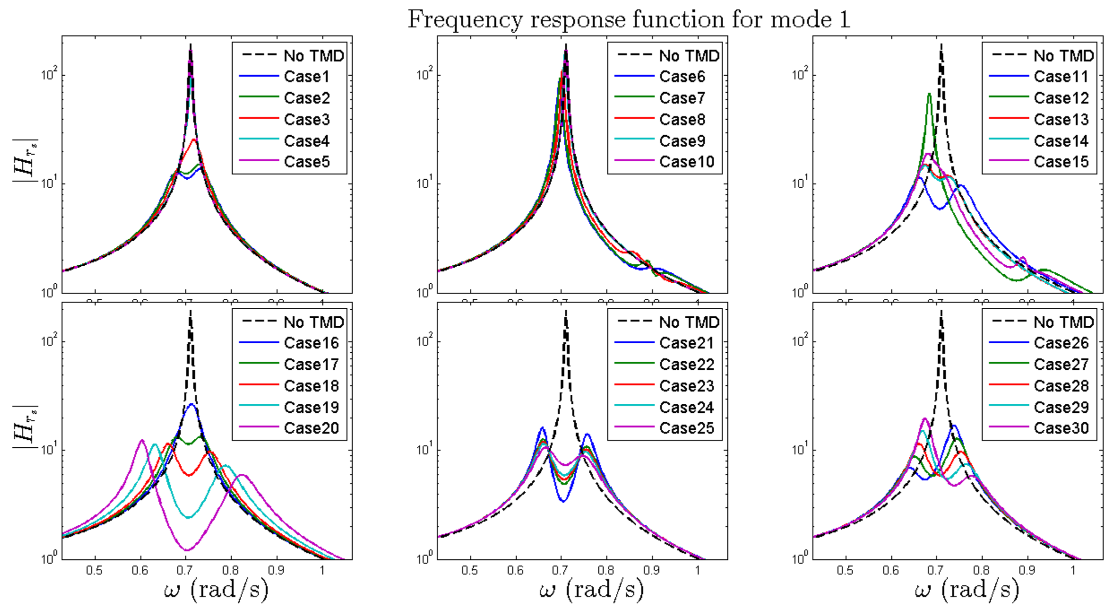

Figure 3.33: Frequency Response Function for the first vertical mode, without TMD and with different cases. The definition of the cases can be found in tables 3.1, 3.2, 3.3, 3.4 and 3.5

In figure 3.34 the frequency response function for all cases for the second vertical mode is given. Many of the results is the same as for mode 1, but the 3 upper plot shows some differences. Since the TMD in the upper left figure is tuned to damp out mode 1, it does not reduce the response at all. For cases 6 to 10, the TMD is tuned to damp out mode 2, and the response is reduced. The most reduction is for case 10 where the TMD is placed at the second modes maximum, at mid-span. Worth noticing is that case 9, where the TMD is placed 55 m from the mode-shape maximum at the mid-span, the FRF is quite similar to case 10. It looks like a small change in where the TMD is placed does not affect the efficiency that much.

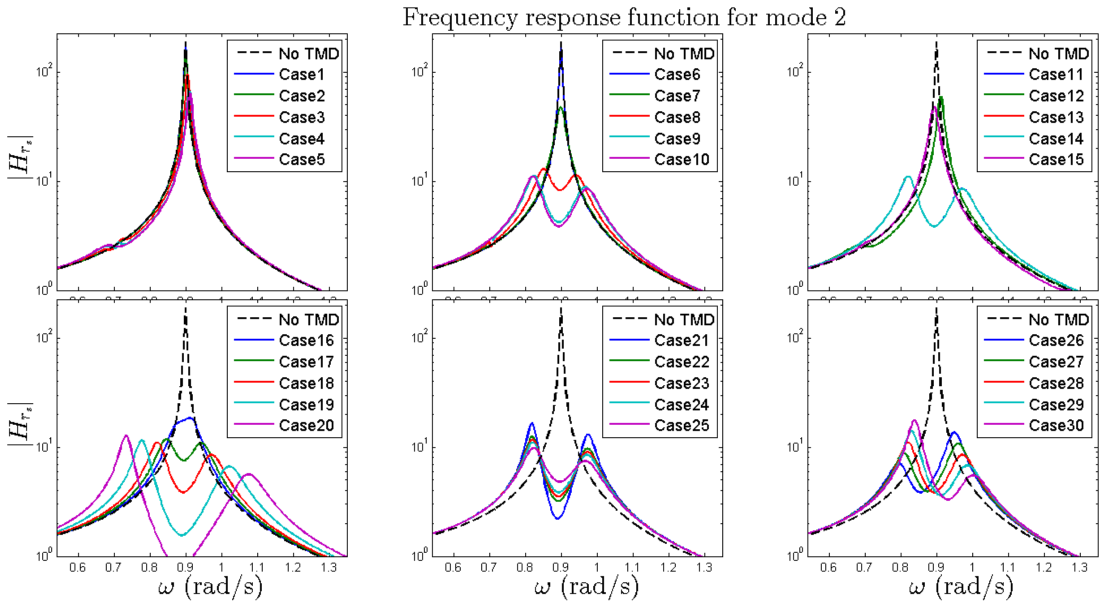

Figure 3.34: Frequency Response Function for the second vertical mode, without TMD and with different cases. The definition of the cases can be found in tables 3.1, 3.2, 3.3, 3.4 and 3.5

3.2.2 Time Domain Simulations

From the spectral density time domain simulations can be produced. Results from 1000 time domain simulations is shown in figure 3.35 for r_z, and in figure 3.36 for r_d. The upper left plot in each of the figures show 999 simulations plotted on top of each other, and one single simulation. As can be seen the single simulation have periods of large deflections, pulses, and periods of almost no movement. This is typical for the most of the time series of from vortex shedding induced vibrations [Strømmen, 2010]. The 999 simulations on top of each other done to visualize that the periods of maximum deflections is not at the same points in each time-series. Keeping in mind that each graph of simulation have some thickness, it looks like the sum of all 999 simulations fill the area between the maximum deflections at positive and negative side. That is, at any point in time at least one of the time series have a maximum value close to the maximum value of all the time series inside the time window considered, in this case 10 minutes. Also at each point in time, any value of deflection less than the maximum is represented by at least one of the time series.

Dividing the results from the 1000 simulations into 100 bars gives the results shown in the other but the uppermost left sub-figure in figure 3.35 and 3.36. Shown is also a normal distribution with mean and standard deviation values based on the simulations. The results indicate that the response is not normal distributed. It looks like there is a certain skewness, especially for the peak factor k_p and some large variations inside between neighbor bars. A reason for why the response does not fit a normal distribution, is that the time series has pulses with large deflections, thus the response is highly dependent on the time window considered. Despite that the response is not normal distributed, it show some of the same tendencies as a normal distribution or a distribution with skewness, like the Weibull distribution; the response from most of the simulations is gathered around a response value. Because of the pulsating art of the time series, it is assumed that the distribution of the simulations will vary, and no further effort is made to try to fit the simulation data to a distribution.

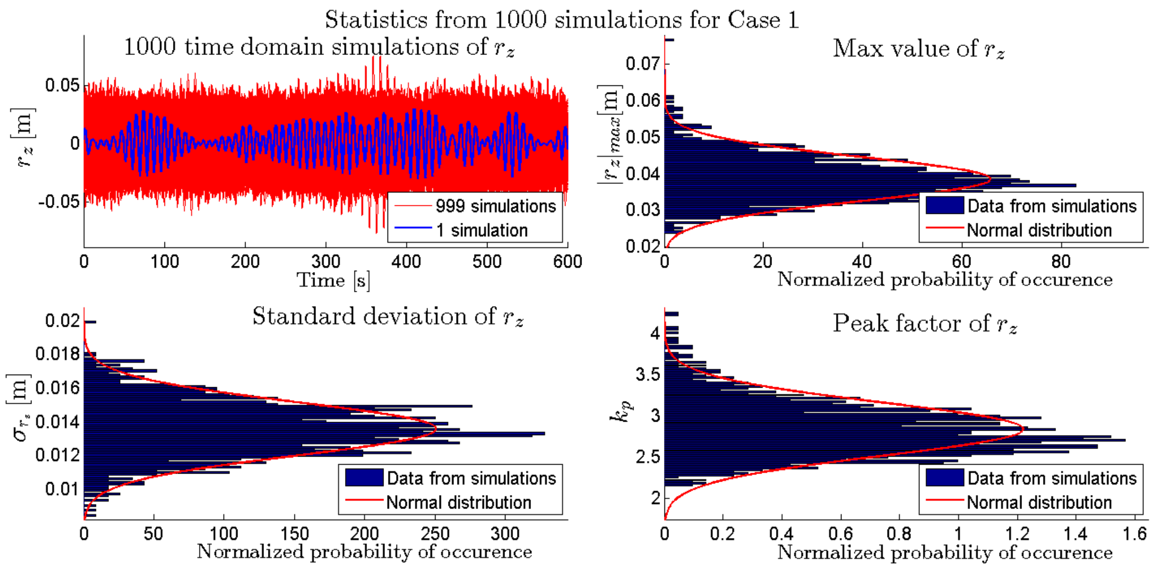

Figure 3.35: Results from 1000 simulations of bridge girder movement

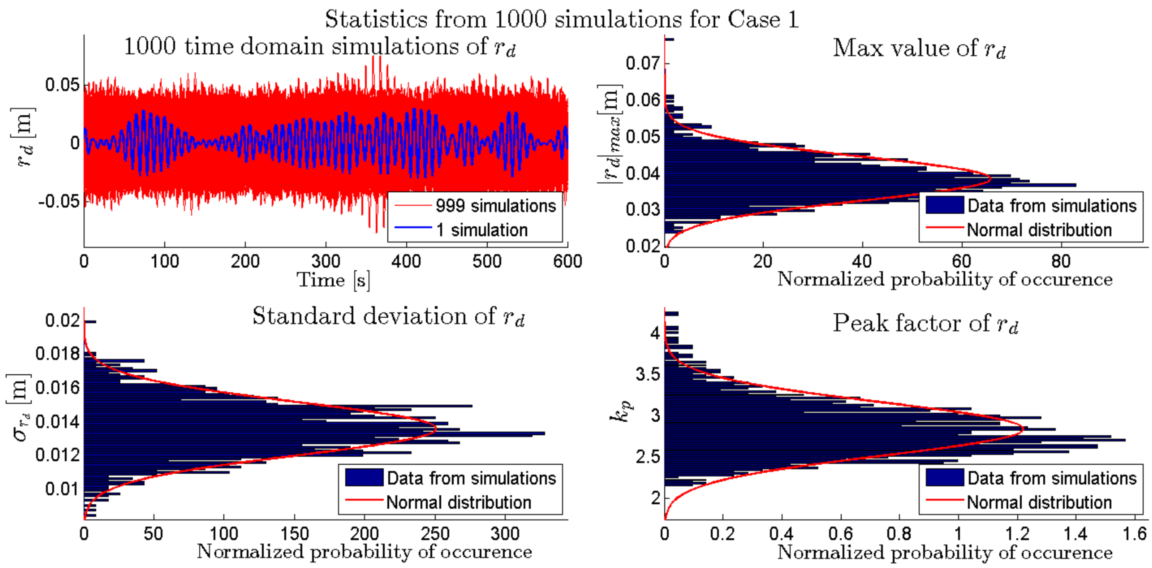

Figure 3.36: Results from 1000 simulations of TMD relative movement

Making a movie of the brigde girder and TMD displacement

As explained in section 2.1.1, it is found useful for the learning process to visualize the displacement response. In the Matlab script, a plot is made for each time instance, and they are put together in Matlab to form a movie. A written report is not a good place to present a movie, however some screen shots is made to illustrate the movie itself, but more importantly the limitations of single point spectral density.

Figure 3.37 shows a screen shot of a movie from case 1, where the wind velocity is the resonant velocity for the first vertical mode. Both sub-figures show the entire span of the bridge. The upper one is of the bridge girder response before installing a TMD. The red circles is the displacement found from time series made from single point calculations of the spectral density, and by using a equal random-vector for each point. As can be seen, the circles have symmetric displacement around the middle, whereas the bridge girder has anti-symmetric and follows the first vertical mode shape. In this case, the circles and the bridge girder would almost coincide if we reflect the right part around the neutral position of the bridge girder. However, the simplified bridge girder displacement seems to give too stiff response in the middle of the span, as it does not coincide with the circles in the middle. The lower sub- plot shows the displacement of the bridge girder after the installation of a TMD, as well as the TMD itself. No single point response is shown in the lower subplots, but in principle it is possible to calculate and show single point response there as well.

In case 5, the TMD is placed at the zero point of the first vertical mode, as shown in figure 3.38, and the response is not reduced, in fact it is slightly increased because some small movement of the TMD. Again the simplified method of finding the bridge girder gives worst results for the middle area, where the bridge girder now is too little stiff.

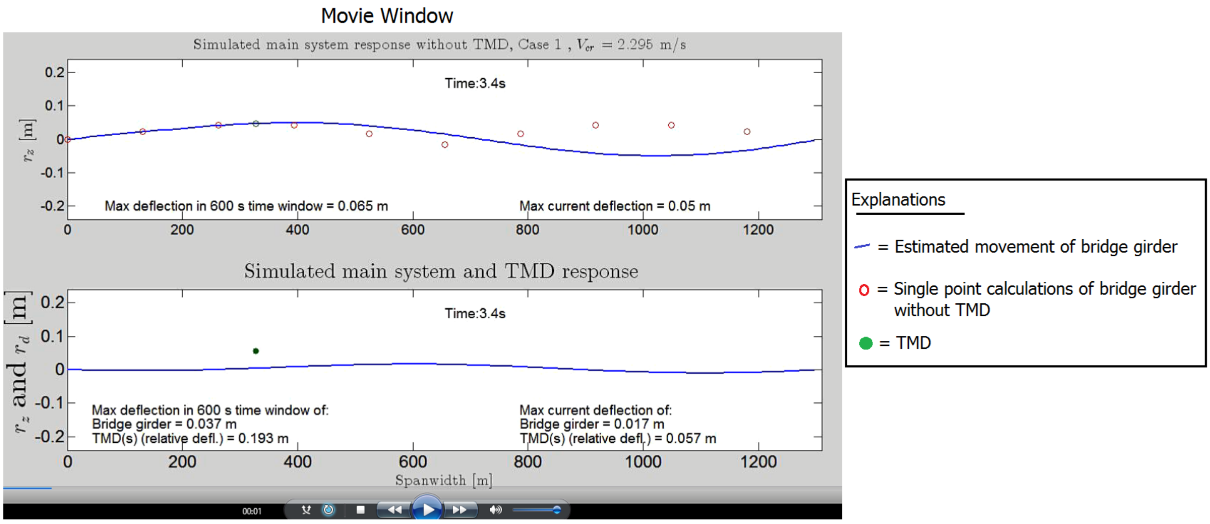

Figure 3.37: Screen-shot of movie of brigde displacement from Case 1

The movies gives a physical meaning of the results. From the "movie-calculations" the maximum displacement response for the bridge girder and the TMDs could be estimated. The best way to estimate the displacement response is directly from time domain simulations, as shown in figure 3.36. However, to get a rough estimate of the response anywhere at the bridge, not just only where the single point response is calculated, the method of using the movie calculations is simple. The maximum response given in figure 5.21, is from just a single time domain simulation, but it is the maximum values of the entire span, not just a chosen point, that

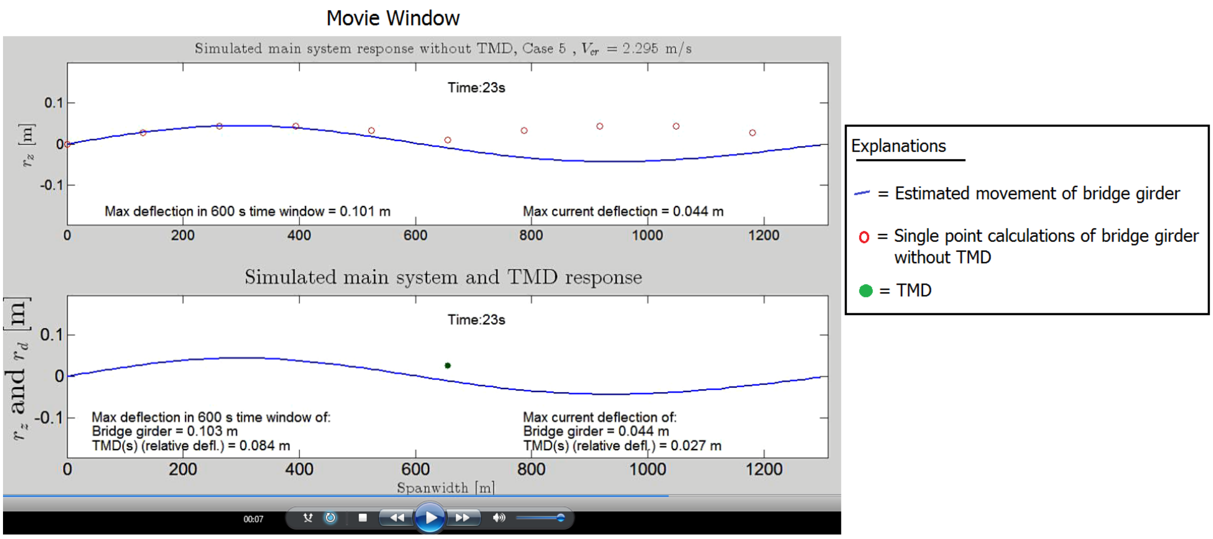

Figure 3.38: Screen-shot of movie of brigde displacement from Case 5

is found. The cases which seems to give lowest response of the bridge girder, case 1, 11, 18, 20, 22 and 29. In the FRF, case 22 and 18 had low response, but the message from the figure must be that if the displacement response should have been trustful, many time domain simulations should have been made. No further effort is made in doing more simulations, since the response is quite well known from frequency response function, spectral density and standard deviation.

