

Modal Expansion of Input Impedance measurements

Albert-Ngabo Niyonsenga

December 14, 2022

1 Introduction

Modal analysis is a technique for deriving the modes of an acoustic system. These modes can then be used to represent the system. In particular we can represent the input impedance of wind instruments as the sum of a set of modes. This can be used for time-domain simulations and linear stability analysis of the input impedance signal. There are many different methods of deriving these modal parameters. In this project we will implement the algorithm proposed by Ablitzer, Frédéric 2021 to derive the modal parameters of the input impedances of wind instruments. This algorithm uses a peak-picking approach to identify peaks.

1.1 Objectives

The goals of this project are to:

- Implement a Peak-Picking Algorithm for modal expansion
- Develop a MATLAB app for users to use the algorithm
- Test the algorithm using measured input impedance of a clarinet.

2 Modal Expansion of Input Impedance

The input impedance of a wind instrument can be represented as a sum of complex modes. with s_n and s_n^* representing the pole and c_n is the associated residue.

$$Z(\omega) = \sum_{n=1}^N \frac{c_n}{j\omega + s_n} + \frac{c_n^*}{j\omega + s_n^*} \quad (1)$$

The poles can be represented in terms of a modal frequency ω_n and the modal damping factor ξ_n .

$$s_n = -\xi_n \omega_n + j\omega_n \sqrt{1 - \xi_n^2} \quad (2)$$

This allows us to rewrite Eq 1 as:

$$Z(\omega) = \sum_{n=1}^N \frac{j\omega A_n + B_n}{\omega_n^2 - \omega^2 + j2\xi_n \omega_n \omega} \quad (3)$$

A_n and B_n are real amplitudes where:

$$\begin{aligned} A_n &= 2\Re(c_n) \\ B_n &= -2\Re(c_n s_n^*) \end{aligned} \quad (4)$$

If we represent frequencies $\vec{\omega}$ as a vector where every row of the vector is the impedance as a given frequency then we can represent Eq. 3 as follow:

$$d(\vec{\omega}) = M(\vec{\omega}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) \vec{x} \quad (5)$$

where

$$\begin{aligned} \vec{x} &= \begin{bmatrix} A_1 \\ \vdots \\ A_N \\ B_1 \\ \vdots \\ B_N \end{bmatrix} \\ M(\vec{\omega}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) &= \begin{bmatrix} \Re(M_a) \Re(M_b) \\ \Im(M_a) \Im(M_b) \end{bmatrix} \\ d(\vec{\omega}) &= \begin{bmatrix} \Re(Z(\vec{\omega})) \\ \Im(Z(\vec{\omega})) \end{bmatrix} \end{aligned} \quad (6)$$

with

$$\begin{aligned} M_a[i, n] &= \frac{j\omega[i]}{\omega_n^2 - \omega[i] + j2\xi_n\omega_n\omega[i]} \\ M_b[i, n] &= \frac{1}{\omega_n^2 - \omega[i] + j2\xi_n\omega_n\omega[i]} \end{aligned} \quad (7)$$

Therefore the identification problem can be posed as solving for four unknowns: $\omega_n, \xi_n, A_n, B_n$. The real and imaginary sections are split so that when we solve for the modal amplitudes and frequencies we can be certain to have real values hence ensuring that Eq. 3 is valid.

2.1 Identifying poles

Identifying the poles can be done by solving a semi-local optimization problem. We keep all the currently identified poles in the basis the same, and optimise for only one pole at a time. We can represent this problem as such:

$$(\hat{\omega}_n, \hat{\xi}_n) = \underset{(\omega_n, \xi_n)}{\operatorname{argmin}} J(\vec{v}, \hat{\omega}_1, \hat{\xi}_1, \dots, \omega_n, \xi_n, \dots, \hat{\omega}_N, \hat{\xi}_N) \quad (8)$$

Where \vec{v} is a column vector containing all the angular frequencies that are considered for the cost function and $\hat{\omega}_n$, and $\hat{\xi}_n$ are estimates of the modal frequency and damping factor. We can define the cost function $J(\vec{v}, \hat{\omega}_1, \hat{\xi}_1, \dots, \omega_n, \xi_n, \dots, \hat{\omega}_N, \hat{\xi}_N)$ as:

$$J(\vec{v}, \hat{\omega}_1, \hat{\xi}_1, \dots, \omega_n, \xi_n, \dots, \hat{\omega}_N, \hat{\xi}_N) = \|M(\vec{v}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) \hat{x}(\vec{v}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) - d(\vec{v})\|^2 \quad (9)$$

2.2 Estimating Modal amplitudes

Given a known set of modal frequencies and damping factors we can estimate the modal amplitudes by Eq. 10 in a least squares fashion.

$$\hat{x}(\vec{v}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) = \underset{x}{\operatorname{argmin}} \|M(\vec{v}, \omega_1, \xi_1, \dots, \omega_N, \xi_N) \vec{x} - d(\vec{v})\|^2 \quad (10)$$

3 Implementation of Algorithm

The algorithm was implemented in MATLAB as a class with methods for the four elementary actions as well as some actions that are shared amongst the actions. This was packaged in a MATLAB App designed in the App Designer to visualise the actions of the algorithm.

