Audio Engineering Society

Convention Paper 10437

Presented at the 149th Convention
Online, 2020 October 27-30

Rapid Approximation of Weakly-Nonlinear Mode Parameters

Travis Skare[1] and Jonathan Abel[1]

[1] *CCRMA, Stanford University*

Correspondence should be addressed to Travis Skare (travissk@ccrma.stanford.edu)

ABSTRACT

Modal processors and synthesizers require determining resonant frequencies and decay rates of a system. We introduce a simple, efficient, and flexible approach to compute these parameters when speed is of the essence, for example modeling a user's sample folder in a realtime mobile synthesizer application. The approach compares DFT peaks in two impulse response windows at different points in time to obtain the modal frequencies and decay rates. The approach trades precision for speed, and should be considered an approximation.

For a selection of samples across tonal/atonal instruments and reverb responses, we consider sensitivity to window choice and window size, and compare results to traditional modeling methods, which operate on the entire impulse response. We propose extensions of the algorithm to more than two windows, and to capture nonlinear, time-varying behavior such as the pitch dive after striking a tom drum at high velocity.

1 Introduction

Room or instrument simulation via modal processors[1] or modal synthesis is summarized by a set of control parameters: N mode frequencies ω_n, a complex modal amplitude γ_n, and a decay rate α_n.

These parameters may be obtained mathematically based on the underlying physics of the system, analytically based on measurements of a real instance of the system, or synthetically by combining/transforming existing models or building them from scratch.

Our recent applications–realtime synthesis and effects plugins–generally use analytical methods, for example computing a room response from an impulse response, or a bar percussion response from a recording of a struck note. In such cases, we typically take an entire sound recording, compute the DFT, find local maxima in the spectrum, and pick the top N maxima based on raw amplitude, optionally after factoring in a equal-loudness curve. Decay rates may be computed based on energy within a band, for example. In some applications, speed and low compute power may be of the essence. Consider a user rapidly browsing through a folder of samples in a modal effects plugin. It is ideal for the analysis step to be imperceptible, ideally significantly faster than real-time.

Similar systems exist in the literature, for example the PARSHL system[2, 3] describes a realtime spectral analysis and resynthesis approach, where a set of sinusoids have parameters adjusted based on results of a realtime analysis step. This approach is recalled in

section 2.6.3 of the thesis of Scott Van Duyne[4]; we expand on it, for example to model small nonlinear effects such as the effect of attack transients on the pitch components, and study pros and cons for tonal sounds and atonal room responses. Note our use cases cover single notes and impulses; for partial tracking over time, full spectrogram data may be used, optionally with strategies such as Linear Programming, covered in detail by Neri and Depalle [5].

Non-Spectral based methods for modal processors for room response have been considered, for example recently by Kereliuk et. al[6], who applied ESPRIT[7, 8] in subbands to reduce computational complexity in both analysis and synthesis. Subbands were also used in the problem of fitting modes to a room response in [9] and [10], with the latter using a frequency-domain pole-zero optimization technique to reduce computation complexity at *synthesis* time.

1.1 Proposed Approaches

In the next section we present a baseline version of a simple algorithm to obtain modal parameters from two windows; such a method is very straightforward with an assumed-linear modal response and static frequencies. We then extend to three windows to attempt to capture nonlinear behaviors such as modes shifting around attack transients before settling. A continuously-evolving sound is outside the problem under study, though we may add additional windows at different times to essentially establish a frequency envelope on different modes and capture moving sounds. In such cases, we again note approaches such as PARSHL or linear-programming partial tracking[2, 3, 5] should be considered; here we study applicability of a lightweight approach.