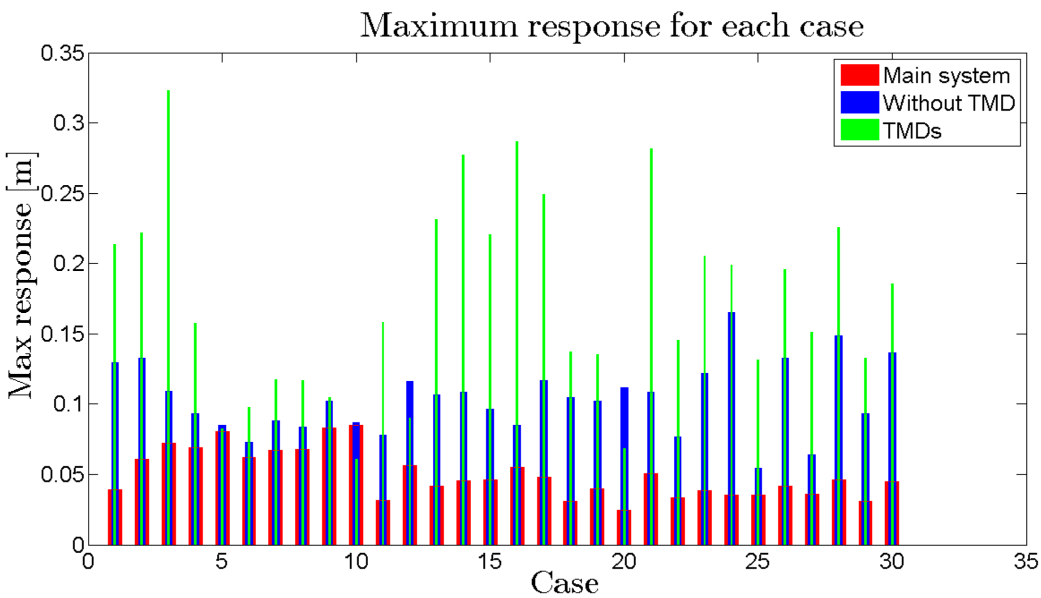

Figure 3.39: Estimated maximum displacement for different cases

Chapter 4

Comments

4.0.3 Why the response of some higher modes tend to be too big

Since the cross sectional load spectrum S_{q_z} is dependent on the variance of the vortex shedding load, which again is dependent on the velocity squared 1.122. Thus the vortex shedding load increases for increasing wind speed values. Another effect of changing the wind velocity, is the effect on the aerodynamic damping. The aerodynamic damping is in itself negative, thus the total damping of the system is reduced. In the equation of aerodynamic damping, equation **??**, the aerodynamic damping coefficient, K_{a_z}, could be dependent on the wind velocity. For the Osterøy bridge, Hjorth-Hansen and Strømmen, [Hjorth-Hansen and Strømmen, 2001], chose to use the following expression of the velocity variation of K_{a_z}:

$$\frac{K_a}{K_{amax}} = \left[\frac{0.9}{(V/V_{cr} - 0.25)^2}\right] \cdot \exp\left[\frac{-1}{(V/V_{cr} + 0.02)^{24}}\right] - 0.18 \tag{4.1}$$

The variation is shown in a graphic format in figure 4.1. The expression is only valid for positive values of K_a, thus it is limited roughly $0.9 \leq \frac{V}{V_{cr}} \leq 2.5$.

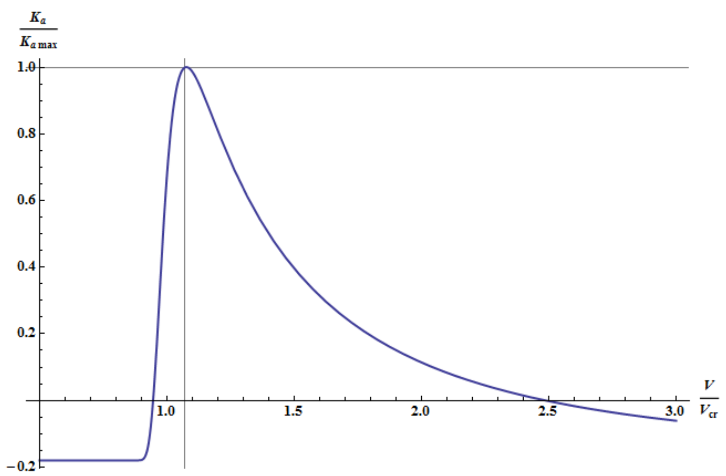

Figure 4.1: K_{az} variation with different velocity

The velocity variation of K_{az} will reduce the (negative) aerodynamic damping, hence increase the total damping, of modes with critical velocity other than the velocity considered. Since $V_{cr1} \leq V_{cr3} \leq V_{cr4}$, the K_{az} value will be reduced from mode 3 where $\frac{V}{V_{cr}} = \frac{V_{cr1}}{V_{cr3}}$, to mode 4 where $\frac{V}{V_{cr}} = \frac{V_{cr1}}{V_{cr4}}$. using equation 4.1. Figure 3.2 is obtained using constant K_{az}, thus it will show larger response than realistic for modes with V_{cr} not equal the velocity, V, considered.

The graph of the velocity variation of K_{az} shown in figure 4.1 is steeper for $V < V_{cr}$, than for $V > V_{cr}$. Also, its maximum is not at velocity $V = V_{cr}$, but slightly over, about $V \approx 1.06 \cdot V_{cr}$. These two curve characteristics comes from two separate, but jet connected reasons. Firstly the slope of the curve could reflect the physical phenomenon called lock-in. That is that the structure and the flow will interact when resonance occurs, and for some range of velocities above V_{cr} the vortex shedding frequency will stay constant equal the shedding frequency at V_{cr} [Strømmen, 2010]. Secondly the Strouhal number is here defined at onset of lock-in, such that the maximum response occurs slightly above V_{cr}.

The consequences of using a constant K_{az} is that some modes other than the one at resonance get higher response than in reality. Especially the response of the lower modes at resonance velocities of the higher modes is overestimated, due to the higher cross sectional load S_{qz} with increased velocity, and that the physical considerable reduction of K_{az} for $V < V_{cr}$ is neglected. However, the maximum response for each mode n is for $V = V_{cr,n}$,and thus the overestimated other modes does not affect the maximum response that much, but rather the frequency content of the response. A response spectra containing overestimated peaks other than the peak for the resonance mode, would produce time series with overestimated frequencies other than the resonance frequency, and make the response more broad banded. While choosing a velocity variation of K_a, Hjorth-Hansen and Strømmen make it clear that the K_a, as well as the band-width parameter b_z does not affect the prediction of the maximum RMS-values much, but rather the broad- or narrow-bandedness of the response [Hjorth-Hansen and Strømmen, 2001].

4.0.4 Why the change in standard deviation does not follow the trend found by Hjorth-Hansen and Strømmen

The expression for aerodynamic damping is dependent on the response σ_{r_z}. Since the aerodynamic damping is used to calculate the response, iterations is needed. Assuming that the tuned mass dampers is effective, the response is small, and it could be neglected from the term of aerodynamic damping to avoid demanding iterations. For a system without tuned mass dampers, or ineffective mass dampers, iterations is needed. In the Matlab script iterations are done for the system without mass dampers, but it is not for the systems with mass dampers. However, since different cases of placement and tuning of the mass dampers is tested, some of them is not effective, and neglecting the response in the calculation of the aerodynamic damping might give to small total damping in the system. Also, the trend presented by Hjorth-Hansen and Strømmen is found by iterating for all cases [Hjorth-Hansen and Strømmen, 2001]. Thus, small changes in the standard deviation might be missed if the response in the aerodynamic damping is neglected.

Chapter 5

Conclusion

In the analysis of the Hardanger Bridge, several concept has been learned. The difference of response with and without one or several tuned mass dampers is clearly when working with the analysis. Tuned mass dampers are very effective to tune out motion at a certain frequency.

The analysis of the Hardanger bridge indicate in the order of 0.1 meters maximum displacement response without the use of tuned mass dampers. It is not certain that this is correct, but assuming the maximum displacement is of that order of magnitude, it would probably not be a problem for the bridge as a construction. However, at many locations wind velocities that gives vortex shedding occurs frequently, and the vibration of the bridge can be a problem for peoples well-feeling and trust of their safety. The purpose of this thesis was not to come up with a conclusion whether or not one or several tuned mass dampers is necessary, but to look at the effect of installing it. The analysis indicate that one or several tuned mass dampers placed at locations where the eigen-mode they are supposed to damp out has its maximum, or is close to its maximum, is an efficient way to reduce the response.

A damper is more effective when it has higher mass ratio, but the ratio of the increasing effect seems to decay for higher mass ratios. The value of the damping seems to adjust the difference between the value of the frequency response at the eigen-frequency, and the peaks around. A higher damping ratio seems to give a smoother curve for the values consider in this case. The value of the frequency seems to adjust the relative height between the two peaks at each side of the eigen-frequency. Den Hartogs optimization gives peaks at almost the same level. Frequency according to Den Hartog, and damping according to Den Hartog or slightly above seems to give effect of the tuned mass dampers.

In the Matlab script, the iteration process to find the response, is done for the case with no damper, but the response term in the aerodynamic damping is neglected for the case of tuned mass dampers installed. That is because the response, if the damper is effective is negligible when used in the expression for aerodynamic damping.

To find the approximate response along the span of the bridge, a simplified method that could be used is to find the response in modal coordinates and multiply it by the mode-shapes. The method does neglect the coupling between the modal degrees of freedom, thus it is best suited for cases where the eigen- frequencies is well separated. Also the coupling is found to be increased with increasing mass ratios.

[]

Bibliography

W.K. George. *Lectures in Turbulence for the 21st Century*. 2013. URL `http://www.turbulence-online.com/index.html`. Accessed March 4, 2013. Availible from: http://www.turbulence-online.com/index.html.

E. Hjorth-Hansen and E. Strømmen. On the use of tuned mass dampers to supress vortex shedding induced vibrations. *Wind Structures, An International Journall*, 4(1):19–30, 2001.

