Machine Learning-based Analysis of a Physics-Informed Data Set of a Viscoelastic Damped Membrane

Cristhiam Martínez¹, Rolf Bader²
1 Institute of Systematic Musicology, 20354 Hamburg, Deutschland, Email: cristhiam.martinez.orellanos@studium.uni-hamburg.de
2 Institute of Systematic Musicology 20354 Hamburg, Deutschland, Email: r_bader@t-online.de

Introduction

Internal damping plays an important role in the sound produced by musical instruments, yet its underlying mechanisms are still debated. In mechanical physics, internal dissipation of energy is traced back to thermodynamics and viscoelasticity [3], the latter crucial in membranes due to material properties [1]. The memory effect of viscoelasticity where stress depends on the history of strains and vice versa [4] determines the behavior of viscoelastic damping because of the energy supplied back to the system, leading to frequency-dependent behavior, non-exponential decay, spectral sidebands and mode coupling [1]. Analysis of viscoelastic damping in musical instruments using physical models are found in [2] [7] and [9] implementing a damping coefficient designed to simulate the exponential decay of a particular material. However, neither a data set accounting for systematic modifications of the parameters that define the behavior of viscoelastic damping is reported, nor, and consequently, no analysis of the influence of said modifications on the sound produced is performed.

To overcome this problem, this work presents a sound data set generated by a Finite-Difference Time Domain viscoelastic model of a membrane. Four parameters can be modified within the model: the viscoelastically damped frequency, the strength of the damping, the rate of decay of the memory effect, and the length of the viscoelastic buffer. To create each sound the parameter space is modified systematically, generating a physics-informed data set. Subsequently, the data set is investigated using a Self-Organizing Map (SOM). The non-linearities of the viscoelastic damping caused by different parameter settings, as well as the clustering performed by the SOM are discussed.

Methods

Finite-Differences Time Domain model of a viscoelastic damped membrane (FDTD)

Based on the differential equation of the membrane

$$\frac{T(x,y)}{\nu(x,y)} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = \frac{\partial^2 u}{\partial t^2} + D \frac{\partial u}{\partial t} \tag{1}$$

where $T(x,y)$, $\nu(x,y)$, and $D$ represent tension, area density, and damping, respectively, a complex and frequency-dependent Young’s Modulus $E(s)$ is implemented for the viscoelastic case

$$s = \alpha + i \omega, \tag{2}$$

which, according to Hook’s Law, expressed in the frequency domain as

$$\sigma(s) = E(s) \epsilon(s) \tag{3}$$

To analyze the development of viscoelastic damping in time, Eq. 3 is transformed into the time domain using the inverse Laplace transformation

$$h(\tau) = \frac{1}{2\pi i} \int_{s=-i\infty}^{s=i\infty} E(s) e^{s\tau} ds \tag{4}$$

where the integral is performed over the range of $s$ and $\gamma$ is a constant for all $E(s)$. Then, Eq. 3 is transferred to the time domain via time convolution

$$\sigma(t) = \int_0^\infty \epsilon(t-\tau)h(\tau) d\tau \tag{5}$$

where $E(s)$ is represented by $h(\tau)$, the viscoelastic function, whereby the memory effect of viscoelasticity is modeled by defining $\sigma(t)$ as the result of all past $\epsilon(t)$ weighted by $h(\tau)$. Furthermore, if in Eq. 5 $\text{Re}\{E(s)\} = 0$, the solution converges to an exponential decay, indicating no viscoelasticity

$$h(\tau) = E_0 \delta(\tau) \tag{6}$$

with $\delta(\tau)$ representing the Kronecker delta function. Based on this case, an imaginary material that presents internal damping only at $\omega_0$ can be modeled as

$$h(\tau) = E_0 \delta(\tau) + \text{Re}\{E(s)\} e^{\gamma \tau} \tag{7}$$

with damping amplitude $\text{Re}\{E(s)\}$. Correspondingly, a frequency-dependent damping spectrum can be expressed as

