

Interactive Neural Resonators

Rodrigo Diaz
Centre for Digital Music
Queen Mary University of
London
r.diazfernandez@qmul.ac.uk

Charalampos Saitis
Centre for Digital Music
Queen Mary University of
London
c.saitis@qmul.ac.uk

Mark Sandler
Centre for Digital Music
Queen Mary University of
London
mark.sandler@qmul.ac.uk

ABSTRACT

In this work, we propose a method for the controllable synthesis of real-time contact sounds using neural resonators. Previous works have used physically inspired statistical methods and physical modelling for object materials and excitation signals. Our method incorporates differentiable second-order resonators and estimates their coefficients using a neural network that is conditioned on physical parameters. This allows for interactive dynamic control and the generation of novel sounds in an intuitive manner. We demonstrate the practical implementation of our method and explore its potential creative applications.

Author Keywords

real-time synthesis, resonators, neural networks

CCS Concepts

•Applied computing → Sound and music computing; •Computing methodologies → Neural networks; •Human-centered computing → User interface toolkits;

1. INTRODUCTION

The synthesis of impact and contact sounds for real-time interaction has been examined using diverse physically-inspired methods. Often the problem is divided into three smaller ones - modelling or analysing a resonating body, modelling the interaction with the body and modelling the object's sound propagation in space.

The division of the action and the object in the synthesis process has a psychoacoustic justification and has been used in several works [7, 5]. Similarly, implementing such a paradigm is usually realised using the source-filter approach, where the source is an excitation signal, and the filter is a bank of resonators [24].

The resonators, in turn, are tuned to the natural modal frequencies of the object. Obtaining such frequencies can be realised by signal-analysis techniques [8, 2], or through modal analysis using the *finite element method* (FEM) [17].

While the latter method is accurate, it is computationally demanding and must be computed anew for objects of different shapes and materials. Furthermore, when the object is assumed to be non-rigid, its mode shapes change under large deformations, and it is necessary to employ non-linear alternatives [16].

Finally, to simulate the sound propagation in space, it is necessary to determine the acoustic transfer function of the object, which can be computationally intensive as it relies on the object's specific geometry [11].

2. RELATED WORK

In the context of real-time interaction, several techniques have been proposed to improve the speed and efficiency of the modes computation [6] and synthesis, such as using statistical methods to generate realistic finite impulse responses and their interpolation for different impact positions [23].

More recently, neural-based methods have been used to predict modal gains based on an object's shape and certain material parameters, where the sound is rendered using an oscillator bank [12, 13]. Our present work builds on a previous approach [9] that uses a neural network and differentiable resonators to learn the frequency response of different shapes and materials without explicitly assuming a modal model.

Similarly, different methods have been proposed for modelling the interaction with the object depending on its type. For impact sounds, a simple model is to model the contact force on the object as a raised cosine function [24]. Non-linear models that consider the local geometric deformation on the surface of the impacted object (i.e., the extended Hertzian contact model) and of continuous frictional contact have been implemented in the works of Avanzini et al. [4, 5] and Papetti [18]. For scraping and sliding sounds, it is also important to consider the micro-collisions between a textured surface and a scraper. The microscopic surface irregularities that constitute a *surface texture* can be modelled using a fractal-noise [24, 20]. However, these textures can be obtained from experimental observations [23, 1], and alternatively, it is also possible to combine both approaches [19], to enhance the realism of such textures. Our method builds upon some of these techniques for synthesising sliding and scraping sounds.

The present work has two main contributions. The first is a practical implementation of a neural-based pipeline for real-time interactive exploration of contact sounds using physical parameters. We focus mainly on impact and scraping interaction on thin membrane-like objects of arbitrary shape and material. Second, we show an analysis of the effect of the modulation of these parameters on the sound, especially regarding their creative use.



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

3. IMPLEMENTATION

To model the response of a resonating body, we base our approach on a previous method [9]. For this, a neural network is trained using a 2D surface with an arbitrary shape and material parameters. The network predicts the filter coefficients of a differential second-order filter bank. The filter bank’s frequency response is compared to a target response obtained from an object using a FEM solver. The network parameters are optimized based on the difference between the reconstructed response and the target.

For its use in real-time, the neural resonator is implemented as a collection of three *externals* in Max/MSP¹. However, since the resonator is written in C++, it can be easily adapted for its use in other interactive applications such as Pure Data² or Unity 3D³. The interface for mesh deformation is implemented using OpenGL externals and Jitter. In Figure 1, a screenshot of the user interface of the implementation is shown.