E. Strømmen. *Theory of Bridge Aerodynamics*. Springer, 2010.

E. Strømmen. *Structural Dynamics*. Springer, 2013.

Appendix

5.0.5 Spectral Density Plot

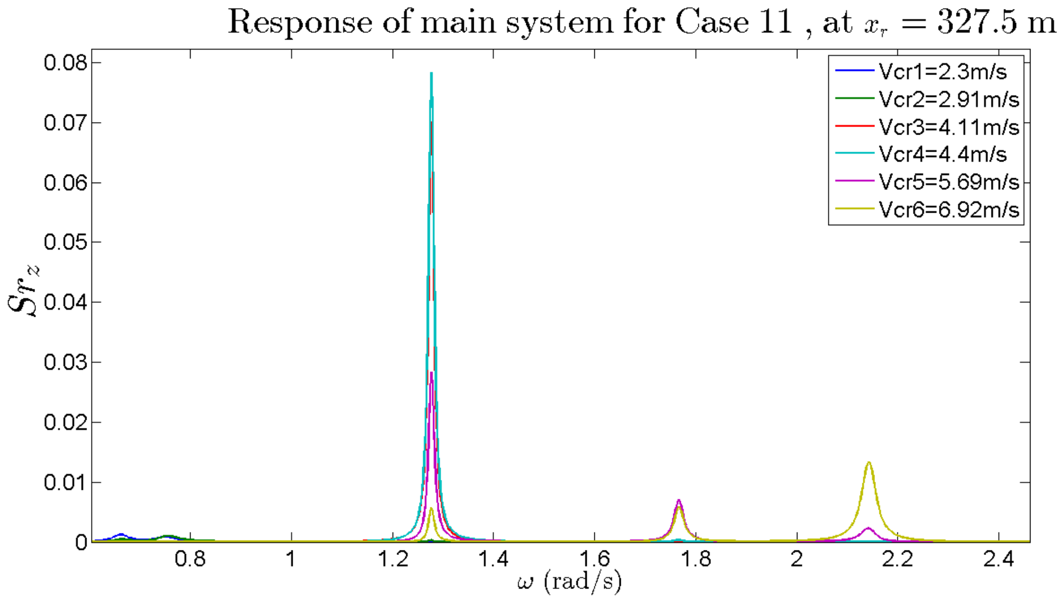

Figure 5.1: Response of bridge girder at x = 327.5 m for Case 11

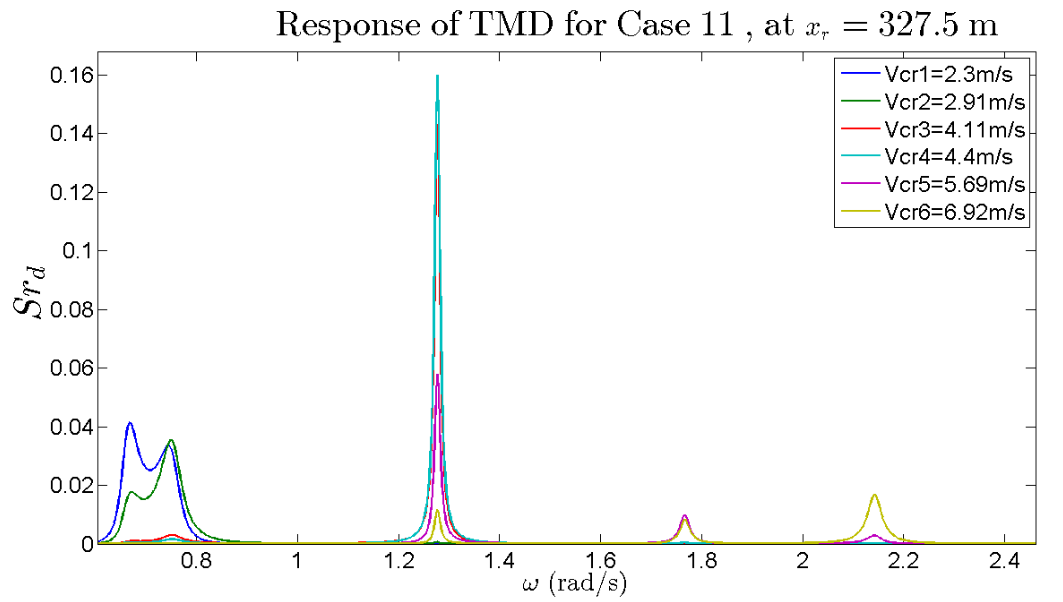

Figure 5.2: Response of TMD at x = 327.5 m for Case 11

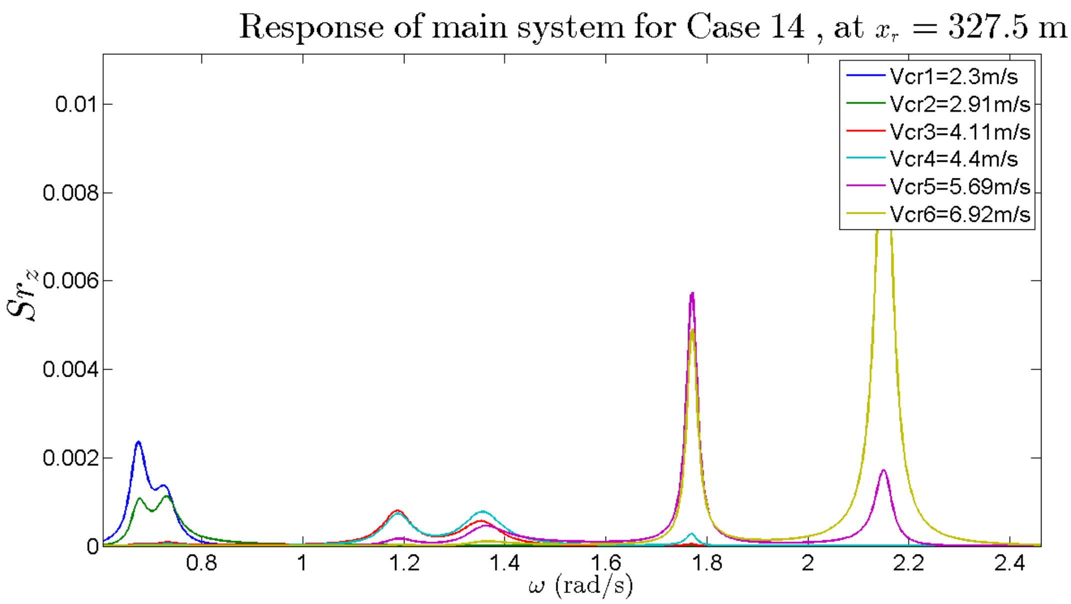

Figure 5.3: Response of bridge girder at x = 327.5 m for Case 14

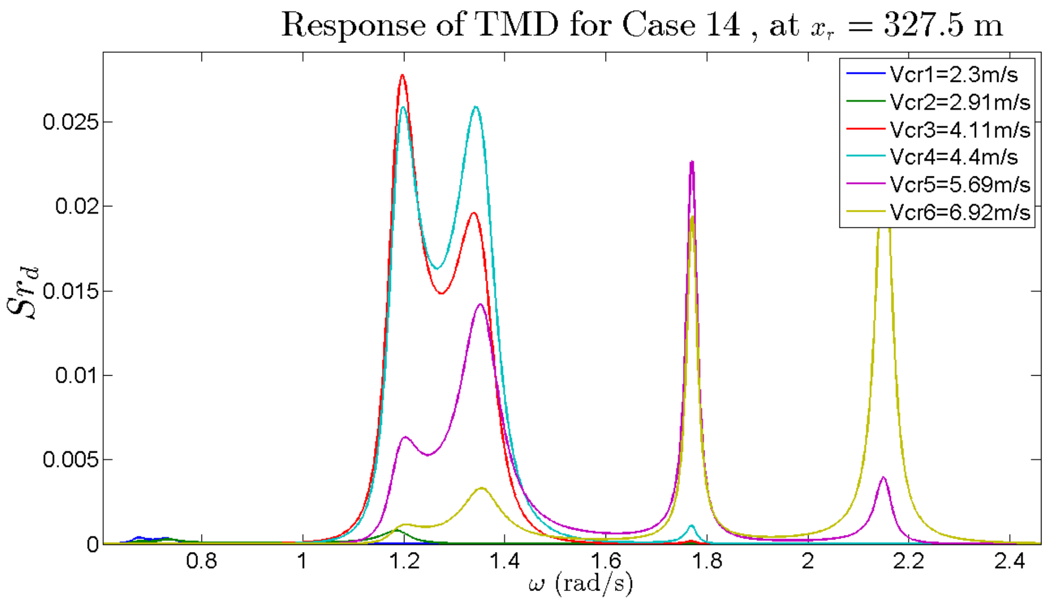

Figure 5.4: Response of TMD at x = 327.5 m for Case 14

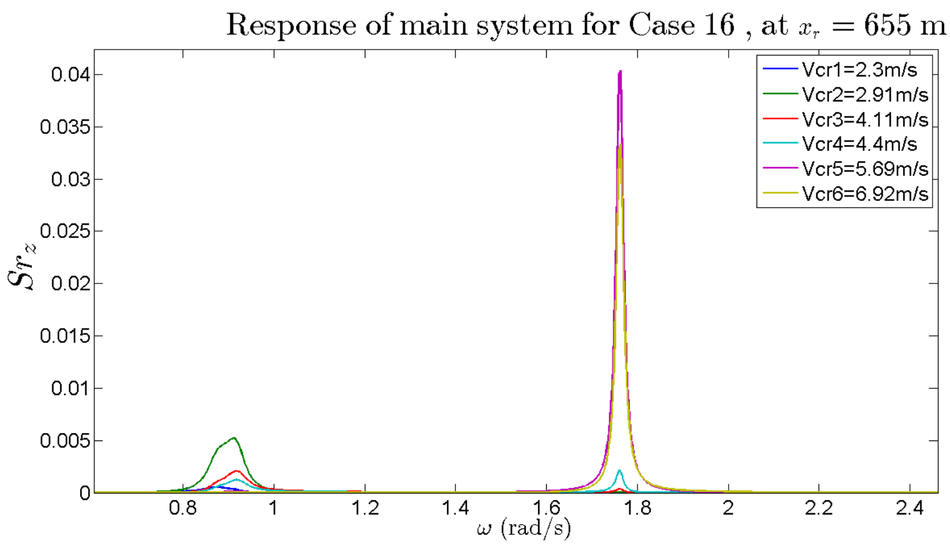

Figure 5.5: Response of bridge girder at x = 655 m Case 16

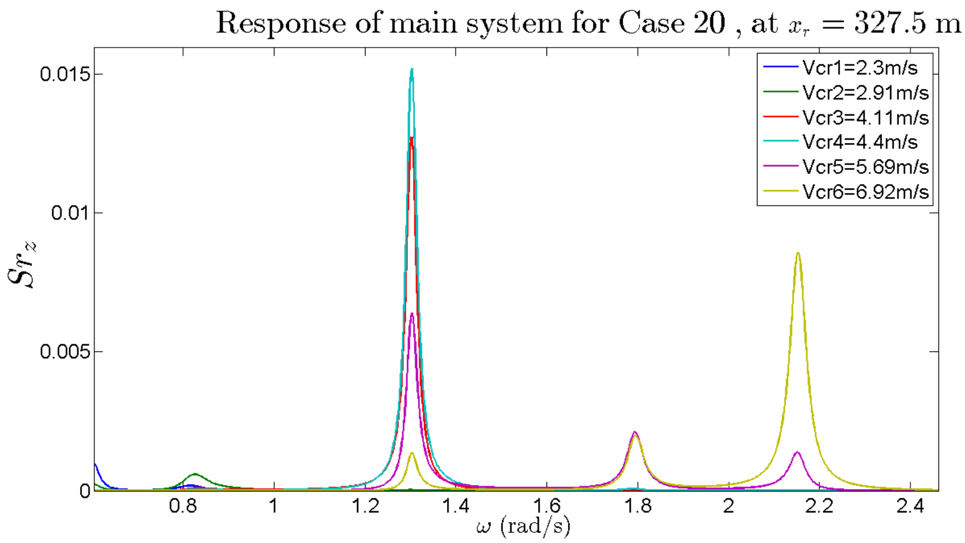

Figure 5.6: Response of bridge girder at x = 327.5 m for Case 20

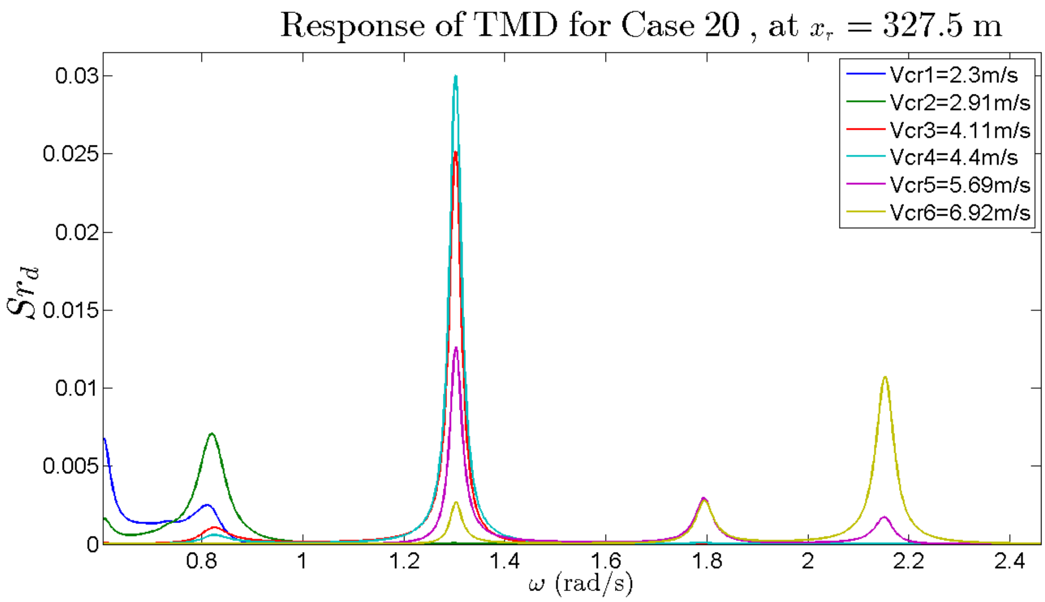

Figure 5.7: Response of TMD at x = 327.5 m for Case 20

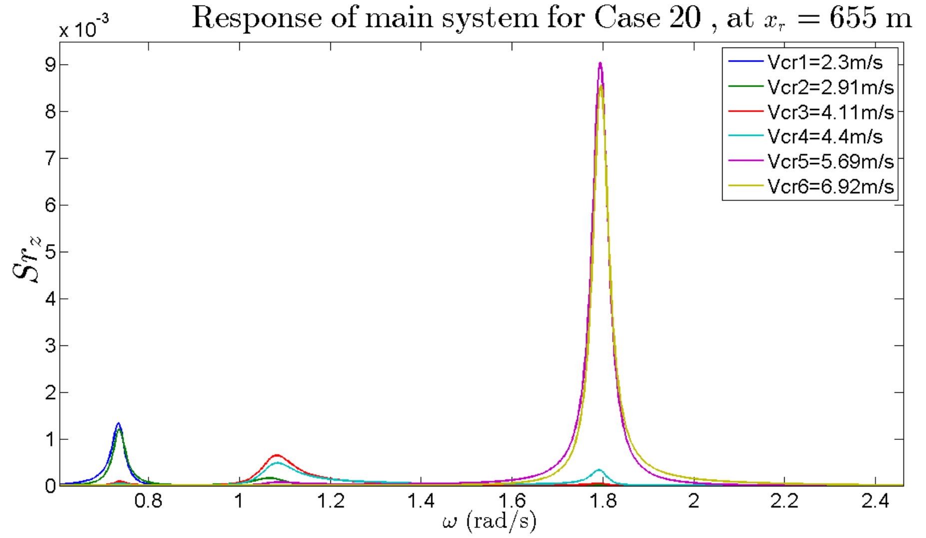

Figure 5.8: Response of bridge girder at x = 655 m

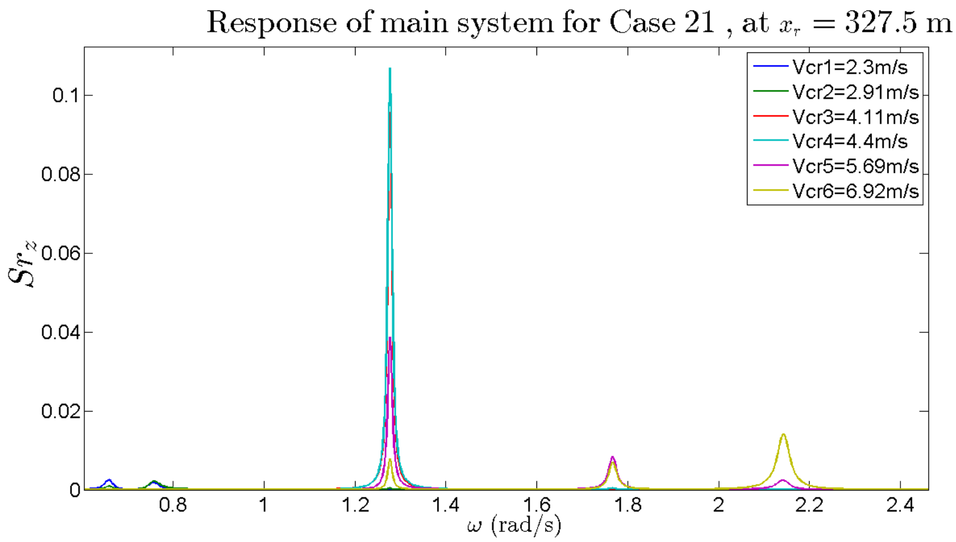

Figure 5.9: Response of the bridge girder at x = 327.5 m

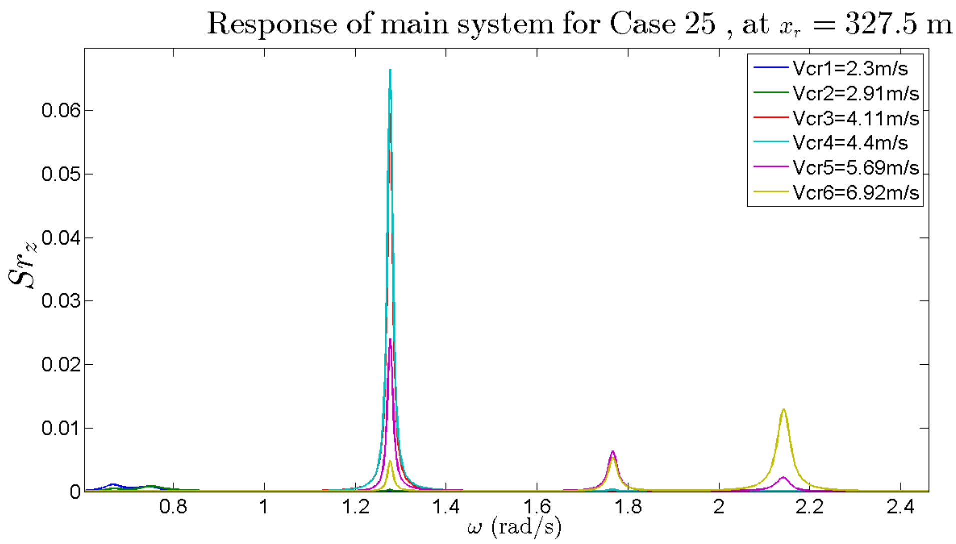

Figure 5.10: Response of TMD at x = 327.5 m

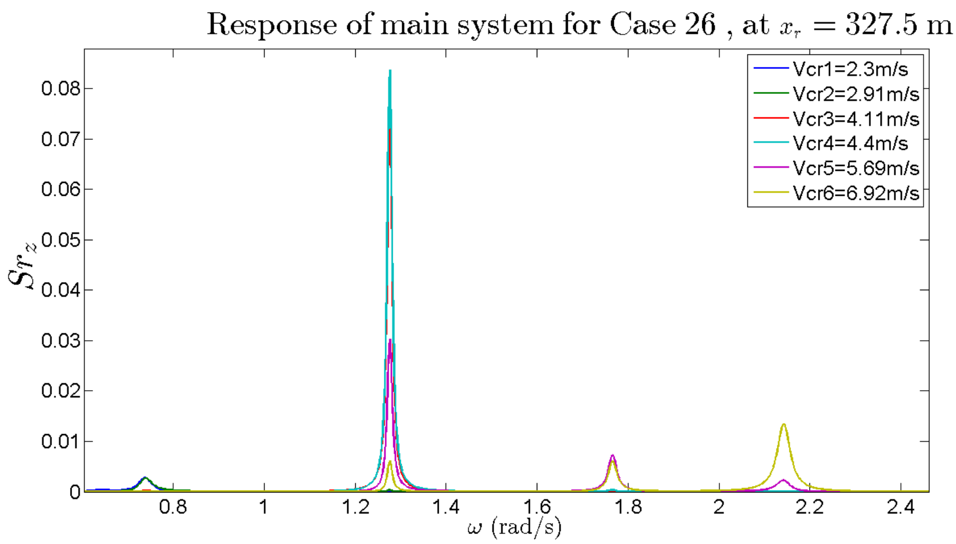

Figure 5.11: Response of the bridge girder at x = 327.5 m

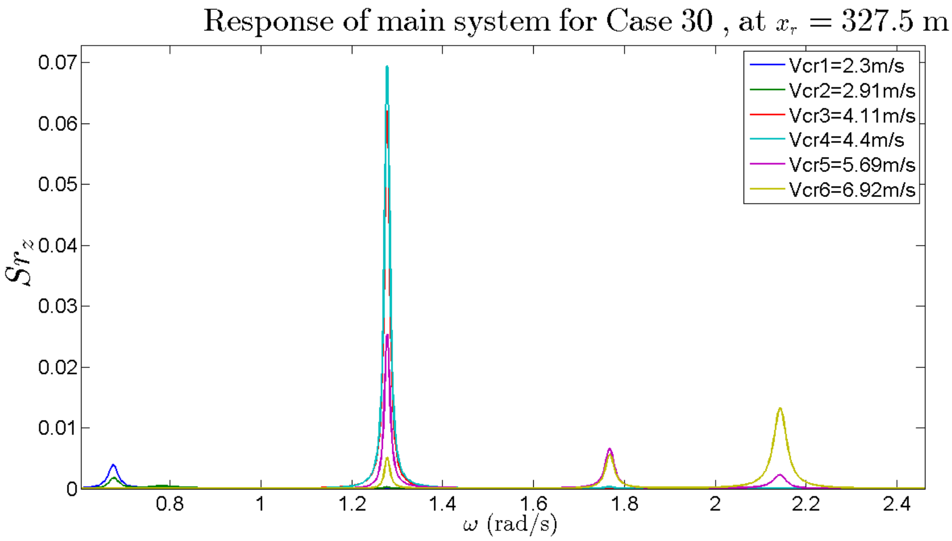

Figure 5.12: Response of the bridge girder at x = 327.5 m

5.0.6 Standard Deviation Plot

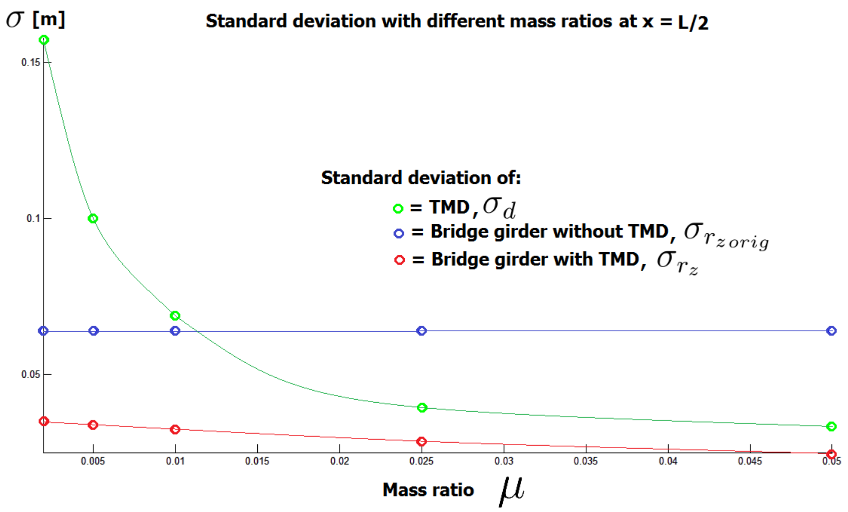

Figure 5.13: Response of TMD

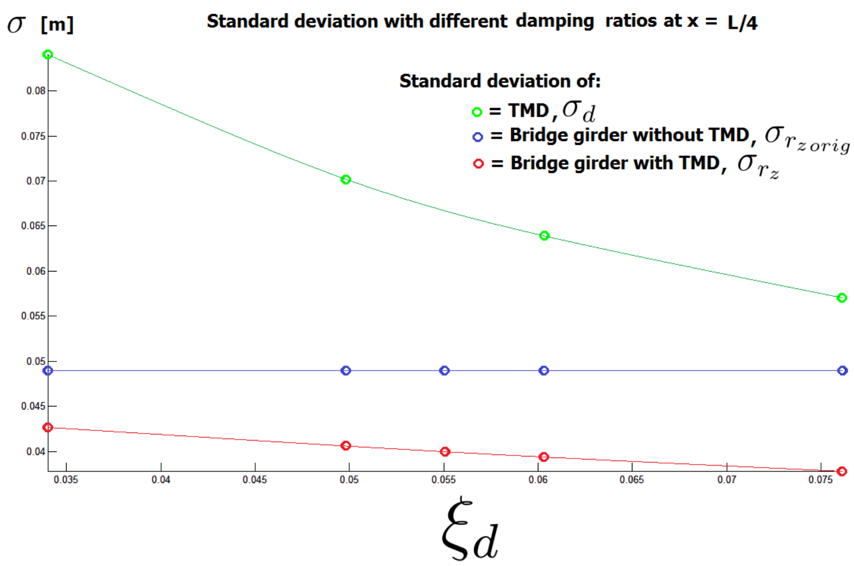

Figure 5.14: Values of standard deviation with different damping ratios at x = L/4

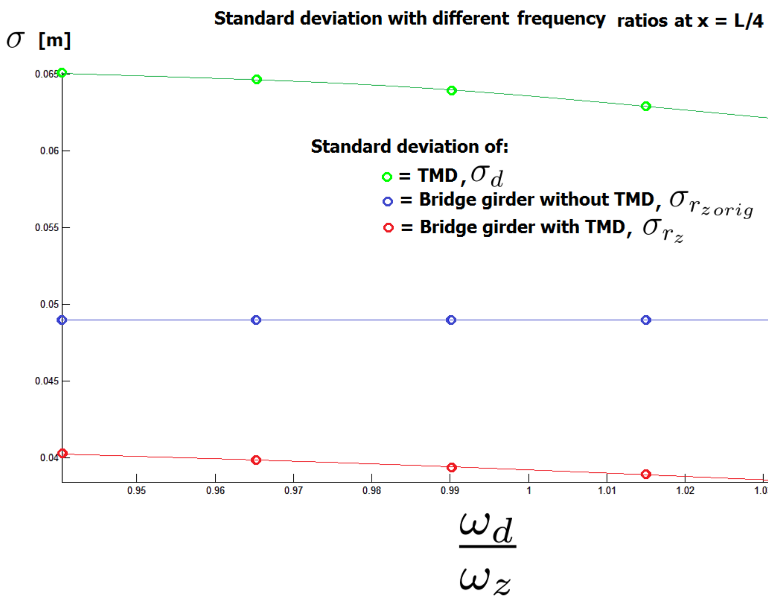

Figure 5.15: Values of standard deviation with different frequency ratios at x = L/4

