The 1st DAFx Parameter Estimation Challenge: A Benchmark for Plate Reverb System Identification

Michele Ducceschi
University of Bologna
michele.ducceschi@unibo.it

Leonardo Gabrielli
Università Politecnica delle Marche
l.gabrielli@staff.univpm.it

***Abstract*—We introduce the 1st DAFx Parameter Estimation Challenge, a benchmark initiative designed to advance reproducible research in the system identification of audio effects. Focusing on plate reverberation—an archetypal example of a dense, modal, and weakly damped acoustic system—the challenge invites participants to tackle two tasks: (A) the estimation of physical parameters of a vibrating plate from audio data, and (B) the recovery of modal parameters from synthetic or measured frequency responses. Both tasks are grounded in a physically informed simulation framework based on the damped Kirchhoff–Love plate equation. Baseline solutions and evaluation metrics are provided to encourage fair and transparent comparison of diverse approaches, from traditional optimisation to machine learning. The challenge aims to bridge the gap between the acoustics, signal processing, and AI communities, fostering open collaboration and establishing a long-term benchmark for identifying physical and perceptual parameters in audio effects.**

***Index Terms*—plate reverb, parameter estimation, system identification, audio effects, DAFx challenge**

I. Introduction

Digital emulations of plate reverberation units have a long history in audio effects. While the physical behaviour of thin plates is well understood, the inverse problem—identifying either material parameters or modal parameters from audio data—remains a challenging task. Dense modal overlaps, frequency-dependent damping, and parameter coupling make the system difficult to invert. In preliminary work by the authors, neural networks were applied with limited success to infer modal frequencies, amplitudes, and decay times from synthetic data. This motivated the creation of a shared benchmark: a well-defined dataset and evaluation framework that allows fair comparisons between diverse methods.

Below is the table of contents for this document. Section II introduces the physical model. Readers already familiar with the model may skip this section, although it also defines some notation that will be useful in the following sections. A detailed understanding of the physical model is unnecessary for solving the challenge tasks, so readers may not devote much time to it.

The challenge tasks are presented in Section III. Evaluation metrics and baseline methods are described in Section IV, and submission guidelines are provided in Section V.

The challege repository can be found at the following GitHub link.

Contents

II. The Physical Model

The reference system consists of a thin rectangular plate undergoing small transverse displacements. Here, the plate is considered rectangular with side lengths L_x, L_y, and its displacement is therefore a function of the spatial coordinates $\mathbf{x} := (x, y)$ as well as time $t \geq 0$. Thus, if u denotes displacement, then:

$$u = u(x, y, t) : [0, L_x] \times [0, L_y] \times \mathbb{R}_0^+ \to \mathbb{R}.$$

The motion follows the damped Kirchhoff–Love equation [1]:

$$\rho h \frac{\partial^2 u}{\partial t^2} = -\rho h \mathcal{K} u - 2\rho h \left(\eta_0 + \eta_1 \mathcal{K}\right) \frac{\partial u}{\partial t} + \delta(\mathbf{x} - \mathbf{x}_i) f(t). \quad (1)$$

Furthermore, the operator $\mathcal{K}$ denotes the Kirchhoff-Love plate operator:

$$\mathcal{K} := -\frac{T_0}{\rho h}\Delta + \frac{D}{\rho h}\Delta\Delta. \quad (2)$$

The notation

$$\begin{aligned}\Delta g &:= \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}\\ \Delta\Delta g &= \frac{\partial^4 g}{\partial x^4} + 2\frac{\partial^4 g}{\partial x^2 \partial y^2} + \frac{\partial^4 g}{\partial y^4}\end{aligned}$$

denotes, respectively, the Laplace and biharmonic operators. In Eq. (1), $\delta(\mathbf{x}-\mathbf{x}_i) := \delta(x-x_i)\delta(y-y_i)$ represents the input distribution in the form of a two-dimensional Dirac's delta for input $\mathbf{x}_i := (x_i, y_i)$, and $p(t)$ is the time-varying input generally representing a dry audio signal into the plate. In all that follows, however, $p(t)$ will itself be represented by a pointwise function for impulse response measurement, so that $p(t) := \delta(t)$. Output is extracted from a single point on the plate, denoted $\mathbf{x}_o := (x_o, y_o)$.