2 Method Description

2.1 Simple Case, Two Windows

We have two windows, w_0 and w_1; these start at t_{w_0} and t_{w_1}. Without loss of generality let the former precede the later; that is $t_{w_0} < t_{w_1}$. They may partially overlap, they may occur immediately serially in time, or there may be a time gap between the windows. For any of these three cases, define:

$$\Delta t_{w_{0,1}} = t_{w_1} - t_{w_0} > 0 \tag{1}$$

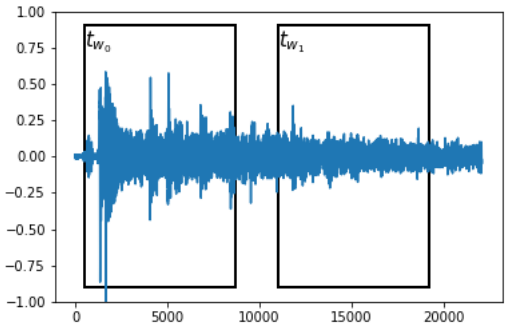

Fig. 1: *Selecting two windows in an impulse response with initial attack and tail components of sufficient length.*

The windows are each of of length M, and have a common windowing function. We discuss qualitatively the sensitivity to windowing functions in Section 3.2 and quantitatively to $\Delta t_{w_{0,1}}$ in 3.3; as a preview, this may be chosen freely by system implementors.

Next, take the DFT of each window to obtain a spectrum. We can compute the top N modes from each, or obtain different-sized maxima sets N_0 and N_1 if needed.

At this point, there are different possibilities for how to analyze our data–in fact, our branching paths actually started earlier, with the choice of t_{w_0} and t_{w_1}. Sometimes we may be operating on our own sampled and curated data and can make informed decisions as to these values, but presented with a directory of context-less user-supplied sounds, we may need to apply a set of general-purpose default values, or apply heuristics.

Without knowing anything about the input ahead of time, a default approach might be to trim silence from the input and obtain DFTs from windows at the onset of the sample t_0 and at the midpoint, separated by some difference (the length of the sound minus a window length), for single-note strikes or room responses, perhaps based on t_{60}. If the sample is not sufficiently long to have separate windows, instead simply take two adjacent windows. If the sample is not sufficiently long for even that, then allow the windows to overlap. If they would need to overlap substantially, consider choosing shorter windows or perhaps choose another approach altogether to estimate decay rates. Moving forward,

we assume the sound is at least long enough to have adjacent or even slightly-separated windows[1].

We use the spectra from the later window to obtain top N modes. In detail, we make an $O(M)$ pass to filter the spectra to only contain local maxima (based on complex amplitude), and then sort to obtain an ordering ($O(MlogM)$).

As a first approach we take the top N modes from that later window, optionally applying an equal-loudness curve, as our set for later synthesis. To obtain decay constants, we compare complex amplitude between the two windows for mode m, to see how quickly it decays over that short time. As the modal synthesis model expects exponentially-decaying partials, we may apply this as our overall damping factor.

That operation is parallelizeable and vectorizeable. However, there are edge cases: what if mode m is not present in the first window? Or what if it *is* present but has lower amplitude than it does in our second window? Both are problematic to our assumption that the system consists of decaying sinusoids at fixed frequencies.

For such cases, we explore choices: we can see if a frequency not in N is nearby and can be used - indicative of a mode that shifted in frequency though we can't be certain. We can average gain for a band of the original sound for decay rate–an approximation though we are synthesizing a mode that didn't exist in the original sound. We could discard mode m altogether. This may be a mistake for a tonal instrument with relatively few modes, like bar percussion, than it is a cymbal with thousands of modes and the rough distribution seems to define the sound. In practice, we apply a heuristic: look for similar effects in the harmonic series, and look for nearby modes that disappear in the opposite direction. Such cases are indicative of a pitch bend, and one example is demonstrated in section 4.

Whatever the choice, we have a set of N modes, and complex amplitudes γ_{m_0} and γ_{m_1}. Computing the damping factor from this information is straightforward: given that the windows are $\Delta t_{w_{0,1}}$ apart:

$$\frac{|\gamma_{m_1}|}{|\gamma_{m_0}|} = (1-\alpha_m)^{\Delta t_{w_{0,1}}} \tag{2}$$

[1]For shorter sounds, such as a snare drum, a full-sound window and a different method of measuring decay should be used.

taking the log of both sides, dividing by Δ_t and re-exponentiating:

$$\alpha_m = 1 - e^{\frac{log\frac{|\gamma_{m_1}|}{|\gamma_{m_0}|}}{\Delta t_{w_{0,1}}}} \tag{3}$$

At this point we have our scalar frequencies ω_m, complex amplitude γ_{m_0}, and damping factor α_m for each mode, at the cost of two DFTs on windows shorter than the sound, plus an $O(MlogM)$ sort (we could even reduce this to $O(M)$ with a radix sort but expect overall runtime to be similar)

We may wish to take a look at modes that were in the first window but not the second, and consider if they might be modeled with noise or a sampled attack, as in PARSHL[3] or with a sample as in some commercial synthesizers of the late 1980s and early 1990s, such as the Roland D-50.

