Fundamentals of Vibration Analysis and Vibroacoustics
Module 2 - Vibroacoustics of Musical Instruments
Assignment 2 - Experimental modal analysis of a violin

Bombaci Nicola 10677942
Fantin Jacopo 10591775
Intagliata Emanuele 10544878

June 2020

1) Matlab code for modes identification

Starting from the experimental data collected into `Data.mat` file we implemented the Matlab script for identifying the modes according to the procedure described in laboratory slides.

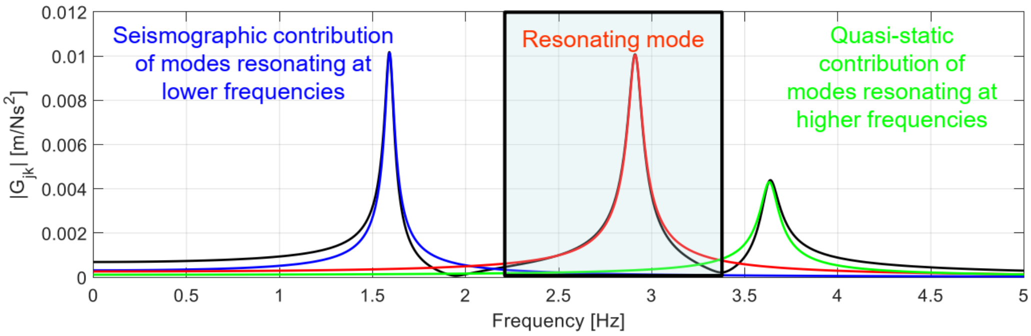
1a) Minimization algorithm

In order to study the vibrational modes of the violin starting from these experimental data, we refer to the **modal approach**, that describes the overall frequency response of a structure as superposition of N modes of vibration, as far as enough modes are considered:

$$G_{jk}^{\text{EXP}}(\omega) \approx \sum_{i=1}^{N} \frac{X_j^{(i)} X_k^{(i)}/m_i}{-\omega^2 + j2\omega_i\omega\xi_i + \omega_i^2} \quad , \quad \xi_i = \frac{c_i}{2m_i\omega_i} \tag{1}$$

For well distinguished peaks and lightly damped structures, the FRF $G_{jk}(\omega)$ can be approximated around a certain ω_i as:

$$G_{jk}^{\text{NUM}}(\omega) \approx \sum_{i=1}^{N} \frac{A_{jk}^{(i)}}{-\omega^2 + j2\omega_i\omega\xi_i + \omega_i^2} + R_{jk}^{H} + \frac{R_{jk}^{L}}{\omega^2} \quad , \quad \omega_{\text{inf}} < \omega_i < \omega_{\text{sup}} \tag{2}$$



The first thing we had to implement was an algorithm in order to find the frequency ranges of interest. We decided to divide the frequency axis in N partitions (where N is the number of identified peaks or considered modes).

Given the N peaks found (i.e. N maxima, see 2b)) we found $N + 1$ minima between those peaks (and the endpoints of the frequency axis).

Listing 1: Frequency range search

```
        wp_dx = iini;
        for yy = 1: (Nmodes + 1) % over the m(+1) peaks
                wp_sx = wp_dx; % left end is former right end

        if yy > Nmodes % right end
                wp_dx = i_max;
        else
                wp_dx = indices(yy); % frequency of next peak (right
                   end)
        end
        [~, min_index] = min(abs(FRF_LP(wp_sx:wp_dx, :)), [], 1);
        rfi(:,yy) = min_index + wp_sx - 1;
end
```

We obtain a set of $N+1$ values of ω (i.e. the endpoints of the frequency ranges) for each measurement.

This is maybe not the most precise way to deal with this kind of problem but it can be seen in the last section that good results are delivered.

For a given set of experimental FRFs G^{EXP}, obtained for a fixed measurement location j and different excitation locations k (grid of points), a **least squares minimization procedure** can be implemented for the estimation of the **modal parameters** $(X^{(i)}, \omega_i, \xi_i, m_i)$.

Considering the experimental FRF matrix $\mathbf{G}^{\text{EXP}}$, it shows:

a number m of rows corresponding to the length of the frequency vector ω;

a number n of columns corresponding to the $k - j$ pairs (i.e. the available FRFs).

