Spectral Analysis for Modal Parameters Linear Estimate

Marco Tiraboschi
University of Milan
Department of Computer Science
LIM - Music Informatics Laboratory
marco.tiraboschi@studenti.unimi.it

Federico Avanzini
and
Stavros Ntalampiras
University of Milan
Department of Computer Science
LIM - Music Informatics Laboratory
[name].[surname]@unimi.it

ABSTRACT

Modal synthesis is used to generate the sounds associated with the vibration of rigid bodies, according to the characteristics of the force applied onto the object. Towards obtaining sounds of high quality, a great quantity of modes is necessary, the development of which is a long and tedious task for sound designers as they have to manually write the modal parameters. This paper presents a new approach for practical modal parameter estimation based on the spectral analysis of a single audio example. The method is based on modelling the spectrum of the sound with a time-varying sinusoidal model and fitting the modal parameters with linear and semi-linear techniques. We also detail the physical and mathematical principles that motivate the algorithm design choices. A Python implementation of the proposed approach has been developed and tested on a dataset of impact sounds considering objects of different shapes and materials. We assess the performance of the algorithm by evaluating the quality of the resynthesised sounds. Resynthesis is carried out via the Sound Design Toolkit (SDT) modal engine and compared to the sounds resynthesised from parameters extracted by SDT's own estimator. The proposed method was thoroughly evaluated both objectively using perceptually relevant features and subjectively following the MUSHRA protocol.

1. INTRODUCTION

Modal synthesis [1] is a physical modelling approach to sound synthesis based on approximating the vibrations of a complex object by decomposing them into a set of independent modes, i.e. oscillations at a single frequency. The physical motivation of this approach is that we can express the the behaviour of a mechanically vibrating object describing it as a lumped mass-spring network [2]. This, in turn, can be modelled by a system of second-order partial differential equations (PDEs). Such a system can be diagonalized and the solution of every factorized PDE is a damped oscillator (function of time) with initial amplitude given by the modal shape (function of space). So, oscillations at any one point can be computed as the linear combination of the damped oscillators, each one weighted by its modal shape at that point.

Despite their potential advantages in terms of interactivity and parametrizability, physically based and procedural approaches have so far gained limited popularity in application domains related to sound design. Böttcher [3] discusses possible reasons for this. One is the need for tools that facilitate the work of the sound designers, without requiring them to deal with low-level technicalities of the sound models.

In the context of modal synthesis, a much needed facilitating tool for the sound designer is one that allows for automatic tuning of the mode parameters in order to reliably resynthesize a target (e.g., recorded) sound. In the constrained case of monophonic audio, every mode is fully characterised by a triplet of scalars: the modal frequency, the decay coefficient and the initial amplitude. This can be generalised to multichannel audio by considering an array of initial magnitudes (each associated to a "pickup point" for the sound), or even to space-continuous processes describing modal shapes as functions of spatial coordinates on the modal object. These parameters have a direct effect on the spectral content of the synthesised sound. Every mode describes how the power of the corresponding modal frequency evolves over time.

This paper presents an approach to the automatic estimation of modal parameters based on a target sound. The proposed approach is abbreviated as SAMPLE (*Spectral Analysis for Modal Parameters Linear Estimate*) as it employs a spectral modelling algorithm to track the variations of energy with respect to every sinusoidal component (partial) and then to perform linear regression to estimate the modal parameters corresponding to the inferred energy function.

SAMPLE is evaluated in combination with the Sound Design Toolkit (SDT), a a software package developed over several years [4, 5], which provides a set of sound models for the interactive generation of several acoustic phenomena, including interactions between solid objects by means of modal synthesis and physically-based interaction force models (impacts and frictions) [6]. Objects used to record an impact sounds evaluation dataset are shown in Fig. 1.

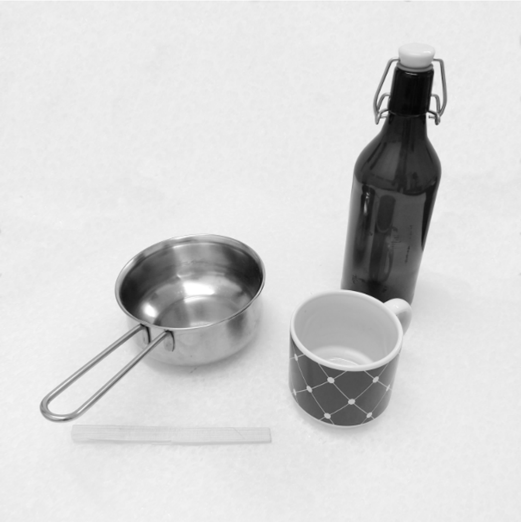

Figure 1. The four object used for recording the dataset: a glass bottle, a metal saucepan, a porcelain mug and a small piece of wood. Recordings by Giulia Clerici, LIM (Music Informatics Laboratory), University of Milan.

