Politecnico di Milano

# Homework Lab#3 - Musical Instruments modeling

Students: Vittoria Malaman [10610731] - Riccardo Martinelli [10456202] - Matteo Pettenò [10868930]

Course: Musical Acoustics – Professor: Fabio Antonacci
Due date: *December 5th, 2022*

## 1. FD scheme explanation

The starting point for the simulation of a C2 piano string is the analytical wave equation for the stiff and lossy string.

$$\frac{\partial^2 y}{\partial t^2} = c^2\frac{\partial^2 y}{\partial x^2} - k^2\frac{\partial^4 y}{\partial x^4} - 2b_1\frac{\partial y}{\partial t} + 2b_2\frac{\partial^3 y}{\partial x^2 \partial t} + \rho^{-1}f(x, x_0, t) \tag{1}$$

To compute the finite difference model, we used the following parameters for the string and the hammer respectively.

| **Piano elements** | **Parameters** | | **C2** |
|---|---|---|---|
| **String** | String fundamental frequency | $f_1$ | $65.4\,\mathrm{Hz}$ |
| | String length | $L$ | $1.92\,\mathrm{m}$ |
| | String mass | $M_S$ | $35 \times 10^{-3}\,\mathrm{Kg}$ |
| | String linear density $(M_S/L)$ | $\rho$ | $18 \times 10^{-3}\,\mathrm{Kg/m}$ |
| | String tension $(\rho(2f_1L)^2)$ | $T_e$ | $1.15 \times 10^{3}\,\mathrm{N}$ |
| | String wave speed $(\sqrt{T_e/\rho})$ | $c$ | $202.84\,\mathrm{m/s}$ |
| | Air damping coefficient | $b_1$ | $0.5\,\mathrm{s}^{-1}$ |
| | String internal friction coefficient | $b_2$ | $6.25 \times 10^{-9}\,\mathrm{s}$ |
| | String stiffness coefficient | $k$ | $7.5 \times 10^{-6}\,[-]$ |
| **Hammer** | Hammer mass | $M_H$ | $4.9 \times 10^{-3}\,\mathrm{Kg}$ |
| | Fluid damping coefficient | $b_H$ | $1 \times 10^{-4}\,\mathrm{s}^{-1}$ |
| | Hammer felt stiffness exponent | $p$ | $2.3\,[-]$ |
| | Hammer felt stiffness | $K$ | $4 \times 10^{8}\,[-]$ |
| | Width of the hammer spatial window $g$ | $w$ | $0.2\,[-]$ |
| | Relative striking position | $a_H$ | $0.12\,[-]$ |
| | Excitation point $(a_HL)$ | $x_0$ | $0.2304\,[-]$ |

To implement the finite difference scheme (FD), we discretized in time and space considering the following relations:

- Temporal discretization
  - Sampling frequency: $F_s = 48\,\mathrm{kHz}$
  - Time step: $T_s = 1/F_s = 2.08 \times 10^{-5}\,\mathrm{s}$
  - Total signal length: $duration = 8\,\mathrm{s}$
  - Number of samples: $N = duration/T_s = 384000$

- Spatial discretization
  The choice of the number of spatial steps is critical for numerical dispersion and stability problems. The following stability condition allows the determination of the maximum number of spatial steps $M_{\max}$ of discrete segments.

$$M_{\max} = \{[-1 + (1 + 16k\gamma^2)^{1/2}]/8k\}^{1/2} = 228.86$$

  where $k$ is the stiffness parameter and $\gamma = F_s/2f_1 = 366.97$.
  In addition to this, to limit the dispersion, we rounded the number of spatial steps $N$ up to the highest possible integer immediately lower than $M_{\max}$, by means of the Matlab function `floor`: $M = 228$.
  The spatial steps are defined as $X_s = L/M = 8.4 \times 10^{-3}\,\text{m}$.

After the definition of these parameters, in order to grant the computational stability of the scheme and to avoid the existence of growing solutions, we impose a boundary for the time step. This conditions is known as Courant-Friedrichs-Lewy condition and can be written as

$$\lambda \leq 1$$

where $\lambda = cT_s/X_s$ is the Courant number. By substituting the values found for $T_s$ and $X_s$, we can see that the stability condition is fulfilled for our scheme: $\lambda = 0.5018 < 1$.