$$h(\tau) = \int_0^\infty \text{Re}\{E(s)\} e^{\gamma \tau} ds \tag{8}$$

Lastly, the complex Young’s Modulus is applied in Eq. 1. The equation is considered as a balanced stress-strain
relation in line with the Newtonian principle of force balance. Here, the stress constitutes the force applied to the structure to induce strain, which in turn represents the system’s potential energy due to variations in displacement \[1\]. In this regard, it is possible to substitute the stress for area density \( T(x,y) \) and strain for the spatial differentiation of the membrane, leading to:

\[
\int_{\tau=0}^{\infty} h(\tau) \frac{T(x,y)}{\nu} \left( \frac{\partial^2 u(x,y,t-\tau)}{\partial x^2} + \frac{\partial^2 u(x,y,t-\tau)}{\partial y^2} \right) d\tau = \frac{\partial^2 u(x,y,t)}{\partial t^2}.
\]

(9)

To discretize the model for computation, Eq. 3 is carried out at discrete time intervals and limited to the integration time \( T \) according to

\[
\sigma_t = \sum_{\tau=0}^{N-1} \epsilon_{t-\tau} h(\tau),
\]

(10)

with \( N \) samples, \( t \) time points and strain \( \sigma_t \). The viscoelastic function (Eq. 4) transforms into a sum

\[
h_\tau = \frac{1}{2\pi i} \sum_{k=1}^{N} E_k e^{\gamma \tau/r} e^{i \omega_k \tau/r}, \text{ with } \tau = 0, 1, 2, 3, \ldots N - 1
\]

(11)

wherein only integer multiples of the periodicity \( T = \frac{N}{r} \) with sampling rate \( r \) can be utilized, and \( E_k \) represents the discrete complex values for frequencies, such that

\[
\omega_k = 2\pi k/r, \text{ with } k = 1, 2, 3, \ldots N
\]

(12)

As the computation remains in the time domain, \( h(\tau) \) must be real. Hence, Eq. 11 is reformulated as:

\[
h_\tau = E_0 \delta(\tau) + \frac{1}{2\pi} \sum_{k=1}^{N} \text{Re} \{ E_k \} e^{\gamma \tau/r} \sin (\omega_k \tau/r + \phi_k)
\]

with \( \tau = 0, 1, 2, 3, \ldots N - 1 \)

(13)

The model has four parameters: \( \omega_k \), the Viscoelastically Damped Frequency (VDF); \( \text{Re} \{ E(s) \} \), the viscoelastic material property; \( \gamma \), the frequency range of the damping; and the integration time \( T \), the length of the viscoelastic function \( h(\tau) \).

Self-Organizing Maps (SOM)

A SOM is a feature map that converts non-linear statistical relations between high-dimensional data into geometrical relations displayed on a low-dimensional array, normally a two-dimensional grid \([5]\) of \( D \times D \) dimensions. To train it, a data set \( Q = [x_1, x_2, \ldots, x_n] \) with \( m \) number of samples is used, where each data sample \( x_i \), represented as a feature vector \( x_i = [x_{i1}, x_{i2}, \ldots, x_{in}] \) with \( n \) number of attributes, is presented to each neuron \( w_{ij} = [w_{ij1}, w_{ij2}, \ldots, w_{ijn}] \) of the SOM, each one also represented as a vector of length \( n \) to match the size of \( x_i \). Both the neurons and the feature vector are normalized. During training at iteration time \( t \), the distance between each sample from \( Q \) and each neuron is calculated. In this study, the Euclidean distance \( d = ||x - w_{ij}|| \) is used. The neuron \( w_{ij} \) with \( d(\text{min}) \) becomes the Best Matching Unit (BMU) for \( x_i \). This process is carried out for all data samples in \( Q \). Subsequently, according to a neighbor function \( N \), the values of the BMU and its neighbor neurons are updated according to \( d \), so that in the next iteration \( t + 1 \), the process is performed based on the weights from \( w_{ij(t)} \). This process is repeated for a \( P \) number of iterations until the map converges.