2. RELATED WORK

Starting with the studies by van den Doel and co-workers [7], modal synthesis has become a popular technique in interactive computer graphics applications, where the aim is to generate a sound that matches the geometry and the material of virtual objects as well as the user's interaction with them. In this context, the geometry of the object is usually known and it provides useful information about the modes of the object.

If both the geometry and the material distribution of the object are known, finite element methods (FEM) can be employed to estimate the modal parameters. Picard *et al.* [8] propose to use finite element methods to compute the masses and stiffnesses of the mass-spring network matrix and to find the modal parameters by eigen-decomposition of that matrix. This approach can compute the complete modal shape for the modes and allows for the synthesis of the sound at any pickup point on the object. However the geometry and the material distribution of the object are assumed to be available *a-priori*.

A conceptually similar approach is taken by Michon *et al.* [9] who use FEM analysis starting from a volumetric mesh of a 3D object, and generate the corresponding modal physical model in the framework of the FAUST programming language for real-time audio processing.

Acoustic information can also be used, as an alternative to, or in combination with object geometry. Ren *et al.* [10] propose a hybrid approach, combining FEM and audio analysis. An initial estimate of the modal parameters is made from the geometrical model, assuming that the material is isotropic and homogeneous along with several starting values with respect to mass density, Young's modulus and Poisson's ratio. Then, convex optimization is used to fit the material parameters to the detected modes in the audio example, which are extracted from an array of varying resolution spectrograms. It also employs deterministic residual compensation for modelling the non-sinusoidal component. This approach has the advantage of working in the absence of any knowledge of the material parameters. Furthermore, it can translate the identified material's parameters to virtual objects with different geometries.

Sterling *et al.* [11] build their approach on top of the feature extraction algorithm proposed by Ren *et al.*, and add a probabilistic model for the damping parameters in order to reduce the effect of external factors and non-linearities on the estimate of damping. External factors that may affect the estimate are the object support (that adds damping on top of the object's natural damping) background noise, feature extraction errors (e.g. spectrogram resolution and windowing sidelobes) and the relative emission and pickup patterns of the object and the microphone. The modal parameters are estimated through maximum-likelihood over an average of nearly 50 impact sounds samples per object, using and exponentially-modified gaussian as the probability density function of the likelihood. This approach requires neither the material parameters nor the geometrical model of the object, and improves robustness to some external factors. However, it is still affected by reverberation and does not estimate the modal shape function. At the same time, it requires multiple audio examples to estimate the parameters.

Kereliuk *et al.* [12] proposed a modification of the ESPRIT algorithm [13] to determine the modal parameters of room impulse responses. Their approach supported a variable number of modes, emphasising on high quality for lower mode counts.

Some approaches address modal parameter estimation via parametric estimate using autoregressive (AR) system fitting. Abel *et al.* [14] used resonant filter parameters fitting to model the modal response of rooms and spring reverberators. Maestre *et al.* [15] used a similar technique to model the response of the body of stringed instruments, such as guitars and violins.

Deep learning approaches are not well suited to address the present problem, for two main reasons: the amount of available labeled data is usually too small to train a deep neural network (DNN) and the number of output features is not known a priori. Owens *et al.* [16] presented a recurrent-convolutional neural network (R-CNN) that directly estimates the sound of a struck or scratched object from a silent video recording of the interaction. The downside of this approach is that, whatever physical properties that DNN has learned to estimate, they are not encoded in a physically meaningful format nor can they be converted into one. Also, in the context of interactive sound, availability of a video registration of the physical interaction corresponding to the sound is a strong assumption.

3. METHODS

The proposed estimation algorithm is based on one single audio example, without any prior knowledge about the geometry or the material properties of the object. The modal parameters are found through linear and non-linear regression on the spectral features obtained by state-of-the-art sinusoidal analysis. The method is summarized in Algorithm 1. An implementation has been developed in Python (Python 3.7.4, NumPy 1.17.2, SciPy 1.3.1). During the development of the approach, we have used MTG's own implementation of SMS, which is available on their GitHub repository [1] .

3.1 Sinusoidal Analysis

Serra [17–19] introduced Spectral Modelling Synthesis (SMS), an analysis and synthesis system for musical sounds based on the decomposition of the sound into a deterministic sinusoidal and a stochastic component. The main components of the sinusoidal analysis are the peak detection and the peak continuation algorithms. The peak detection algorithm detects peaks in each STFT frame of the analysed sound as a local maximum in the magnitude spectrum: zero-phase windowing is employed, so, local flatness of the phase spectrum can be used to detect the peaks. The peak continuation algorithm organizes the peaks into temporal trajectories, with every trajectory representing the time-varying behaviour of a partial. Subsequently, for every peak in a trajectory, the instantaneous frequency, magnitude and phase are stored to allow further manipulation and resynthesis. The residual power-spectral density is estimated with a line-segment approximation of the difference between the original spectrum and the sinusoidal component.