3.1 Max/MSP Externals

The first external, `encoder`, is a 2D convolutional encoder based on EfficientNet [22]. It takes as an input a 64×64 grid of binary values that correspond to the two-dimensional shape and outputs a vector $\mathbf{v} \in \mathbb{R}^{1000}$ that encodes the shape information. This is concatenated with two additional vectors - a position vector $\mathbf{x} \in \mathbb{R}^2$ and a material parameter vector $\phi \in \mathbb{R}^5$ that comprises material density (ρ), Young’s modulus (E), Poisson’s ratio (ν) and damping coefficients (α, β). The resulting vector is normalized in the range $[0, 1]$. The ranges for the unnormalized material parameters are set in advance, and these are $\rho \in [500, 15000]$, $E \in [8 \times 10^9, 5 \times 10^{10}]$, $\nu \in [0.1, 0.4]$, $\alpha \in [1, 10]$ and $\beta \in [3 \times 10^{-7}, 2 \times 10^{-6}]$.

The second part of the pipeline is a fully-connected network (`fc`) external. It receives the concatenated vector previously computed by the `encoder` and outputs a set of $5 \times M \times L$ coefficients, where M is the cascade depth of the filter (second order sections), and L is the number of parallel filters. Then each section transfer function is given by:

$$H_{l,m}(z) = g \frac{1 + b_{1,l,m}z^{-1} + b_{2,l,m}z^{-2}}{1 + a_{1,l,m}z^{-1} + a_{2,l,m}z^{-2}}$$

The coefficients are fed to a configurable `filterbank~` external that implements the second-order resonators using the Direct-Form-II realization [21]. The modular design of the pipeline enables us to predict sounds based on changes in position and material parameters separately from changes in shape, reducing the computational load required for the predictions.

3.2 Excitation signals

The resonator bank can be excited in different ways. In the case of impact sounds, we use a kaiser window-like impulse with a controllable β parameter [25]. This single parameter can be used to control the apparent hardness of the impact. Note that methods for non-linear contact excitation [4] can also be applied with a modified configuration of the `filterbank~` [25]. Alternatively, the user can also manually draw the shape of the impulse as shown in Figure 1.

For the simulation of scraping sounds, a method based on previous works [1, 23] is implemented. To simulate sur-

faces of different roughness and structure, we utilize the *pre-quilted* signals [10] provided by the authors⁴. Assuming the contact with surface $S(x, y)$ is continuous, we compute the vertical and horizontal forces F_v, F_h given by:

$$F_v(t) = m_p \left(\frac{\partial^2 S(x, y)}{\partial x^2} |v_x(t)|^2 + \frac{\partial^2 S(x, y)}{\partial y^2} |v_y(t)|^2 \right)$$

$$F_h(t) = v_x(t) \frac{\partial S(x, y)}{\partial x} + v_y(t) \frac{\partial S(x, y)}{\partial y}$$

$$F = \alpha F_v + \beta F_h$$

where $v_x(t), v_y(t)$ is the velocity scraper in each direction and m is the mass of the scraper. The parameters α and β are set arbitrarily by the user to scale the vertical and horizontal forces, respectively.

3.3 Control Parameters

Inference of coefficients for different materials, positions and shapes can be computed in real-time [9], allowing dynamic modulation of the parameters.

When the user creates or modifies a shape, each position within the shape results in a different set of coefficients for the `filterbank~` object. Positional values are in the *window* coordinate space and are scaled to the range $[0, 1]$, where the origin is located at the bottom left of the window. These values are streamed at Max’s *control-rate* (typically 1000Hz). The network predicts smooth coefficient values (as shown in Fig. 4); thus, additional interpolation of the coefficients is not strictly necessary for a smooth transition in the output sound, unlike previous works [1]. At the same time, the position and velocity are also used to generate a corresponding signal to excite the newly configured `filterbank~`.

Similarly, we can simulate non-linear behaviour by modulating the material parameters. For example, we can modulate the Young modulus manually to roughly approximate dynamic elasticity after an impact. This could be used to approximate the *pitch glide* [14] effect observed after large impacts due to non-linear materials and large geometric deformation. Naturally, it can also be modulated in more creative and less physically plausible ways (as shown in Figure 3).

4. EXPERIMENTS AND RESULTS

The implementation, further results and an interactive demo can be found at <https://interactive-neural-resonator.s.com>.

In Figure 2, we demonstrate the effect of modulating the Young’s Modulus parameter. It should be noted that the network was not trained with dynamically modulated input. The reference results, in this case, were obtained by solving a different system at each modulation step and rendering the sound using an impulse. The final modulated result is generated by combining the individual renderings using the overlap-add method. Conversely, the results obtained using our network were achieved by simply filtering an impulse and modulating the parameters at each time step.