Here we have that:

$G_r^{\text{EXP}}(\omega_s)$ is a generic element of the experimental FRF matrix $\mathbf{G}^{\text{EXP}}$ corresponding to the r-th column (FRF) evaluated in correspondence of the frequency ω_s;

$G_r^{\text{NUM}}(\omega_s)$ is the numerical FRF estimation around a certain ω_i, corresponding to the r-th column (FRF) evaluated in correspondence of the frequency ω_s, i.e.

$$G_k^{\text{NUM}}(\omega_s) = \sum_{i=1}^{N} \frac{A_r^{(i)}}{-\omega_s^2 + j2\omega_i\omega_s\xi_i + \omega_i^2} + R_r^H + \frac{R_r^L}{\omega_s^2}$$

1b) Error minimization problem

The **error function** to be minimized should then be:

$$\varepsilon = \sum_{r=1}^{N} \sum_{s=s_{\text{inf}}}^{s_{\text{sup}}} \Re^2\{G_r^{\text{EXP}}(\omega_s) - G_r^{\text{NUM}}(\omega_s)\} + \Im^2\{G_r^{\text{EXP}}(\omega_s) - G_r^{\text{NUM}}(\omega_s)\} \tag{3}$$

Since the error function depends non-linearly from the unknown parameters, an *iterative* minimization procedure must be used.

For an iterative procedure an **initial guess vector** x_0 is required, consisting of a preliminary estimate of:

ω_i which is found from the maximum peak in the considered frequency range (see section 2b);

ξ_i which is found through some simplified method, assuming that just the resonating mode is contributing to the system response. We used the **phase derivative method**:

$$\xi_i = -\frac{1}{\omega_i \frac{\partial \angle G_{jk}}{\partial \omega}\Big|_{\omega=\omega_i}}$$

$A_r^{(i)}$ which is found considering each FRF at resonance and assuming real valued mode shapes:

$$A_r^{(i)} \approx -\Im(2\xi_i\omega_i^2 G_r^{\text{EXP}}(\omega_i))$$

R_r^L and R_r^R are set to zero under the assumption of sufficiently distinguished peaks.

It can be seen that the non-linear minimization procedure elaborates *simultaneously* the whole set of FRFs, leading to an estimate of $\omega_i, \xi_i, A_r^{(i)}$. This is not a trivial problem from a computational point of view.

Firstly we tried to implement this procedure using `fminsearch` function. It finds the minimum of unconstrained multivariable function using derivative-free method. It is a nonlinear programming solver and searches for the minimum of a problem specified by

$$\min_x f(x)$$

where `f(x)` is a function that returns a scalar, and `x` is a vector or a matrix.

In our case `x` is a set of modal parameters vectors (one for each measurement) and `f(x)` should be the error function seen in eq. (3).

This approach led to a runtime of almost an hour.

In our second attempt we used the MATLAB function `lsqnonlin` (as suggested in laboratory slides). This function solves a non-linear least-squares curve fitting problem of the form:

$$\min_x ||f(x)||_2^2 = \min_x \left(f_1(x)^2 + f_2(x)^2 + \ldots + f_n(x)^2\right)$$

Rather than computing the value $||f(x)||_2^2$ (the sum of squares), `lsqnonlin` requires the user-defined function to compute the *vector-valued* function:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

In other words, `x = lsqnonlin(fun, x0)` starts at the point `lsqnonlin(fun,x0)` and finds a minimum of the sum of squares of the functions described in `fun`. The function `fun` should return a vector (or array) of values and not the sum of squares of the values. The algorithm implicitly computes the sum of squares of the components of `fun(x)`.

This function forces us to use a slightly simplified version of the **error function**:

$$\varepsilon = \sum_{s=s_{\text{inf}}}^{s_{\text{sup}}} \Re^2\{G^{\text{EXP}}(\omega_s) - G^{\text{NUM}}(\omega_s)\} + \Im^2\{G^{\text{EXP}}(\omega_s) - G^{\text{NUM}}(\omega_s)\} \tag{4}$$

In this way the non-linear minimization procedure elaborates the FRFs *one-by-one*.

This leads us to a less precise minimization algorithm. On the other hand now it is much faster.