Another point of the FD implementation, regards the spatial limitation of the interaction between the hammer and the string, not being the hammer's action punctual.
The force density term $f(x, x_0, t)$ represents the excitation of the string by the hammer, and can be written as:

$$f(x, x_0, t) = f_H(t)g(x, x_0)$$

where:
- $g(x, x_0)$ is a spatial window responsible for the spatial limitation of the exerted force, which can be modeled as an Hanning window of length $L_g$.
- $f_H(t)$ refers to the time history of the hammer force on the string $F_H(t)$ and is defined as:

$$f_H(t) = F_H(t)\left(\int_{x_0-\delta x}^{x_0+\delta x} g(x, x_0)dx\right)$$

In order to make the length discrete and to center the window with respect to the excitation point $x_0$, we started by computing $x_0$ as function of the relative striking position $a = x_0/L$, then discretized ($m_0$ is the closest integer to $x_0$) and used it for the definition of $L_g$.
Despite the function $g$ is different from 0 only along $L_g$, we defined it for the whole string length, so to compute the FD loop.
Once we obtained $L_g$, $g$ was defined exploiting the Matlab `hann` function.

**FD implementation**

The main variable of the simulation is the transverse string displacement $y$, which is discretized both in space and time and represented as a matrix of MxN elements. Before describing the FD loop, initial and boundary conditions must be taken into account.

***Boundary and initial conditions***

- Initial string displacement: $y = 0_{MxN}$
- Initial hammer velocity: $V_{H_0} = 2.5\,\mathrm{m/s}$
  As for the string, also the hammer's force (meaning the force exerted on the string) has been computed by means of $MxN$ matrix. The hammer initial displacement and the string initial velocity are discretized as linear arrays of N elements, referred to as $\eta(N)$ and soundwave$(N)$ respectively.
- String initial velocity: soundwave $= \underline{0}$
- Hammer initial displacement: $\eta(1) = V_{H_0} \cdot T_S$
- Hammer initial force: $F_{H_0} = 0_{MxN}$
- Number of spatial samples: $X_{av} = 12$

Since the transfer of the force only happens when the string and the hammer are in contact, we added a boolean variable `hammer_on`, to determine whether the hammer has left the string or not. The interaction process ends when the displacement of the hammer head becomes smaller than displacement of the string at the excitation point i.e. when $\eta(n) < y(m_0, n) \Rightarrow$ `hammer_on` = false.
At $t = 0$, `hammer_on` = true.
When the hammer leaves the string, the forcing term is set equal to zero.

***Loop computation***

We computed the hammer force $F_H(n)$ and the hammer displacement referred to following step as:

$$F_H(n) = K|\eta(n) - y(m_0, n)|^P$$

$$\eta(n+1) = d_1\eta^n + d_2\eta^{n-1} + d_F F_H(n)$$

where the coefficients $d_1$, $d_2$, $d_F$ are given by:

$$d_1 = \frac{2}{1 + b_H T_s/2M_H} \qquad d_2 = \frac{-1 + b_H T_s/2M_H}{1 + b_H T_s/2M_H} \qquad d_F = \frac{-T_s^2/M_H}{1 + b_H T_s/2M_H}$$

Then, we focused on the discrete-time expression of the analytical wave equation for the stiff and lossy string (equation 1), which can be formulated as follows:

$$y_m^{n+1} = a_1(y_{m+2}^n + y_{m-2}^n) + a_2(y_{m+1}^n + y_{m-1}^n) + a_3 y_m^n + a_4 y_m^{n-1} + a_5(y_{m+1}^{n-1} + y_{m-1}^{n-1}) + a_F F_m^n$$

Since the discrete-time expression for $y$ is only valid for the interior sections of the string ($m = 2, .., M-2$), we had to refine the model taking into account different expressions for $m = 1$, $m = 2$, $m = M-1$, $m = M$, which comply with the boundary conditions:

- $m = 1$:

$$y_m^{n+1} = b_1^l y_m^n + b_2^l y_{m+1}^n + b_3^l y_{m+2}^n + b_4^l y_m^{n-1} + b_F^l F_m^n$$

