POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

HOVENRING

Assessment of the tension in the stay cables
DMS - Mechanical Engineering, Politecnico di Milano

Authors:
Paolo Trenta, Student ID: 10718668
Matteo Trotta, Student ID: 10337755
Lorenzo Vignoli, Student ID: 10712405

Professor: Alberto Zasso
Co-professor: Giulia Pomaranzi
Academic Year: 2023-24

Contents

1. Introduction

1.1. Hovenring bridge: system description

This assignment assesses the tension of the stay cables of the Hovenring bridge through an experimental campaign. The main parts of this structure are:

- 70 m high central tower
- Cable connectors at two levels (different attachment points result in different nominal lengths)
- Dampers (low frequencies and high frequencies)
- 24 cables (Ø 50 mm)
- Deck-cable connectors
- Circular deck (Ø 72 m)

It's important to note that the cables nearer to the land abutments are the ones provided with the biggest nominal tension.

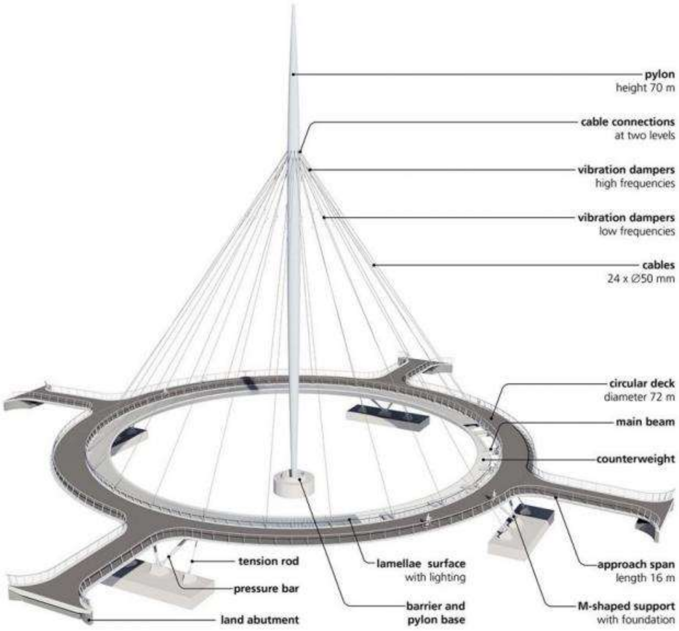
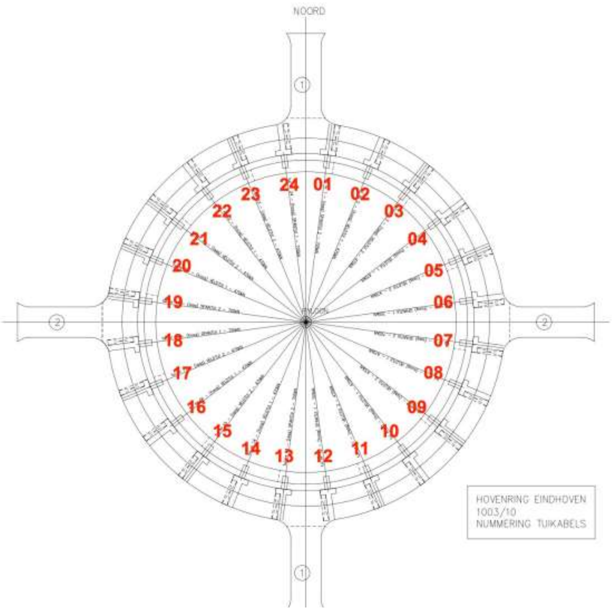


Figure 1: system description



Number	Type	Nominal Tension [kN]	Nominal total length [m]
Cable 01	Spantui 2	705	52.647
Cable 02	Veldtui 1	470	53.423
Cable 03	Veldtui 2	470	52.647
Cable 04	Veldtui 1	470	53.423
Cable 05	Veldtui 2	470	52.647
Cable 06	Spantui 1	705	53.423
Cable 07	Spantui 2	705	52.647
Cable 08	Veldtui 1	470	53.423
Cable 09	Veldtui 2	470	52.647
Cable 10	Veldtui 1	470	53.423
Cable 11	Veldtui 2	470	52.647
Cable 12	Spantui 1	705	53.423
Cable 13	Spantui 2	705	52.647
Cable 14	Veldtui 1	470	53.423
Cable 15	Veldtui 2	470	52.647
Cable 16	Veldtui 1	470	53.423
Cable 17	Veldtui 2	470	52.647
Cable 18	Spantui 1	705	53.423
Cable 19	Spantui 2	705	52.647
Cable 20	Veldtui 1	470	53.423
Cable 21	Veldtui 2	470	52.647
Cable 22	Veldtui 1	470	53.423
Cable 23	Veldtui 2	470	52.647
Cable 24	Spantui 1	705	53.423

Figure 2: cables' properties and nominal tension

A FE model for the cables is created, including the effects due to the TMDs of the structure together with a suitable representation of the constraints. For each cable, the first eigenfrequency was measured by mean of free motion tests (in the neighborhood of that value). No measurements of the tension of all the cables have been performed because it would have been too expensive and time-consuming; the values of the tension were directly measured with a jacking system for only four cables to validate the FE model, which is very accurate.