For the problem of modal analysis, our method relies only on the deterministic sinusoidal component of SMS. The general-purpose analysis of SMS enables recycling of the peak trajectories: if one trajectory becomes inactive, it can be later picked up when a newly detected partial arises. Thus, after SMS analysis, one first post-processing step amounts to splitting mixed trajectories into trajectories that represent only one partial. Subsequently, two trajectories that do not overlap in time but have approximately the same average frequency (any thresholding function could be used, in the default implementation of the algorithm it is set to 1 mel) can be considered as belonging to the same partial and merged into the same trajectory.

Some simplifying assumptions and ad-hoc refinements can be made for the sounds considered in this work, i.e. sounds of resonating objects produced by mechanical impacts. First, all partials start at the same onset, so any identified trajectory that starts significantly after the onset can be discarded. Second, we found that feeding the audio signal into the algorithm with the time axis reversed (i.e. starting from the end of the sound) avoids the detection of spurious spectral peaks in the attack transient of the sound, while allowing it to pick up the longest-lasting partials more robustly. This is a common trick used in audio analysis for additive synthesis [20], that is also used in MPEG Audio [21]. It doesn't impose any further limitation to the approach, because processing is always assumed to be batch.

```
Data: the array of audio samples $x$,
      the sampling frequency $f_s$
Result: list of frequencies $f$,
        list of decays $d$,
        list of magnitudes $m$
// Sinusoidal analysis
$x \leftarrow \text{flip}(x)$
$\tau \leftarrow \text{sinusoidal\_analisis}(x, f_s)$
for $i := 0$ to $\#\tau - 1$ do
    $\tau[i] \leftarrow \text{flip}(\tau[i])$
end

// Split trajectories
$\tau \leftarrow \bigcup_{i=0}^{\#\tau-1} \text{split\_trajectory}(\tau[i])$

// Merge trajectories
$i := 0$
$\text{offset} := +\infty$
while $i < \#\tau$ do
    $j := i + 1$
    while $j < \#\tau$ do
        if $\text{mergeable}(\tau[i], \tau[j])$ then
            $\tau[i] \leftarrow \text{merge\_trajectories}(\tau[i],\ \tau[j])$
            delete $\tau[j]$
            $j \leftarrow i + 1$
        else
            $j \leftarrow j + 1$
        end
    end
    $\text{offset} \leftarrow \min(\text{offset},\ \tau[i][0].\text{time})$
    $i \leftarrow i + 1$
end

// Discard trajectories starting
   late
$i := 0$
while $i < \#\tau$ do
    if $\tau[i][0].\text{time} \leq \text{offset} + \text{off\_thresh}$ then
        $i \leftarrow i + 1$
    else
        delete $\tau[i]$
    end
end

// Regression
initialize empty lists: $f, d, m$
for $i := 0$ to $\#\tau - 1$ do
    $(k, q) \leftarrow \text{regression}(\tau[i].\text{time}, \tau[i].\text{db})$
    $f.\text{append}(\text{mean}(\tau[i].\text{freq}))$
    $d.\text{append}\,(-40 \log_{10}(e)/k)$
    $m.\text{append}\,(\exp_{10}(q/20))$
end

// Filter
$i := 0$
while $i < \#f$ do
    if $\text{is\_plausible}(f[i], d[i], m[i])$ then
        $i \leftarrow i + 1$
    else
        delete $f[i], d[i], m[i]$
    end
end
```

Algorithm 1: Summary of the SAMPLE algorithm.

[1] https://github.com/MTG/sms-tools

3.2 Parameter Regression

Partials of a modal impact sound are characterized by exponentially decaying amplitudes. In its general form, the oscillation related to the i^{th} modal partial is defined as

$$p_i(x,t) = s_i(x)e^{-\frac{t}{d_i}} \cos\left(2\pi \int_0^t f_i(t)dt + \phi_0\right), \quad (1)$$

where f_i is the instantaneous frequency of the mode, d_i is the associated damping, and s_i is the modal shape at the spatial point x. In the case of static modes (constant modal frequency) and for one pickup point, Eq. (1) can be simplified as follows:

$$p_i(t) = m_i e^{-\frac{t}{d_i}} \cos\left(2\pi f_i t + \phi_0\right). \quad (2)$$

SMS trajectories carry information about instantaneous frequency, magnitude and phase; however, phase information will not be considered. Hence, the trajectory τ_i relative to the i^{th} partial will be regarded as a function that associates to every frame index the instantaneous amplitude and frequency of the partial:

$$\tau_i : N(i) \subset \mathbb{N} \to \mathbb{R}^2, n \mapsto \left(m_i e^{-\frac{t(n)}{d_i}}, f_i(t(n))\right), \quad (3)$$

$$\tau_i(n) =: \left(A(\tau_i(n)), f(\tau_i(n))\right), \quad (4)$$

where $t(n)$ is the time instant associated to frame n and $N(i)$ is the set of indices of the frames where there is a peak belonging to τ_i.