5.0.7 Frequency response function (FRF) for vertical modes 3 - 6

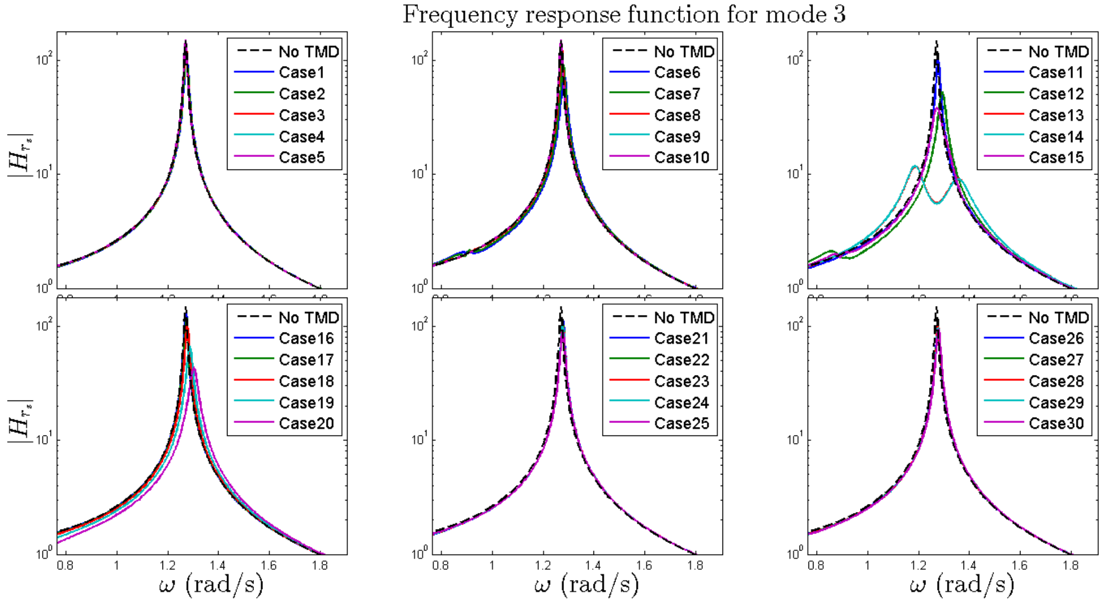

Figure 5.16: FRF for the third vertical mode, without TMD and with different cases. The definition of the cases can be found in tables 3.1, 3.2, 3.3, 3.4 and 3.5

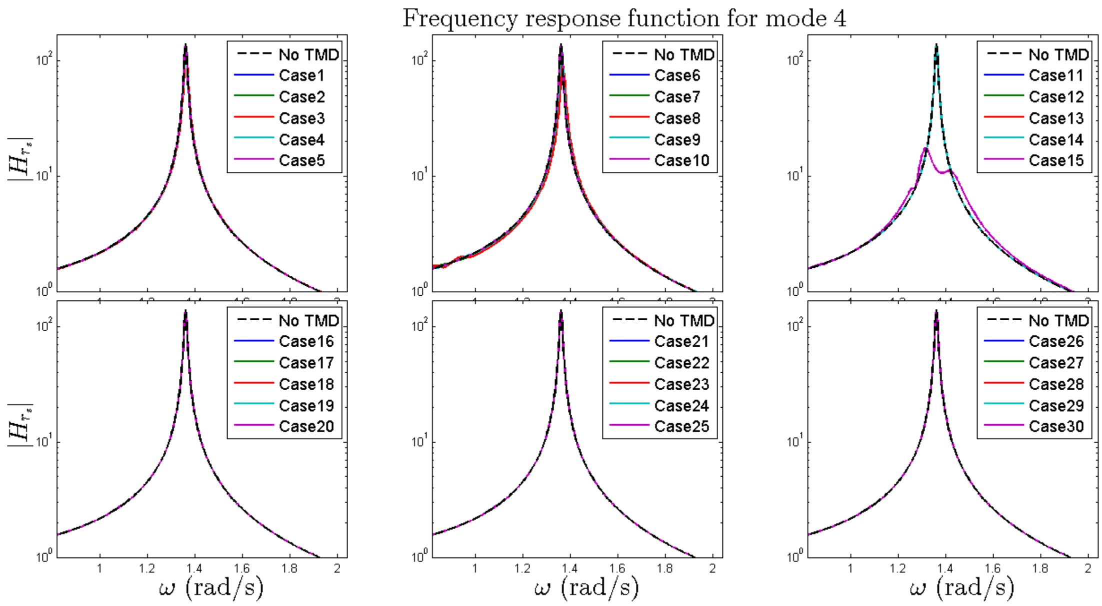

Figure 5.17: FRF for the fourth vertical mode without TMD and with different cases. The definition of the cases can be found in tables 3.1, 3.2, 3.3, 3.4 and 3.5