1.2. Finite Element Model

The stay cables are modelled using an equivalent tensioned beam finite-element scheme (including the bending stiffness too, since they're stiffer than cable elements). For each cable, the following elements are modelled (from deck to pylon):

- Threaded bar M100
- Socket
- Cable
- Socket
- Threaded bar M72

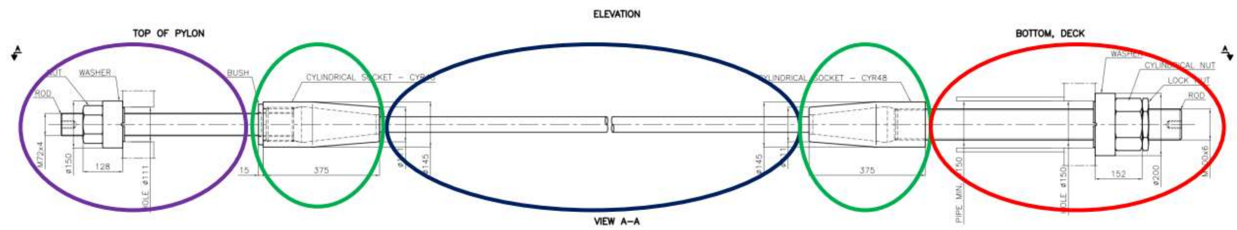
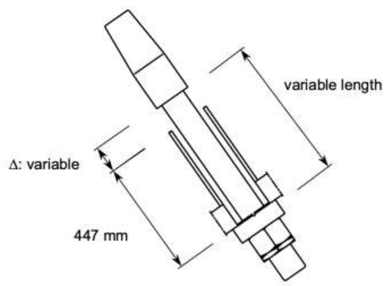


Figure 3: cable elements



Moreover, the threaded bar is inserted within the coil in the deck, so the actual length of each cable can be different from the nominal one. The threaded bar is 447 mm plus a variable length (Δ), which is measured for each cable.

1.3. FEM: constraints and dampers

Boundary conditions are basically defined by the constraints (washer-nut interface). Because of friction and high-tension force, the spherical nuts and washers act more as clamps than as hinges. Therefore, a hinge and a lumped torsional stiffness have been considered for a realistic model of the constraints.

To compute K_T, a best fitting procedure between the experimental and numerical modal shapes of the system is used.

Damping action is provided by TMDs: one-d.o.f.-systems which, coupled with the main one, can dampen a particular frequency's vibration (in this case, wind's one).

In *Figure 4* dampers for different frequencies are shown, together with the modelling that can be used (parameters defined through best fitting, always in the first frequency range).

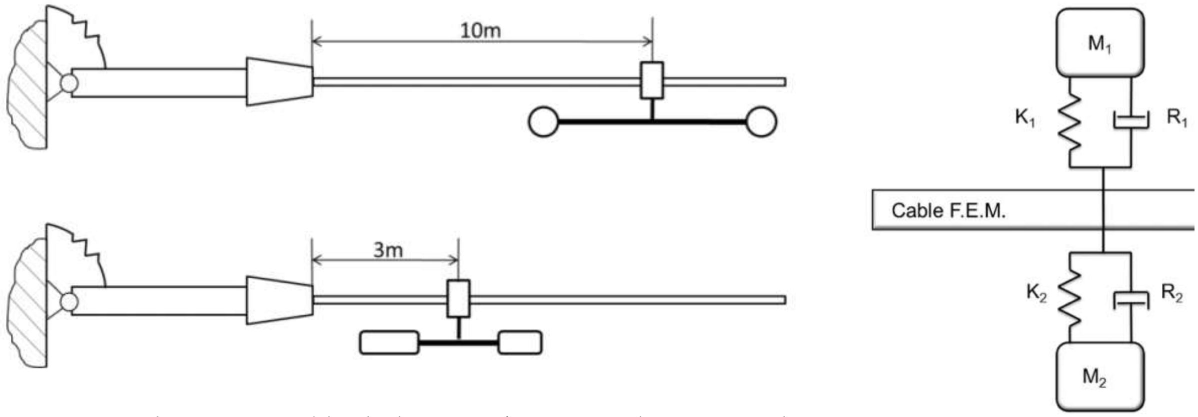


Figure 4: low (top) and high (bottom) frequency dampers and modelling

1.4. Experimental setup: system excitation

Each cable is instrumented with a triaxial accelerometer placed at 10 m from the cable socket. The sampling frequency is 128 Hz and the acquisition time is 100 s. The cable is excited by hand according to the frequency of first vibration, then it is let to move freely.

The output's the acceleration, as a function of time. Three different regions are identified:

1. Large amplitudes region (non-linear effects)
2. Smaller amplitudes and so slightly smaller damping ratio
3. Lowest damping ratio (TMD is not efficient)

In the large amplitudes region, the small displacements hypothesis disappears. In fact, in this case:

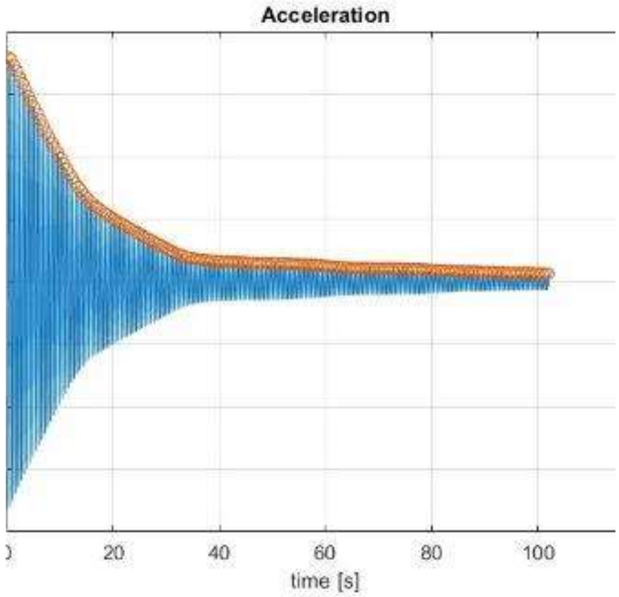
$dL = dx\sqrt{1 + (\frac{\partial w}{\partial x})^2}$. Then: $L_{final} = \int_0^L \sqrt{1 + (\frac{\partial w}{\partial x})^2}\, dx$.

But: $\varepsilon = \frac{L_{final} - L}{L}$, and $\Delta T = EA \cdot \varepsilon$, obtaining a variation of the tension of the cable.