A. Plate Parameters

Many constants appear in the model described by Eq. (1). These are: ρ is the volumetric density, h the thickness, T_0 the applied in-plane tension per unit length, $D := Eh^3/12(1-\nu^2)$ the flexural rigidity, with E being Young's modulus and ν being Poisson's ratio. η_0 and η_1 are two loss parameters measured in s^{-1} and s, respectively. Thus, the set of plate parameters is:

$$P = \{\rho, E, \nu, L_x, L_y, h, T_0, \eta_0, \eta_1, x_i, y_i, x_o, y_o\}, \quad (3)$$

in which the input location, also appearing in Eq. (1), and the measured output location were listed. A summary of all the plate parameters is given in Table I, along with the parameter upper and lower bounds.

B. Modal Representation

Plate reverb units are typically suspended from tensioned springs attached to a heavy frame, featuring free edges. Here, for simplicity, the effect of the spring tension has been incorporated into the global parameter T_0, and the boundaries of the plate are assumed simply-supported, such that:

$$u = \frac{\partial^2 u}{\partial n^2} = 0 \text{ along the boundary, } \forall\, t \quad (4)$$

with n denoting the direction perpendicular to each edge. This simplification allows for the analytical solution of the eigenproblem. The eigenproblem is formulated in the absence of losses and external forcing in (1). Furthermore, a particular solution is:

$$\begin{aligned}u(x,y,t) &:= \Phi_m(x,y)e^{-j\Omega_m t}, \quad \text{with:} & (5a)\\ \Phi_m &:= \frac{2}{\sqrt{L_x L_y}} \sin\frac{m_x \pi x}{L_x} \sin\frac{m_y \pi y}{L_y}, & (5b)\end{aligned}$$

where m_x, m_y are positive integers (i.e., 1,2,3,...). Note the important *orthonormality property* for Φ:

$$\int_0^{L_x}\int_0^{L_y} \Phi_m\, \Phi_{m'}\, \mathrm{d}x\mathrm{d}y = \kappa_{m,m'}, \quad (6)$$

with $\kappa_{m,m'} = 1$ if $m = m'$, and zero otherwise. Example modal shapes are given in Fig. 1.

Here, $\Phi_m(x,y)$, depending only on the spatial coordinates, is the *modal shape*, associated with a modal frequency Ω_m. The modal shape clearly satisfies the boundary conditions (4), and it solves the spatial eigenproblem analytically. In fact, computing $\mathcal{K}\Phi_m$, from (2), gives:

$$\begin{aligned}\mathcal{K}\Phi_m &= (\rho h)^{-1}\left(T_0\gamma_m^2 + D\gamma_m^4\right)\Phi_m \quad \text{with} & (7a)\\ \gamma_m &= (p_x^2\pi^2/L_x^2 + p_y^2\pi^2/L_y^2)^{\frac{1}{2}}. & (7b)\end{aligned}$$

Inserting such a solution in (1) with $\eta_0 = \eta_1 = f = 0$, thus, results in:

$$\Omega_m = (\rho h)^{-\frac{1}{2}}\gamma_m(T_0 + D\gamma_m^2)^{\frac{1}{2}} \quad (8)$$

Inserting back losses and forcing in (1), the general solution is given as a superposition of modes.

$$u(x,y,t) = \sum_m \Phi_m(x,y)q_m(t), \quad (9)$$

where the $q_m(t)$'s are now unknown, and must be solved for.

C. Modal system and decay constant selection

An equation for $q_m(t)$ is obtained easily after inserting the general solution (9) into (1), multiplying by $\Phi_{p'}$, integrating over the plate's domain, and using the orthonormality property (6) and result (8):

$$\ddot{q}_m(t) = -\Omega_m^2 q_m - 2\sigma_m\dot{q}_m + (\rho h)^{-1}\Phi_m(x_i,y_i)\delta(t), \quad (10)$$

for $p = 1, ..., M$, and with $\sigma_m := \eta_0 + \eta_1\Omega_m^2$ being a frequency-dependent loss coefficient. This is the standard equation of a harmonic oscillator with loss and Dirac input.