Using `lsqnonlin` we provide a `f(x)` of this form:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = \begin{bmatrix} \Re\{G^{\text{exp}}(\omega_s) - G^{\text{num}}(\omega_s)\} \\ \Im\{G^{\text{exp}}(\omega_s) - G^{\text{num}}(\omega_s)\} \end{bmatrix}$$

Remember: the algorithm implicitly computes the sum of squares of the components of `f(x)`.

Listing 2: Snippet of code from `err_i.m`

```
        % error computation
        c_e = H_exp - H_num; %complex error
        real_e = (real(c_e));
        imag_e = (imag(c_e));
        error = [real_e; imag_e];
```

In the end the minimization process gives us an estimation of the modal parameters $\xi_i, \omega_i, X^{(i)}, R_r^L$ and R_r^R.

Setting $m_i = X_r^{(i)}$, the normalized mode shape at each of the k locations is given by the gain $A_r^{(i)} = X_r^{(i)}$ Other useful modal parameters can be found in function of the previous ones:

$$c_i = 2m_i\omega_i\xi_i$$
$$k_i = \omega_i^2 m_i$$

1c) FRF Reconstruction

The FRFs are reconstructed using the custom function `reco.m`. It takes as input a vector with modal parameters and gives as output the reconstructed FRF, using the formulation seen in eq. (1).

Listing 3: Snippet of code from `reco.m`

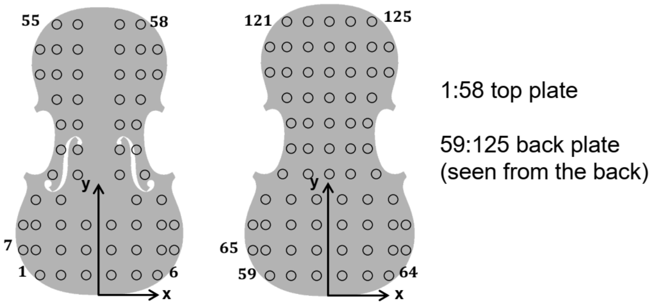
```
        for mm = 1:meas
                for pp = 1:nPeaks
                        FRF_reco(:, mm) = FRF_reco(: , mm) + ...
                                (Aj(pp,mm) + 1i*Bj(pp,mm))./(- m_i(pp
                                   ,mm).*omega.^2 + 1i*c_i(pp,mm).*
                                   omega +
                                k_i(pp,mm));
                end
        end
```

Quality of the estimates can be visually assessed comparing in a plot the identified FRFs G^{NUM} with the experimental ones G^{EXP} . We will see it in the next sections.

2) Software testing

2a) Data pre-processing

The given `Data.mat` file contains experimental FRFs $[m/s^2N]$ that have been obtained performing a hammer test on a violin, following a roving hammer procedure, that consists in fixing the measurement position(s) while varying excitation location. The coordinates of the measuring grid can be found in the variable `xy`. The measuring grid is reported in the figure below.



To sum up, `Data.mat` contains the frequency vector (`freq`), the experimental FRF matrix (`FRF`, measurement direction positive outward with respect to the top plate), the measurement grid (`xy`), the contours of front plate (`xy_bt`) and back plate (`xy_bf`).