- $m = 2$:

$$\begin{aligned} y_m^{n+1} =& a_1(y_{m+2}^n - y_m^n + 2y_{m-1}^n) + a_2(y_{m+1}^n + y_{m-1}^n) + a_3 y_m^n + a_4 y_m^{n-1} + \\ &+ a_5(y_{m+1}^{n-1} + y_{m-1}^{n-1}) + a_F F_m^n \end{aligned}$$

- $m = M - 1$:

$$\begin{aligned} y_m^{n+1} =& a_1(2y_{m+1}^n - y_m^n + y_{m-2}^n) + a_2(y_{m+1}^n + y_{m-1}^n) + a_3 y_m^n + a_4 y_m^{n-1} + \\ &+ a_5(y_{m+1}^{n-1} + y_{m-1}^{n-1}) + a_F F_m^n \end{aligned}$$

- $m = M$ correspond to the bridge position (right end):

$$y_m^{n+1} = b_1^r y_m^n + b_2^r y_{m+1}^n + b_3^r y_{m+2}^n + b_4^r y_m^{n-1} + b_F^r F_m^n$$

All the coefficients $(a_1, a_2, .., a_5, a_F, b_1^r, b_2^r, .., b_F^r, b_1^l, b_2^l, .., b_F^l)$ are related to the starting PDE and the boundary conditions for both ends respectively. In order to compute them, we used the following parameters:

$$\mu = \frac{k^2}{c^2 X_s^2} \qquad v = \frac{2b_2 T_s}{X_s^2}$$

together with the normalized impedances of the left and right end (bridge position):

$$\nu_l = 1 \times 10^{20} \qquad \nu_b = 1 \times 10^3$$

The corresponding expressions are reported in the tables below.
As regards the forcing term $F_H(m, n)$, we updated it for each $m$ value exploiting its definition ( $F_H(m, n) = F_H(n) \cdot g(m)$). To ultimate the description of the string velocity at each time step $n$, we computed the average of the sound over $X_{av}$, centering in in correspondence of the excitation point.

| **PDE coefficients** |
|---|
| $a_1 = \frac{-\lambda^2\mu}{1+b_1T_S}$ |
| $a_2 = \frac{\lambda^2+4\lambda^2\mu+v}{1+b_1T_S}$ |
| $a_3 = \frac{2-2\lambda^2-6\lambda^2\mu-2v}{1+b_1T_S}$ |
| $a_4 = \frac{-1+b_1T_S+2v}{1+b_1T_S}$ |
| $a_5 = \frac{-v}{1+b_1T_S}$ |
| $a_F = \frac{T_S^2\rho}{1+b_1T_S}$ |

| **Hinged end boundary coefficients (left)** |
|---|
| $b_1^l = \frac{2-2\lambda^2\mu-2\lambda^2}{1+b_1T_S+\nu_l\lambda}$ |
| $b_2^l = \frac{4\lambda^2\mu+2\lambda^2}{1+b_1T_S+\nu_l\lambda}$ |
| $b_3^l = \frac{-2\lambda^2\mu}{1+b_1T_S+\nu_l\lambda}$ |
| $b_4^l = \frac{-1+b_1T_S+\nu_l\lambda}{1+b_1T_S+\nu_l\lambda}$ |
| $b_F^l = \frac{T_S^2/\rho}{1+b_1T_S+\nu_l\lambda}$ |

| **Bridge boundary coefficients (right)** |
|---|
| $b_1^r = \frac{2-2\lambda^2\mu-2\lambda^2}{1+b_1T_S+\nu_b\lambda}$ |
| $b_2^r = \frac{4\lambda^2\mu+2\lambda^2}{1+b_1T_S+\nu_b\lambda}$ |
| $b_3^r = \frac{-2\lambda^2\mu}{1+b_1T_S+\nu_b\lambda}$ |
| $b_4^r = \frac{-1+b_1T_S+\nu_b\lambda}{1+b_1T_S+\nu_b\lambda}$ |
| $b_F^r = \frac{T_S^2/\rho}{1+b_1T_S+\nu_b\lambda}$ |