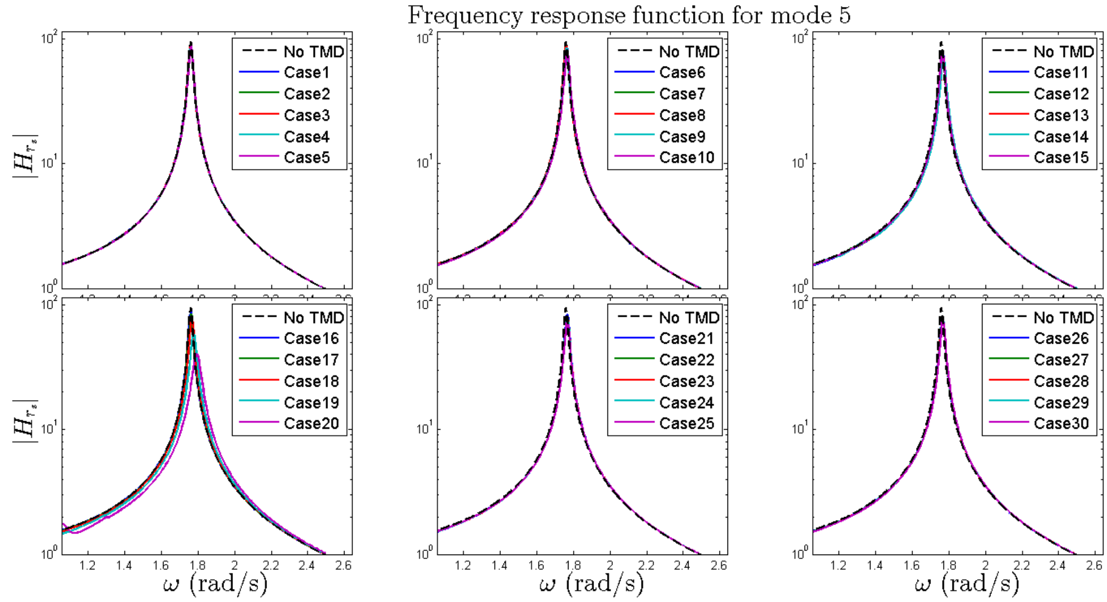

Figure 5.19: FRF vertical mode 5

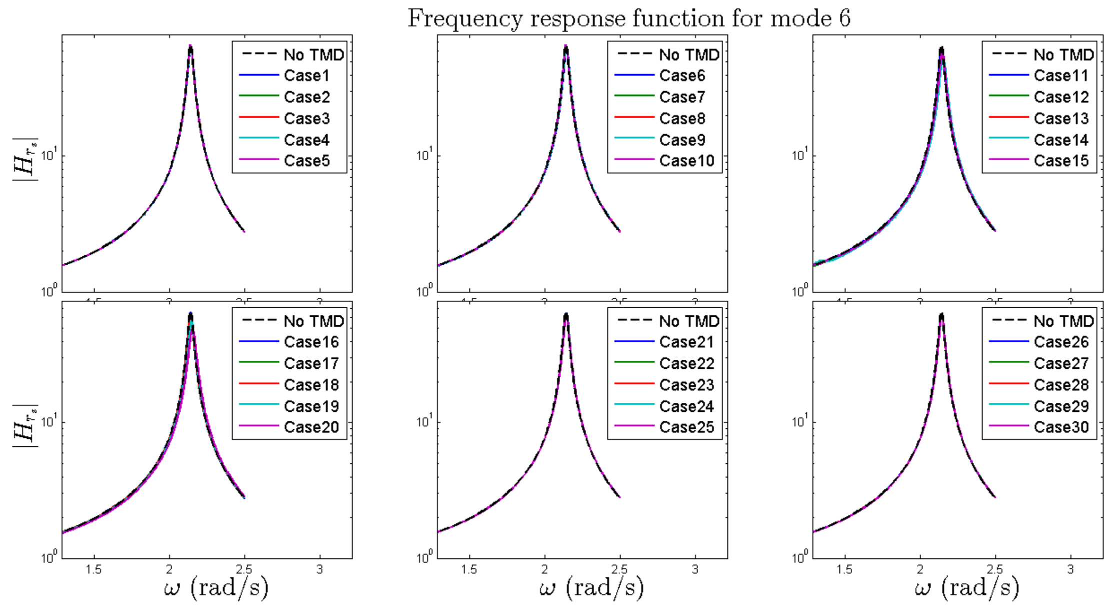

Figure 5.21: FRF vertical mode 6

5.0.8 Copy of the main part of the Matlab Script

```
% Matlab Script for Determine the Response of Vortex Induced Vibrations in Vertical
Modes at a Suspension Bridge
% Developed by Jens Einar Aaland during the master-thesis 2014
% All use at own risk.

clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUT PART
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L = 1310;
nxval = 100; dxval = (L+1)/nxval;
xval = 0:dxval:L;
mz_vec = [12937, 12937, 12937, 12937, 12937, 12937];

rho = 1.25; B = 18.3;   bz = 0.2; St = 0.16; lambda = 3.5; D = 3.25;   aL = 0.233;
sigma_CL = 3.9234/(sqrt(lambda/bz));
sigma_qz_hatt = sigma_CL; %*(D/B);

% Use constant Kaz ?
Kamax = 2.4112;
Constant_Kaz = Kamax*4*(D/B)^2;          % If Constant_Kaz=0, it will compute Kaz for
every wind-speed setting
                                 % based on Kamax. If constant Kaz should be used,
assign its value.

N_mod = 6;               % Number of eigenmodes
%N_j = 3;                % Number of tuned mass dampers (given in text-file)



% Eigenfrequencies [omega_z1,omega_z2,....,omega_z(N_mod)]:
omega_z = [0.71,0.9,1.27,1.36,1.76,2.14];


% Damping [xi_z1,xi_z2,...,xi_z(N_mod)]:
xi_z = [0.005,0.0051,0.0058,0.006,0.0077,0.01];

% Eigenmodes:
v = [3 4 6 7 10 12];             % Which of the eigenmodes is vertical
k_pi = 1:1:16;                   % Value of sine wave

% Load text file with eigenmodes:
load -ascii MatLabInPutFiles\AA.txt
K=length(AA(1,:));
N=length(AA(:,1));
ak = zeros(length(v),K);
for k = 1:length(v)
ak(k,:) = AA(4*(v(k))-1,:);
end

% Eigenmodes [phi_z1, phi_z2, ...., phi_z(N_mod)]:
phi_z = @(x)
[sum(ak(1,:)*sin(k_pi(:)*pi*x/L)),sum(ak(2,:)*sin(k_pi(:)*pi*x/L)),sum(ak(3,:)*sin(k_pi
(:)*pi*x/L)),

sum(ak(4,:)*sin(k_pi(:)*pi*x/L)),sum(ak(5,:)*sin(k_pi(:)*pi*x/L)),sum(ak(6,:)*sin(k_pi(
:)*pi*x/L))];

phi_val = [];
```

```
phi_xval = xval';
for j = 1:length(xval)
phi_val = [phi_val; phi_z(xval(j))];
end
clear AA SS
[max_val_phi,first_element_nr_max] = max(abs(phi_val));
location_first_max = xval(first_element_nr_max);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input data which in this case is given in files:
%
% N_j                     Number of TMD's
% x_d                     Position(s) of TMD(s)
% TMD_mode_damp           Which vertical mode the TMD(s) is tuned to damp
% my_d                    TMD(s) mass ratio
% xi_d                    TMD(s) damping ratio
% omega_d                 TMD(s) eigen-frequency
%                         (The files is loaded in the "output-part")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Make movie?
make_movie = 0;          % If make_movie=1, it will make movie, if make_movie=0 it will
not.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OUTPUT PART
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Plot of vertical eigenmodes:
vm = length(v);
fig = figure;
FigProperties(fig)
for n=1:vm,
    j=v(n);
    str=['\phi_{' num2str(j) '}'];
    Phizi = phi_val(:,n)/max(abs(phi_val(:,n)));
    subplot(vm,1,n);
    p = plot(xval,Phizi,'-');
    pPlotProperties(p);
    axis([0, L, -1.2, 1.2])
    grid
    hYLabel = ylabel(str);
    set(hYLabel, ...
    'FontSize',18);
    if n == vm
        hXLabel = xlabel('x');
        hXLabelProperties(hXLabel)
    end
     if n ==1
        hTitle = title('Vertical mode shapes, $$\phi_{z}\left(\frac{x}{L}\right)$$');
        set(hTitle, ...
    'interpreter','latex','FontName','Times','FontSize',16);
    end
    hold on
end
screen2png(['MatLabOutPutFiles\VerticalModes'])

% Get N_j, TMD_mode_damp and x_d from text file:
% One single TMD:
     load -ascii MatLabInPutFiles\TMDnumberAndPositionsSingleMode.txt
        N_j_vec_single_mode = TMDnumberAndPositionsSingleMode(:,1);
        TMD_mode_damp_vec_single_mode = TMDnumberAndPositionsSingleMode(:,2);
```

```matlab
        x_d_vec_single_mode = TMDnumberAndPositionsSingleMode(:,3);
        clear TMDnumberAndPositionsSingleMode
% Multiple TMD's:
     load -ascii MatLabInPutFiles\TMDnumberAndPositionsMultiMode.txt
        N_j_vec_multi_mode = TMDnumberAndPositionsMultiMode(:,1);
        TMD_mode_damp_vec_multi_mode = TMDnumberAndPositionsMultiMode(:,2:4);
        x_d_vec_multi_mode = TMDnumberAndPositionsMultiMode(:,5:7);
        clear TMDnumberAndPositionsMultiMode

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   % Making omega- axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   nomega = 20;
   diff_small_large = 4; % Difference between small and large omega steps
   step_omega_n = 0.25*((omega_z(1)+omega_z(N_mod))/2);
   omega_min = omega_z(1)-step_omega_n;
   omega_max = omega_z(1)+step_omega_n;

   domega_large_approx = 0.5*step_omega_n/(nomega); % Approximate domega_large
   domega_small = 0.5*step_omega_n/(diff_small_large*nomega+0.5);
   n_steps_small = ceil((2*step_omega_n/domega_small)+1);
   omega_small = zeros(N_mod,n_steps_small);
   for n = 1:N_mod
     omega_small(n,:) = omega_z(n)-step_omega_n:domega_small:omega_z(n)+step_omega_n;
   end
   Omega_start = omega_z(1)-
2*step_omega_n+0.5*domega_large_approx:domega_large_approx:omega_z(1)-step_omega_n-
0.5*domega_large_approx;
      Omega = Omega_start;
      omega_small_n = omega_small(1,:);
      m_max = length(omega_small(1,:))-1;
   for n = 1:N_mod-1
       omega_small_n_plus_1 = omega_small(n+1,:);
       for m = 1:m_max
       if omega_small_n(m)>omega_small_n_plus_1(m_max-m)
           omega_small_n = omega_small_n(1:m-1);
           omega_small_n_plus_1 = omega_small_n_plus_1(m_max-
m+1:numel(omega_small_n_plus_1));
           break
       end
       end

       delta_large = omega_small_n_plus_1(1)-omega_small_n(end);
       n_steps_large = ceil(delta_large/domega_large_approx);
       domega_large = delta_large/n_steps_large;
       omega_large =
omega_small_n(end)+0.5*domega_large:domega_large:omega_small_n_plus_1(1)-domega_large;

       if n == N_mod-1
           omega_large = [omega_large, omega_small_n_plus_1];
       end
       Omega =[Omega, omega_small_n, omega_large];
       omega_small_n = omega_small_n_plus_1;
       m_max = length(omega_small_n)-1;
   end
   dOmega = zeros(size(Omega));
   for n = 2:length(Omega)-1
       dOmega(n) = 0.5*(Omega(n+1)-Omega(n-1));
   end
   dOmega(1) = 2*0.5*(Omega(2)-Omega(1));
   dOmega(end) = 2*0.5*(Omega(end)-Omega(end-1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Running different cases
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LocationMode1 = [1;0;0;0;0;0]; LocationMode2 = [0;1;0;0;0;0]; LocationSeveralModes =
[0;0;1;0;0;0];
MassRatio = [0;0;0;1;0;0]; DampingTMD = [0;0;0;0;1;0]; FrequencyTMD = [0;0;0;0;0;1];
CaseStudy = [LocationMode1, LocationMode2, LocationSeveralModes, MassRatio, DampingTMD,
FrequencyTMD];
n_parameters = length(CaseStudy(1,:));

TotCaseNr = 0;
number_of_cases = 5;

Information_matrix = zeros(n_parameters*number_of_cases,17);
% Write information in the following format:
% Case , N_j , x_d (1x(1->3)-vector) , TMD_mode_damp (1x(1->3)-vector) , my_d
% (1x(1->3)-vector) , xi_d (1x(1->3)-vector) , omega_d (1x(1->3)-vector)

% Make the size of the critical-wind-speed for each mode (Vr),
% and the omega_s - vectors:
Vr = zeros(1,N_mod);
Omega_s = zeros(1,N_mod);

H_rz = zeros(length(Omega), N_mod, N_mod, n_parameters*number_of_cases);
H_orig = zeros(length(Omega), N_mod, N_mod, n_parameters*number_of_cases);

maxmaxrz_orig = zeros(1,number_of_cases*n_parameters);
maxstdrz_orig = zeros(1,number_of_cases*n_parameters);
maxkp_orig = zeros(1,number_of_cases*n_parameters);

maxmaxrz = zeros(1,number_of_cases*n_parameters);
maxstdrz = zeros(1,number_of_cases*n_parameters);
maxkp = zeros(1,number_of_cases*n_parameters);

maxmaxrd = zeros(1,number_of_cases*n_parameters);
maxstdrd = zeros(1,number_of_cases*n_parameters);
maxkp_rd = zeros(1,number_of_cases*n_parameters);

max_y_orig = zeros(number_of_cases*n_parameters,N_mod);
maxy_rz = zeros(number_of_cases*n_parameters,N_mod);
maxy_damper = zeros(number_of_cases*n_parameters,N_mod);
j_max_rz = zeros(number_of_cases*n_parameters,N_mod);

SIGMAhalf_rz = zeros(1,number_of_cases*n_parameters);
SIGMAhalf_rd = zeros(1,number_of_cases*n_parameters);
SIGMAquart_rz = zeros(1,number_of_cases*n_parameters);
SIGMAquart_rd = zeros(1,number_of_cases*n_parameters);

SIGMA_ORIGhalf = zeros(1,number_of_cases*n_parameters);
SIGMA_ORIGquart = zeros(1,number_of_cases*n_parameters);

for Parameter = 1:n_parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   CaseToStudy = CaseStudy(Parameter,:);

    if CaseToStudy(1)==1
    % Get N_j, TMD_mode_damp and x_d from text file:
        N_j_vec = N_j_vec_single_mode(1:number_of_cases,1);
        TMD_mode_damp_vec = TMD_mode_damp_vec_single_mode(1:number_of_cases,:);
        x_d_vec = x_d_vec_single_mode(1:number_of_cases,:);

    elseif CaseToStudy(2)==1
    % Get N_j, TMD_mode_damp and x_d from text file:
        N_j_vec = N_j_vec_single_mode(number_of_cases+1:2*number_of_cases,1);
        TMD_mode_damp_vec =
TMD_mode_damp_vec_single_mode(number_of_cases+1:2*number_of_cases,:);
        x_d_vec = x_d_vec_single_mode(number_of_cases+1:2*number_of_cases,:);
```

```
    elseif CaseToStudy(3)==1
        % Get N_j, TMD_mode_damp and x_d from text file:
        N_j_vec = N_j_vec_multi_mode(1:number_of_cases,1);
        TMD_mode_damp_vec = TMD_mode_damp_vec_multi_mode(1:number_of_cases,:);
        x_d_vec = x_d_vec_multi_mode(1:number_of_cases,:);
    else
        N_j_vec = 3*ones(number_of_cases,1);
        TMD_mode_damp_vec = repmat([1,2,1],[number_of_cases,1]);
        x_d_vec = repmat([327.5, 655, 982.5],[number_of_cases,1]);
    end
    for Case = 1:number_of_cases %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     TotCaseNr = TotCaseNr + 1;

    if CaseToStudy(4)==1
    % Get weight between TMD's mass ratios from text file:
        load -ascii MatLabInPutFiles\TMDmassRatioRUN2.txt
        my_d_vec = TMDmassRatioRUN2;
        clear TMDmassRatioRUN2
    else
        my_d_vec_first_row = zeros(1,N_j_vec(Case));
        for n = 1:N_j_vec(Case)
        my_d_vec_first_row(n) = 0.010;
        end
        my_d_vec = repmat(my_d_vec_first_row,[number_of_cases,1]);
    end

        my_d = my_d_vec(Case,:);
    if CaseToStudy(5)==1
        % Calculate TMD damping ratio(s) in a seperate Matlab script (Xis.m)
        xi_d_vec = XisRUN2(my_d_vec);
    else
        % Use Den Hartog for my = 0.01:
        xi_d_vec_first_row = zeros(1,N_j_vec(Case));
        for n = 1:N_j_vec(Case)
        xi_d_vec_first_row(n) = sqrt((3*0.01)/(8*(1+0.01)^3));
        end
        xi_d_vec = repmat(xi_d_vec_first_row,[number_of_cases,1]);
    end
    if CaseToStudy(6)==1
        % Calculate TMD frequency in a seperate Matlab script (Omegas.m):
    omega_d_vec = OmegasRUN2(my_d_vec,TMD_mode_damp_vec,omega_z);
    else
        % Use Den Hartog for my = 0.01:
        omega_d_vec = zeros(number_of_cases,N_j_vec(Case));
        for ncase = 1:number_of_cases
        for n = 1:N_j_vec(Case)
        omega_d_vec(ncase,n) = omega_z(TMD_mode_damp_vec(ncase,n))/(1+0.01);
        end
        end
    end

% All parameters to be used:
    N_j = N_j_vec(Case,:);
    TMD_mode_damp = TMD_mode_damp_vec(Case,:);
    x_d = x_d_vec(Case,:);
    my_d = my_d_vec(Case,:);
    xi_d = xi_d_vec(Case,:);
    omega_d = omega_d_vec(Case,:);

% Values of eigenmodes at damper positions:
phi_d = zeros(N_j,N_mod);
    for n = 1:N_j
          phi_d(n,:)=phi_z(x_d(n));
    end
```

```
% Integral of phi_z^2
int_phi_z_sq = zeros(1,N_mod);
dn = 0.01;
for n = 0:dn:L
    int_phi_z_sq = int_phi_z_sq + (phi_z(n)).^2*dn;
end

% Modal mass matrix:
Mz = zeros(N_mod);
for n = 1:N_mod
    Mz(n,n) = mz_vec(n)*int_phi_z_sq(n);
end
% TMD modal mass matrix:
Md = zeros(N_j);
for m = 1:N_j
       Md(m,m) = my_d(m)*Mz(TMD_mode_damp(m),TMD_mode_damp(m));
end

% Some matrixes to be used later on:

  SR_original = zeros(N_mod,N_mod,length(Omega),N_mod);
  Hinv_original = zeros(N_mod,N_mod,length(Omega),N_mod);
  H_original = zeros(N_mod,N_mod,length(Omega),N_mod);
  Sr_original = zeros(length(Omega),N_j,N_mod);
  S_eta_original = zeros(N_mod,N_mod,length(Omega),N_mod);
  Psi_original = zeros(1,N_mod,N_j);
  sigma_rz_original = zeros(N_j,N_mod);
  xi_aez_original = zeros(1,N_mod);
  xiz_original = zeros(N_mod,N_mod,N_mod);
  xi_aez_orig = zeros(N_j,N_mod,N_mod);

  Sqz = zeros(length(Omega),N_mod);
  SRzRz = zeros(N_mod,length(Omega),N_mod);
  SR = zeros(N_mod+N_j,N_mod+N_j,length(Omega),N_mod);
  H11 = zeros(N_mod,N_mod,length(Omega),N_mod);
  H12 = zeros(N_mod,N_j,length(Omega),N_mod);
  H21 = zeros(N_j,N_mod,length(Omega),N_mod);
  H22 = zeros(N_j,N_j,length(Omega),N_mod);
  Hinv = zeros(N_mod+N_j,N_mod+N_j,length(Omega),N_mod);
  H = zeros(N_mod+N_j,N_mod+N_j,length(Omega),N_mod);
  Sr = zeros(1+N_j,1+N_j,length(Omega),N_mod,N_j);
  S_eta = zeros(N_mod+N_j,N_mod+N_j,length(Omega),N_mod);
  xi_aez = zeros(N_mod,N_mod);

  Psi = zeros(N_j+1,N_mod+N_j,N_j);
  % Make the bottom part of the Psi - matrix:
  for n = 1:N_j
     Psi(1+n,N_mod+n,:) = 1;
  end

Kaz = zeros(N_mod,N_mod);
sigma_rz_guess_original = 0;
n_loops = zeros(N_j,N_mod);
Convergence_sigma_rz_all = [];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For every wind-speed setting
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for v = 1:N_mod
        Vr(v) = (D*omega_z(v))/(2*pi*St);

if Constant_Kaz == 0
for n = 1:N_mod
    if Vr(v)/Vr(n) < 0.9 || Vr(v)/Vr(n) > 2.45
        Kaz(n,v) = 0;
```

```
    else
        Kaz(n,v) = (4*(D/B)^2)*Kamax*(0.9/(((Vr(v)/Vr(n))-0.25)^(2)))*exp(-
1/(((Vr(v)/Vr(n))+0.02)^(24)))-0.18;
    end
end
else
    for n = 1:N_mod
    Kaz(n,v) = Constant_Kaz;
    end
end
      V = Vr(v);

      Omega_s(v) = (2*pi*V*St)/D;
      omega_s = Omega_s(v);


      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      % Response without mass damper:
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     Kzinv = zeros(N_mod,N_mod);
    for n = 1:N_mod
    Kzinv(n,n) = (1/(omega_z(n)^2*Mz(n,n)));
    end
    for xr = 1:N_j
        error = 1;
        Convergence_sigma_rz = [sigma_rz_guess_original];
while error > 0.001
        n_loops(xr,v) = n_loops(xr,v) + 1;
    for n = 1:N_mod
    xi_aez_original(1,n) = ((rho*B^2)/(4*mz_vec(v)))*Kaz(n,v)*(1-
(sigma_rz_guess_original/(aL*D)).^2);
    xiz_original(n,n,v) = (xi_z(n)-xi_aez_original(1,n));
    end
    omega_z_rel = zeros(N_mod,N_mod,length(Omega));
    % For every omega- setting:
   for o = 1:length(Omega)
  omega = Omega(o);

   for n = 1:N_mod
   omega_z_rel(n,n,o) = omega*(1/(omega_z(n)));
   end

   Sqz(o,v) = (((0.5*rho*B*sigma_qz_hatt*V^2)^2)/(sqrt(pi)*omega_s*bz))*exp(-((1-
omega/omega_s)/bz)^2);
   for m = 1:N_mod
   SRzRz(m,o,v) = 2*lambda*D*Sqz(o,v)*int_phi_z_sq(m);

   SR_original(m,m,o,v) = (Kzinv(m,m)^2*SRzRz(m,o,v));
   end

   Hinv_original(:,:,o,v) = eye(N_mod)-
omega_z_rel(:,:,o)^2*eye(N_mod)+2*1i*omega_z_rel(:,:,o)*xiz_original(:,:,v);
   H_original(:,:,o,v) = inv(Hinv_original(:,:,o,v));


   Psi_original(1,1:N_mod,xr) = phi_z(x_d(xr));

   Sr_original(o,xr,v) =
Psi_original(:,:,xr)*conj(H_original(:,:,o,v))*SR_original(:,:,o,v)*
   transpose(H_original(:,:,o,v))*transpose(Psi_original(:,:,xr));
   S_eta_original(:,:,o,v) =
conj(H_original(:,:,o,v))*SR_original(:,:,o,v)*transpose(H_original(:,:,o,v));
  end

    % Computing the standard deviation:
```

```
      sigma_rzrz_original_itt =
abs(sqrt(trapz(transpose(Sr_original(:,xr,v)).*dOmega)));

        error = abs(sigma_rzrz_original_itt - sigma_rz_guess_original);
        sigma_rz_guess_original = sigma_rzrz_original_itt;
        Convergence_sigma_rz = [Convergence_sigma_rz, sigma_rzrz_original_itt];
end
if xr == 1
Convergence_sigma_rz_all = [Convergence_sigma_rz_all, Convergence_sigma_rz];
end
xi_aez_orig(xr,:,v) = xi_aez_original(1,:);
sigma_rz_original(xr,v) = sigma_rzrz_original_itt;
if sigma_rzrz_original_itt < 0
    return
end
    end

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % Response with TMD (Tuned mass damper):
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % For every omega- setting:
   omega_z_rel = zeros(N_mod,N_mod,length(Omega));
   omega_d_rel = zeros(N_j,N_j,length(Omega));
  for o = 1:length(Omega)
  omega = Omega(o);

    xiz = zeros(N_mod,N_mod);
    Kzinv = zeros(N_mod,N_mod);
   for n = 1:N_mod
   omega_z_rel(n,n,o) = omega*(1/(omega_z(n)));
   xi_aez(n,v) = (rho*B^2/(4*mz_vec(v)))*Kaz(n,v);
   xiz(n,n) = (xi_z(n)-xi_aez(n,v));
   Kzinv(n,n) = (1/(omega_z(n)^2*Mz(n,n)));
   end

   xid = zeros(N_j,N_j);
   for n = 1:N_j
       omega_d_rel(n,n,o) = omega*(1/omega_d(n));
       xid(n,n) = xi_d(n);
   end

   Sqz(o,v) = (((0.5*rho*B*sigma_qz_hatt*V^2)^2)/(sqrt(pi)*omega_s*bz))*exp(-((1-
omega/omega_s)/bz)^2);
   for m = 1:N_mod
   SRzRz(m,o,v) = 2*lambda*D*Sqz(o,v)*int_phi_z_sq(m);

   SR(m,m,o,v) = (Kzinv(m,m)^2*SRzRz(m,o,v));
   end

   Dhatt = inv(Mz)*transpose(phi_d)*Md;
   H11(:,:,o,v) = eye(N_mod)-
omega_z_rel(:,:,o)^2*(eye(N_mod)+Dhatt*phi_d)+2*1i*omega_z_rel(:,:,o)*xiz;
   H12(:,:,o,v) = -omega_z_rel(:,:,o)^2*Dhatt;
   H21(:,:,o,v) = -omega_d_rel(:,:,o)^2*phi_d;
   H22(:,:,o,v) = eye(N_j)-omega_d_rel(:,:,o)^2+2*1i*omega_d_rel(:,:,o).*xid;

   Hinv(:,:,o,v) = [H11(:,:,o,v),H12(:,:,o,v);H21(:,:,o,v),H22(:,:,o,v)];
   H(:,:,o,v) = inv(Hinv(:,:,o,v));

    for xr = 1:N_j
   Psi(1,1:N_mod,xr) = phi_z(x_d(xr));

   Sr(:,:,o,v,xr) =
Psi(:,:,xr)*conj(H(:,:,o,v))*SR(:,:,o,v)*transpose(H(:,:,o,v))*transpose(Psi(:,:,xr));
```

```
   S_eta(:,:,o,v) = conj(H(:,:,o,v))*SR(:,:,o,v)*transpose(H(:,:,o,v));
    end

  end

end

% Saving the standard deviation for later use:
for xr = 1:N_j
    if x_d(xr) == L/2
        SIGMA_ORIGhalf(TotCaseNr) = max(abs(sigma_rz_original(xr,:)));
    end
    if x_d(xr) == L/4
        SIGMA_ORIGquart(TotCaseNr) = max(abs(sigma_rz_original(xr,:)));
    end
end

% End of "for all wind- speed setting"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear omega_z_rel Sqz SRzRz SR_original Hinv_original Psi_original Dhatt H11 H12 H21
H22 Psi


%%%%%%%%%%%%%
% Make a plot of convergence of original response:
if TotCaseNr ==1
    fig = figure;
    start = 1;
for v = 1:min(N_mod,6)
    subfig = subplot(3,2,v);
yplotval = Convergence_sigma_rz_all(start:start+n_loops(1,v));
xplotval = 0:1:n_loops(1,v);
start = start + length(yplotval);
p = plot(xplotval,yplotval);
set(p,'LineWidth',2.5);
    if v == 1
    hTitle = title('Convergence of $$\sigma_{r_{z}}$$ itterations of system without
TMD','HorizontalAlignment','left');
    set(hTitle, ...
    'interpreter','latex','FontName','Times','FontSize',22);
    end
hXLabel = xlabel('Number of itterations');
hYLabel = ylabel('$$\sigma_{r_{z}}$$    [m]');
set([hXLabel, hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16);
end
FigProperties(fig)

end

 % Splitting Sr_rz and Sr_rd:
  Sr_rz = zeros(length(Omega),N_j, N_mod);
  S_eta_rz = zeros(N_mod,length(Omega), N_mod);
  S_eta_rd = zeros(N_j,length(Omega), N_mod);
  Sr_rd = zeros(length(Omega),N_j, N_mod);
  for o = 1:length(Omega)
  for v = 1:N_mod
      for xr = 1:N_j
      Sr_rz(o,xr,v) = Sr(1,1,o,v,xr);
      Sr_rd(o,xr,v) = Sr(1+xr,1+xr,o,v,xr);
      end
      for n = 1:N_mod
      S_eta_rz(n,o,v) = S_eta(n,n,o,v);
      end
```

```
      for n = 1:N_j
      S_eta_rd(n,o,v) = S_eta(N_mod+n,N_mod+n,o,v);
      end
  end
  end
   %clear Sr S_eta

  % Computing the standard deviation:
  sigma_rzrz = zeros(N_j, N_mod);
  sigma_rdrd = zeros(N_j, N_mod);
  for v = 1:N_mod
  for xr = 1:N_j
      sigma_rzrz(xr, v) = abs(sqrt(trapz(transpose(Sr_rz(:,xr,v)).*dOmega)));
      sigma_rdrd(xr, v) = abs(sqrt(trapz(transpose(Sr_rd(:,xr,v)).*dOmega)));
  end
  end
  % Store the standard deviations for x = L/4 and x = L/2:
  for xr = 1:N_j
  if x_d(xr)==L/2
  SIGMAhalf_rz(TotCaseNr) = max(abs(sigma_rzrz(xr, :)));
  SIGMAhalf_rd(TotCaseNr) = max(abs(sigma_rdrd(xr, :)));
  end
  if x_d(xr)==L/4
  SIGMAquart_rz(TotCaseNr) = max(abs(sigma_rzrz(xr, :)));
  SIGMAquart_rd(TotCaseNr) = max(abs(sigma_rdrd(xr, :)));
  end
  end
  % For every "mean wind speed at resonance"- setting:
  %.......................................................
  % Compute:
  % * Response spectral densities
  % * Max/min values to be used in plotting
    min_x =  0.85*min(omega_z(1)); max_x =  1.15*min(omega_z(N_mod));


x_d_legend_line_rd_matr = zeros(length(Omega),N_j,N_mod);
x_d_legend_line_matr = zeros(length(Omega),N_j,N_mod);

S_rzrz_original_matr = zeros(length(Omega),N_j,N_mod);

S_rzrz_line_matr = zeros(length(Omega),N_j,N_mod);
S_rdrd_line_matr = zeros(length(Omega),N_j,N_mod);
  for v = 1:N_mod

% Store the computed S-specra in a matrix for each v- and xr- setting:
for xr = 1:N_j
x_d_legend_line_rd_matr(:,xr,v) = x_d(xr);
x_d_legend_line_matr(:,xr,v) = x_d(xr);

S_rzrz_line_matr(:,xr,v) = Sr_rz(:,xr,v);
S_rdrd_line_matr(:,xr,v) = Sr_rd(:,xr,v);

S_rzrz_original_matr(:,xr,v) =  Sr_original(:,xr,v);
end
     for n = 1:N_mod
     for o = 1:length(Omega)
         H_rz(o,n,v,TotCaseNr) = H(n,n,o,v);
         H_orig(o,n,v,TotCaseNr) = H_original(n,n,o,v);
     end
     end
  end
  clear S_rzrz_rel S_rdrd_rel
  clear H H_original

  for xr = 1:N_j
```

```
    S_rdrd_line_plot = [];
    S_rzrz_original_plot = [];

    S_rzrz_line_plot = [];
    legend_wind_speed1 = [];
    legend_wind_speed2 = [];

  for v = 1:N_mod
        S_rdrd_line_plot = [S_rdrd_line_plot, S_rdrd_line_matr(:,xr,v)];
        S_rzrz_line_plot = [S_rzrz_line_plot, S_rzrz_line_matr(:,xr,v)];

        S_rzrz_original_plot = [S_rzrz_original_plot, S_rzrz_original_matr(:,xr,v)];
      legend_wind_speed1 = [legend_wind_speed1, v];
      legend_wind_speed2 = [legend_wind_speed2, round(100*Vr(v))/100];
  end
  if x_d(xr) == L/2 || x_d(xr) == L/4
 % Plotting of S_rzrz:
        titletextS = ['Response of main system for Case ' num2str(TotCaseNr) ' , at
$$x_{r}$$ = ' num2str(x_d(xr)) ' m'];
        ylabeltextS = ['$$ Sr_{z}$$'];
        saveAsRz = ['MatLabOutPutFiles\VSTMDmmResponceMainSystemXr'
num2str(floor(x_d(xr))) 'Case' num2str(TotCaseNr) ''];
     fig =
PlotResponseSpectra(S_rzrz_line_plot,Omega,legend_wind_speed1,legend_wind_speed2,min_x,
max_x,titletextS,ylabeltextS,saveAsRz);
     close(fig)
 % Plotting of S_rdrd:
        titletextS = ['Response of TMD for Case ' num2str(TotCaseNr) ' , at $$x_{r}$$ =
' num2str(x_d(xr)) ' m'];
        ylabeltextS = ['$$ Sr_{d}$$'];
        saveAsRd = ['MatLabOutPutFiles\VSTMDmmResponceTMDXr' num2str(floor(x_d(xr)))
'Case' num2str(TotCaseNr) ''];
     fig =
PlotResponseSpectra(S_rdrd_line_plot,Omega,legend_wind_speed1,legend_wind_speed2,min_x,
max_x,titletextS,ylabeltextS,saveAsRd);
     close(fig)
 % Plotting of S_rzrz original (without TMD):
        titletextS = ['Response of original system without TMD at $$x_{r}$$ = '
num2str(x_d(xr)) ' m'];
        ylabeltextS = ['$$ Sr_{z}$$'];
        saveAsRz_orig = ['MatLabOutPutFiles\VSTMDmmOrigResponceMainSystemXr'
num2str(floor(x_d(xr))) ''];
     fig =
PlotResponseSpectra(S_rzrz_original_plot,Omega,legend_wind_speed1,legend_wind_speed2,mi
n_x,max_x,titletextS,ylabeltextS,saveAsRz_orig);
     close(fig)

  end
  end
clear S_rdrd_line_matr S_rzrz_line_matr S_rzrz_original_matr S_rdrd_line_plot
S_rzrz_line_plot S_rzrz_original_plot
clear legend_wind_speed1   legend_wind_speed2

 % Time domain simulation
jn_min = 12;
if TotCaseNr == 1 || TotCaseNr == 11 ||TotCaseNr == 20 || TotCaseNr == 25 ||TotCaseNr
== 26 || TotCaseNr == 27
    || TotCaseNr == 28 || TotCaseNr == 29 || TotCaseNr == 30
jn = 1000; % jn = number of time domain simulations
else
jn = jn_min;
end
dt = 0.1;
time = 0:dt:600;
```

```
t_sim = 0:dt:30;

tn = length(time);
t_simn = length(t_sim);

rz = zeros(jn,tn,N_j,N_mod);     rd = zeros(jn,tn,N_j,N_mod);
k_p = zeros(jn,N_j,N_mod);       k_p_rd = zeros(jn,N_j,N_mod);
maxrz = zeros(jn,N_j,N_mod);     maxrd = zeros(jn,N_j,N_mod);
meanrz = zeros(jn,N_j,N_mod);    meanrd = zeros(jn,N_j,N_mod);
stdrz = zeros(jn,N_j,N_mod);     stdrd = zeros(jn,N_j,N_mod);

rz_orig = zeros(jn,tn,N_j,N_mod);
k_p_orig = zeros(jn,N_j,N_mod);
maxrz_orig = zeros(jn,N_j,N_mod);
meanrz_orig = zeros(jn,N_j,N_mod);
stdrz_orig = zeros(jn,N_j,N_mod);

 phi_mode = phi_val;
  % Make the rand-matrix
 rand_matrix = zeros(jn,length(Omega));
 for j = 1:jn
     for o = 1:length(Omega)
         rand_matrix(j,o) = 2*pi*rand;
     end
 end
 for v =1:N_mod

 for j = 1:jn
    rz_sum = zeros(1,tn);
    rd_sum = zeros(1,tn);
    rz_sum_orig = zeros(1,tn);

 for xr = 1:N_j;

 for k=1:length(Omega)
    omegak=Omega(k);
    dOmegak = dOmega(k);
    Srzk = Sr_rz(k,xr,v);

    rz_sum = rz_sum+sqrt(2*Srzk*dOmegak)*cos(omegak*time+rand_matrix(j,k));

    Srdk = Sr_rd(k,xr,v);
    rd_sum = rd_sum+sqrt(2*Srdk*dOmegak)*cos(omegak*time+rand_matrix(j,k));

    Srzorigk = Sr_original(k,xr,v);
    rz_sum_orig =
rz_sum_orig+sqrt(2*Srzorigk*dOmegak)*cos(omegak*time+rand_matrix(j,k));
 end

rz(j,:,xr,v) = rz_sum;
maxrz(j,xr,v) = max(abs(rz(j,:,xr,v)));
meanrz(j,xr,v) = mean((rz(j,:,xr,v)));
stdrz(j,xr,v) = std(rz(j,:,xr,v));
k_p(j,xr,v) = max(abs(rz(j,:,xr,v)))/std(rz(j,:,xr,v));

rz_sum = zeros(1,tn);

rd(j,:,xr,v) = rd_sum;
maxrd(j,xr,v) = max(abs(rd(j,:,xr,v)));
meanrd(j,xr,v) = mean((rd(j,:,xr,v)));
stdrd(j,xr,v) = std(rd(j,:,xr,v));
k_p_rd(j,xr,v) = max(abs(rd(j,:,xr,v)))/std(rd(j,:,xr,v));

rd_sum = zeros(1,tn);
```

```
rz_orig(j,:,xr,v) = rz_sum_orig;
maxrz_orig(j,xr,v) = max(abs(rz_orig(j,:,xr,v)));
meanrz_orig(j,xr,v) = mean((rz_orig(j,:,xr,v)));
stdrz_orig(j,xr,v) = std(rz_orig(j,:,xr,v));
k_p_orig(j,xr,v) = max(abs(rz_orig(j,:,xr,v)))/std(rz_orig(j,:,xr,v));

rz_sum_orig = zeros(1,tn);
 end
 end

 for xr = 1:N_j
if v == 1
if TotCaseNr == 1 || TotCaseNr == 20

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Statistics on original system without TMD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
titletext_orig = ['Statistics from   ' num2str(jn) '  simulations for main system
without TMD'];
saveAsPsRzor = ['MatLabOutPutFiles\DistributionPeaksRzOrigAtXpos'
num2str(floor(x_d(xr))) ''];
fig =
PlotStatistics(maxrz_orig,stdrz_orig,k_p_orig,N_j,v,jn,time,rz_orig,titletext_orig,save
AsPsRzor);
close(fig)

maxmaxrz_orig(TotCaseNr) = max(maxrz_orig(:,xr,v));
maxstdrz_orig(TotCaseNr) = max(stdrz_orig(:,xr,v));
maxkp_orig(TotCaseNr) = max(k_p_orig(:,xr,v));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Statistics system with TMD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
titletext_rz = ['Statistics from   ' num2str(jn) '  simulations for Case '
num2str(TotCaseNr) ''];
saveAsPsRz = ['MatLabOutPutFiles\DistributionPeaksRz Case' num2str(TotCaseNr) ' atXpos'
num2str(floor(x_d(xr))) ''];
fig = PlotStatistics(maxrz,stdrz,k_p,N_j,v,jn,time,rz,titletext_rz,saveAsPsRz);
close(fig)

maxmaxrz(TotCaseNr) = max(maxrz(:,xr,v));
maxstdrz(TotCaseNr) = max(stdrz(:,xr,v));
maxkp(TotCaseNr) = max(k_p(:,xr,v));

titletext_rd = ['Statistics from   ' num2str(jn) '  simulations for Case '
num2str(TotCaseNr) ''];
fig = PlotStatistics_Rd(maxrz,stdrz,k_p,N_j,x_d,v,jn,time,rz,titletext_rd,TotCaseNr);
close(fig)
maxmaxrd(TotCaseNr) = max(maxrd(:,xr,v));
maxstdrd(TotCaseNr) = max(stdrd(:,xr,v));
maxkp_rd(TotCaseNr) = max(k_p_rd(:,xr,v));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plotting of time series:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if x_d(xr) == L/2 || x_d(xr) == L/4
  % Plotting of r_z with TMD installed (Done in a separate Matlab- script
(PlotTimeSeries.m)):
  titletextTS = ['Time domain simulation of $r_{z}$ with TMD installed, Case'
num2str(TotCaseNr)
      ' at x = ' num2str(x_d(xr)) ' m, $$V_{cr}$$ =' num2str(round(1000*Vr(v))/1000)
''];
  fig =
PlotTimeSeries(time,real(rz),maxrz,stdrz,k_p,TotCaseNr,jn,xr,x_d,v,titletextTS);
  close(fig)
```

```
  % Plotting of r_d with TMD installed (Done in a separate Matlab- script
(PlotTimeSeriesRd.m)):
  titletextTS_Rd = ['Time domain simulation of $r_{d}$ with TMD installed, Case'
num2str(TotCaseNr)
      ' at x = ' num2str(x_d(xr)) ' m, $$V_{cr}$$ =' num2str(round(1000*Vr(v))/1000)
''];
  fig =
PlotTimeSeriesRd(time,real(rd),maxrd,stdrd,k_p,TotCaseNr,jn,xr,x_d,v,titletextTS_Rd);
  close(fig)

  % Plotting of original r_z without TMD installed (Done in a separate Matlab- script
(PlotTimeSeriesOrig.m)):
  titletextTS_orig = ['Time domain simulation of original $r_{z}$ without TMD at x = '
num2str(x_d(xr))
      ' m, $$V_{cr}$$ =' num2str(round(1000*Vr(v))/1000) ''];
  fig =
PlotTimeSeriesOrig(time,real(rz_orig),real(maxrz_orig),stdrz_orig,k_p_orig,TotCaseNr,jn
,xr,x_d,v,titletextTS_orig);
    close(fig)
    end
end
end
 end
 end
 clear titletext_orig titletext_rd
clear Sr_rz Sr_rd Sr_original
clear rz maxrz stdrz k_p titletextTS
clear rd maxrd stdrd k_p titletextTS_Rd
clear maxrz_orig stdrz_orig k_p_orig titletextTS_orig
%%%%%%%%%%%%%%%%%

 eta_n   = zeros(length(time),N_mod,jn_min);
 eta_rd_n = zeros(length(time),N_j,jn_min);
 eta_orig_n = zeros(length(time),N_mod,jn_min);


%%%%%%%%%%%%%%%%%%%%%%%%
% Finding max response
%%%%%%%%%%%%%%%%%%%%%%%

 for v = 1:N_mod

for j = 1:jn_min
 for n = 1:N_mod
     eta_sum = zeros(1,tn);
     eta_sum_original = zeros(1,tn);

 for k=1:length(Omega)
    omegak=Omega(k);
    dOmegak = dOmega(k);

    Setak = S_eta_rz(n,k,v);
    eta_sum = eta_sum + sqrt(2*Setak*dOmegak)*cos(omegak*time+rand_matrix(j,k));

    Setak_original = S_eta_original(n,n,k,v);
    eta_sum_original = eta_sum_original +
sqrt(2*Setak_original*dOmegak)*cos(omegak*time+rand_matrix(j,k));
 end
 eta_n(:,n,j) = eta_sum;
 eta_orig_n(:,n,j)  = eta_sum_original;
 end
 for m = 1:N_j
      eta_sum_rd = zeros(1,tn);
  for k=1:length(Omega)
    omegak=Omega(k);
```

```
    dOmegak = dOmega(k);
    Setak_rd = S_eta_rd(m,k,v);
    eta_sum_rd = eta_sum_rd +
sqrt(2*Setak_rd*dOmegak)*cos(omegak*time+rand_matrix(j,k));
  end
  eta_rd_n(:,m,j) = eta_sum_rd;
 end

 r_z_sum_of_modes = zeros(length(xval),tn,jn_min);
 r_z_sum_of_modes_original = zeros(length(xval),tn,jn_min);
 r_d_sum_of_modes = zeros(tn,N_j,jn_min);

 for x = 1:length(xval)
for n = 1:N_mod
    r_z_sum_of_modes(x,:,j) = r_z_sum_of_modes(x,:,j) +
phi_mode(x,n)*transpose(eta_n(:,n,j));
    r_z_sum_of_modes_original(x,:,j) = r_z_sum_of_modes_original(x,:,j) +
phi_mode(x,n)*transpose(eta_orig_n(:,n,j));
end
 end
for m = 1:N_j
     r_d_sum_of_modes(:,m,j) = 1*transpose(eta_rd_n(:,m,j));
end

  damper_displacement_x = zeros(N_j);

for xr = 1:N_j
damper_displacement_x(xr) = x_d(xr);
end
end
rz_orig_plot = rz_orig(:,1:length(t_sim),:,:);
[max_y_orig(TotCaseNr,v), j_max_rz(TotCaseNr,v)] =
max(max(max(abs(r_z_sum_of_modes_original(:,:,1:jn_min)))));
maxy_rz(TotCaseNr,v) = max(max(max(abs(r_z_sum_of_modes(:,:,1:jn_min)))));
maxy_damper(TotCaseNr,v) = max(max(max(abs(r_d_sum_of_modes(:,:,1:jn_min)))));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MOVIE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if make_movie == 1;
    if v == 1 || v == 2
    for j = j_max_rz(TotCaseNr,v)
% Original response, several points:
[rz_orig_for_multiple_xr,x_points] =
OriginalResponseSeveralPoints(xval,N_mod,Omega,dOmega,t_sim,rho,B,mz_vec,v,Kaz,j,xi_z,a
L,D,omega_z,sigma_qz_hatt,Omega_s,bz,Vr,lambda,int_phi_z_sq,phi_z,Kzinv,rand_matrix);

MakeMovie(max_y_orig(:,v),maxy_rz(:,v),TotCaseNr,maxy_damper(:,v),r_z_sum_of_modes_orig
inal,r_z_sum_of_modes,r_d_sum_of_modes,v,xval,t_sim,time,j,damper_displacement_x,rz_ori
g_plot,rz_orig_for_multiple_xr,x_points,L,Vr,N_j,x_d);
    end
    end
end
 end
clear S_eta_rz S_eta_rd r_z_sum_of_modes_original r_z_sum_of_modes
clear rz_orig_for_multiple_xr x_points
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Information_matrix(TotCaseNr,1) = TotCaseNr;
Information_matrix(TotCaseNr,2) = N_j;
Information_matrix(TotCaseNr,3:2+N_j) = x_d;
Information_matrix(TotCaseNr,6:5+N_j) = TMD_mode_damp;
Information_matrix(TotCaseNr,9:8+N_j) = my_d;
Information_matrix(TotCaseNr,12:11+N_j) = xi_d;
Information_matrix(TotCaseNr,15:14+N_j) = omega_d;
```

```
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot of max statistical values in different cases:
for v = 1:N_mod
Y1 = [];
Y2 = [];
Y3 = [];
for casenr = 1:TotCaseNr
Y1 = [Y1; maxy_rz(casenr,v)];
Y2 = [Y2; max_y_orig(casenr,v)];
Y3 = [Y3; maxy_damper(casenr,v)];
 end
width1 = 0.5; width2 = width1/2; width3 = width2/4;

fig = figure;
bary1 = bar(Y1,width1);
hold on
bary2 = bar(Y2,width2);
bary3 = bar(Y3,width3);

hold off
set(bary1,'FaceColor','r','EdgeColor','r');
set(bary2,'FaceColor','b','EdgeColor','b');
set(bary3,'FaceColor','g','EdgeColor','g');
hLegend = legend('Main system','Without TMD' , 'TMDs');
hTitle = title('Maximum response for each case');
hYLabel = ylabel('Max response    [m] ');
hXLabel = xlabel('Case');
BarPlotProperties(fig, hTitle, hXLabel, hYLabel);
screen2png(['MatLabOutPutFiles\MaxResponseRzRdCasesVcr' num2str(v) '']);
close(fig)
end
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plotting of standard deviation:
SigmaPlot(SIGMAhalf_rz, SIGMAhalf_rd, SIGMAquart_rz,SIGMAquart_rd,SIGMA_ORIGhalf,
SIGMA_ORIGquart,Information_matrix)

% Plotting the FRF at:
Hplot(H_rz,TotCaseNr,H_orig,Omega,omega_z)
```