The network can also extrapolate to some degree outside the training range. For example, in Figure 3, we show the simultaneous modulation of all material parameters in time. Position parameters can be modulated likewise over time.

Modulating material and positional parameters within the training range yields results consistent with the FEM simulation. However, as the modulation falls outside this range, the predicted sound increasingly deviates from the

¹<https://cycling74.com/>

²<https://puredata.info/>

³<https://unity.com/>

⁴<https://github.com/threedworld-mit/tdw>

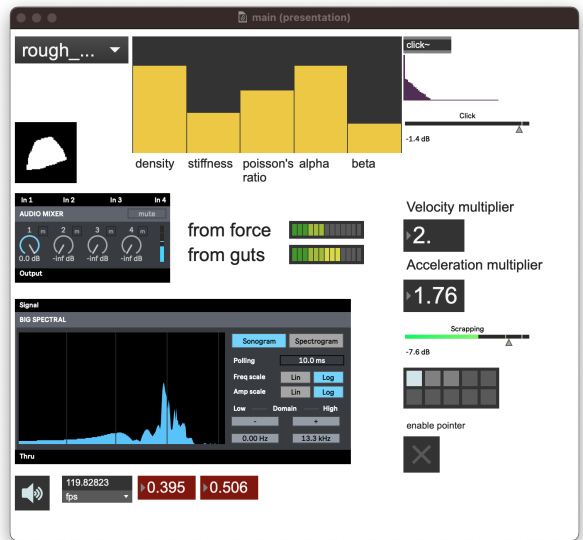
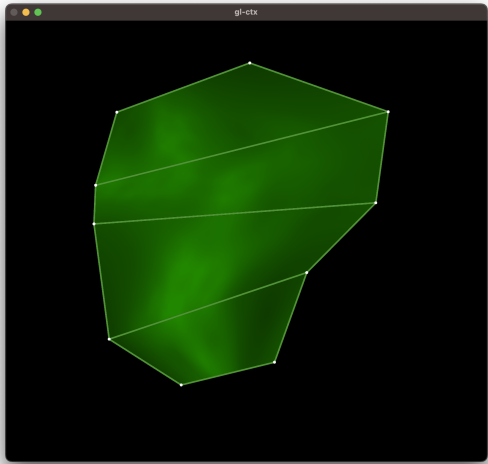


Figure 1: Prototypical user interface in Max/MSP. On the left, the interface for shape deformation is displayed. The user is able to deform a 2D mesh by moving vertex handles or generating a random mesh. They can also interact with the resonator by sliding or scraping inside the shape with the mouse or by clicking an arbitrary point (hitting). On the right, the interface for controlling excitation and material parameters is shown.

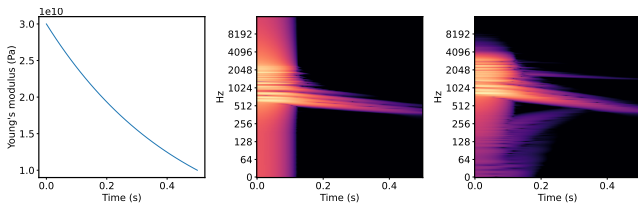


Figure 2: Dynamic control of the Young's Modulus parameter. Left, we show the parameter curve. The middle and rightmost plots show the spectrogram of the prediction and the FEM results, respectively.

one produced using the numerical simulation, i.e., the error with respect to the target frequency response increases. Despite these inaccuracies, the network's predictions confirm that it has acquired an understanding of certain aspects of the sounds. For example, changes in density and Young's modulus influence the pitch, while variations in Poisson's ratio affect the perceived inharmonicity. The damping parameters α and β also correspond to longer decay and less damped frequencies, respectively, when outside the training range.

An interesting effect is observed at the boundary and outside the shape. During training, the network is supervised using nodal positions inside the boundaries, as interaction outside the shape is physically impossible; therefore, the border itself is a discontinuity not present during training. However, the network appears to learn a *softer* approximation to the function near the boundary, as shown in Figure 4. This, in turn, generates interesting ringing effects in the output sound if one scrapes rapidly outside the shape.