## 2. Guitar model

### 2.1. Voltage generator model

The two main vibrational elements of a guitar are the top plate and the cavity. They represent a two-mass model, whose behavior can be analyzed by studying its electrical equivalent circuit (shown in fig.1). The equivalent voltage of the circuit is the force applied to the top plate divided by the effective top plate area, while the other elements of the circuit are defined as follows:

- $M_p = m_p/A_p^2$ is the inertance (mass/area) of the top plate ($kg/m^4$),
- $M_h = m_h/A_h^2$ is the inertance of air in the soundhole ($kg/m^4$),
- $C_p = A^2/K_p$ is the compliance of the top plate ($N/m^5$),
- $C_v = V/\rho c^2$ is the compliance of the enclosed air ($N/m^5$),
- $U_p$ is the volume velocity of the top plate ($m^3/s$),
- $U_h$ is the volume velocity of air in the soundhole ($m^3/s$),
- $R_p$ is the loss (mechanical and radiative) in the top plate,
- $R_h$ is the due to radiation from the soundhole,
- $R_v$ is the loss in the enclosure.

This coupled system presents two resonances, divided by an antiresonance, which corresponds to the Helmholtz resonance of the enclosure.

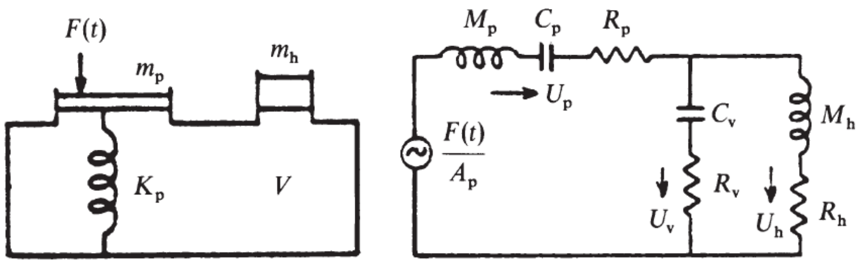


Figure 1: (Left) Two-mass model representing the motion of a guitar with a rigid back plate and ribs; (Right) Equivalent electric circuit

As mentioned before, this model is oversimplified, since the guitar top plate has more than one resonances, whose positions change depending on the forcing and measurement points and on how the guitar is supported.
In order to include 20 resonances in our analysis, we refined it, measuring both the force and the response in correspondence of the guitar's bridge (meaning we considered a driving point admittance).
The guitar sound can be computed starting from the bridge velocity, which is in turn related to the pressure variation (the current of the electric analog) by means of the Euler's equation

$$\rho_0 \frac{\partial v}{\partial t} = -\nabla p$$

As regards the circuit, we modeled the resonances by inserting a filter bank: each RLC branch corresponds to a particular filter with one resonance; the values of each element derive from empirical analysis aimed to obtain the resonances measured from a real guitar.

The input signal (fig.2) is given by a damped square wave, whose expression is:

$$V_{in} = sgn(sin(2\pi f_0 t))e^{-\beta t}$$

with $f_0 = 300\,\text{Hz}$ and $\beta = 3$.
The current sensor is placed where the current represents the bridge velocity; the resistors act as damping sources, attenuating the input signal.