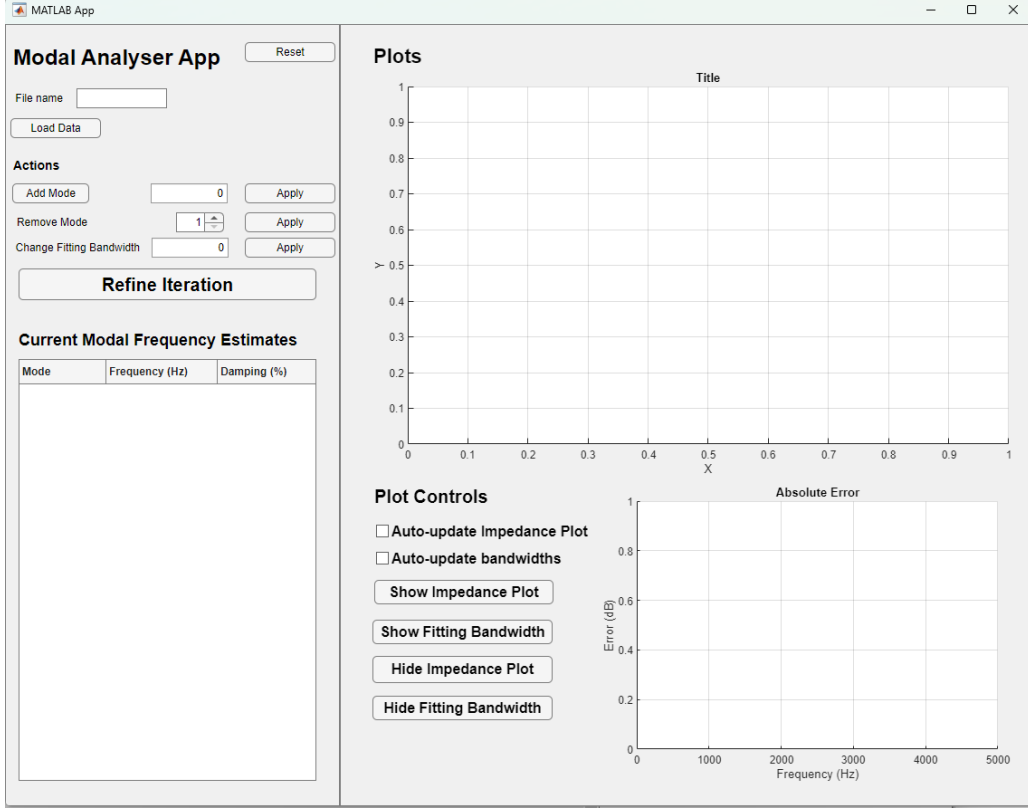


Figure 1: User Interface of the App

The Load Data button allows the user to select a ".mat" file containing there impedance data. On the left hand side of the app, all the actions for the algorithm are present. Clicking the Add mode button allows the user to manually select a peak from the impedance plot. Clicking the "Apply" button applies the action. The table below the Refine Iteration button automatically updates to show the most recent estimate for the poles and damping.

The top graph on the right displays the measured impedance data in blue, and the simuated impedance in black. It can also show the current bandwidth with red dotted lines. The error plot on the bottom right hows the difference between the measured and simulated impedance in dB (ie: the ratio between the measured and simulated impedance). This provides a visual cue to guide users for how well there current estimate fits the measurements.

4 Results and Discussion

The clarinet impedance data provided by Gary Scavone was used to do preliminary analysis of the algorithm. The analysis consisted of looking at the accuracy of the reconstructed impedance in terms of peak frequency location and magnitude. This was done by visually inspecting the error graph and the impedance plot.