In the code, the decay constants η_0, η_1 are not selected directly, but rather by choosing the decay times τ_1, τ_2 at two specific frequencies Ω_0, Ω_1. For model (10), the amplitude decays as $e^{-\sigma_m t}$, and hence, for a 60 dB drop, the decay time is $\tau_m = 3\log(10)/\sigma_m$.

$$\begin{aligned}\eta_0 &= \frac{3\log(10)}{\Omega_1^2 - \Omega_0^2}\left(\frac{\Omega_1^2}{\tau_0} - \frac{\Omega_0^2}{\tau_1}\right), & (11a)\\ \eta_1 &= \frac{3\log(10)}{\Omega_1^2 - \Omega_0^2}\frac{\tau_0 - \tau_1}{\tau_0\tau_1}. & (11b)\end{aligned}$$

In the code $\Omega_0 = 0$ (DC), and $\Omega_1 = 1000\pi$ (i.e., a linear frequency of 500 Hz), so that only two time parameters τ_0, τ_1 are required to set the decay constants. Furthermore, for energy dissipation (passivity) at all frequencies, one must select the decay times such that $\tau_0 > \tau_1$ (i.e., the decay at DC must be slower than the decay at 500 Hz). This condition is enforced in the code provided on GitHub.

D. Frequency domain representation and output extraction

Output is computed simply by evaluating (9) at $\mathbf{x} = \mathbf{x}_o$. Thus, transforming (10) in the frequency domain by means of $\frac{d}{dt} \to j\omega$, the measured output in the frequency domain, for a Dirac input at (x_i, y_i) is:

$$\begin{aligned}\hat{U}(x_o,y_o,\omega|(x_i,y_i)) &= \sum_m \hat{Q}_m(x_o,y_o,\omega|(x_i,y_i)), \text{ with:}\\ \hat{Q}_m(x_o,y_o,\omega|(x_i,y_i)) &:= \frac{\Phi_m(x_i,y_i)\Phi_m(x_o,y_o)}{\Omega_m^2 - \omega^2 + 2j\sigma_m\omega} \quad (12)\end{aligned}$$

Thus, the output in the frequency domain is expressed as the sum of simple *second-order all-pole filters*. All such filters are arranged in a parallel topology.

Table I
PLATE PARAMETER RANGES.

Parameter	Symbol	Min	Max	Notes / Units
Plate length (x)	L_x	0.5	1.0	m
Plate length (y)	L_y	1.1	4.0	m
Thickness	h	0.001	0.005	m
Tension per unit length	T_0	0.01	1000.0	N/m
Material density	ρ	2430.0	21230.0	kg/m^3
Young's modulus	E	6.7×10^{10}	2.2×10^{11}	Pa
Poisson's ratio	ν	0.25	0.25	Fixed
Decay time (0 Hz)	τ_0	6.0	10.0	s
Decay time (at f_1)	τ_1	1.0	5.0	s
Reference loss frequency	f_1	500.0	500.0	Hz (Fixed)
Input position (x)	x_i	0.00	$0.49 \cdot L_x$	(in the code, expressed as a fraction of L_x)
Input position (y)	y_i	0.00	$0.49 \cdot L_y$	(in the code, expressed as a fraction of L_y)
Output position (x)	x_o	$0.49 \cdot L_x$	L_x	(in the code, expressed as a fraction of L_x)
Output position (y)	y_o	$0.51 \cdot L_y$	L_y	(in the code, expressed as a fraction of L_y)