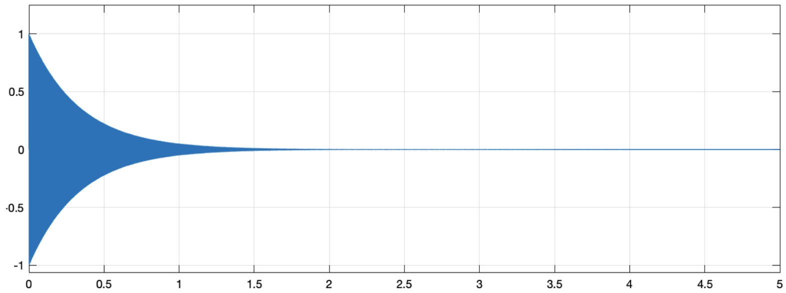


Figure 2: Input signal: damped square wave

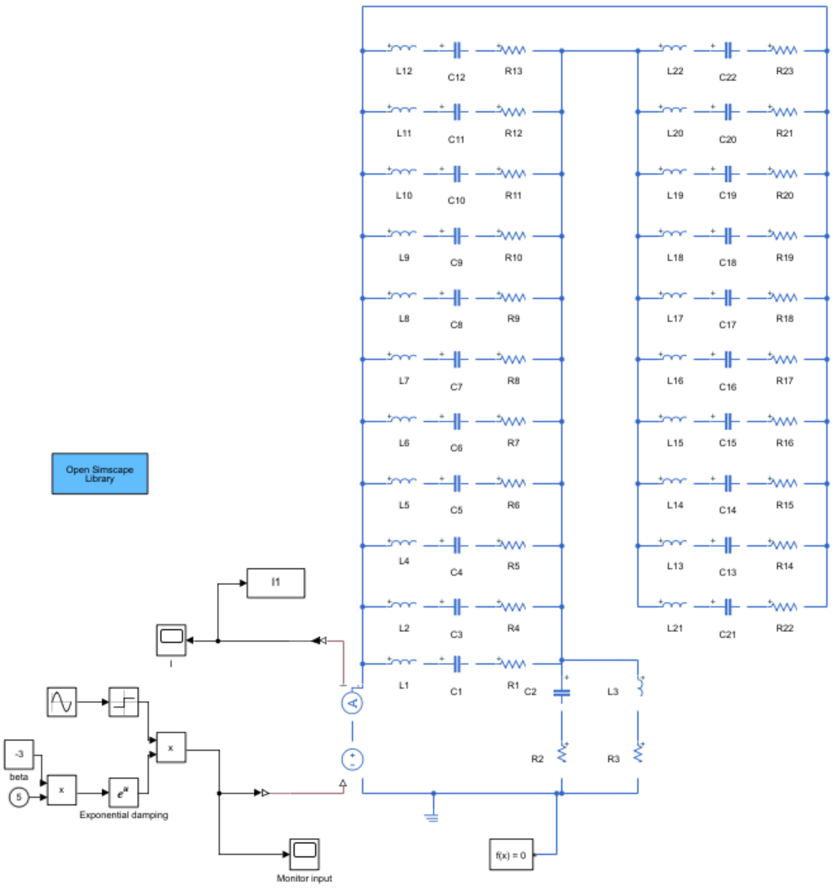


Figure 3: Extended equivalent electric circuit of a guitar

The output of this circuit ($I1$), sent from *Simulink* to *Matlab*, contains non constant time intervals between samples: in order to plot the correspondent signal, we had to resample it exploiting the `resample` Matlab function.
After resampling and normalizing the signal, we computed the corresponding FFT and plotted it in terms of magnitude and phase (fig. 4). To do so, we considered a total signal length of 5 s and a sampling frequency $F_s = 44\,100\,\mathrm{Hz}$.
Figure 4 shows the signal representation into the frequency domain, as magnitude and phase.