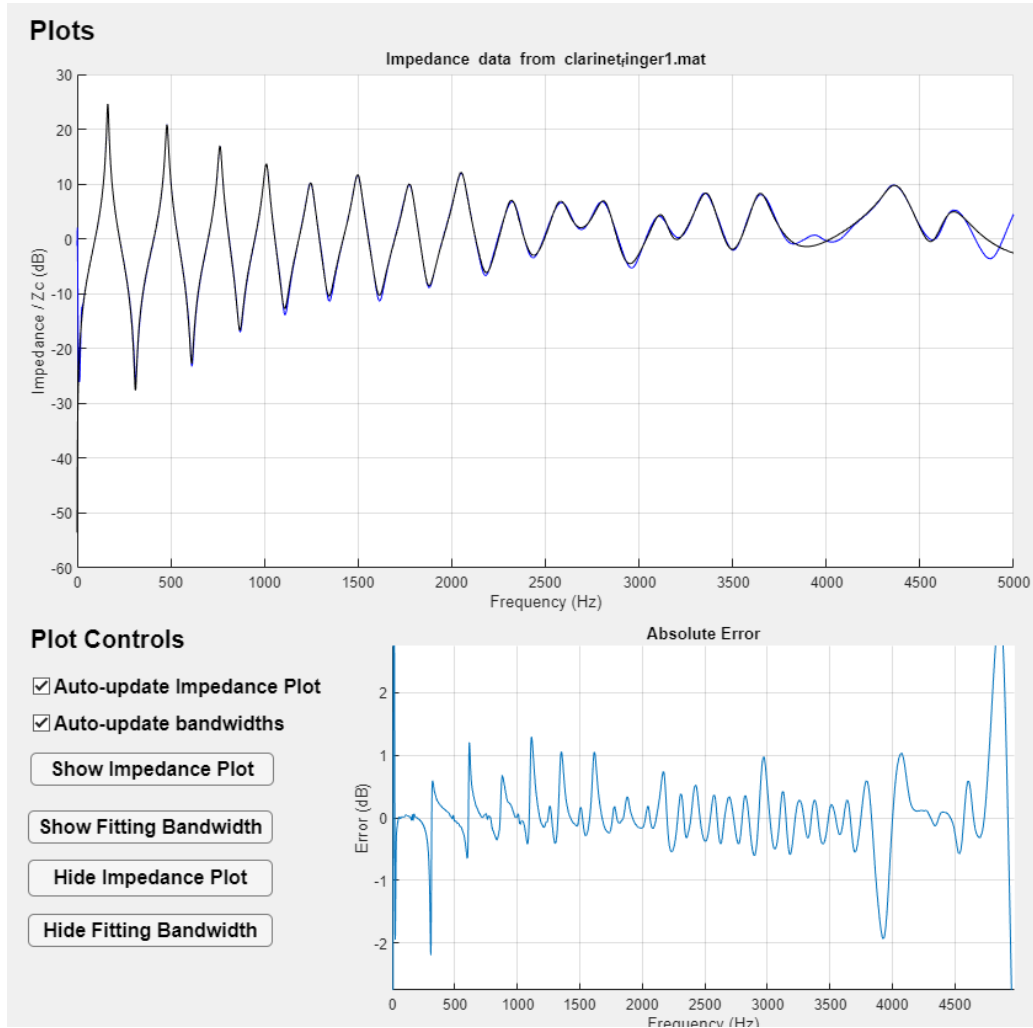
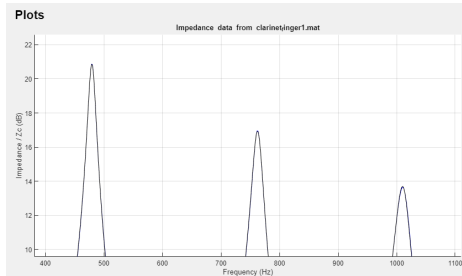
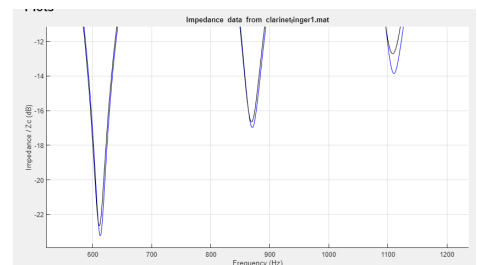


Figure 2: Simulated Impedance of Clarinet using 18 modes

As can be seen from fig.2 the locations of the peak frequencies for the simulated and measured impedance are in agreement. The magnitudes at the peaks are in agreement for the peaks. However, the valleys tend to have a higher error. Fig. 3 shows the bigger error in the valleys. This error is what changes the most during the refine iteration steps and rarely goes to zero even for the lower frequency modes.



(a) Zoomed in graph of the 3rd, 4th and 5th peak



(b) Zoomed in graph of the 3rd, 4th and 5th troughs

Figure 3: Comparison between magnitude estimate of peaks and troughs

5 Conclusion

This project focused mostly on the implementation of the algorithm and the design of the user interface. The actions of the algorithm were implemented in MATLAB and an app was designed to allow users to experiment and play around with the algorithm. In that sense I would say the project was a success. The app mostly functions although the lack of error handling means it is not robust to user error. The visual update of the simulated impedance offers a good visual aid to users to see how accurate their current estimate is. Furthermore, the error graph shows where the biggest errors in magnitude lie.

Future work includes improving the basic functionality of the app. This includes improving error handling, especially in terms of loading input data. Furthermore, a graph showing the value of the cost function, with each iteration will improve the feedback to users to see when refining the iteration provides no additional value. In addition, adding functionality to allow MATLAB to guess a specified amount of modes will help simplify adding a large amount of modes.

6 Code

The code can be found in this public Github Repository. Run `peakPickingApp.mlapp` to use the app. Opening this file in MATLAB's app designer will allow you to see the code used to make the GUI.

References

Ablitzer, Frédéric (2021). "Peak-picking identification technique for modal expansion of input impedance of brass instruments". In: *Acta Acust.* 5, p. 53. DOI: 10.1051/aacus/2021046. URL: <https://doi.org/10.1051/aacus/2021046>.