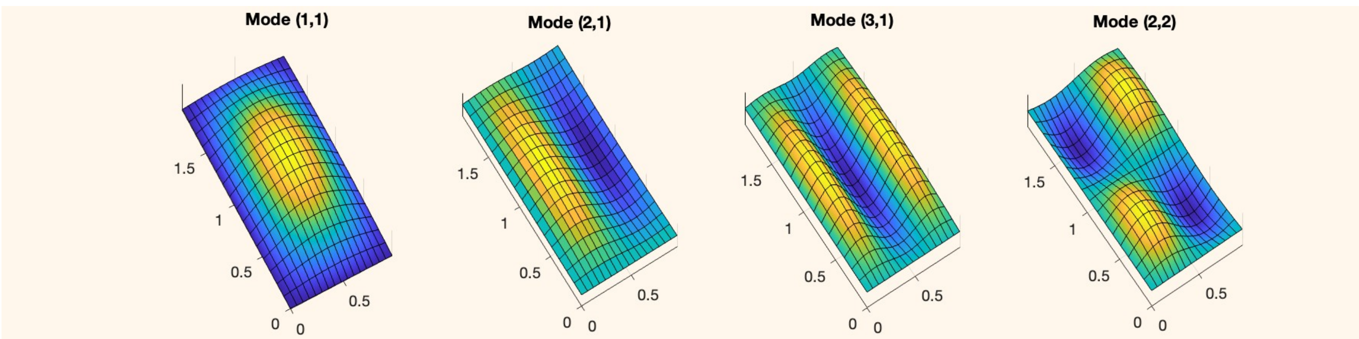

Figure 1. First four modal shapes $\Phi_m(x, y)$ of a plate with $L_x = 1$ m, $L_y = 2$. The corresponding modal indices m_x, m_y are given.

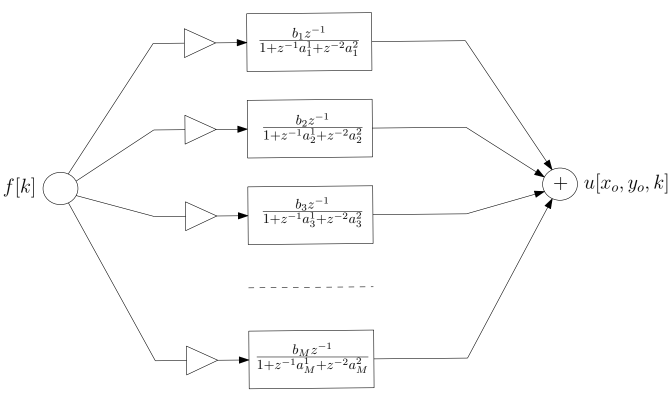

Figure 2. Modal reverb architecture. A sampled input $f[k]$ is projected onto a bank of modes, described by an all-pole biquad filterbank. Each filter is represented by three coefficients, which are related to the mode's natural frequency, decay time, and gain. Output at location x_o, y_o is then reconstructed by summing up all the modal contributions.

E. Discrete-time model

Updating equation (10) in time is accomplished easily by discretising each $q_m(t)$ with a *time series* $q_m[k]$, where k is the *time index* such that $q_m[k] \approx q_m(t_k)$, with $t_k := Tk$, for sampling interval T.

In the Python code provided on GitHub, an update scheme is already implemented using the exact form for an all-pole biquad discretisation [2]. It requires no further intervention from the reader. The update equation is of the form:

$$q_m[k+1] = 2r_m \cos(\Omega_m T) q_m[k] - r_m^2 q_m[k-1] + b_m f[k], \tag{13}$$

with:

$$\begin{aligned} r_m &:= e^{-\sigma_m T} \\ b_m &:= \frac{4T^2 \Phi_m(x_i, y_i)\Phi_m(x_o, y_o) r_m}{\rho h L_x L_y}. \end{aligned}$$

In the z- domain, the frequency response is thus obtained as:

$$H_m(e^{j\omega T}) = \frac{b_m e^{j\omega T}}{1 - 2r_m \cos(\Omega_m T) e^{j\omega T} + r_m^2 e^{2j\omega T}} \tag{14}$$

This is the discrete-time Fourier transform of mode m for scheme (13). The modal workflow is summarised in Fig. 2.

III. CHALLENGE TASKS

The challenge consists of two branches, both of which rely on synthetic data generated by the reference model. The two tasks are similar in requiring the inverse estimation of the plate parameters, but differ in the type of parameters:

- Task A asks to estimate a subset of the plate's parameters P as listed in Eq. (3). The output is a handful of physical, geometrical and material parameters defining the plate.
- Task B, on the other hand, asks to estimate the modal parameters making up the plate's response. This is a larger

scale problem, with thousands of modal parameters to estimate.

IMPORTANT NOTICE. A priori knowledge of the plate's modal distribution—such as the analytical frequency formula (8) or the behaviour of decay times with frequency described in Section II-C—**must not** be used to solve the challenge tasks. Only the provided synthetic impulse or frequency responses should be employed, as if they were obtained from actual measurements rather than from an analytical model.