The modal frequency is then estimated as the average frequency of the trajectory:

$$\hat{f}_i := \frac{1}{\#N(i)} \sum_{n \in N(i)} f(\tau_i(n)). \quad (5)$$

The instantaneous amplitude of the partial can be derived from Eq. (2), excluding the oscillatory component, as defined previously in Eqs. (3) and (4):

$$A_i(t) = m_i e^{-\frac{t}{d_i}}. \quad (6)$$

Taking the logarithm of the instantaneous amplitude, the function becomes linear.

$$\ln A_i(t) = -\frac{t}{d_i} + \ln m_i, \quad (7)$$

Thus, $\ln m_i$ and d_i^{-1} can be estimated as the coefficients of a linear model via ordinary linear regression (least-squares estimate, LSE) [22]. In the implementation of the method, amplitude is expressed in decibel.

$$A_i^{(dB)}(t) = k_i t + q_i \Rightarrow A_i(t) = 10^{\frac{k_i t + q_i}{20}}. \quad (8)$$

The estimates of the linear coefficients are

$$\hat{k}_i, \hat{q}_i = \underset{k_i, q_i}{\arg\min} \sum_{n \in N(i)} \left| A^{(dB)}(\tau_i(n)) - k_i t(n) - q_i \right|^2, \quad (9)$$

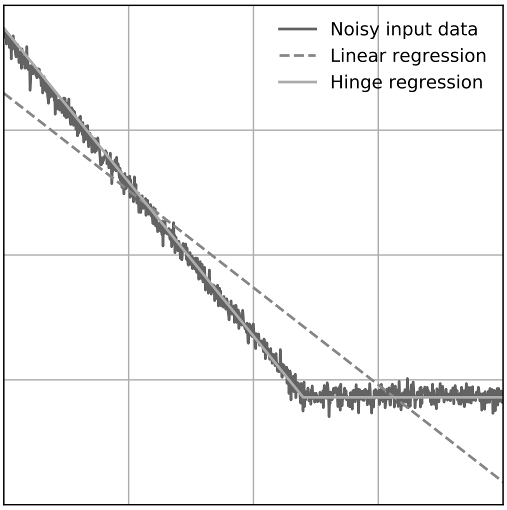

Figure 2. Comparison of fitting a linear function and a hinge function to noisy amplitude trajectory data when the noise floor stops the linear decay. Axes are time (abscissa) and amplitude (ordinate, logarithmic). The hinge regression allows for a better estimate of the linear parameters, while the linear regression is biased towards a lower value for the intercept q and a less steep slope k.

where $A^{(dB)}$ denotes the trajectory point amplitude expressed in decibel. These are the conversion formulae.

$$m_i = 10^{\frac{q_i}{20}}, \quad (10)$$

$$d_i = -\frac{20 \log_{10} e}{k_i}. \quad (11)$$

When the noise floor is high, it could be detected by SMS and its magnitude would contribute to the magnitude of the trajectory. In order to obtain a finer estimate of the linear parameters, the amplitude of the trajectory can be fitted to a hinge function using non-linear least squares estimate (as shown in Fig. 2). A hinge function $h(t)$ is a function that is linear for $t < \alpha$ and then continues as a constant.

$$h_{k,q,\alpha}(t) = k \cdot \min(t, \alpha) + q. \quad (12)$$

Non-linear LSE is not guaranteed to converge to a global optimum. Matti *et al.* [23] included the noise in the model and defined a customized optimization method. In this approach most of the noise is excluded by the sinusoidal analysis and the optimizer is initialized with the linear parameters values found via linear LSE, with α equal to half of the duration of the signal. Parameter α is discarded after the estimate.

3.3 Feature elimination

Triplets of parameters not corresponding to plausible modal features are discarded. Partials whose frequency is not in the audible band (default 20 Hz ∼ 20 kHz) are filtered out because they are not of interest for audio resynthesis.