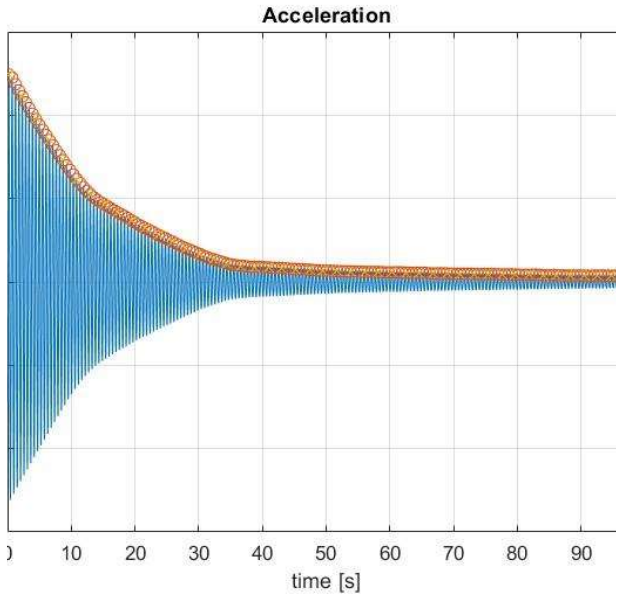
In each case (damping and frequency computations) the FEM model can reproduce the system behavior assuming a constant tension force and dampers' behavior in their operative condition (in our case, in the interval 15-35 s, as we can guess by observing the acceleration slope).

Through the MatLab function *findpeaks* the acceleration peaks are found; then only one peak every four is considered, providing better accuracy. They can be used then to compute the nondimensional damping (through the logarithmic decrement) and the oscillation period (thus the first eigenfrequency).

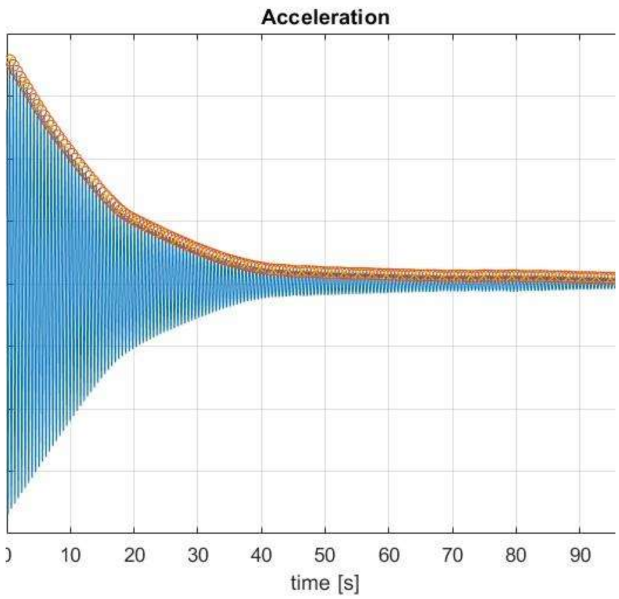
Cable 05



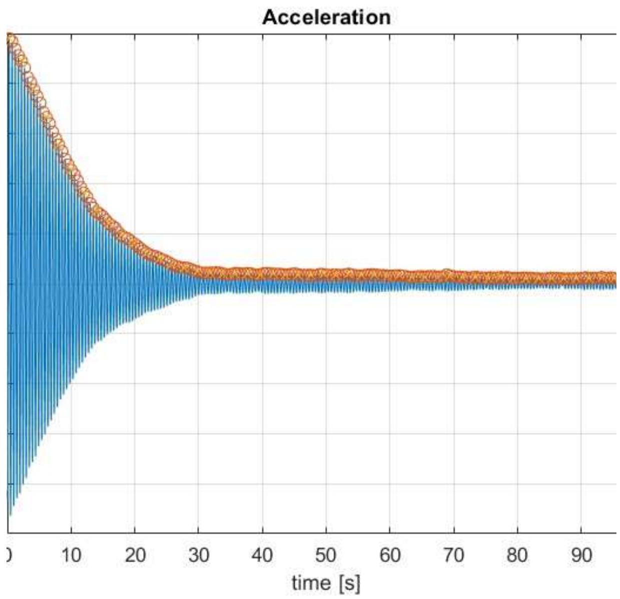
Cable 06



Cable 08



Cable 13



2.Identification of the damping ratio

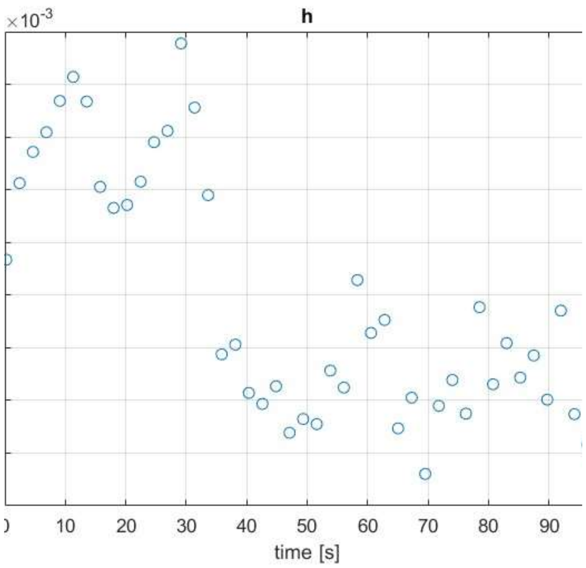
The non-dimensional damping ratio h_i can be computed as:

$$h_i = \frac{\delta_i}{2\pi}$$

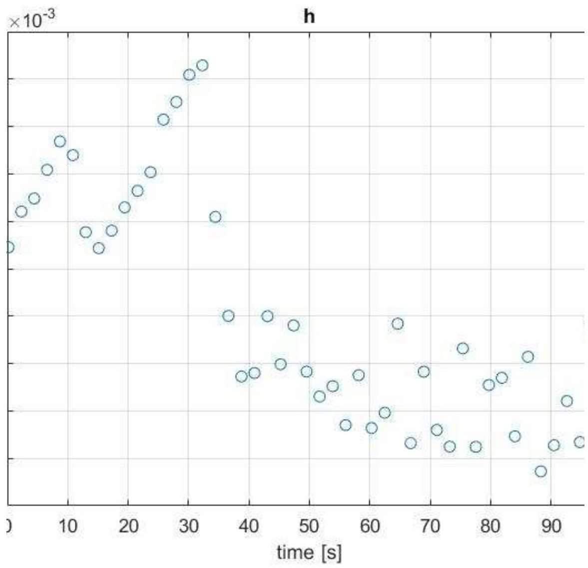
Being δ_i logarithmic decrement of the acceleration:

$$\delta_i = \frac{1}{4}\log\left(\frac{\ddot{y}(t)}{\ddot{y}(t(i+4))}\right)$$

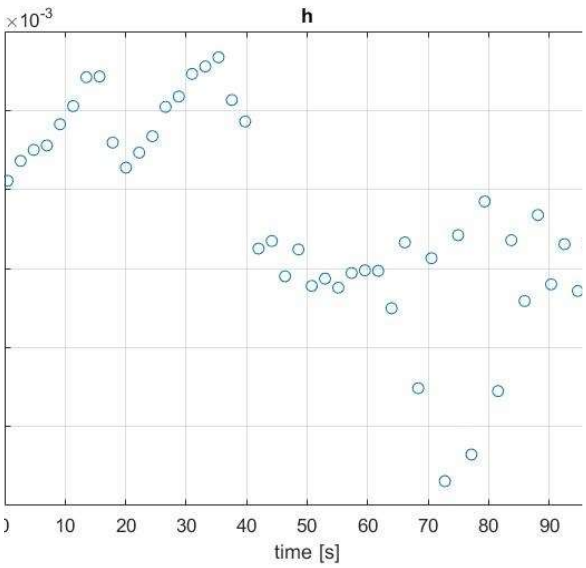
Cable 05



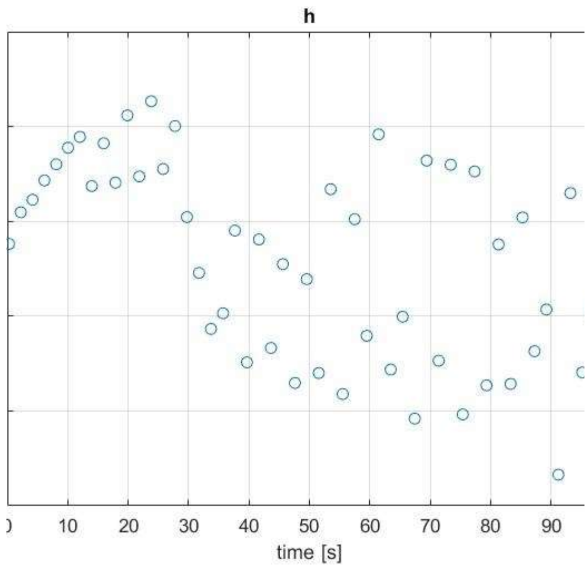
Cable 06



Cable 08



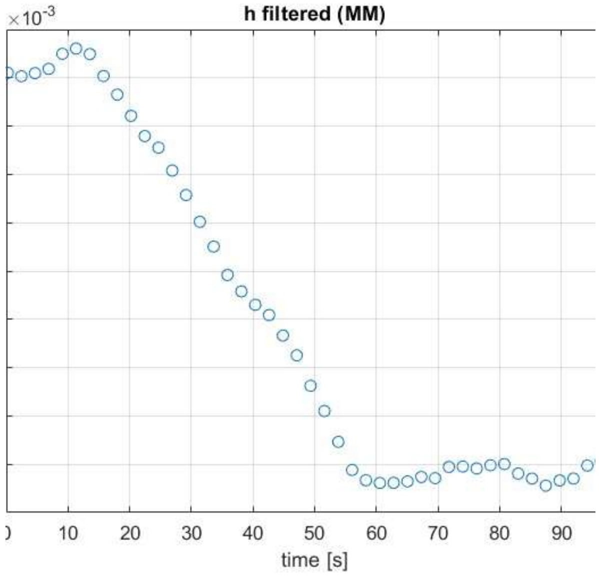
Cable 13



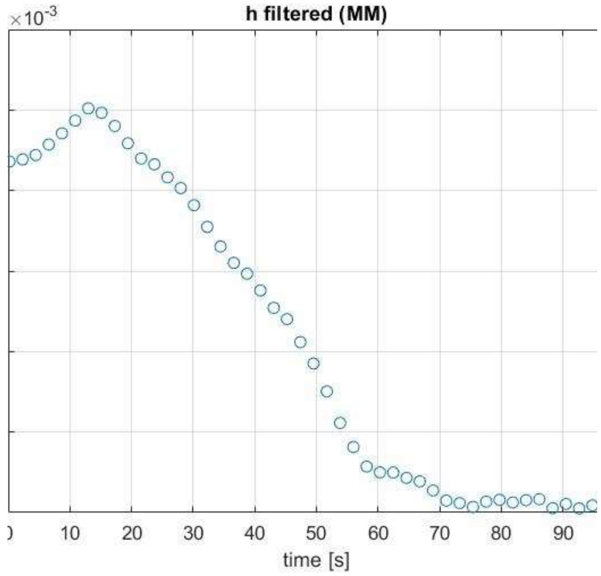
Performing the calculations, it's important to remember that only one peak every four is used to gain better accuracy.

To get a smoother shape, a moving average (MM) filter is applied to the results, with a window size equal to 20 (using the MatLab function *movmean*).