A. Task A: Physical Parameter Identification

Given an impulse, obtainable via the code shared on the GitHub repo, infer the subset $S(P)$ of the physical plate parameters in (3):

$$S(P) = \{\rho, E, L_y, h, T_0, x_o, y_o\}. \tag{15}$$

The following parameters, belonging to P, have been excluded from S: L_x, ν, η_0, η_1, x_i, y_i, and are *fixed* for each simulation. This is partly to avoid ambiguities and redundancies and partly because some parameters, such as the modal decay times, are estimated in Task B. The fixed values for the excluded parameters are as follows:

Table II
FIXED VALUES OF PLATE PARAMETERS FOR TASK A.

L_x	ν	τ_0 (at 0 Hz)	τ_1 (at 500 Hz)	x_i	y_i
1.0 m	0.25	6 s	2 s	$0.335\,L_x$	$0.467\,L_y$

B. Task B: Modal Parameter Estimation

The modal decomposition (12). Given an impulse or a frequency response, retrieve, for each mode, its modal parameters

$$M_m = \{\Omega_m, \sigma_m, b_m\},\ m = 1, ..., M, \tag{16}$$

corresponding to the mode's natural frequency, decay constant (in s^{-1}) and modal gain, as they appear in the update equation (13).

IV. Datasets, Baseline and Evaluation Metric

Participants can generate training and validation data using the official Python simulator, which is provided in the challenge repository. The simulator generates an IR of a given duration starting from the physical parameters, while a Python script is provided to generate your own dataset with audio and data (including physical parameters and mode parameters).

The target dataset for the final evaluation will be provided at a later time. It will consist of a sufficient number of plates to enable a statistically significant ranking to be computed. Furthermore, the target dataset size will discourage methods that require excessive computational time.

A. Evaluation for Task A

For Task A, the main metric for the plate parameter estimation is the Normalized Mean Squared Error (NMSE):

$$\mathrm{NMSE} = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{(\alpha_i^{(\mathrm{est})} - \alpha_i^{(\mathrm{ref})})^2}{\alpha_i^{(\max)} - \alpha_i^{(\min)}}. \tag{17}$$

Here, N_S is the number of parameters appearing in $S(P)$, as per (15), and $\alpha_i \in S(P)$, while $\alpha_i^{(max)} - \alpha_i^{(min)}$ is the range of the parameter.

The final ranking will be compiled based on the NMSE. However, the organisers reserve the right to include additional metrics if they believe that they can provide additional details of scientific interest.

The participants are required to provide information on the total runtime and, if the algorithm employs an iterative approach or involves a training phase, to specify the number of iterations/epochs and the hardware used to execute the algorithm. This information will not modify the ranking but will allow the organisers to grant a special mention to proposals that are both particularly efficient in terms of computational time/power and good in the estimation task. The format for supplying this information can be found in the baseline script provided on GitHub. Please note that brute force methods will not be considered. We consider brute-force methods those that do not rely on physical information or on a loss surface obtained by the physical model. Search methods informed by the loss surface should not take excessive computation time (e.g. more than 1 hour).

B. Baseline for Task A

The baseline for Task A is a Particle Swarm Optimization (PSO) tailored to work for the task. PSO is based on a swarm of particles that explore the N_S-dimensional parameter space by exploiting a social behaviour, i.e. by sharing knowledge of the explored loss surface with the other particles, in order to find the global minimum. Each particle has velocity and acceleration, as well as a direction vector, which is updated iteratively based on three coefficients: the inertia w, the self-trust coefficient c_1, and the neighbour-trust coefficient c_2. The error surface is computed by a popular loss in the field of Deep Learning, i.e. the Multi-scale Spectral Loss (MSS) [3]. PSO operates within a normalised parameter range of $[0, 1]$, to ensure consistency in particle velocity and acceleration across all dimensions.

The baseline is provided to demonstrate how to handle the input data and format the results, enabling the organisers to evaluate the proposed methods accurately. The baseline results will stand as a reference for the participants to surpass.

Please note: a constant predictor method will also stand as a lower bound for performance. When the parameters range is known, a lazy method to obtain a MSE-sense optimal parameter estimation is that of providing constant values to each input, that is the average of the parameter distribution. The NMSE score of a constant predictor (averaged by N_S) is 1/12, i.e. the variance of the random variable across the uniform distribution $[0, 1]$.