2.2 Extension to Three or More Windows

Some of our sounds have nonlinear effects such as striking a tom drum at high velocity. A spectrogram of this sound is shown in Figure 2

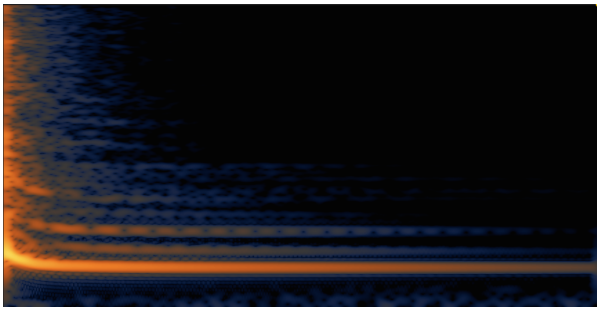

Fig. 2: *Rack tom hit at high velocity; 2.5 second spectrogram from [0Hz .. 1.5kHz]. Note the dense attack, downward pitch glide during first 200ms, and a tail that is comprised mostly of the fundamental and few overtones.*

The approximation method may be extended to more than two windows, and overall frequencies/decay rates may be obtained via least-squares matrix formulation. We may wish to hold the last window as the steady-state pitch, consider the prior grouped frequency points on a logarithmic scale, then run linear least squares on that scale to obtain the pitch dive over time. Here,

the approximation nature of the algorithm should be re-emphasized. Depending where we choose our window, we will likely find inexact peaks to the modal pitch envelopes. This is especially true if the peaks for different modes are separate in time. Given more compute cycles, we may place a middle window over peaks obtained from the spectrum.

While this strategy attempts to deal with nonlinear pitch bends, it still requires the pitch to 'settle' to something that can be synthesized by summing partials: the model will give incorrect results in the case where the tail of a sound continues to vary in pitch–a guitar with a vibrato effect, for example, or a drum loop that made its way into a sample directory. In such cases, something that tracks evolving partials over time we again suggest an analysis-resynthesis system such as PARSHL or continuous tracking via linear programming Neri and Depalle [5].

3 Applications and Examples

We apply this approach to different sound types. The following eight samples represent our test set, intended to cover use cases of tonal instruments, a sample with a clear nonlinear effect, and three different reverb types. The first word of each bullet point is an identifier which will be used moving forward.

- Piano - tonal, undamped piano note, rich spectrum with aurally different attack and decay segments, lengthy tail.
- Cowbell - pitched percussion strike; quick decay makes this a challenging or degenerate case.
- Tom - Rack Tom drum strike exhibiting nonlinear pitch dive (as in Figure 2). Many modes decay quickly, presenting a potential challenge.
- CymbalA - many modes, qualitatively simpler, "bright" response - Zildjian A custom 16" Crash, medium-high velocity strike
- CymbalB - qualitatively more complex, "dark" response - Zildjian K Custom Dark China cymbal, medium-high velocity strike
- Spring - Spring reverb IR based on an impulse response of a Fender Deluxe Reverb reverb unitvia YouTube [11].
- Hall - Long, dense church hall reverb IR fromRoos) [12].
- Plate - Vocal plate reverb IR fromRoos) [12].

We use some of these examples to explore effects and sensitivity of the algorithm's parameters.

3.1 Choice of Comparative Metrics

An immediate question is: which metric should we use to gauge how close two sets of mode parameters are?

An ideal metric would have one accurate, repeatable, intuitive value to capture how close one set of modes is to another.

However, in some applications we may find that specific modal frequencies and their harmonic relationship matter (tonal instrument samples), and sometimes it is the overall distribution, and specific frequencies can be shifted slightly without humans noticing (cymbal tails, room responses). We may or may not care about decay rate precision–in some applications we may want an exact match; in some applications we may want the modes to decay somewhat proportionally to their true physical values, and in some applications we may be planning to enforce our own decay rates based on a predetermined waterfall plot. For our applications we would like to determine accuracy of decay rates.