The time-constant d is not conventionally used to describe the decay time of a sound. A more common descriptor is

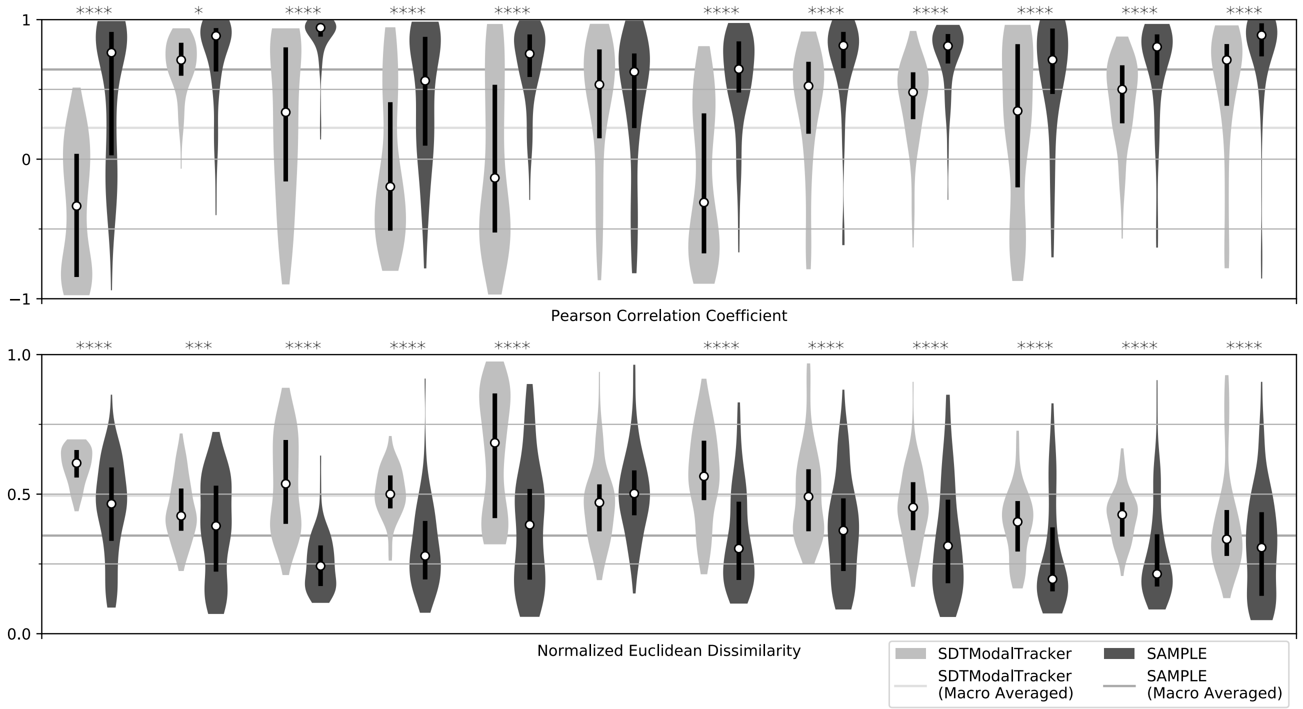

Figure 3. Violin plots for the Pearson correlation coefficient and the normalized Euclidean dissimilarity between the original sounds MFCCs and the resynthesised sounds. The twelve pairs of violin plots correspond to the MFCC index (one to twelve). The body of the violin plot is the histogram, the black rectangle is the interquartile range and the white dot is the median value. Stars above the groups are indicative of the *p*-value of the paired-difference test as explained in equation (17). Macro averages are drawn as horizontal lines of the same colour of the corresponding class.

t_{60}, the time to decay by 60 dB [24].

$$\frac{A(t_{60})}{A(0)} = 10^{-\frac{60}{20}} \Rightarrow t_{60} = d \cdot 3 \ln 10 = -\frac{60}{k}. \qquad (13)$$

A threshold value on the t_{60} can be set to filter out rapidly decaying partials. Setting the threshold to 0 has the effect of filtering out only misbehaving partials. If $k > 0$ it, then the partial is increasing in magnitude instead of decreasing.

A threshold can be also set on the relative magnitude. Considering the partial with the largest magnitude m^* as a reference, the relative magnitude (in decibel) is

$$m_i^{(r)} := 20 \log_{10} \frac{m_i}{m^*}. \qquad (14)$$

In this way, very soft partials can be discarded.

3.4 Resynthesis

To resynthesise modal sounds, the PureData implementation of SDT [5] was employed.[2] A PureData patch has been assembled to synthesise sounds using previously extracted modal parameters. Also, a Python interface was developed with Cython [25] to wrap the relevant C API of SDT, which makes it possible to write complete analysis/synthesis pipelines entirely in Python.

An important technical caveat is that SDT resonator objects require a decay parameter δ that is twice the decay parameter d defined in Eq. (1):

$$\delta_i = 2d_i = -\frac{40 \log_{10} e}{k_i}. \qquad (15)$$

4. RESULTS

A dataset was collected, containing 20 audio recordings from each of the 4 objects in Fig. 1 (a bottle, saucepan, plank, and mug), for a total of 80 samples. They have been recorded in an acoustically isolated room at LIM with a Zoom H4 recorder. The sounds have been produced by manually hitting the objects with a wooden stick.