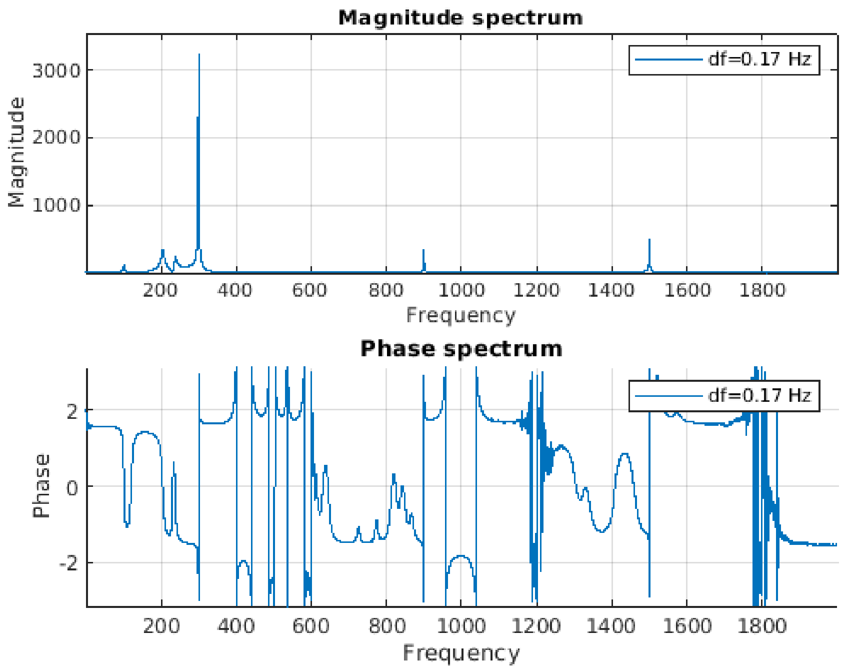


Figure 4: Magnitude and phase of the signal frequency content with voltage generator

### 2.2. Plucked string model

In order to further refine the previous model, we took into account the plucked string action. In particular, we considered the response of the guitar when the $E1$ string is plucked, since its the one which comes closest to the representation of an ideal string with no damping or stiffness.
To obtain the refined electric analog circuit (fig. 6), we substituted the voltage generator with a transmission line, which allows to take into account wave reflections at the string boundaries.
The input of the transmission line is obtained exploiting the D'Alembert solution to the wave equation: the input is given by two symmetrical triangular current pulses (each with a duration equal to half of the string fundamental frequency), whose shape simulates the solicitation coming from a plucked string on a guitar body. To implement this we exploited the `Signal Builder` Simulink block, shown in fig. 5.

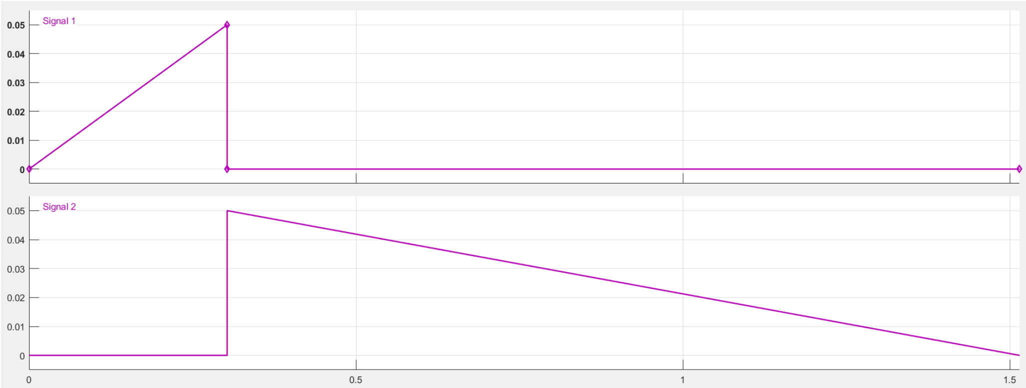


Figure 5: Input signal of the transmission line

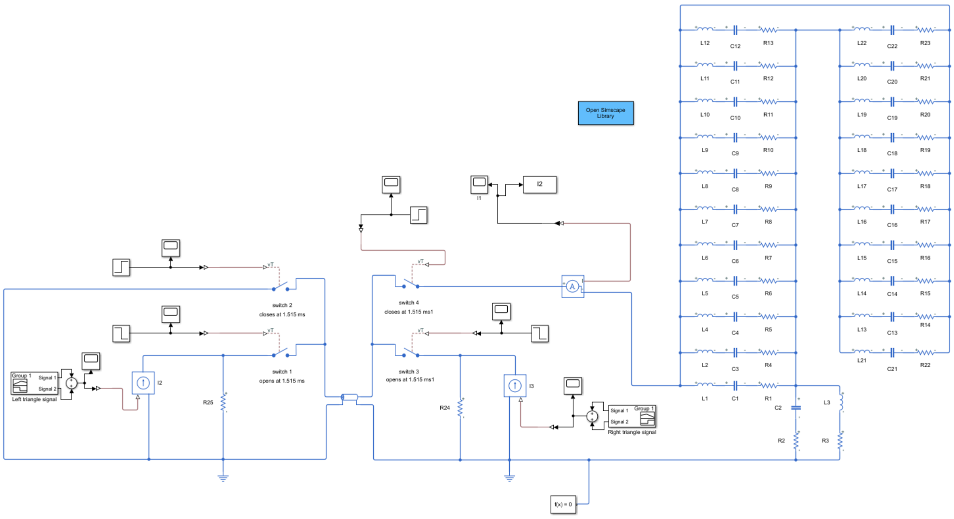

Figure 6: Equivalent electric circuit of the guitar when considering the plucked string