Frequency matching for Tonal cases:

In cases where we would want our model to obtain the correct top frequencies (piano, brass, winds...), we order our matching modes by amplitude in the steady-state based on their amplitudes: $\gamma_0 \geq \gamma_1 \geq \gamma_2 ...$ and create a vector made up of repeating entries:

$$< \omega_m, k_0\gamma_m, (2\pi k_1)\alpha_m >, m \in [0, N'-1] \quad (4)$$

Where N' is either all modes N, or a truncation providing the subset of the N' top modes, $N' < N$.

k_0 and k_1 represent how important the amplitude and decay constant accuracy are.

Each set of statistics may be normalized to $[0,1]$ and we may compute vector distance between two sets of mode parameters; we use L2 norm and set $k_0, k_1 = 0$ to study found frequencies, and call this metric found-mode-distance $MD()$ for short. Establishing an improved metric for comparing sets of modal parameters is an area of exploration; we also experimented with e.g. Jaccard similarity for the frequency sets but found strategies with exact comparisons sensitive to window placement in the face of nonlinearities.

3.2 Choice of Window Type

During experimentation the approach did not seem as sensitive to window type as other spectral analyses; we initially explored using a rectangular window and compared with Blackman windows and variants, settling on a Kaiser window of order 5 for the analyses in this section.

3.3 Sensitivity to choice of t_{w_0}

We check the sensitivity of starting position for our attack window. This is done by capturing a "reference" set of modes obtained using the approach with windows at 0.5 and 1 second into each sound. Next, we apply the approach with the first window starting at t_0 and the second at 0.5 seconds. The difference MD() metric shows how close our set is to reference. This is performed for a selection of our sample set, and considering the top 25 modes in Figure 3 and the top 200 modes in Figure 4. A 8192-point DFT was used.

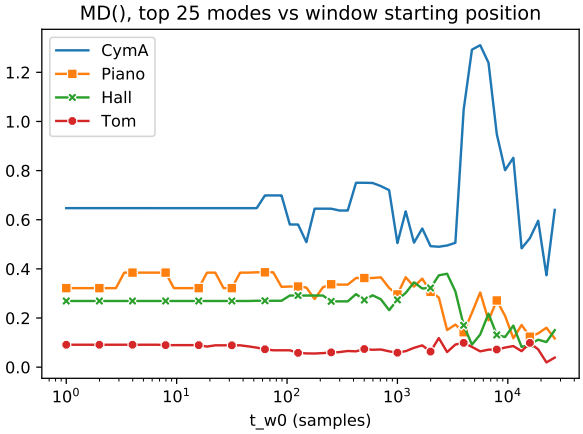

Fig. 3: Effect on mode vector distance metric for the top 25 modes of sliding our starting window over times from 0 to 0.25 seconds. Beyond t=20000 samples, analysis windows start to overlap.

In the first plot, we chose to study the sensitivity of window start position on the most prominent 25 modes. As we progress along the x-axis (moving t_0 towards the second reference window), we do see some variation, which is expected since our sound samples are not completely modeled using steadily-decaying fixed sinusoids. This is overall L2 vector distance for 25-element vectors of frequency in radians (Figure 3), so the variance is not enormous, but we do note there is some sensitivity to where we pick our windows.

We see a behavior of becoming more accurate past the nonlinear portion of sounds (piano/cymbal attacks), which is expected since our reference modes were captured during the steadier tail. There is an exception: CymbalA has a range where it becomes less accurate around 4000-8000. Comparing with the sound and listening, we believe this is related to the highest-energy section of the cymbal attack where nonlinear effects are the strongest, which does not occur at the very beginning of the sound. This makes sense with background literature on plate physics, for exampleChaigne et al. [13].