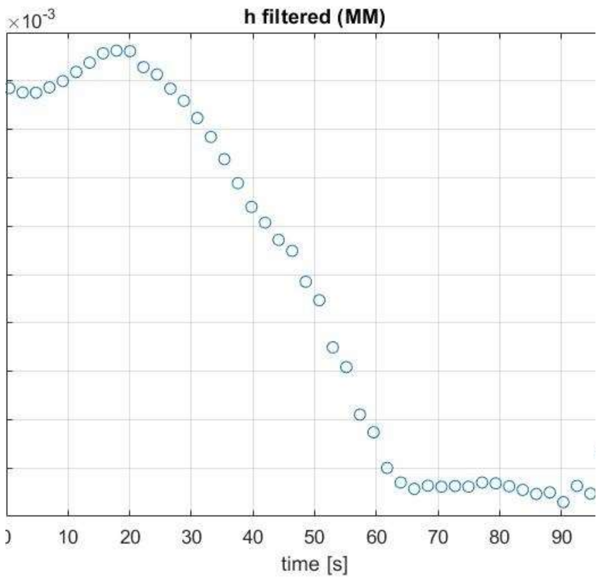
Cable 05



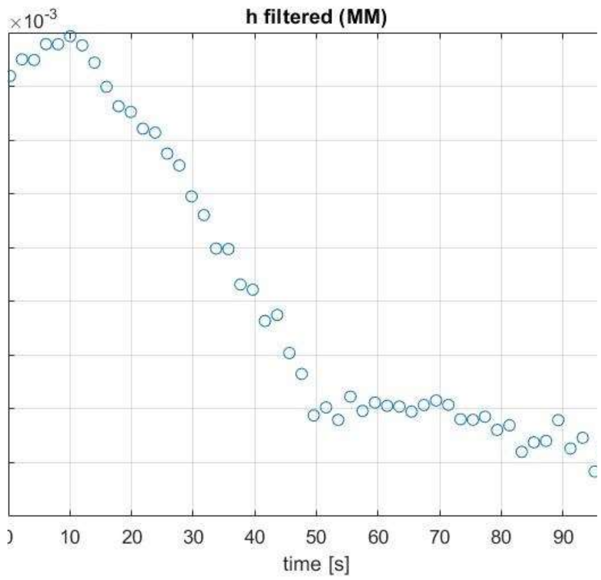
Cable 06



Cable 08



Cable 13



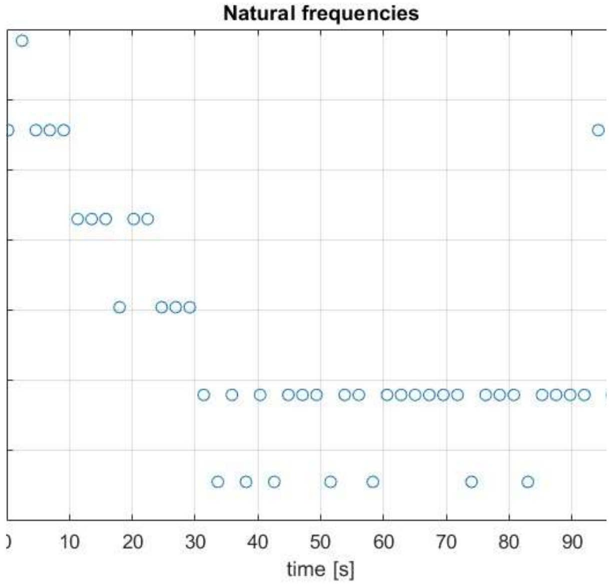
The final damping values for each cable are then computed averaging the (filtered) values included between 15 s and 35 s.

3. First natural frequency

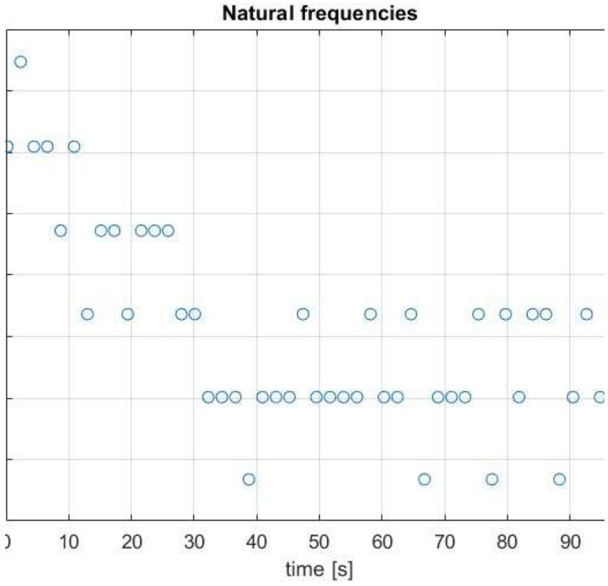
The natural frequencies of the cables are assumed to be equal to the damped ones since the factor $\sqrt{1-h_i^2}$ is really close to 1.