5. CONCLUSION AND FUTURE WORK

We have presented an interactive method for synthesizing various contact sounds using neural network-controlled resonators and a practical implementation of this pipeline in Max/MSP. We believe this implementation can be used as a neural instrument for exploratory music performance and

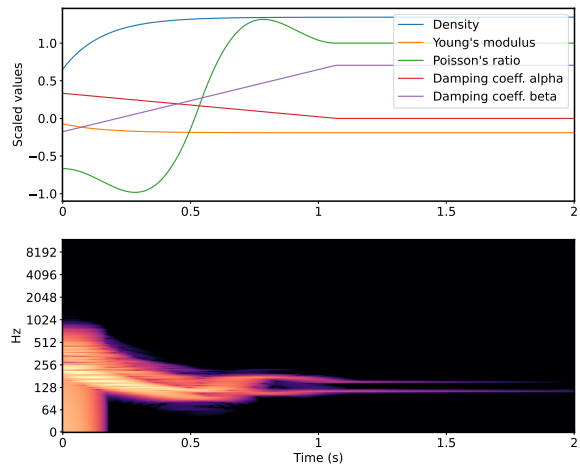


Figure 3: Simultaneous modulation of the parameters of the neural resonator model. On top, a visualization of the modulated parameters in time, where the y-axis is the scaled value of the parameters according to the ranges described in Section 3.1. The parameters are modulated beyond the $[0,1]$ range. On the bottom a spectrogram of the generated sound.

sonic art. To gain more insights into the expressive and subtle design possibilities of our method, we plan to conduct a user study where participants will be assigned *tuning*, and *sound-matching tasks* [3] using our neural instrument.

Given that the network architecture of our pre-trained model does not require significant computational resources, it can easily be adapted to run on embedded devices such as Bela⁵. We plan to develop an embedded version of our pipeline and complement the interaction with our neural synthesizer using audio excitation signals generated from hardware devices such as the Tickle [15] instead of modelling physical excitation signals. This would increase the

⁵<https://bela.io/>

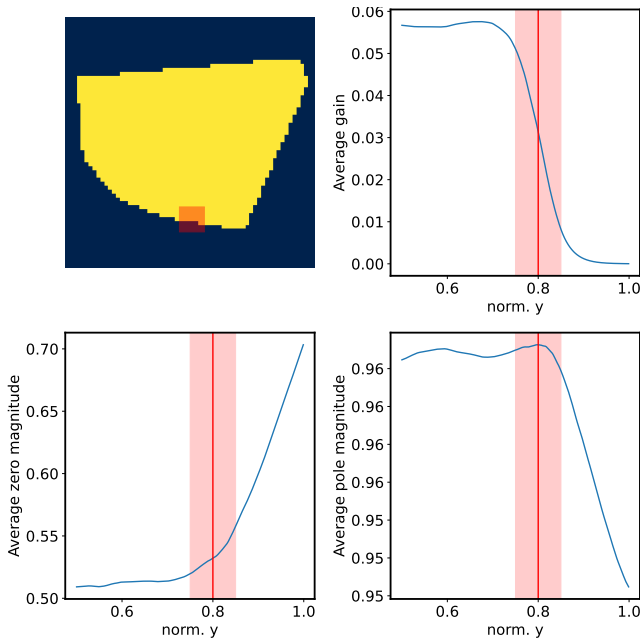


Figure 4: Average magnitude of the estimated parameters (gains, zeros and poles), as a function of the normalized vertical coordinate position. The top left panel displays the shape used for inference with the shaded region indicating the boundary’s location. The estimated parameters are displayed on the top right and bottom panels, with the solid red line representing the shape’s boundary and the shaded area indicating the region within the shape adjacent to the boundary (corresponding to the shaded patch in the top left panel).

accessibility, interactivity, and frugality of our approach. Furthermore, we plan to adapt our model for inference on the web and mobile devices.

We also intend to address the shortcomings of generalization outside the domain of our training set and its robustness against approximation errors in the synthetic training set (e.g., refinement of the tessellation of the input mesh and finite element type). We believe this could be alleviated primarily through the use of real-world labelled recordings.

While we have shown the ability to control the parameters of our neural resonator manually, we believe that it could also be helpful to approximate the parameters using other target sounds as a starting point for creative exploration. This could be achieved, for example, by freezing our model’s parameters and optimizing material parameters using gradient descent. Similarly, we plan to explore the possibility of optimizing the vertex parameters of an initial template mesh as a starting point for shape deformation.

6. ACKNOWLEDGEMENTS

Rodrigo Diaz is supported by UK Research and Innovation [grant number EP/S022694/1].

7. ETHICAL STANDARDS

The focus of this paper is the implementation of a software pipeline utilizing neural techniques trained on numerical simulations of vibration, with no ethical implications or considerations arising from the research and content presented.