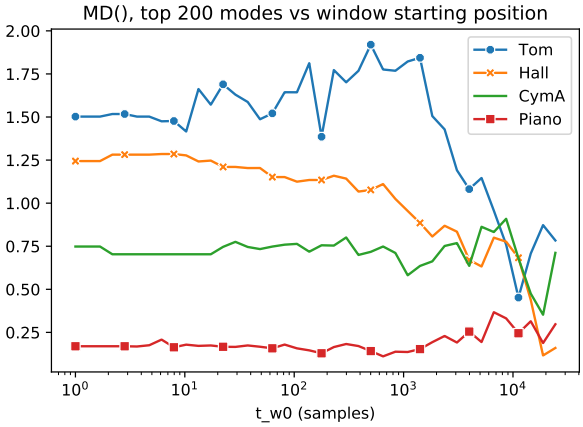

Fig. 4: Same sliding-starting-time experiment, considering a full complement of 200 modes. Beyond t=20,000 samples, analysis windows start to overlap.

In Figure 4, we compare found sets of 200 modes. Our metric shows the tom drum as least accurate, since it does not have that many prominent modes and we are comparing some low-amplitude modes and noise, and as a result absolute distance between modes determined by different approaches harms our overall metric. If we were to use this metric in a closed system, we would wish to remove such rough edges, likely by scaling by amplitude over an equal-loudness curve and zeroing out frequencies below some threshold.

On a positive note, we see the approach behaving accurately and as expected for sources with larger number of modes: rooms, piano, etc.

3.4 Sensitivity to Window Size

Finally, we consider specializing our algorithm for modal reverberators and modally-dense samples. We run the same trial with a 16384-point DFT for three of our modally-dense sources: the three reverbs and CymbalB. Results are in Figure 5.

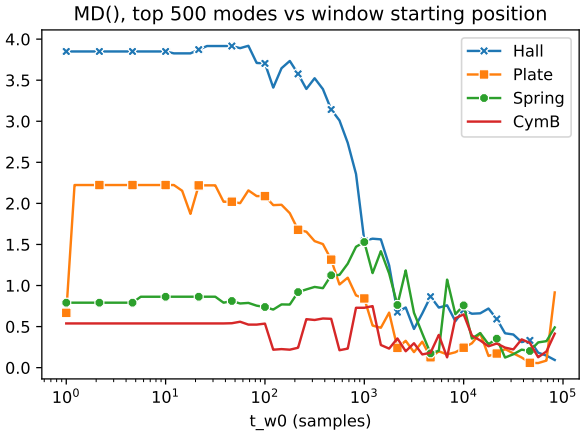

Fig. 5: Use with 2^{14}-point DFT on sources with high number of modes

Note now our analysis window is significantly longer; windows are overlapping by 10,000 samples. The final spike in the Plate line is due to running into silence. The Hall sample qualitatively has an imperfect noise burst vs. a clean pop or impulse at the start of the response. Reliability improves once we're past the artifact, but we note sensitivity to such behaviors.

The spring is an interesting case; there is a section of the impulse similar to the nonlinear cymbal where we introduce new modes past $t = 0$ before settling into the steady-state region of the system. We can synthesize modes based on a window covering this region, but may have simultaneously-playing partials that do not exist simultaneously in the natural response, instead presenting as a frequency shift. Sensitivity to this phenomenon must also be noted.

4 Application to Nonlinear Behaviors

For perfect captures of decaying sinusoids in linear systems, two-window comparison should find our resonant modes. Looking to quickly capture short-term nonlinear behavior in e.g. attacks of percussion instruments, we extend the process to obtain three or more windows, and pair frequencies between pairs of windows based on a metric. Perhaps the most efficient is the mode's magnitude in each response, as we already have the data available from our analysis step. However we note this would immediately break for cross-fading sounds with modes traveling in opposite directions.

Considering the tom-tom sample with a pitch glide up, a plot of the first five modes, after filtering to obtain only local maxima in the spectrum, is presented in Figure 6; darker lines (which in this case correspond to lower frequencies) are more prominent. We note that a direct implementation of the algorithm orders modes such that two mode frequency envelopes travel from high frequencies to low frequencies and back; this behavior can quickly be seen to be incorrect by looking at the spectrogram.

To work around this, we might choose to use a more standard approach such as tracking several windows through a spectrogram. Alternatively we could impose heuristics, for example placing a limit on pitch guide effects when building chains of windows, either forcing a choice of different points, or discarding windows that have too much movement. Either approach may lead to incorrect mode frequency envelopes or a loss of modes present in the output.