$$f_i = \frac{4}{t(i+4)-t(i)}$$

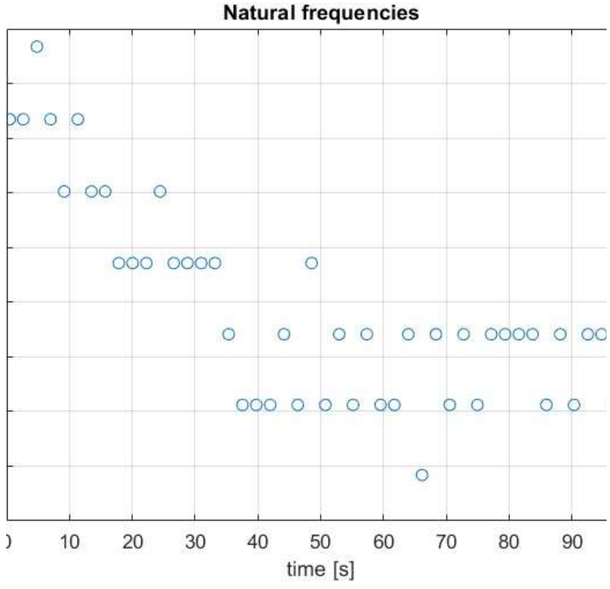
Cable 05



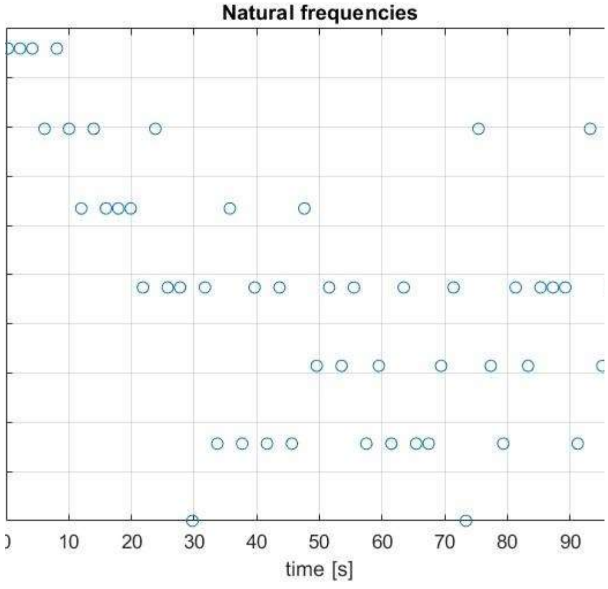
Cable 06



Cable 08

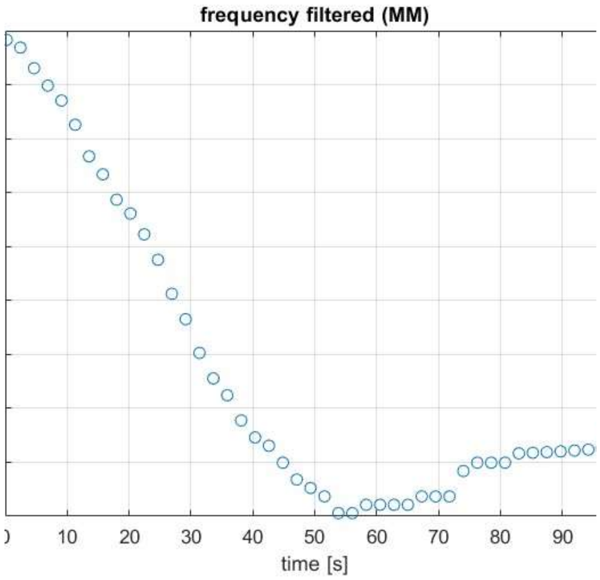


Cable 13

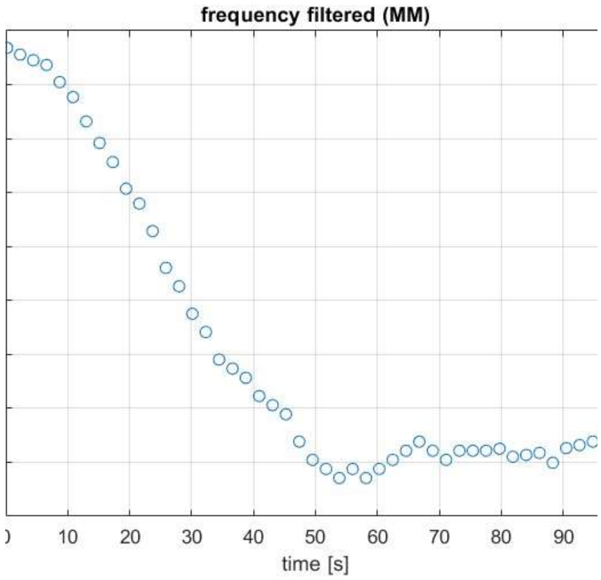


As done for the frequency, a MM filter is used (window size equal to 20): the final frequency values for each cable are the computed in the same way, averaging the values included between 15 s and 35 s.

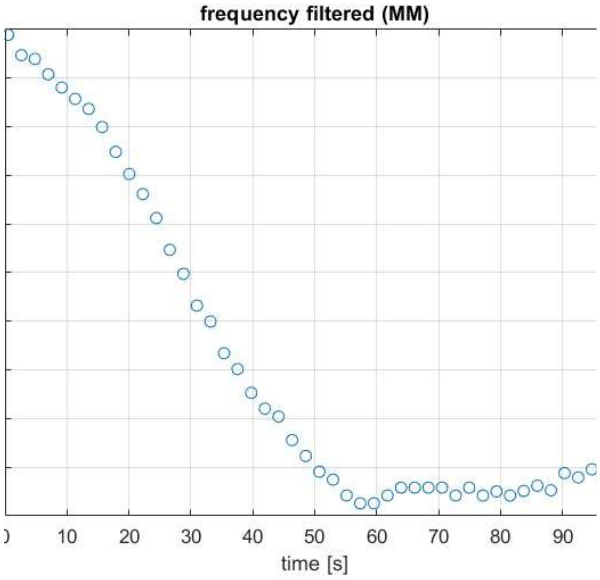
Cable 05



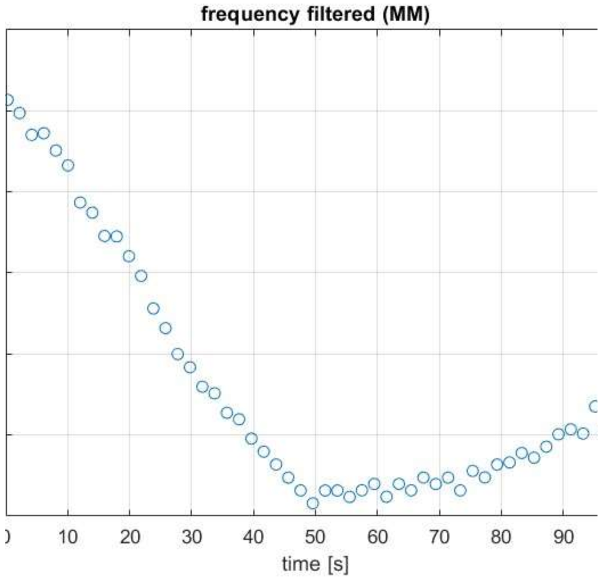
Cable 06



Cable 08



Cable 13



4. Estimation of the actual tension force

A further MatLab code has been written to evaluate each cable's tension according to the FEM model, opportunely modifying the *HovenringMain* script provided.
Knowing the (experimental) natural frequency of each cable, a cycle (*for*) was implemented in the code, varying each time the cable's tension provided as an input to the function *loadstructure*: a different cable structure was loaded each time, collecting as output all the first eigenfrequencies of the different systems.
Discretization steps were coarse at the beginning and finer each time the code was run. Thus, a vector of tensions was used, obtaining a vector of frequencies: the ones with the least absolute differences with the actual one individuated next tension vector's extremes, being aware that the FEM tension of the cable would have been for sure between these values.
The process has been repeated several times, decreasing each time discretization steps' amplitude and changing the maximum and minimum possible tension.