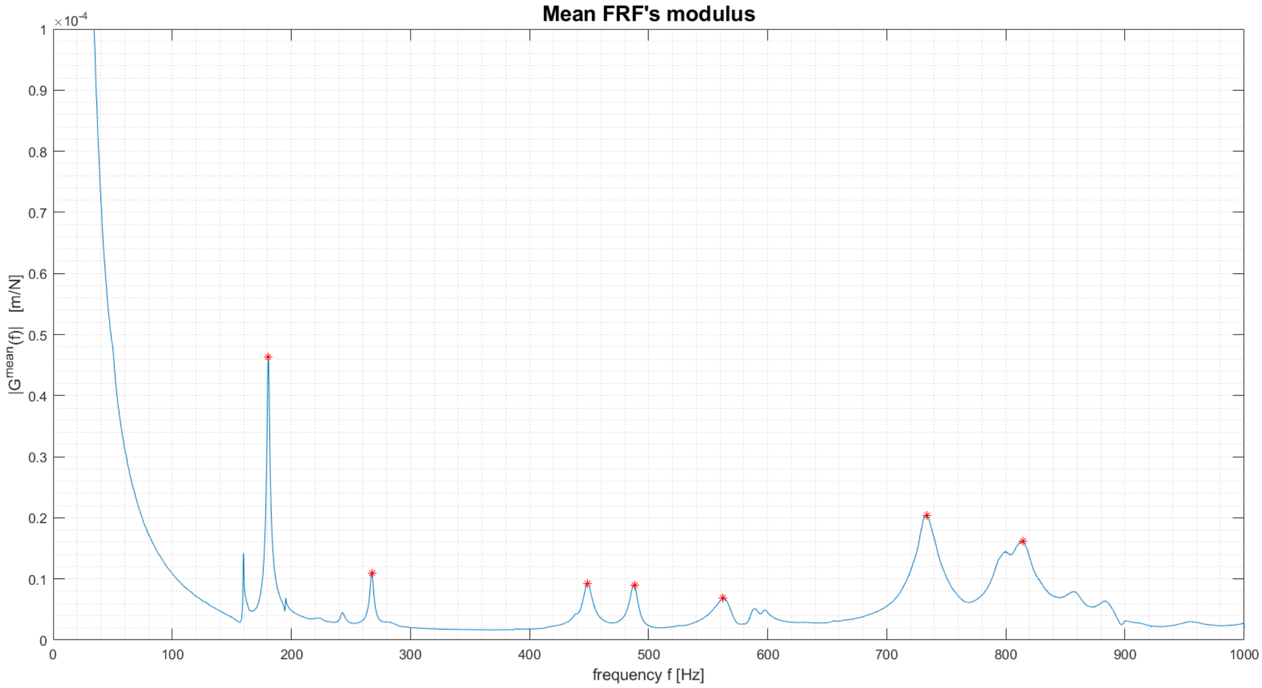
First of all we converted the unit of measurement of the FRFs from an acceleration over a force $[m/s^2N]$ to a displacement over a force $[m/N]$. To do so we divided the FRFs by the vector ω^2 using the function `repmat`.

Being the frequency response spectra very noisy, we decided to smooth them out, extracting only the envelope.

2b) Peak search

The steps we followed to identify the resonance frequencies, which correspond to the relative maxima of the module diagram of the FRFs:

- Computing the mean FRF over all the low-passed FRFs, considering those measured for points on the front plate together with those for points on the back plate, and we low-passed again the result. This way, we would be quite sure to identify vibration modes in common between the two plates.



- Using `findpeaks` to identify the most evident maxima in the mean FRF. In particular, the following snippet of code in used for searching for peaks.

Listing 4: Peaks' search

```
prom = 0.000003;
width = 8;
tr = 0;
iini = 500;
[~, indices] = findpeaks(FRFmean(iini:end), 'MinPeakProminence', prom,
    'MinPeakWidth', width, 'Threshold', tr);
Nmodes = length(indices);
indices = indices + iini;
```