Additionally, five publicly available recordings of different bells sounds were used: they have been recorded by Daniel Simion and are available under Creative Commons Attribution 3.0[3] licence. Every recording has been cropped to a single repetition, starting right at the onset.

The two datasets were analysed with both the proposed approach and SDTModalTracker, a modal parameters estimator recently added to SDT. The sounds resynthesised using the parameters extracted with the two approaches were compared both objectively and subjectively.

All sounds have been analysed using the same hyperparameters. This serves as a reference for the baseline performance of the approach. An experienced sound-designer should tweak those hyperparameters based on the nature of

[2] `http://soundobject.org/SDT/`

[3] `http://soundbible.com/tags-bell.html`

the sound being analysed. The hyperparameters we used are the following: Hamming window (2048 points), FFT size 16384, hop size 256, magnitude threshold -80 dB, minimum sine duration 0.02 s, maximum number of sines 64, frequency deviation offset 10 and slope 0.001, time delay threshold 0.1 s, initial magnitude threshold -60 dB (absolute), frequency boundaries 20 Hz and 18 kHz, t_{60} threshold 0 s, using time-reversed audio.

The original audio files and the outputs can be found on the GitLab Pages website of the development repository [4], as well as time-domain and time-frequency domain plots [5]. Plots for the in-between analysis steps are also available at the same page.

4.1 Objective Evaluation

To assess the resynthesis accuracy of the two methods, the resynthesised sounds are compared to the original sounds using the mel-frequency cepstrogram (12 MFCCs are computed for every STFT frame, resulting in a time-cepstrum representation of the sound). For every MFCC, the Pearson correlation coefficient is computed between the original and the resynthesised sound. Since the correlation is invariant to scale factors, another metric is employed to account for absolute differences, the normalized Euclidean dissimilarity (NED). This is a variant on the RMSE that is insensitive to bins for which both vectors are zero-valued and is always in the interval between zero and one.

$$\mathrm{NED}(A,B) = \frac{\|A-B\|_2}{\sqrt{2(\|A\|_2^2+\|B\|_2^2)}}. \tag{16}$$

The first dataset was used for this evaluation. For each of the 48 output values (12 MFCCs, 2 evaluation metrics, 2 analysis methods) the Shapiro-Wilk test for normality [26] was performed to ascertain which paired-difference statistical test could be used for comparison. The only output value that did not fail the Shapiro-Wilk test for normality was the NED for MFCC-6.

For MFCC-6, the paired-samples t-test rejected the hypothesis that the two methods have different means for the NED. Also, the Wilcoxon signed-rank test rejects the same hypothesis for the PCC. For every other MFCC, the difference in the empirical values is significant and in favour of the proposed method, according to the Wilcoxon signed-rank test. Fig. 3 shows violin plots with the distributions of the metrics for all MFCCs and for both methods. Stars indicate the level of significance for the p-value of the corresponding paired-difference test.

$$\begin{array}{llll} * & p \leq 0.05 & ** & p \leq 0.01 \\ *** & p \leq 0.005 & **** & p \leq 0.001 \end{array} \tag{17}$$

The macro-average values are summarized in table 1. It should be noted that the PCC is a measure of similarity and the NED is a measure of dissimilarity.

	PCC μ	PCC σ	NED μ	NED σ
SDTModalTracker	0.225	0.293	0.494	0.077
SAMPLE	**0.643**	0.132	**0.351**	0.069

Table 1. Objective evaluation metrics macro-average and macro-standard-deviation.

4.2 Subjective Evaluation

A MUSHRA test (*MUltiple Stimuli with Hidden Reference and Anchor*) [27] has been set up to evaluate the subjective quality of the resynthesised sounds. It has been developed using webMUSHRA, a MUSHRA compliant web audio API based experiment software [28], and deployed to the web [6].

The listening test has 9 pages, one for each of the different sounds (one sample for each of the four objects in Fig. 1, chosen at random, and the five bell sounds). In each page the listener is presented with one reference sound sample and 5 approximations: the reference, two anchors (the reference sound filtered with a low-pass filter at 7 kHz and 3.5 kHz) and two synthetic sound generated with SDT using the modal parameters extracted with SAMPLE and SDTModalTracker.

The listener is asked to report on the quality of each approximation with respect to the reference sound using a scale from 0 to 100. The listener is informed that the reference sound is hidden among the approximations. They are also encouraged to give a prefect score to a sample if they think that it is the reference. The listener can listen to the reference and to the approximations in any order and as many times as they like.

The listener is also asked to use headphones or studio monitors instead of their computer or phone built-in speakers, if possible. At the end of the test the listener must state what listening device they used (headphones, earphones, studio monitors or built-in speakers) at what is their audio background: no background, intermediate (e.g. student or amateur musician or audio engineer) or expert (e.g. musician or audio engineer). Optional information about gender and age can be input, along with a feedback message.