Moreover, the nominal frequency associated to each cable's nominal tension has been computed using the same code.

5. Conclusions and error estimation

We can report all the obtained results in a tabular form:

	Cable 05	**Cable 06**	**Cable 08**	**Cable 13**
f_{nom} [Hz]	1.7635	2.0376	1.7343	2.0527
f_{exp} [Hz]	1.7911	1.8639	1.8300	2.0275
h [‰]	4.7	5.7	5.5	5.7
e_f [%]	1.57	- 8.52	5.52	- 1.23
T_{nom} [kN]	470.000	705.000	470.000	705.000
T_{FEM} [kN]	485.822	549.736	527.484	661.340
e_T [%]	3.37	- 22.02	12.23	- 6.19

The estimated tension is expected to be within the range ± 5% of the actual tension value. For a tensioned cable:

$$f_1 = \frac{1}{2L} \cdot \sqrt{\frac{T}{m}}$$

$$f_1 + \Delta f = \sqrt{\frac{T + \Delta T}{m}} \cdot \frac{1}{2L} = \frac{1}{2L} \cdot \sqrt{\frac{T}{m} \cdot (1 + \frac{\Delta T}{T})}$$

From McLaurin's expansion:

$$f_1 + \Delta f \sim \frac{1}{2L} \cdot \sqrt{\frac{T}{m}} \cdot (1 + \frac{\Delta T}{2T})$$

$$\frac{\Delta f}{f_1} \sim \frac{1}{2} \cdot \frac{\Delta T}{T}$$

In the table the ratio ${}^{\Delta f}/_{f_1} = \frac{f_{exp}-f_{nom}}{f_{nom}}$ is called e_f and ${}^{\Delta T}/_{T} = \frac{T_{FEM}-T_{nom}}{T_{nom}}$ is called e_T. In fact, f_{exp} is the frequency associated to T_{FEM}.
Since the error related to T should be below 5%, the one related to the frequency should be below 2.5% (taking 2% as threshold here). As a consequence, for cables 06 and 08 tension differs from the nominal one for sure, while cable 13 is borderline.

However, it must be said that the approximation is valid only for the model of pinned-pinned tensioned cable, while we know that the system is much more complex. Moreover, only four cables have been considered by the jacking system to validate the FE model: maybe if different cables were chosen the outcome would have been different.

6. MATLAB script (cable 05_VELDTUI2)

6.1. Hovenring_cable05_VELDTUI2

```
close all;
clear all;
clc;
%% 1. Identification of the damping ratio
load("Cable05.mat");
dt = 1/fsamp;
t_max = (length(acc)-1)*dt;
% Definition of the time (t):
t = 0:dt:t_max;

% There's a problem: the mean value is not null. Thus we need to
% "filter" our signal:
acc = acc - mean(acc);

% We find the acc peaks and the relative time
[acc_peaks, i_vect_peaks] = findpeaks(acc);
t_peaks = t(i_vect_peaks);

% We select every N = 4 peaks
```

```
N = 4;
i_jump_peaks = i_vect_peaks(1:N:length(i_vect_peaks));
% We select the "jumping" acceleration peaks
acc_jump_peaks = acc(i_jump_peaks);
% We do the same for the time
t_jump_peaks = t(i_jump_peaks);

% We need to compute damping
h = [];
for ii = 1: length(i_jump_peaks)-1
h(end+1) = log(acc_jump_peaks(ii)/acc_jump_peaks(ii+1))/(2*N*pi);
end

% Moving average filter (to neutralize the noise)
windowSize = 20;
filtered_h = movmean(h, windowSize);

% We compute natural frequency
freq = [];
for ii = 1:length(i_jump_peaks)-1
freq = [freq, N/(t_jump_peaks(ii+1)-t_jump_peaks(ii))];
% We are assuming the natural frequency equal to the damped frequency,
% since sqrt(1-h^2) ~ 1.
end

% We plot the acceleration
figure(1);
plot(t, acc);
hold on;
title('Acceleration');
plot(t_peaks, acc_peaks, 'o');
grid on;
xlabel('time [s]');
ylabel('acceleration [m/s^2]');
plot(t_jump_peaks, acc_jump_peaks, 'x');
hold off;

% We now plot the non-filtered damping ratio (without MM)
figure(2);
```

```
% We need to plot only until (end - 1) since the logathmic decrement needs
% two consecutive values to compute one
plot(t_jump_peaks(1:end-1), h, 'o');
hold on;
title('h');
grid on;
xlabel('time [s]');
ylabel('damping ratio [-]');
hold off;

% We now plot the filtered damping ratio
figure(3);
plot(t_jump_peaks(1:end-1), filtered_h, 'o');
hold on;
title('h filtered (MM)');
grid on;
xlabel('time [s]');
ylabel('filtered h [-]');
hold off;

% Plot of the natural frequencies
figure(4);
plot(t_jump_peaks(1:end-1), freq, 'o')
hold on;
title('Natural frequencies');
grid on;
xlabel('time [s]');
ylabel('frequency [Hz]');
hold off;

% Plot of the natural frequencies (MM)
windowSize = 20;
filtered_freq = movmean(freq, windowSize);
figure(5);
plot(t_jump_peaks(1:end-1), filtered_freq, 'o');
hold on;
title('frequency filtered (MM)');
grid on;
xlabel('time [s]');
```

```
ylabel('filtered frequency [Hz]');
hold off;

% Three different regions are identified: large amplitudes, with non-linear
% effects (1), smaller amplitudes in the standard operative amplitude (2),
% no more efficient TMD with lowest damping ratio (3). We look for the second
% region from the vector "acc_peaks", looking at the "change in the
% derivative".
first_extremity = 15;
second_extremity = 35;
damping_reliable = filtered_h(t_jump_peaks >= first_extremity &...
t_jump_peaks <= second_extremity);
damping_ratio = mean(damping_reliable)

%% 2. Identification of the first natural frequency
freq_reliable = filtered_freq(t_jump_peaks >= first_extremity & t_jump_peaks <=
second_extremity);
freq1 = mean(freq_reliable)

%% Tension force estimated by the FE model, thanks to frequencies
% We use the HovenringMain script changing time by time the tension
% We slightly changed the script "HovenringMain" (keeping the same
% "load structure") to use a proper algorithm.

% freq1 = 1.7911, T_nom = 470000
% NOTE: being the nominal tension already known, the excitation
% frequencies that the cable undergoes surely are in the neighbourhood
% of the theoretical first natural one

% Using HovenringMain_05, we extract the tension derived from the
% FEM modelling, using values which are comparable to the nominal one as
% extremities at the beginning and steps which are increasingly narrower.
T_FEM = 485822
% increment of 3.3664%
```