8. REFERENCES

- [1] V. Agarwal, M. Cusimano, J. Traer, and J. McDermott. Object-Based Synthesis of Scraping and Rolling Sounds Based on Non-Linear Physical Constraints. In *2021 24th International Conference on Digital Audio Effects (DAFx)*, pages 136–143, Vienna, Austria, Sept. 2021. IEEE.
- [2] M. Aramaki and R. Kronland-Martinet. Analysis-synthesis of impact sounds by real-time dynamic filtering. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(2):695–705, Mar. 2006. Conference Name: IEEE Transactions on Audio, Speech, and Language Processing.
- [3] J. Armitage. *Subtlety and detail in digital musical instrument design*. Thesis, Queen Mary University of London, Apr. 2022.
- [4] F. Avanzini and D. Rocchesso. Controlling material properties in physical models of sounding objects. 2001.
- [5] F. Avanzini, S. Serafin, and D. Rocchesso. Interactive Simulation of rigid body interaction with friction-induced sound generation. *IEEE Transactions on Speech and Audio Processing*, 13(5):1073–1081, Sept. 2005.
- [6] N. Bonneel, G. Drettakis, N. Tsingos, I. Viaud-Delmon, and D. James. Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics*, 27(3):1–9, Aug. 2008.
- [7] S. Conan, E. Thoret, M. Aramaki, O. Derrien, C. Gondre, S. Ystad, and R. Kronland-Martinet. An Intuitive Synthesizer of Continuous-Interaction Sounds: Rubbing, Scratching, and Rolling. *Computer Music Journal*, 38(4):24–37, Dec. 2014.
- [8] P. R. Cook. Physically Informed Sonic Modeling (PhISM): Synthesis of Percussive Sounds. *Computer Music Journal*, 21(3):38, 1997.
- [9] R. Diaz, B. Hayes, C. Saitis, G. Fazekas, and M. Sandler. Rigid-Body Sound Synthesis with Differentiable Modal Resonators, Oct. 2022. arXiv:2210.15306 [cs, eess].
- [10] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, Aug. 2001.
- [11] D. L. James, J. Barbič, and D. K. Pai. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics*, 25(3):987–995, July 2006.
- [12] X. Jin, S. Li, T. Qu, D. Manocha, and G. Wang. Deep-Modal: Real-Time Impact Sound Synthesis for Arbitrary Shapes. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1171–1179, Seattle WA USA, Oct. 2020. ACM.
- [13] X. Jin, S. Li, G. Wang, and D. Manocha. NeuralSound: learning-based modal sound synthesis with acoustic transfer. *ACM Transactions on Graphics*, 41(4):1–15, July 2022.
- [14] T. Kirby and M. Sandler. The evolution of drum modes with strike intensity: Analysis and synthesis using the discrete cosine transform. *The Journal of the Acoustical Society of America*, 150(1):202–214, July 2021.
- [15] M. Neupert and C. Wegener. Interacting with digital resonators by acoustic excitation. 2019.

- [16] J. F. O'Brien, P. R. Cook, and G. Essl. Synthesizing sounds from physically based motion. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 529–536, Los Angeles, California, United States of America, 2001. ACM Press.
- [17] J. F. O'Brien, C. Shen, and C. M. Gatchalian. Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, pages 175–181, New York, NY, USA, July 2002. Association for Computing Machinery.
- [18] S. Papetti, F. Avanzini, and D. Rocchesso. Numerical Methods for a Nonlinear Impact Model: A Comparative Study With Closed-Form Corrections. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2146–2158, Sept. 2011.
- [19] Z. Ren, H. Yeh, and M. C. Lin. Synthesizing contact sounds between textured models. In *2010 IEEE Virtual Reality Conference (VR)*, pages 139–146, Boston, MA, USA, Mar. 2010. IEEE.
- [20] D. Rocchesso, R. Bresin, and M. Fernstrom. Sounding objects. *IEEE Multimedia*, 10(2):42–52, Apr. 2003.
- [21] J. O. Smith. *Introduction to Digital Filters: With Audio Applications*. W3K Publishing, 2007.
- [22] M. Tan and Q. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, May 2019. ISSN: 2640-3498.
- [23] J. Traer, M. Cusimano, and J. H. McDermott. A perceptually inspired generative model of rigid-body contact sounds. 2019.
- [24] K. van den Doel, P. G. Kry, and D. K. Pai. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, pages 537–544, Not Known, 2001. ACM Press.
- [25] S. Zambon. *Accurate Sound Synthesis of 3D Object Collisions in Interactive Virtual Scenarios*. PhD thesis, Universita degli Studi di Verona, 2012.