We received 12 entries, two of which have been discarded according to the MUSHRA guideline because they rated the hidden reference below 90 MUSHRA points for more than 15 percent of all test items. One of them reported having no audio background, the other reported being an intermediate. The remaining 10 participants were so divided: 2 no-background, 3 intermediate and 5 experts. Only the discarded intermediate used studio monitors, all other participants used either headphones or earphones. Although the test was not conducted in a controlled environment, these conditions and support have been considered sufficient to draw macroscopic conclusions.

The results for all the valid entries are summarized using violin plots in Fig. 4, similarly to Fig. 3. The sound resynthesised using the parameters extracted with SAMPLE have been considered as *Fair* in the average case.

[4] chromaticisobar.gitlab.io/pyaprsi2/audio

[5] chromaticisobar.gitlab.io/pyaprsi2/plots

[6] www.lim.di.unimi.it/mushra/?config=sample.yaml

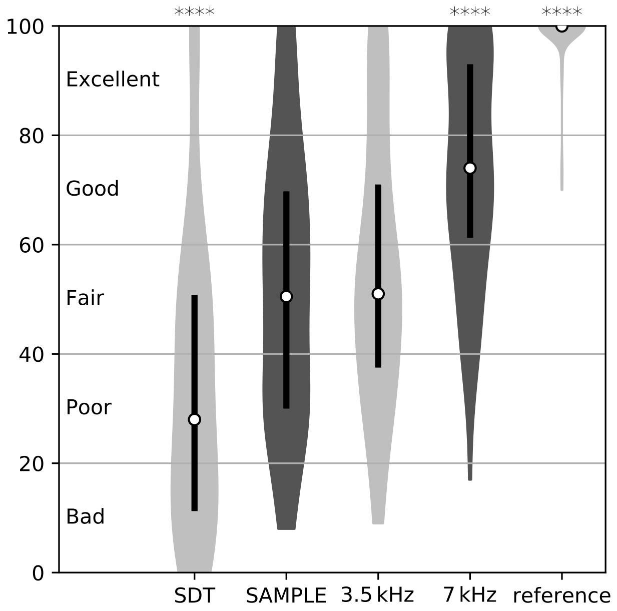

Figure 4. Violin plots for the MUSHRA test scores. The body of the violin plot is the histogram, the black rectangle is the interquartile range and the white dot is the median value. Stars above the groups are indicative of the p-value of the paired-difference test as explained in equation (17). The audio samples resynthesised with parameters extracted with the proposed method have been judged similar to the 3.5 kHz anchor and better than SDT.

The distribution of the scores is also very similar to the 3.5 kHz low-pass filtered anchor and their differences are not significant ($p > 0.05$ for the Wilcoxon signed-rank test). The 7 kHz low-pass filtered anchor and the reference collected better scores (*Good* and *Excellent*, respectively). The sound resynthesised using the parameters extracted with SDTModalTracker have been considered as *Poor* in the average case, that is, significantly worse than SAMPLE ($p \leq 0.001$ for the Wilcoxon signed-rank test).

5. CONCLUSION

A method for estimating modal parameters has been presented, which uses only one audio recording as a sample. The method has been compared to a publicly available open-source method with the same premises in the task of parameter estimation for modal synthesis. The two methods were compared with computational and subjective tests.

The proposed method does not require any previous knowledge of the object geometry or materials and also requires no more than one audio recording of the sound of the impacted object. However, it is only applicable to rigid resonators and it is not robust to external factors such as reverberation. Furthermore, the system that contributes to the sound is modelled as a whole: the contribution of the resonator, the impacting object and any support are not disentangled. Also, the method only models the mode contribution when impacted at one impact point and recorded at one pickup point.

Non-linear phenomena, such as coupling, are not modelled in this approach. Modal systems that have a significantly non-linear behaviour should be analysed by addressing the specific non-linearities. Those non-linearities should also be implemented in the synthesis model.

The parametrization of transients is not addressed and left to the sound-designers. The main motivation of this choice is that it would require a complementary approach (e.g. it could be addressed using information from the residual component). Also, in modal synthesisers such as SDT's there is a small number of transient parameters, that can easily be tweaked by the sound-designer.

Assumptions of modal analysis could be implemented directly in the sinusoidal model, instead of after the analysis, such as stationarity of modes and monotonic decreasing amplitude. A stand-alone GUI will be developed to allow for the fine-tuning of hyperparameters and modal parameters and ease the access for sound-designers who are not familiar with Python.

Beats could be included in the model, allowing for the resolution of partials that are grouped together in the STFT. This could be done either in the time domain or in the time-frequency domain, as a successive fitting step after hinge regression.