As can be seen from fig.6, the two ends of the transmission line are provided with two switches, each controlled by a `Step signal` block: they allow us to switch off the current sources once the input signal has elapsed, and to feed the rest of the circuit with the desired triangular waveform.
After the circuit implementation, we handled the corresponding output $I2$ as we did for the previous model, and obtained the results shown in fig.7
By comparing the two signal frequency contents obtained with the two different models, some comments can be made. When including the transmission line, we can clearly observe the presence of a set of overtones, almost absent in the voltage generator case. In particular, we can detect the first six harmonics, with only a slight inharmonicity ($\simeq 1\%$). As regard the peak that occurs at around 23 Hz, it's a consequence of the FFT computation and doesn't belong to the harmonic series. On the other hand, in the plot referred to the voltage generator input, the peaks identification is much more complex: the two main visible peaks correspond to the 3rd and 5th harmonics, but the inharmonicity is significantly higher ($\simeq 9\%$). All the other remaining resonances correspond to lower magnitude peaks, which can only be seen by drastically increasing the scaling of the y-axis.

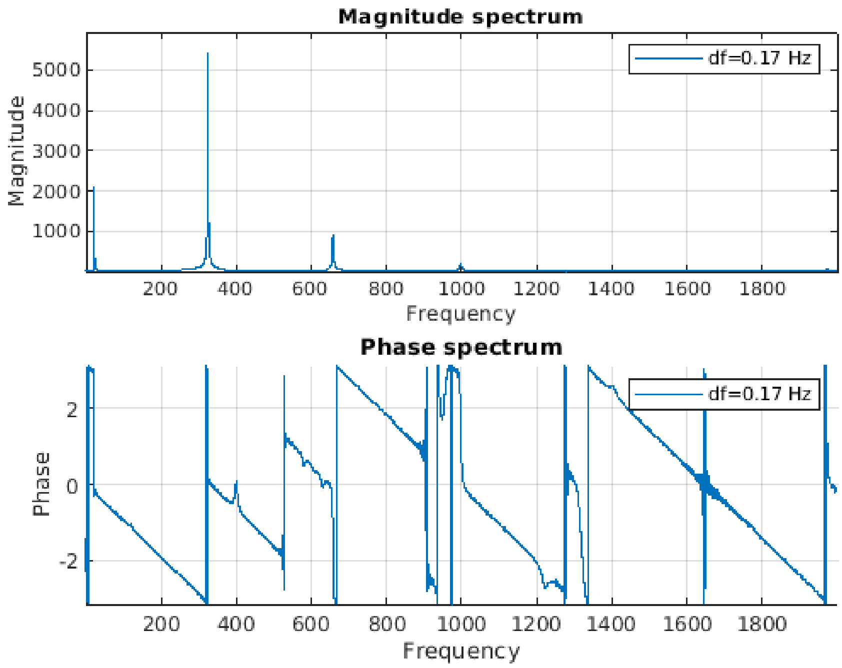


Figure 7: Magnitude and phase of the signal frequency content with transmission line

We also plotted the resampled and normalized signals obtained in 2.1 and 2.2, shown in fig.8 and 9. Once we obtained the normalized signals, we used the Matlab `sound` function to send them to the speaker at the desired sample rate.
We noticed that the sound obtained with a variable solver step and the transmission line block results glitchy, due to the presence of phase distortion. This can be adjusted by adopting a fixed solver step (equal to the sampling frequency), at the expense of the computational efficiency. We run the simulations for both these scenarios, and included all the samples in the samples folder.
The sound produced including the plucked string action is much more realistic with respect to the one deriving from the first circuit, confirming the validity of the plucked string extension.