Once converged, if the algorithm finds underlying patterns within the data, it is expected to create clusters according to such patterns. To visualize said clusters, the map is calibrated before training, so that certain characteristics of the plot correspond to values provided as metadata. In this study, the colors and the sizes of each data point correspond to \( \text{Re} \{ E(s) \} \); the darker the color, the higher the values; and to \( \gamma \): the bigger the point, the narrower the frequency range of the damping, respectively. Furthermore, a U-matrix is derived by calculating the distances between neurons, where 1 stands for the maximum distance and 0 for complete similarity. The calculation is performed w.r.t. \( d \) and \( N \). In the present paper, the background color of each map represents the U-Matrix, where total similarity is symbolized by black neurons and dissimilarity by white ones. Additionally, for this study the initial weights of \( w_{ij} \) are obtained by performing a Principal Component Analysis (PCA) over \( Q \), and interpolating the values of the grid according to the first two principal components along the \( x \) and \( y \) axes, respectively. A gaussian neighborhood function with a width \( = 1.5 \) is used, number of iterations \( P = 1000 \) and the learning rate is set to 0.5.

Data pipeline

\[\text{FDTD SIMULATION} \rightarrow \text{STFT} \rightarrow \text{PEAK PICKING} \rightarrow \text{ANALYSIS FREQUENCY RANGE} \rightarrow \text{Z-SCORE NORMALIZATION} \rightarrow \text{SOM}\]

**Figure 1:** Data pipeline. In the fourth step, an analysis Frequency Range (FR) is defined around the Viscoelastically Damped Frequency (VDF). The frequencies and amplitudes from the peaks within said FR are placed into the feature vector, whereby the frequencies, together with \( \text{Re} \{ E(s) \} \), \( \gamma \) and \( T \) values from each data sample are used as metadata, and the normalized amplitudes as the actual data fed to the SOM.

For the FDTD, a membrane was simulated with tension
= 2284 [Pa], volume density = 300kg/m³, thickness = 3mm, radius = 10cm and boundary conditions \( U_{rim} = 0 \). Three vectors of length 104 × 104 = 10816 vector entries for displacement, velocity and displacement storage, and an acceleration memory of 104 × 104 × 1000 = 10816000 entries were utilized. The CUDA language on a Graphics Processing Unit (GPU) in a desktop with a NVIDIA GeForce RTX 3060, 16GB RAM, windows 10 and an intell-17-12700 2.1GHz CPU. The parameter space set for the simulation is presented in Table 1. The processing time for one sound 500ms long with a sample rate of 96kHz was about 10 seconds. The data pipeline, shown in Figure 1, is implemented in Python. For each sound of the dataset a Short-Time Fourier Transform (STFT) [6] with a window size of 4096 samples and a hop length of 96 samples is executed, followed by a peak picking [8] over the entire audible spectrum, set to the minimum distance and width. An analysis Frequency Range (FR) of 1KHz around the VDF is then selected, e.g. for the VDF = 700Hz, FR = 200Hz - 1.2KHz, and the frequencies and amplitudes from the peaks within it are extracted and placed in the feature vector. The frequencies, together with \( Re\{E(s)\} \), \( \gamma \) and \( T \) values are used for data annotation, and the actual amplitude values from the peaks, once normalized, as data to train the SOM [10].

Table 1: Parameter space of the simulation.

<table>
<thead>
<tr>
<th>VDF (Hz)</th>
<th>T (ms)</th>
<th>( Re{E(s)} )</th>
<th>( \gamma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>4</td>
<td>0</td>
<td>1/100</td>
</tr>
<tr>
<td>6400</td>
<td>7</td>
<td>0.003</td>
<td>1/10</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.0003</td>
<td>1/10</td>
</tr>
<tr>
<td>N. of steps</td>
<td>10</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

Results and discussion

Physics-informed sound data set

A physics-informed sound data set with 660 sounds was generated w.r.t. the set parameter space (see Table 1). As expected, the energy supply provided by the memory effect causes a complex non-linear behavior. Figure 3 presents the example for the VDF = 700Hz, where an increase in the material property of the damping \( Re\{E(s)\} \) intensifies the complex amplitude response of the extracted peaks, reaching their maximum value at mid values of \( \gamma \) and not at \( \gamma (max) \).