C. Baseline and Evaluation for Task B

For Task B, a baseline metric is provided based on a two-stage process:

- Stage 1: identification of modal peaks in the magnitude response, from which modal frequencies and modal bandwidths are estimated. These yield approximate values for Ω_m and σ_m. More specifically, the modal frequencies Ω_m are obtained by fitting a *parabola* around each identified peak in the magnitude response, and by computing the analytic maximum of the parabola. The loss coefficient σ_m is obtained via the *half power bandwidth* rule, that is, given the radian frequencies ω_m^+, ω_m^- found at $H(\Omega_m)/\sqrt{2}$, and defining $\Delta\Omega_m := \omega_m^+ - \omega_m^-$, then $\sigma_m \approx \Delta\omega_m/2$.
- Stage 2: estimation of the modal gains b_m by probing the imaginary part of the frequency response in correspondence with the peak locations in the magnitude response. The frequency response for a single mode is given in Eq. (14). Evaluating the frequency response around the modal frequency, i.e. for $\omega \approx \Omega_m$, one obtains the approximate formula:

$$H(e^{j\Omega_m T}) \approx -j\frac{b_m}{2\sigma_m T\sin(\Omega_m T)}. \tag{18}$$

Thus, the frequency response is almost purely imaginary, and the modal gain b_m is easily estimated by knowing the modal frequency and decay constant, as calculated in the previous step. This strategy performs satisfactorily when the peaks are sparse and clearly visible in the spectrum, but misses many modes in highly overlapping modal scenarios, as exemplified by the provided baseline code. Note that here, as opposed to Task A, all parameters are allowed to vary within their ranges, as per Table I.

The evaluation is based on the similarity between the discovered modal parameters and the actual modal parameters of the simulation. The relative error is defined for the three modal parameters:

$$\mathrm{RE}_\Omega = \frac{1}{M}\sum_{m=1}^{M}\frac{|\Omega_m^{(\mathrm{est})} - \Omega_m^{(\mathrm{ref})}|}{\Omega_m^{(\mathrm{ref})}}, \tag{19}$$

$$\mathrm{RE}_\sigma = \frac{1}{M}\sum_{m=1}^{M}\frac{|\sigma_m^{(\mathrm{est})} - \sigma_m^{(\mathrm{ref})}|}{\sigma_m^{(\mathrm{ref})}}, \tag{20}$$

$$\mathrm{RE}_b = \frac{1}{M}\sum_{m=1}^{M}\frac{|b_m^{(\mathrm{est})} - b_m^{(\mathrm{ref})}|}{b_m^{(\mathrm{ref})}}. \tag{21}$$

The combined relative error is then used as a starting metric for assessing the goodness of the estimation algorithm:

$$\mathrm{RE}_0 = \frac{1}{3}\left(\mathrm{RE}_\Omega + \mathrm{RE}_\sigma + \mathrm{RE}_b\right). \tag{22}$$

Note that, contrary to the previous task, the total number of modes M and, hence, the total number of modal parameters is *unknown*. Thus, if $\tilde{M}$ represents the identified number of modes, it will generally be true that:

$$\Delta M := |M - \tilde{M}| > 0. \tag{23}$$

Thus, the evaluation metrics (22) is augmented as:

$$\mathrm{RE} := \mathrm{RE}_0 + \Delta M, \tag{24}$$

which corresponds to assigning a value of zero to each unidentified modal parameter. The correspondence between the identified and actual parameters will be determined by frequency, with the identified parameters sorted according to their closest frequency match in the actual set.

V. Participation and Submission

Teams may participate in one or both tasks. The submission package must include:

- A short technical report (2–4 pages) describing the method.
- Source code and instructions for reproducibility.
- Result files for the official test set.

Additional information will be shared on the dedicated GitHub repository. The organising committee will store all code and reports in a public archive after evaluation.

The challenge runs throughout the academic year 2025–2026. Results will be presented, and winners announced, at DAFx26 in Boston.

References

[1] S. Bilbao, "A digital plate reverberation algorithm," *Journal of the Audio Engineering Society*, vol. 55, pp. 135–144, March 2007.

[2] M. H. Haupt, *Introduction to Digital Signal Processing and Digital Filtering*. Clifton Park, NY: Delmar Thomson Learning, 2001.

[3] J. Engel, C. Gu, A. Roberts *et al.*, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.