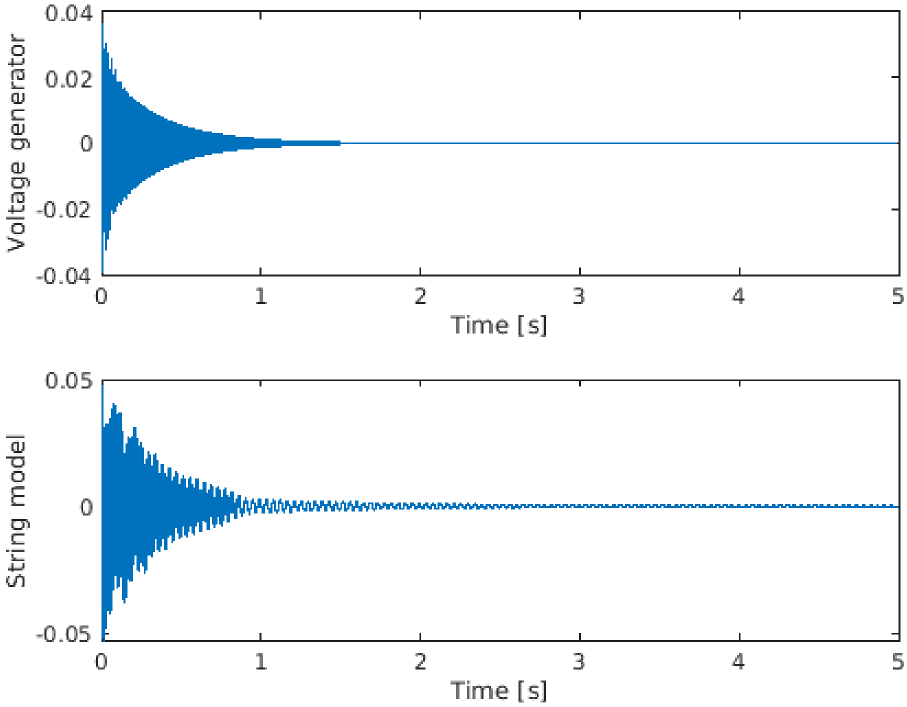


Figure 8: Resampled signals

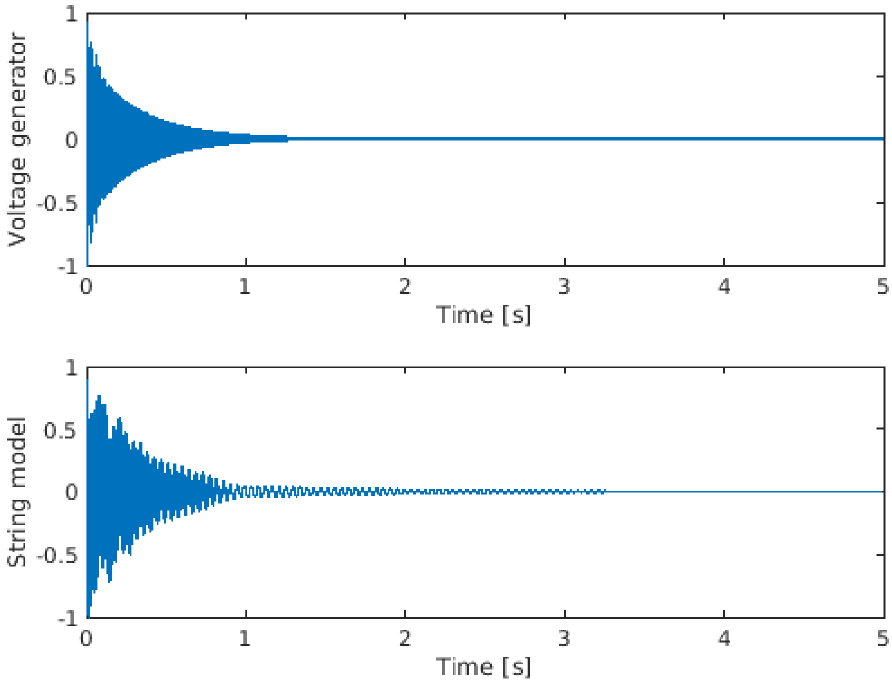


Figure 9: Normalized signals