5.0.9 Copy of input-data and sub- Script used in the Matlab Script

MatLabInPutFiles\AA.txt

1	0	0.0383	0	-0.0021	0	0.0004	0	-0.0001	0	0	0	0	0	0	0
0.8955	0	-0.0287	0	0.0048	0	-0.0016	0	0.0006	0	-0.0003	0	0.0001	0	-0.0001	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0.0402	0	-0.0039	0	0.0008	0	-0.0002	0	0.0001	0	0	0	0
0	0.4818	0	-0.0555	0	0.0128	0	-0.0048	0	0.0020	0	-0.0010	0	0.0005	0	-0.0003
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	-0.0058	0	-0.0009	0	-0.0002	0	-0.0001	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.8010	0	1	0	0.0521	0	0.0141	0	0.0054	0	0.0025	0	0.0013	0	0.0007	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1416	0	1	0	0.0198	0	-0.0026	0	0.0006	0	-0.0002	0	0	0	0	0
0.4937	0	0.2171	0	-0.0444	0	0.0118	0	-0.0048	0	0.0021	0	-0.0011	0	0.0005	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0.8131	0	-0.2185	0	-0.0439	0	-0.0157	0	-0.007	0	-0.0036	0	-0.002	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0058	0	1	0	-0.0004	0	-0.0001	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-0.1817	0	-0.2059	0	0.006	0	-0.0012	0	0.0003	0	-0.0001	0	0	0	0
0	1	0	0.1406	0	-0.0206	0	0.0073	0	-0.0028	0	0.0014	0	-0.0007	0	0.0004
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0.2884	0	0.1161	0	0.011	0	-0.0015	0	0.0003	0	-0.0001	0	0	0	0	0
0.8232	0	1	0	-0.0239	0	0.0073	0	-0.0024	0	0.0012	0	-0.0005	0	0.0003	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1572	0	0.0742	0	1	0	-0.0194	0	-0.0057	0	-0.0024	0	-0.0012	0	-0.0006	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-0.1016	0	1	0	0.212	0	-0.0035	0	0.0009	0	-0.0003	0	0.0001	0	0
0	0.386	0	0.5646	0	-0.0712	0	0.0225	0	-0.0091	0	0.0041	0	-0.0021	0	0.0011
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.0009	0	0.0004	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-0.0297	0	-0.0056	0	-0.002	0	-0.0009	0	-0.0005	0	-0.0003	0	-0.0002	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.0297	0	1	0	-0.003	0	-0.0009	0	-0.0004	0	-0.0002	0	-0.0001	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.0057	0	0.0028	0	1	0	-0.0008	0	-0.0003	0	-0.0001	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

MatLabInPutFiles\TMDnumberAndPositionsSingleMode.txt

1	1	327.5
1	1	400
1	1	500
1	1	600
1	1	655
1	2	327.5
1	2	400
1	2	500
1	2	600
1	2	655

MatLabInPutFiles\TMDnumberAndPositionsMultiMode.txt

3	1,2,1	327.5,655,982.5
3	2,1,2	327.5,655,982.5
3	1,2,3	327.5,655,982.5
3	1,2,3	327.5,655,327.5
3	1,2,4	200,400,500

MatLabInPutFiles\TMDmassRatioRUN2.txt

0.002	0.002	0.002
0.005	0.005	0.005
0.01	0.01	0.01
0.025	0.025	0.025
0.05	0.05	0.05