6.2. HovenringMain_05

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mechanical System Dynamics
% FEM script
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```matlab
clear all;
close all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
min_T = 485000;
max_T = 486000;
step_T = 100;
Tension = min_T:step_T:max_T;
freq1 = 1.7911;
frequencies1 = [];
for tension = Tension
[file_i,xy,nnod,sizee,idb,ndof,incid,l,gamma,m,EA,EJ,T,posit,nbeam,pr]=loadstructu
re(tension, 'Cable05_VELDTUI2');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Draw structure
% dis_stru(posit,l,gamma,xy,pr,idb,ndof);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Assemble mass and stiffness matricies
[M,K] = assem(incid,l,m,EA,EJ,T,gamma,idb);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Add concentrated elements
% Torsional spring - deck side
kT1 = 9e6;
i_ndof_spring = idb(1,3);
K(i_ndof_spring,i_ndof_spring) = K(i_ndof_spring,i_ndof_spring)+kT1;

% Torisional spring - tower side
kT2 = 9e6;
i_ndof_spring = idb(230,3);
K(i_ndof_spring,i_ndof_spring) = K(i_ndof_spring,i_ndof_spring)+kT2;

% TMD 1 - 04R36 - Mass and stiffness
m_tmd1 = 4.93;
k_tmd1 = 2.12E+03;
i_ndof1 = idb(149,1);
i_ndof2 = idb(231,1);
i_dof_tmd1 = [i_ndof1 i_ndof2];

M(i_ndof2,i_ndof2) = M(i_ndof2,i_ndof2) + m_tmd1;

K_tmd1 = [k_tmd1 -k_tmd1; -k_tmd1 k_tmd1];
K(i_dof_tmd1,i_dof_tmd1) = K(i_dof_tmd1,i_dof_tmd1)+K_tmd1;

% TMD 2 - 04R36 - Mass and stiffness
m_tmd1 = 5.1;
k_tmd1 = 888;
i_ndof1 = idb(149,1);
i_ndof2 = idb(232,1);
i_dof_tmd1 = [i_ndof1 i_ndof2];

M(i_ndof2,i_ndof2) = M(i_ndof2,i_ndof2) + m_tmd1;

K_tmd1 = [k_tmd1 -k_tmd1; -k_tmd1 k_tmd1];
K(i_dof_tmd1,i_dof_tmd1) = K(i_dof_tmd1,i_dof_tmd1)+K_tmd1;
```

```
% TMD 1 - 4RZ11 - Mass and stiffness
m_tmd1 = 5;
k_tmd1 = 4.00E+03;
i_ndof1 = idb(164,1);
i_ndof2 = idb(233,1);
i_dof_tmd1 = [i_ndof1 i_ndof2];

M(i_ndof2,i_ndof2) = M(i_ndof2,i_ndof2) + m_tmd1;

K_tmd1 = [k_tmd1 -k_tmd1; -k_tmd1 k_tmd1];
K(i_dof_tmd1,i_dof_tmd1) = K(i_dof_tmd1,i_dof_tmd1)+K_tmd1;

% TMD 2 - 4RZ11 - Mass and stiffness
m_tmd1 = 7;
k_tmd1 = 1.17E+04;
i_ndof1 = idb(164,1);
i_ndof2 = idb(234,1);
i_dof_tmd1 = [i_ndof1 i_ndof2];

M(i_ndof2,i_ndof2) = M(i_ndof2,i_ndof2) + m_tmd1;

K_tmd1 = [k_tmd1 -k_tmd1; -k_tmd1 k_tmd1];
K(i_dof_tmd1,i_dof_tmd1) = K(i_dof_tmd1,i_dof_tmd1)+K_tmd1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute natural frequencies and mode shapes
MFF = M(1:ndof,1:ndof);
MCF = M(ndof+1:end,1:ndof);
MFC = M(1:ndof,ndof+1:end);
MCC = M(ndof+1:end,ndof+1:end);

KFF = K(1:ndof,1:ndof);
KCF = K(ndof+1:end,1:ndof);
KFC = K(1:ndof,ndof+1:end);
KCC = K(ndof+1:end,ndof+1:end);

[modes, omega] = eig(MFF\KFF);
omega = sqrt(diag(omega));
[omega,i_omega] = sort(omega);
freq0 = omega/(2*pi);
modes = modes(:,i_omega);
% We take the first frequency for each value of tension then:
frequencies1 = [frequencies1, freq0(1)];
end
% We look for the closest values to the freq1, computed through the main
% code, and we get the corresponding extremities for the next tension's
% iteration
difference_f = abs(frequencies1-freq1);
[ordered_differences, place] = sort(difference_f);
T_extremity1 = Tension(place(1))
T_extremity2 = Tension(place(2))
% Then we go on using these extremities and a narrower step_T
```