SOM clustering

Figure 2 (top) shows the SOM for the VDF = 700Hz at \( T = 10\)ms. The U-Matrix generates four regions placed on each corner of the map. Although no enclosed clusters are generated, the data distribution shows a transition corresponding to an increase in the \( Re\{E(s)\} \) values, starting at the bottom right region, where also the reference cases are placed clearly separated from the rest of the data by the U-Matrix, and following an inverse “n” shape across the SOM. Note that the data samples crossing the boundaries created by the U-matrix have in all cases mid to high \( \gamma \) values. SOM for the same VDF with shorter \( T \) display a similar data distribution, yet with more data samples spread across the map regardless of their \( Re\{E(s)\} \) values. On the other hand, for the VDF = 6.4KHz at \( T = 10\)ms (see Figure 2 (bottom)) the U-Matrix creates three regions: two small ones placed at the bottom and top of the right side, and a larger one on the left side, with different levels of data density. Compared to the previous case, the SOM generates cohesive clusters that clearly correlate with \( Re\{E(s)\} \), and whose internal structure relates to \( \gamma \). Still, some data points, specially for \( Re\{E(s)\} = 0.0024 \), are still scattered. The overall clustering does not follow a particular pattern. Lastly, as in the previous case, clustering at shorter \( T \) becomes blurry and the internal structures weaker.

The performance of the map varies depending on the frequency range and the integration time analyzed. The resulting U-Matrix is strongly influenced by the frequency range: while at VDF = 700Hz clear regions are formed that correlate with different \( Re\{E(s)\} \) values, at VDF = 6.4KHz regions do not correlate with an increase or de-
Figure 3: Non-linearities caused by viscoelastic damping. Peaks within the frequency range 500Hz - 900Hz for the Viscoelastic Damped Frequency (VDF) 700Hz for three different $Re\{E(s)\}$ values are presented. In case, six different $\gamma$ values are plotted. When $Re\{E(s)\}$ is low, viscoelastic damping behaves mostly in a linear manner, decreasing the amplitude of the peaks according to the increments in $\gamma$. When $Re\{E(s)\}$ is increased, middle values of $\gamma$ produce major increases in amplitude of the peaks than higher values (see plots b) and c) freqs. 797Hz and 820Hz). Furthermore, high values of $Re\{E(s)\}$ suppress peaks present in the other cases (see freq. 727Hz in cases a) and b) ) and generate new peaks, again with mid values of $\gamma$ producing the highest amplitudes (see freq. 656Hz in case c) ).

increase in $Re\{E(s)\}$. Regarding $Re\{E(s)\}$, the formation of clusters clearly correlates strongly with this variable, which is to be expected because $Re\{E(s)\}$ is a material property and is neither transformed nor influenced by the viscoelastic damping and therefore does not depend on any other variable. In relation to $\gamma$, the formation of internal structures suggest an underlying pattern despite the strong non-linearities related to the $Q$-factor caused by the filter-like behavior of the damping [1], still further studies are required to understand how it is calculated. Lastly, in the cases in which clusters are created, both the cohesiveness of the clustering, *i.e.* how well they are separated from each other, as well as the development of the internal structure of each cluster, the latter regardless of the frequency range, show a dependency on the length of the integration time. However, a more detailed evaluation of this parameter, as well as the influence of each parameter along the data pipeline, remain for future work.

Conclusions

This study addressed the problem concerning the lack of a systematic generated data set for the study of viscoelastic damping in musical instruments, on the one hand, and the need for an analysis method appropriate for such highly complex data, on the other. By using a Finite-Differences Time Domain model and Self-Organizing Maps, we were able to shed light into the behavior of viscoelastic damping. Further studies are needed to analyze the influence of the parameters governing these acoustic phenomena upon the perception of the resulting sound.

References