XisRUN2.m

```
function [xis] = XisRUN2(my_d_vec)
my_d = my_d_vec(1,:);
xis = zeros(size(my_d_vec));
for n = 1:length(my_d_vec(1,:))
myd = my_d(n);

% R. Luft:
xis(2,n) = sqrt((myd/4)*(1-((3*myd)/4)));

% Den Hartog:
xis(4,n) = sqrt((3*myd)/(8*(1+myd)^3));

% Middle between R.Luft and Den Hartog:
xis(3,n) = xis(2,n)+(xis(4,n)-xis(2,n))/2;

% R. Luft - 10%
xis(1,n) = xis(2,n)-3*(xis(4,n)-xis(2,n))/2;

% Den Hartog + 10%
xis(5,n) = xis(4,n) + 3*(xis(4,n)-xis(2,n))/2;
end
end
```

OmegasRUN2.m

```
function [omegas] = OmegasRUN2(my_d_vec,TMD_mode_damp_vec,omega_z)
omegas = zeros(size(my_d_vec));
my_d = my_d_vec(1,:);
for n = 1:length(TMD_mode_damp_vec(1,:))
    omegaz = omega_z(TMD_mode_damp_vec(1,n));
    myd = my_d(n);

% Den Hartog:
omegas(3,n) =  omegaz/(1+myd);
omegasDH =  omegaz/(1+myd);
% R. Luft:
omegasL = omegaz/sqrt(1+(3/2)*myd);

deltaDH_L = omegasL - omegasDH;


% Den Hartog - 20*deltaDH_L
omegas(1,n) = omegasDH - 20*deltaDH_L;

% Den Hartog - 10*deltaDH_L
omegas(2,n) = omegasDH - 10*deltaDH_L;

% Den Hartog + 10*deltaDH_L
omegas(4,n) = omegasDH + 10*deltaDH_L;

% Den Hartog + 20*deltaDH_L
omegas(5,n) = omegasDH + 20*deltaDH_L;
end
end
```

PlotResponseSpectra.m

```
function fig =
PlotResponseSpectra(S,Omega,legend_wind_speed1,legend_wind_speed2,min_x,max_x,titletext
S,ylabeltextS,saveAs)

        max_y = 1.05*max(max(S));

        fig = figure;
        p = plot(Omega,real(S));
        str = strcat('Vcr',strtrim(cellstr(num2str(legend_wind_speed1.'))),
'=',strtrim(cellstr(num2str(legend_wind_speed2.'))), 'm/s');
        axis([min_x max_x 0 max_y])
        hTitle = title([titletextS]);
        hXLabel = xlabel('$$\omega$$ (rad/s)');
        hYLabel = ylabel(ylabeltextS);
        hLegend = legend(str{:});
        PlotProperties(fig, p, hTitle, hXLabel, hYLabel);
        screen2png(saveAs);

end
```

PlotStatistics.m

```
function fig = PlotStatistics(maxrz,stdrz,k_p,N_j,v,jn,t,rz,titletext,saveAsPs)

for xr = 1:N_j
% r_z Bar- Chart, making of vectors:
n_bars = 100;
r_z_bar = sort(maxrz(:,xr,v));
r_z_bar_min = r_z_bar(1); r_z_bar_max = r_z_bar(jn); delta_r_z_bar = (r_z_bar_max-
r_z_bar_min)/n_bars;
r_z_val = zeros(1,n_bars);
r_z_count = zeros(1,n_bars);
    count = 1;
    for k = 1:min(n_bars,jn);
for js = 1:length(r_z_bar)
    if r_z_bar(js) > r_z_bar_min + delta_r_z_bar*(k-1) && r_z_bar(js) <= r_z_bar_min +
delta_r_z_bar*(k)
        count = count + 1;
    end
end
        r_z_count(k) = count;
        r_z_val(k) = (r_z_bar_min+delta_r_z_bar*(k-0.5));
        count = 0;
    end

% \sigma Bar- Chart, making of vectors:
std_bar = sort(stdrz(:,xr,v));
std_bar_min = std_bar(1); std_bar_max = std_bar(jn); delta_std_bar = (std_bar_max-
std_bar_min)/n_bars;
std_val = zeros(1,n_bars);
std_count = zeros(1,n_bars);

    count = 1;
    for k = 1:min(n_bars,jn);
for js = 1:length(std_bar)
    if std_bar(js) > std_bar_min + delta_std_bar*(k-1) && std_bar(js) <= std_bar_min +
delta_std_bar*(k)
        count = count + 1;
    end
end
        std_count(k) = count;
        std_val(k) = (std_bar_min+delta_std_bar*(k-0.5));
        count = 0;
    end

% k_p Bar- Chart, making of vectors:
k_p_bar = sort(k_p(:,xr,v));
k_p_bar_min = k_p_bar(1); k_p_bar_max = k_p_bar(jn); delta_k_p_bar = (k_p_bar_max-
k_p_bar_min)/n_bars;
k_p_val = zeros(1,n_bars);
k_p_count = zeros(1,n_bars);

    count = 1;
    for k = 1:min(n_bars,jn);
for j = 1:length(k_p_bar)
    if k_p_bar(j) > k_p_bar_min + delta_k_p_bar*(k-1) && k_p_bar(j) <= k_p_bar_min +
delta_k_p_bar*(k)
        count = count + 1;
    end
end
        k_p_count(k) = count;
        k_p_val(k) = (k_p_bar_min+delta_k_p_bar*(k-0.5));
        count = 0;
```

```
    end


n_norm = length(r_z_bar);   norm_x_min = 0.75*r_z_bar(1);       norm_x_max =
1.25*r_z_bar(end);
d_norm_x = (norm_x_max-norm_x_min)/(n_norm-1);   norm_x =
norm_x_min:d_norm_x:norm_x_max;
norm_rz = normpdf(norm_x, mean(r_z_bar),std(r_z_bar));

n_norm_std = length(std_bar);   norm_x_min_std = 0.75*std_bar(1);   norm_x_max_std =
1.25*std_bar(end);
d_norm_x_std = (norm_x_max_std-norm_x_min_std)/(n_norm_std-1);      norm_x_std =
norm_x_min_std:d_norm_x_std:norm_x_max_std;
norm_std = normpdf(norm_x_std, mean(std_bar),std(std_bar));

n_norm_kp = length(k_p_bar);   norm_x_min_kp = 0.75*k_p_bar(1);   norm_x_max_kp =
1.25*k_p_bar(end);
d_norm_x_kp = (norm_x_max_kp-norm_x_min_kp)/(n_norm_kp-1);    norm_x_kp =
norm_x_min_kp:d_norm_x_kp:norm_x_max_kp;
norm_kp = normpdf(norm_x_kp, mean(k_p_bar),std(k_p_bar));
minvalrz = -1.15*max(abs(maxrz(:,xr,v))); maxvalrz =  1.2*max(abs(maxrz(:,xr,v)));
pm =    [0.07, 0.55, 0.4, 0.35;
         0.55, 0.55, 0.4, 0.35;
         0.07, 0.10, 0.4, 0.35;
         0.55, 0.10, 0.4, 0.35];


fig = figure;
   FigProperties(fig)
subfig = subplot(2,2,1);
set(subfig, 'position', pm(1,:))
hold on;
for js = 2:jn
p1 = plot(t,real(rz(js,:,xr,v)));
set(p1,'Color','red','LineWidth',0.2)
end
p = plot(t,real(rz(1,:,xr,v)));
set(p,'Color','blue','LineWidth',2)
axis([0 600 minvalrz maxvalrz]);
hTitle = title(titletext,'HorizontalAlignment','left');
hXLabel = xlabel('Time [s]');
hYLabel = ylabel('$$r_{z} $$    [m]');
SubPlotProperties(subfig,hXLabel, hYLabel);
SubPlotTitleProperties(subfig, hTitle);
hold off
str1=['' num2str(jn) '  time domain simulations of $r_{z}$ '];
text(50,0.88*maxvalrz,str1,'interpreter','latex','FontName','Times','FontSize',16);
hLegend = legend([p1,p],'999 simulations', '1 simulation','Location','SouthEast');

r_z_count_norm = r_z_count/(delta_r_z_bar*sum(r_z_count));
maxrzbar_x = 1.1*max(max(r_z_count_norm),max(norm_rz)); minrzbar_y =
0.92*min(min(r_z_val),0.91*min(r_z_bar)); maxrzbar_y =
1.01*max(max(r_z_val),1.001*max(r_z_bar));

subfig = subplot(2,2,2);
set(subfig, 'position', pm(2,:))
hold on
bar1 = barh(r_z_val,r_z_count_norm,1);
stp1 = plot(norm_rz,norm_x);
set(stp1,'Color','red','LineWidth',2)
%fit1 = histfit(norm_rz,100,'normal');
axis([0 maxrzbar_x minrzbar_y maxrzbar_y]);
hXLabel = xlabel('Normalized probability of occurence');
hYLabel = ylabel('$$|r_{z}|_{ max} $$    [m]');
SubPlotProperties(subfig,hXLabel, hYLabel);
```

```
SubPlotTitleProperties(subfig, hTitle);
hold off
str2=['Max value of $r_{z}$'];
text(0.35*maxrzbar_x,0.95*maxrzbar_y,str2,'interpreter','latex','FontName','Times','Fon
tSize',16);
hLegend = legend([bar1,stp1],'Data from simulations', 'Normal
distribution','Location','SouthEast');

std_count_norm = std_count/(delta_std_bar*sum(std_count));
maxstdbar_x = 1.05*max(max(std_count_norm),max(norm_std)); minstdbar_y =
0.99*min(min(std_val),0.99*min(std_bar)); maxstdbar_y =
1.02*max(max(std_val),1.02*max(std_bar));
subfig = subplot(2,2,3);
set(subfig, 'position', pm(3,:))
hold on
bar2 = barh(std_val,std_count_norm,1);
stp2 = plot(norm_std,norm_x_std);
set(stp2,'Color','red','LineWidth',2)
axis([0 maxstdbar_x minstdbar_y maxstdbar_y]);
hXLabel = xlabel('Normalized probability of occurence');
hYLabel = ylabel('$$\sigma_{r_{z}} $$     [m]');
SubPlotProperties(subfig,hXLabel, hYLabel);
SubPlotTitleProperties(subfig, hTitle);
hold off
str3=['Standard deviation of $r_{z}$'];
text(0.35*maxstdbar_x,0.97*maxstdbar_y,str3,'interpreter','latex','FontName','Times','F
ontSize',16);
hLegend = legend([bar2,stp2],'Data from simulations', 'Normal
distribution','Location','SouthEast');


k_p_count_norm = k_p_count/(delta_k_p_bar*sum(k_p_count));
maxkpbar_x = 1.05*max(max(k_p_count_norm),max(norm_kp)); minkpbar_y =
0.82*min(min(k_p_val),0.99*min(k_p_bar)); maxkpbar_y =
1.005*max(max(k_p_val),1.01*max(k_p_bar));
subfig = subplot(2,2,4);
set(subfig, 'position', pm(4,:))
hold on
bar3 = barh(k_p_val,k_p_count_norm,1);
stp3 = plot(norm_kp,norm_x_kp);
set(stp3,'Color','red','LineWidth',2)
axis([0 maxkpbar_x minkpbar_y maxkpbar_y]);
hXLabel = xlabel('Normalized probability of occurence');
hYLabel = ylabel('$$k_{p}$$');
SubPlotProperties(subfig,hXLabel, hYLabel);
SubPlotTitleProperties(subfig, hTitle);
hold off
str4=['Peak factor of $r_{z}$'];
text(0.35*maxkpbar_x,0.95*maxkpbar_y,str4,'interpreter','latex','FontName','Times','Fon
tSize',16);
hLegend = legend([bar3,stp3],'Data from simulations', 'Normal
distribution','Location','SouthEast');
screen2png(saveAsPs);
end
end
```

PlotTimeSeries.m

```
function fig = PlotTimeSeries(t,rz,maxrz,stdrz,k_p,TotCaseNr,jn,xr,x_d,v,titletextTS)

maxy = 1.25*max(max(maxrz(1:min(jn,12),xr,v)));
fig = figure;
for j=1:min(jn,12)
    h1 = maxrz(j,xr,v);
    h2 = stdrz(j,xr,v);
    h3 = k_p(j,xr,v);

subfig = subplot(4,3,j);
plot(t,real(rz(j,:,xr,v)))
SubfigProperties(subfig);
axis([0 600 -maxy maxy]);
str1=['r_{z, max} = {' num2str(round(1000*h1)/1000) '} m'];
str2=['\sigma_{rz} = {' num2str(round(1000*h2)/1000) '} m'];
str3=['k_{p} = {' num2str(round(1000*h3)/1000) '}'];

if j<10
text(320,0.77*maxy,str1);
text(20,-1.25*maxy,str2);
text(320,-1.25*maxy,str3);
else
text(320,0.77*maxy,str1);
text(20,-0.77*maxy,str2);
text(320,-0.77*maxy,str3);
end

if j==1 || j==4 || j==7 || j==10
hYLabel = ylabel('$$r_{z} (m/s)$$');
hYLabelProperties(hYLabel);
end
if j==2
    hTitle = title(titletextTS);
    SubPlotTitleProperties(subfig, hTitle);
end
if j==10 || j==11 || j==12
hXLabel = xlabel('Time (s)');
hXLabelProperties(hXLabel);
else
    set(gca,'XTick',[]);

end
end
FigProperties(fig)
screen2png(['MatLabOutPutFiles\TimeSerie_rzCase' num2str(TotCaseNr) ' atXpos'
num2str(floor(x_d(xr))) '']);
end
```

MakeMovie.m

```
function MakeMovie(max_y_orig,maxy_rz,
TotCaseNr,maxy_damper,r_z_sum_of_modes_original,r_z_sum_of_modes,r_d_sum_of_modes,v,xva
l,t_sim,time,j,damper_displacement_x,rz_orig_plot,rz_orig_for_multiple_xr,x_points,L,Vr
,N_j,x_d)

maxY = 1.05*max(max(max_y_orig,maxy_rz(TotCaseNr)+maxy_damper(TotCaseNr)));
fig = figure;
 writerObj = VideoWriter(['MatLabOutPutFiles\TMDmmResponceTimeSimulationMovieVR '
num2str(v) ' TotCaseNr ' num2str(TotCaseNr) '.avi']);
open(writerObj)
       for t = 1:length(t_sim)
           max_y_orig_current = max(abs(r_z_sum_of_modes_original(:,t,j)));
           max_y_rz_current = max(abs(r_z_sum_of_modes(:,t,j)));

   subplot(2,1,1)
   p = plot(xval,real(r_z_sum_of_modes_original(:,t,j)));
            axis([0 L -maxY maxY])
                str1=['Time:' num2str(round(10*t_sim(t))/10) 's'];
                text(L/2,0.7*maxY,str1,'Fontsize',14)
                    str2=['Max deflection in ' num2str(round(10*time(end))/10) ' s time
window = ' num2str(round(1000*max_y_orig(TotCaseNr))/1000) ' m '];
                    text(L/20,-0.83*maxY,str2,'Fontsize',14)
                    str3=['Max current deflection = '
num2str(round(1000*max_y_orig_current)/1000) ' m '];
                    text(6*L/10,-0.83*maxY,str3,'Fontsize',14)
                    set(gca,'Fontsize',14,'LineWidth',1)
                    hTitle = title(['Simulated main system response without TMD, Case '
num2str(TotCaseNr) ' , $$V_{cr}$$ = ' num2str(round(1000*Vr(v))/1000) ' m/s ']);
                    hYLabel = ylabel('$$r_{z}$$  [m]');
                    hYLabelProperties(hYLabel);
                    FigAndTitleProperties(fig, hTitle)
                    pPlotProperties(p);
                    hold on
    for xr = 1:N_j
    scatter(damper_displacement_x(xr),rz_orig_plot(j,t,xr,v));
    end
    for xr = 1:length(x_points)
    scatter(x_points(xr),rz_orig_for_multiple_xr(xr,t),'r');
    end
    hold off
   subplot(2,1,2)
   p = plot(xval,real(r_z_sum_of_modes(:,t,j)));
        hold on
            axis([0 L -maxY maxY])
                    str4=['Bridge girder = '
num2str(round(1000*maxy_rz(TotCaseNr))/1000) ' m '];
                    text(L/30,-0.67*maxY,str4,'Fontsize',14)
                    str5=['Bridge girder = ' num2str(round(1000*max_y_rz_current)/1000)
' m '];
                    text(6*L/10,-0.67*maxY,str5,'Fontsize',14)

                set(gca,'Fontsize',14,'LineWidth',1)
                hTitle = title('Simulated main system and TMD response');
                hYLabel = ylabel('$$r_{z}$$ and $$r_{d}$$  [m]');
                hXLabel = xlabel('Spanwidth   [m]');
                str1=['Time:' num2str(round(10*t_sim(t))/10) 's'];
                text(L/2,0.7*maxY,str1,'Fontsize',14)
                PlotProperties(fig, p, hTitle, hXLabel, hYLabel);

            max_y_damper_current = zeros(1,N_j);
```

```
    for xr = 1:N_j
        xpos = floor(x_d(xr)/(xval(end)/length(xval)));
       max_y_damper_current(xr) =
max(abs(real(r_d_sum_of_modes(t,xr)+r_z_sum_of_modes(xpos,t,j))));

scatter(damper_displacement_x(xr),real(r_d_sum_of_modes(t,xr)+r_z_sum_of_modes(xpos,t,j
)),'filled');
    axis([0 L -maxY maxY])
    end
                    current_max_y_damp = max(max_y_damper_current);
                    str6=['Max deflection in ' num2str(round(10*time(end))/10) ' s time
window of: '];
                    text(L/30,-0.5*maxY,str6,'Fontsize',14)
                    str7=['TMD(s) (relative defl.) = '
num2str(round(1000*maxy_damper(TotCaseNr))/1000) ' m '];
                    text(L/30,-0.85*maxY,str7,'Fontsize',14)
                    str8=['Max current deflection of: '];
                    text(6*L/10,-0.5*maxY,str8,'Fontsize',14)
                    str9=['TMD(s) (relative defl.) = '
num2str(round(1000*current_max_y_damp)/1000) ' m '];
                    text(6*L/10,-0.85*maxY,str9,'Fontsize',14)
          frame = getframe(fig);
      hold off
    writeVideo(writerObj,frame);
       end
  close(writerObj);
 %movie(frame,1,1)
end
```