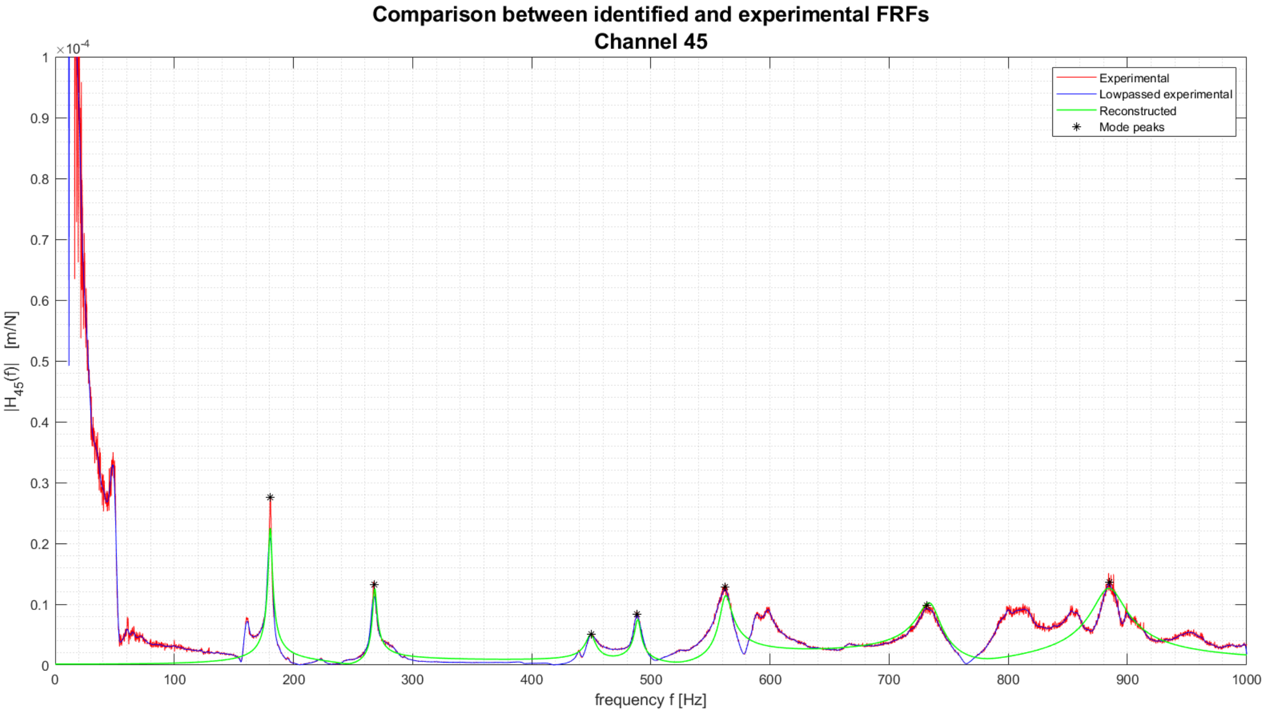
As it can be seen the first 500 samples are discarded in order not to consider the initial part of the FRF that diverges. This can be attributed to the structure used for holding the violin, so we're allowed to pretend the FRF starts after this region.

- Finding the frequency value associated with each peak. These constitute the first-guess-values for the minimization process as for the natural frequencies.

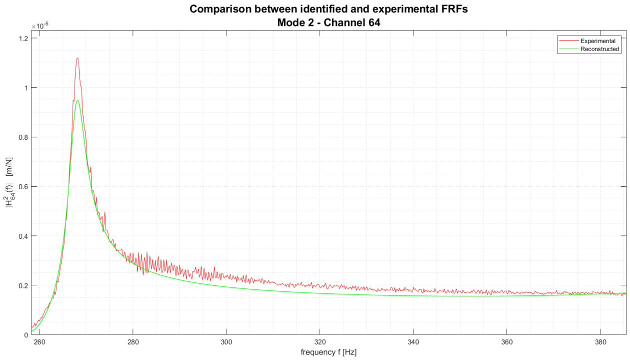
3) Results

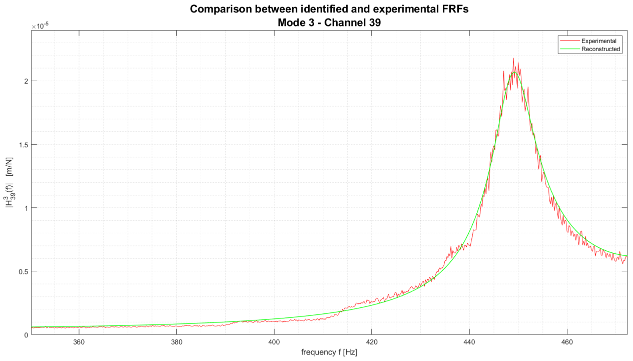
3a) Comparison between reconstructed and experimental FRFs

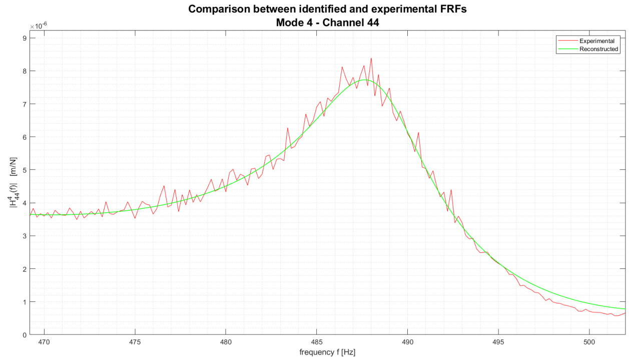
Our MATLAB script identifies 7 modes. Considering one of the 125 input-output couples, we report the reconstructed Frequency Response Function compared to the corresponding experimental one. The numerical one provides a good approximation of the original one, following its amplitude envelope. We also remind that, in case the value of the modulus for one of the identified natural frequencies is not the maximum value in the considered frequency range, we considered the actual maximum value, and its corresponding frequency, for the reconstruction of that portion of FRF. This is the case for the last peak in the showed diagram, whose frequency is 884 Hz in place of 814 Hz.