The model could be generalized to the case in which more than one example is available, to improve robustness. Similarly, a generalized model could accept optional prior information about the geometry and the material distribution of the object, to extend the model to a space-time process.

Acknowledgments

The sound dataset used for the objective evaluation is courtesy of Ms. Giulia Clerici, LIM (Music Informatics Laboratory), University of Milan.

6. REFERENCES

[1] J.-M. Adrien, "The missing link: Modal synthesis," in *Representations of musical signals*. MIT Press, 1991, pp. 269–298.

[2] C. Cadoz, A. Luciani, J.-L. Florens, C. Roads, and F. Chadabe, "Responsive input devices and sound synthesis by stimulation of instrumental mechanisms: The cordis system," *Computer music journal*, vol. 8, no. 3, pp. 60–73, 1984.

[3] N. Böttcher, "Current problems and future possibilities of procedural audio in computer games," *J. of Gaming & Virtual Worlds*, vol. 5, no. 3, pp. 215–234, 2013.

[4] S. Delle Monache, P. Polotti, and D. Rocchesso, "A toolkit for explorations in sonic interaction design," in *Proc. Int. Conf. Audio Mostly (AM2010)*, Piteå, 2010, pp. 1–7.

[5] S. Baldan, S. Delle Monache, and D. Rocchesso, "The sound design toolkit," *SoftwareX*, vol. 6, pp. 255–260, 2017.

[6] F. Avanzini, M. Rath, and D. Rocchesso, "Physically-based audio rendering of contact," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME2002)*, vol. 2, Lausanne, 2002, pp. 445–448.

[7] K. van den Doel, P. G. Kry, and D. K. Pai, "Foleyautomatic: Physically-based sound effects for interactive simulation and animation," in *Proc. Int. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*, Los Angeles, 2001, pp. 537–544.

[8] C. Picard, F. Faure, G. Drettakis, and P. Kry, "A robust and multi-scale modal analysis for sound synthesis," in *DAFx-09-12th International Conference on Digital Audio Effects*, 2009, pp. 1–7.

[9] R. Michon, S. R. Martin, and J. O. Smith, *MESH2FAUST: a Modal Physical Model Generator for the Faust Programming Language-Application to Bell Modeling*, Shanghai, 2017.

[10] Z. Ren, H. Yeh, and M. C. Lin, "Example-guided physically based modal sound synthesis," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, p. 1, 2013.

[11] A. Sterling, N. Rewkowski, R. L. Klatzky, and M. C. Lin, "Audio-material reconstruction for virtualized reality using a probabilistic damping model," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 5, pp. 1855–1864, 2019.

[12] C. Kereliuk, W. Herman, R. Wedelich, and D. J. Gillespie, "Modal analysis of room impulse responses using subband esprit," in *Proceedings of the International Conference on Digital Audio Effects*, 2018.

[13] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[14] J. S. Abel, S. A. Coffin, and K. S. Spratt, "A modal architecture for artificial reverberation with application to room acoustics modeling," in *Proc. AES 137th Conv.*, Los Angeles, CA, USA, Oct. 2014.

[15] E. Maestre, G. P. Scavone, and J. O. Smith, "Joint modeling of bridge admittance and body radiativity for efficient synthesis of string instrument sound by digital waveguides," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1128–1139, May 2017.

[16] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman, "Visually indicated sounds," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2405–2413.

[17] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, CCRMA Department of Music, Stanford University, 1989.

[18] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[19] X. Serra *et al.*, "Musical sound modeling with sinusoids plus noise," *Musical signal processing*, pp. 91–122, 1997.

[20] J. A. Moorer, "Signal processing aspects of computer music—a survey," vol. 1, no. 1, pp. 4–37, 1977.

[21] H. Purnhagen and N. Meine, "Hiln-the mpeg-4 parametric audio coding tools," in *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*, vol. 3. IEEE, 2000, pp. 201–204.

[22] A. S. Goldberger, "Classical linear regression," *Econometric theory*, pp. 156–212, 1964.

[23] M. Karjalainen, P. Antsalo, A. Mäkivirta, T. Peltonen, and V. Välimäki, "Estimation of modal decay parameters from noisy response measurements," vol. 2002, no. 11, pp. 867–878, 2002.

[24] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT)*. `http://ccrma.stanford.edu/~jos/mdft/`, 2003, online book, 2007 edition.

[25] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, "Cython: The best of both worlds," *Computing in Science & Engineering*, vol. 13, no. 2, p. 31, 2011.

[26] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[27] I. Recommendation, "1534-1,"method for the subjective assessment of intermediate sound quality (mushra)"," *International Telecommunications Union, Geneva, Switzerland*, 2001.

[28] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webMUSHRA - A comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, 2018.