SigmaPlot.m

```
function [fig1,fig2,fig3,fig4,fig5,fig6] = SigmaPlot(SIGMAhalf_rz, SIGMAhalf_rd,
SIGMAquart_rz,SIGMAquart_rd,SIGMA_ORIGhalf, SIGMA_ORIGquart,Information_matrix)
% Different my_d (Case 16-20)
fig1 = figure;
scatter(Information_matrix(16:20,10),SIGMAhalf_rz(16:20),'r')
hold on
scatter(Information_matrix(16:20,10),SIGMAhalf_rd(16:20),'g')
scatter(Information_matrix(16:20,10),SIGMA_ORIGhalf(16:20),'b')
hold off
x1min = min(Information_matrix(16:20,10));
x1max = max(Information_matrix(16:20,10));
y1min =
min(min(SIGMAhalf_rz(16:20),min(min(SIGMAhalf_rd(16:20)),min(SIGMA_ORIGhalf(16:20)))));
y1max =
max(max(SIGMAhalf_rz(16:20),max(max(SIGMAhalf_rd(16:20)),max(SIGMA_ORIGhalf(16:20)))));
axis([x1min x1max y1min y1max]);
set(fig1,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaHalfMy']);

fig2 = figure;
scatter(Information_matrix(21:25,13),SIGMAhalf_rz(21:25),'r')
hold on
scatter(Information_matrix(21:25,13),SIGMAhalf_rd(21:25),'g')
scatter(Information_matrix(21:25,13),SIGMA_ORIGhalf(21:25),'b')
hold off
x2min = min(Information_matrix(21:25,13));
x2max = max(Information_matrix(21:25,13));
y2min =
min(min(SIGMAhalf_rz(21:25),min(min(SIGMAhalf_rd(21:25)),min(SIGMA_ORIGhalf(21:25)))));
y2max =
max(max(SIGMAhalf_rz(21:25),max(max(SIGMAhalf_rd(21:25)),max(SIGMA_ORIGhalf(21:25)))));
axis([x2min x2max y2min y2max]);
set(fig2,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaHalfXi']);

fig3 = figure;
scatter(Information_matrix(26:30,16)/0.9,SIGMAhalf_rz(26:30),'r')
hold on
scatter(Information_matrix(26:30,16)/0.9,SIGMAhalf_rd(26:30),'g')
scatter(Information_matrix(26:30,16)/0.9,SIGMA_ORIGhalf(26:30),'b')
hold off
x3min = min(Information_matrix(26:30,16)/0.9);
x3max = max(Information_matrix(26:30,16)/0.9);
y3min =
min(min(SIGMAhalf_rz(26:30),min(min(SIGMAhalf_rd(26:30)),min(SIGMA_ORIGhalf(26:30)))));
y3max =
max(max(SIGMAhalf_rz(26:30),max(max(SIGMAhalf_rd(26:30)),max(SIGMA_ORIGhalf(26:30)))));
axis([x3min x3max y3min y3max]);
set(fig3,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaHalfOmega']);


fig4 = figure;
scatter(Information_matrix(16:20,9),SIGMAquart_rz(16:20),'r')
hold on
scatter(Information_matrix(16:20,9),SIGMAquart_rd(16:20),'g')
scatter(Information_matrix(16:20,9),SIGMA_ORIGquart(16:20),'b')
hold off
x4min = min(Information_matrix(16:20,9));
x4max = max(Information_matrix(16:20,9));
```

```
y4min =
min(min(SIGMAquart_rz(16:20),min(min(SIGMAquart_rd(16:20)),min(SIGMA_ORIGquart(16:20)))
));
y4max =
max(max(SIGMAquart_rz(16:20),max(max(SIGMAquart_rd(16:20)),max(SIGMA_ORIGquart(16:20)))
));
axis([x4min x4max y4min y4max]);
set(fig4,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaQuartMy']);

fig5 = figure;
scatter(Information_matrix(21:25,12),SIGMAquart_rz(21:25),'r')
hold on
scatter(Information_matrix(21:25,12),SIGMAquart_rd(21:25),'g')
scatter(Information_matrix(21:25,12),SIGMA_ORIGquart(21:25),'b')
hold off
x5min = min(Information_matrix(21:25,12));
x5max = max(Information_matrix(21:25,12));
y5min =
min(min(SIGMAquart_rz(21:25),min(min(SIGMAquart_rd(21:25)),min(SIGMA_ORIGquart(21:25)))
));
y5max =
max(max(SIGMAquart_rz(21:25),max(max(SIGMAquart_rd(21:25)),max(SIGMA_ORIGquart(21:25)))
));
axis([x5min x5max y5min y5max]);
set(fig5,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaQuartXi']);

fig6 = figure;
scatter(Information_matrix(26:30,15)/0.71,SIGMAquart_rz(26:30),'r')
hold on
scatter(Information_matrix(26:30,15)/0.71,SIGMAquart_rd(26:30),'g')
scatter(Information_matrix(26:30,15)/0.71,SIGMA_ORIGquart(26:30),'b')
hold off
x6min = min(Information_matrix(26:30,15)/0.71);
x6max = max(Information_matrix(26:30,15)/0.71);
y6min =
min(min(SIGMAquart_rz(26:30),min(min(SIGMAquart_rd(26:30)),min(SIGMA_ORIGquart(26:30)))
));
y6max =
max(max(SIGMAquart_rz(26:30),max(max(SIGMAquart_rd(26:30)),max(SIGMA_ORIGquart(26:30)))
));
axis([x6min x6max y6min y6max]);
set(fig6,'units','normalized','outerposition',[0 0 1 1]);
screen2png(['MatLabOutPutFiles\SigmaQuartOmega']);
end
```

Hplot.m

```
function [fig1, fig2,fig3,fig4,fig5,fig6] = Hplot(H_rz,TotCaseNr,H_orig,Omega,omega_z)

H_plot_mode1 = [];       H_orig_plot_mode1 = [];
H_plot_mode2 = [];       H_orig_plot_mode2 = [];
H_plot_mode3 = [];       H_orig_plot_mode3 = [];
H_plot_mode4 = [];       H_orig_plot_mode4 = [];
H_plot_mode5 = [];       H_orig_plot_mode5 = [];
H_plot_mode6 = [];       H_orig_plot_mode6 = [];
Legend_H_plot = [];      Legend_H_orig_plot =  [];

for m = 1:TotCaseNr
    H_plot_mode1 = [H_plot_mode1, H_rz(:,1,1,m)];
    H_plot_mode2 = [H_plot_mode2, H_rz(:,2,2,m)];
    H_plot_mode3 = [H_plot_mode3, H_rz(:,3,3,m)];
    H_plot_mode4 = [H_plot_mode4, H_rz(:,4,4,m)];
    H_plot_mode5 = [H_plot_mode5, H_rz(:,5,5,m)];
    H_plot_mode6 = [H_plot_mode6, H_rz(:,6,6,m)];
    Legend_H_plot = [Legend_H_plot, m];
end
H_orig_plot_mode1 = H_orig(:,1,1,1);
H_orig_plot_mode2 = H_orig(:,2,2,1);
H_orig_plot_mode3 = H_orig(:,3,3,1);
H_orig_plot_mode4 = H_orig(:,4,4,1);
H_orig_plot_mode5 = H_orig(:,5,5,1);
H_orig_plot_mode6 = H_orig(:,6,6,1);


       max_y_H = real(1.2*max(max(abs(H_orig_plot_mode1))));
       min_y_H = 1;
       min_x_H = 1.2*omega_z(1)/2;
       max_x_H = 3*omega_z(1)/2;
       fig1 = figure;
        grid
        for j = 1:6
        subfig = subplot(2,3,j);
        set(subfig,'units','normalized');
        %p(j,:) =  get(subfig, 'position');
        p = [0.06,0.53,0.24, 0.425;
            0.37,0.53, 0.24, 0.425;
            0.68,0.53,0.24,0.425;
            0.06,0.09 ,0.24,0.425;
            0.37,0.09,0.24,0.425;
            0.68,0.09,0.24,0.425];
        set(subfig, 'position', p(j,:))
        p_or = semilogy(Omega, abs(H_orig_plot_mode1),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        hold on
        semilogy(Omega, abs(H_plot_mode1(:,j*5-4:j*5)),'LineWidth',1.5);
        p_or = semilogy(Omega, abs(H_orig_plot_mode1),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
        str = [str2];
        axis([min_x_H max_x_H min_y_H max_y_H]);
        if j ==2
        hTitle = title(['Frequency response function for mode 1']);
        FigAndTitleProperties(fig1,hTitle);
        end
        if j == 1 || j == 4
        hYLabel = ylabel('$$ |H_{r_{z}}|$$');
        set([hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
        end
```

```
    if j == 4 || j == 5 || j == 6
        hXLabel = xlabel('$$\omega$$ (rad/s)');
        set([hXLabel], ...
'interpreter','latex','FontName','Times','FontSize',16)
    end
    hLegend = legend('No TMD', str{:});
    set(hLegend,'FontSize',12)
     %SubPlotProperties(subfig,hXLabel, hYLabel);
    hold off
    end
    screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode1'])
     close(fig1)

   max_y_H2 = 1.2*max(max(abs(H_orig_plot_mode2)));
   min_y_H2 = 1;
   min_x_H2 = 1.2*omega_z(2)/2;
   max_x_H2 = 3*omega_z(2)/2;
   fig2 = figure;
    grid
    for j = 1:6
    subfig = subplot(2,3,j);
    set(subfig,'units','normalized');
    %p(j,:) =  get(subfig, 'position');
    p = [0.06,0.53,0.24, 0.425;
        0.37,0.53, 0.24, 0.425;
        0.68,0.53,0.24,0.425;
        0.06,0.09 ,0.24,0.425;
        0.37,0.09,0.24,0.425;
        0.68,0.09,0.24,0.425];
    set(subfig, 'position', p(j,:))
    p_or = semilogy(Omega, abs(H_orig_plot_mode2),'--');
    set(p_or,'LineWidth',1.5,'Color','black')
    hold on
    semilogy(Omega, abs(H_plot_mode2(:,j*5-4:j*5)),'LineWidth',1.5);
    p_or = semilogy(Omega, abs(H_orig_plot_mode2),'--');
    set(p_or,'LineWidth',1.5,'Color','black')
    %str1 = strcat('Without TMD ');
    str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
    str = [str2];
    axis([min_x_H2 max_x_H2 min_y_H2 max_y_H2]);
    if j ==2
    hTitle = title(['Frequency response function for mode 2']);
    FigAndTitleProperties(fig2,hTitle);
    end
    if j == 1 || j == 4
    hYLabel = ylabel('$$ |H_{r_{z}}|$$');
    set([hYLabel], ...
'interpreter','latex','FontName','Times','FontSize',16)
    end
    if j == 4 || j == 5 || j == 6
        hXLabel = xlabel('$$\omega$$ (rad/s)');
        set([hXLabel], ...
'interpreter','latex','FontName','Times','FontSize',16)
    end
    hLegend = legend('No TMD', str{:});
    set(hLegend,'FontSize',12)
     %SubPlotProperties(subfig,hXLabel, hYLabel);
    hold off
    end
    screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode2'])
     close(fig2)

   max_y_H3 = 1.2*max(max(abs(H_orig_plot_mode3)));
   min_y_H3 = 1;
   min_x_H3 = 1.2*omega_z(3)/2;
```

```
  max_x_H3 = 3*omega_z(3)/2;
 % p = zeros(6,4);
  fig3 = figure;
   grid
   for j = 1:6
   subfig = subplot(2,3,j);
   set(subfig,'units','normalized');
   %p(j,:) =  get(subfig, 'position');
   p = [0.06,0.53,0.24, 0.425;
       0.37,0.53, 0.24, 0.425;
       0.68,0.53,0.24,0.425;
       0.06,0.09 ,0.24,0.425;
       0.37,0.09,0.24,0.425;
       0.68,0.09,0.24,0.425];
   set(subfig, 'position', p(j,:))
   p_or = semilogy(Omega, abs(H_orig_plot_mode3),'--');
   set(p_or,'LineWidth',1.5,'Color','black')
   hold on
   semilogy(Omega, abs(H_plot_mode3(:,j*5-4:j*5)),'LineWidth',1.5);
   p_or = semilogy(Omega, abs(H_orig_plot_mode3),'--');
   set(p_or,'LineWidth',1.5,'Color','black')
   %str1 = strcat('Without TMD ');
   str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
   str = [str2];
   axis([min_x_H3 max_x_H3 min_y_H3 max_y_H3]);
   if j ==2
   hTitle = title(['Frequency response function for mode 3']);
   FigAndTitleProperties(fig3,hTitle);
   end
   if j == 1 || j == 4
   hYLabel = ylabel('$$ |H_{r_{z}}|$$');
   set([hYLabel], ...
'interpreter','latex','FontName','Times','FontSize',16)
   end
   if j == 4 || j == 5 || j == 6
       hXLabel = xlabel('$$\omega$$ (rad/s)');
       set([hXLabel], ...
'interpreter','latex','FontName','Times','FontSize',16)
   end
   hLegend = legend('No TMD', str{:});
   set(hLegend,'FontSize',12)
    %SubPlotProperties(subfig,hXLabel, hYLabel);
   hold off
   end
   screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode3'])
    close(fig3)

  max_y_H4 = 1.2*max(max(abs(H_orig_plot_mode4)));
  min_y_H4 = 1;
  min_x_H4 = 1.2*omega_z(4)/2;
  max_x_H4 = 3*omega_z(4)/2;
 % p = zeros(6,4);
  fig4 = figure;
   grid
   for j = 1:6
   subfig = subplot(2,3,j);
   set(subfig,'units','normalized');
   %p(j,:) =  get(subfig, 'position');
   p = [0.06,0.53,0.24, 0.425;
       0.37,0.53, 0.24, 0.425;
       0.68,0.53,0.24,0.425;
       0.06,0.09 ,0.24,0.425;
       0.37,0.09,0.24,0.425;
       0.68,0.09,0.24,0.425];
   set(subfig, 'position', p(j,:))
```

```
        p_or = semilogy(Omega, abs(H_orig_plot_mode4),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        hold on
        semilogy(Omega, abs(H_plot_mode4(:,j*5-4:j*5)),'LineWidth',1.5);
        p_or = semilogy(Omega, abs(H_orig_plot_mode4),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        %str1 = strcat('Without TMD ');
        str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
        str = [str2];
        axis([min_x_H4 max_x_H4 min_y_H4 max_y_H4]);
        if j ==2
        hTitle = title(['Frequency response function for mode 4']);
        FigAndTitleProperties(fig4,hTitle);
        end
        if j == 1 || j == 4
        hYLabel = ylabel('$$ |H_{r_{z}}|$$');
        set( [hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
        end
        if j == 4 || j == 5 || j == 6
            hXLabel = xlabel('$$\omega$$ (rad/s)');
            set([hXLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
        end
        hLegend = legend('No TMD', str{:});
        set(hLegend,'FontSize',12)
         %SubPlotProperties(subfig,hXLabel, hYLabel);
        hold off
        end
        screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode4'])
close(fig4)

          max_y_H5 = 1.2*max(max(abs(H_orig_plot_mode5)));
       min_y_H5 = 1;
       min_x_H5 = 1.2*omega_z(5)/2;
       max_x_H5 = 3*omega_z(5)/2;
      % p = zeros(6,4);
       fig5 = figure;
        grid
        for j = 1:6
        subfig = subplot(2,3,j);
        set(subfig,'units','normalized');
        %p(j,:) =  get(subfig, 'position');
        p = [0.06,0.53,0.24, 0.425;
            0.37,0.53, 0.24, 0.425;
            0.68,0.53,0.24,0.425;
            0.06,0.09 ,0.24,0.425;
            0.37,0.09,0.24,0.425;
            0.68,0.09,0.24,0.425];
        set(subfig, 'position', p(j,:))
        p_or = semilogy(Omega, abs(H_orig_plot_mode5),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        hold on
        semilogy(Omega, abs(H_plot_mode5(:,j*5-4:j*5)),'LineWidth',1.5);
        p_or = semilogy(Omega, abs(H_orig_plot_mode5),'--');
        set(p_or,'LineWidth',1.5,'Color','black')
        %str1 = strcat('Without TMD ');
        str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
        str = [str2];
        axis([min_x_H5 max_x_H5 min_y_H5 max_y_H5]);
        if j ==2
        hTitle = title(['Frequency response function for mode 5']);
        FigAndTitleProperties(fig5,hTitle);
        end
        if j == 1 || j == 4
```

```
        hYLabel = ylabel('$$ |H_{r_{z}}|$$');
        set([hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
        end
        if j == 4 || j == 5 || j == 6
            hXLabel = xlabel('$$\omega$$ (rad/s)');
            set([hXLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
        end
        hLegend = legend('No TMD', str{:});
        set(hLegend,'FontSize',12)
         %SubPlotProperties(subfig,hXLabel, hYLabel);
        hold off
        end
        screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode5'])
         close(fig5)

        max_y_H6 = 1.2*max(max(abs(H_orig_plot_mode6)));
       min_y_H6 = 1;
       min_x_H6 = 1.2*omega_z(6)/2;
       max_x_H6 = 3*omega_z(6)/2;
      % p = zeros(6,4);
       fig6 = figure;
        grid
       for j = 1:6
       subfig = subplot(2,3,j);
       set(subfig,'units','normalized');
       %p(j,:) =  get(subfig, 'position');
       p = [0.06,0.53,0.24, 0.425;
           0.37,0.53, 0.24, 0.425;
           0.68,0.53,0.24,0.425;
           0.06,0.09 ,0.24,0.425;
           0.37,0.09,0.24,0.425;
           0.68,0.09,0.24,0.425];
       set(subfig, 'position', p(j,:))
       p_or = semilogy(Omega, abs(H_orig_plot_mode6),'--');
       set(p_or,'LineWidth',1.5,'Color','black')
       hold on
       semilogy(Omega, abs(H_plot_mode6(:,j*5-4:j*5)),'LineWidth',1.5);
       p_or = semilogy(Omega, abs(H_orig_plot_mode6),'--');
       set(p_or,'LineWidth',1.5,'Color','black')
       %str1 = strcat('Without TMD ');
       str2 = strcat('Case ',strtrim(cellstr(num2str(Legend_H_plot(j*5-4:j*5).'))));
       str = [str2];
       axis([min_x_H6 max_x_H6 min_y_H6 max_y_H6]);
       if j ==2
       hTitle = title(['Frequency response function for mode 6']);
       FigAndTitleProperties(fig6,hTitle);
       end
       if j == 1 || j == 4
       hYLabel = ylabel('$$ |H_{r_{z}}|$$');
       set([hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
       end
       if j == 4 || j == 5 || j == 6
           hXLabel = xlabel('$$\omega$$ (rad/s)');
           set([hXLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16)
       end
       hLegend = legend('No TMD', str{:});
       set(hLegend,'FontSize',12)
        %SubPlotProperties(subfig,hXLabel, hYLabel);
       hold off
       end
       screen2png(['MatLabOutPutFiles\TMDmmFRFWithAndWithoutTMDMode6'])
       close(fig6)
end
```

PlotProperties.m

```
function PlotProperties(fig, p, hTitle, hXLabel, hYLabel)
set( gca                        , ...
    'FontName'   , 'Helvetica' );
set([hTitle, hXLabel, hYLabel], ...
    'interpreter','latex','FontName','Times','FontSize',16);
    %'FontName'   , 'Helvetica') ;
   % set(hLegend              , ...
    %'FontSize'   , 12           );
    set(gca               , ...
    'FontSize'   , 18           );
set( hYLabel                     , ...
    'FontSize'   , 23  );
%set([hXLabel, hYLabel]  , ...
%    'FontSize'   , 13           );
set( hTitle                     , ...
    'FontSize'   , 22           , ...
    'FontWeight' , 'bold'       );
set(p,'LineWidth',1.8)
set(fig,'units','normalized','outerposition',[0 0 1 1]);
```