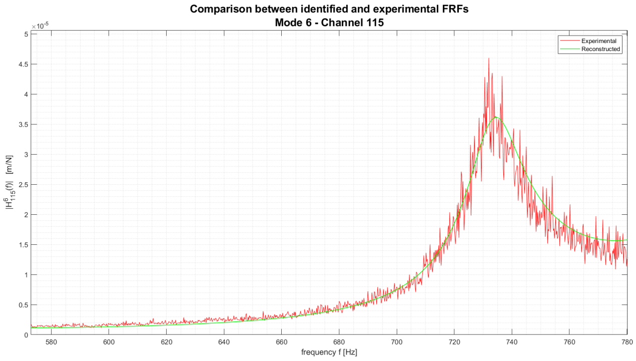


Taking a closer look at some of the identified vibration modes for some of the measurements, we may appreciate even more the fact a non-noisy FRF has been analitically obtained:









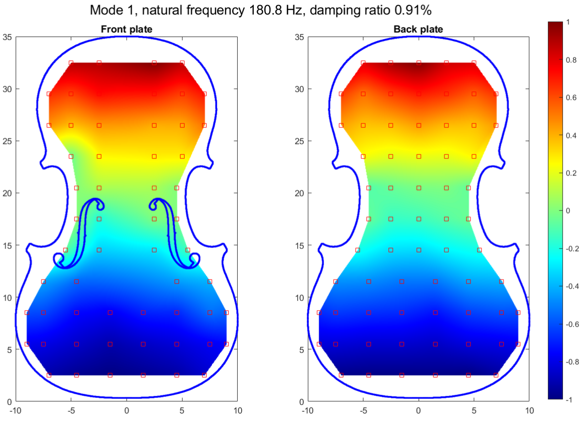
3b) Modal parameters

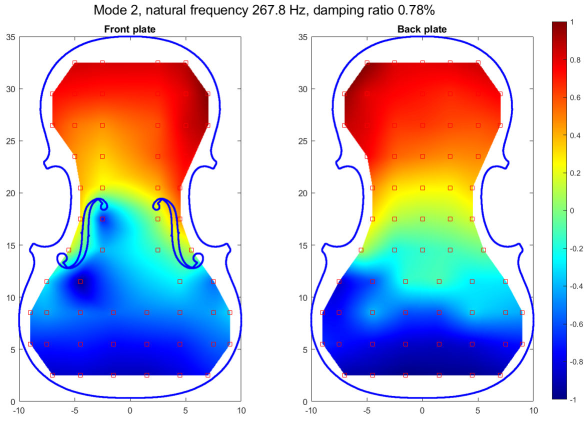
The results in terms of natural frequencies and damping ratios for the 7 identified modes can be summarized in the following table.

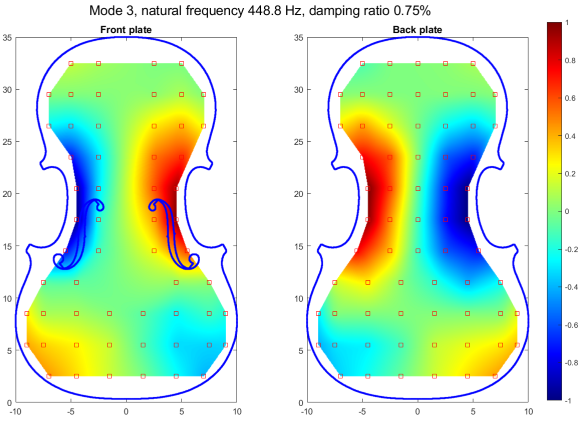
Mode	Frequency [Hz]	Damping ratio (%)
1	180.8	0.91%
2	267.8	0.78%
3	448.8	0.75%
4	488.2	0.58%
5	562.6	1.08%
6	733.6	1.22%
7	814.2	1.49%

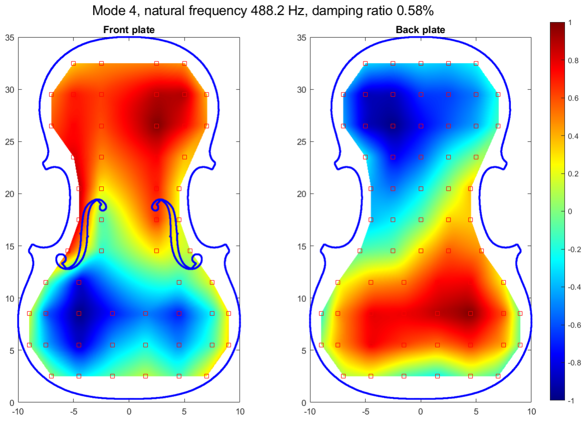
3c) Identified modes

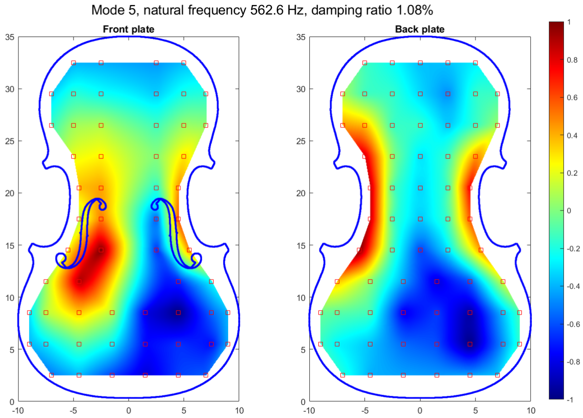
The following plots summarize the characteristics of the vibration modes of the violin plates: for each of them, we're displaying natural frequency, adimensional damping ratio (percentage) and a graphical representation of the mode shape, indicating the displacement of the points on the surface with respect to each other for that particular oscillation mode.

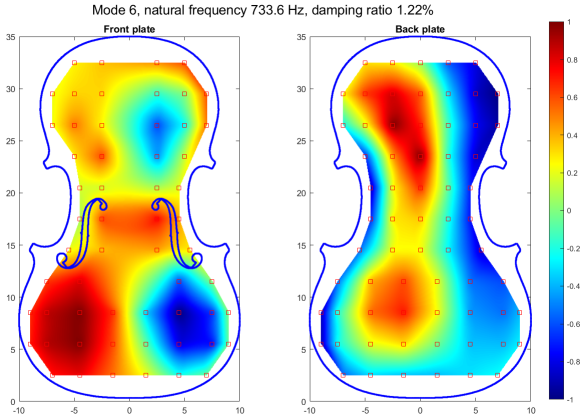


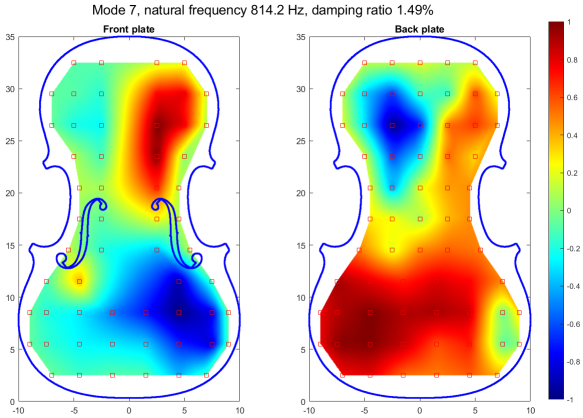







Mode 5, natural frequency 562.6 Hz, damping ratio 1.08%


Mode 6, natural frequency 733.6 Hz, damping ratio 1.22%


Mode 7, natural frequency 814.2 Hz, damping ratio 1.49%

As we could infer a priori, the mode shape is more and more complex as the resonating frequency increases: in the first two modes, we can easily identify two main oscillating areas (the upper part and the lower part of the plate), in accordance between top and back plate. In the third one, there are four, with crossed polarization and accordance between top and back plate, and so on. As the mode number increases, the two plates start decouple one from the other, and the oscillation modes may not correspond anymore. This is due to the fact the higher the frequencies, the more top and back plate exhibit different modes. Talking about the damping ratio, this is on average lower at lower frequencies than at higher ones.