Imposing a heuristic of limiting linking windows to a musical interval of a fifth, we obtain the new trajectories in Figure 7; we see that all points follow a more expected envelope. As a result, one set of points from the prior figure no longer met the decay requirements so we do not plot it. Of course, this hand-adjustment is against the goal of avoiding any hand-tuning, so a full system should consider the applicability to the domain of tones or impulses in question. There may be conflicting behaviors to model; note that even when limiting discussion to tom drums, pitch glide effects may go up *or* down depending on whether the bottom resonant head is tuned tighter or looser than the top batter head, respectively.

As another note on heuristics: this section used 8192-point DFTs; the first two windows overlapped by half and the third window followed 400ms after the second when the sound has settled. Note the choice of these constants fits some sound sources - drum set attacks - but we would want a longer set of time constants to capture a cymbal swell, and an even shorter one for string nonlinearities (such as in the piano case). As such, there remains work to be done toward establishing heuristics for processes with three or more windows.

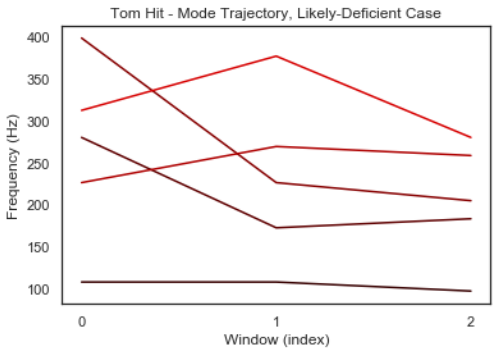

Fig. 6: *Top Mode trajectories for Tom Pitch Glide effect, for three successive windows.*

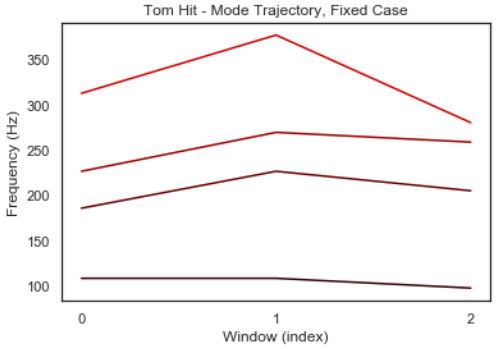

Fig. 7: *Case of Figure 6, with heuristics applied.*

5 Summary

We considered and extended a method for rapidly obtaining modal frequencies and parameters from sample content, for use in modal processors or synthesizers. It is straightforward, can easily be tweaked during offline, interactive analyses when a human is in the loop, but may also be used in scenarios such as fast batch sample processing. Sensitivity analysis was performed, with a metric proposed to capture both strengths and deficiencies of these cases. Overall, while other approaches should be used to obtain a more stable or exact set of metrics when possible, this straightforward approximation can obtain qualitatively-similar results when resources are constrained, especially for situations where modeling the distribution of many modes (room response, cymbal sounds) is more important than getting a few modes exactly correct. Extensions to chains of analysis windows are able to capture weakly-nonlinear behaviors, though development of heuristics is required for robustness.

References

[1] Abel, J. S., Coffin, S., and Spratt, K., "A Modal Architecture for Artificial Reverberation with Application to Room Acoustics Modeling," in *Audio Engineering Society Convention 137*, 2014.

[2] Smith, J. O. and Serra, X., "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proceedings of the 1987 International Computer Music Conference, ICMC; 1987 Aug 23-26; Champaign/Urbana, Illinois.[Michigan]: Michigan Publishing; 1987. p. 290-7.*, International Computer Music Conference, 1987.

[3] Serra, X., *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*, Ph.D. thesis, Stanford University, 1989.

[4] Van Duyne, S. A., *Digital Filter Applications to Modeling Wave Propagation in Springs, Strings, Membranes and Acoustical Space*, Ph.D. thesis, Stanford University, 2007.

[5] Neri, J. and Depalle, P., "Fast Parital Tracking of Audio with Real-Time Capability Through Linear Programming," in *21st International Conference on Digital Audio Effects (DAFx-18)*, 2018.

[6] Kereliuk, C., Herman, W., Wedelich, R., and Gillespie, D. J., "Modal Analysis of Room Impulse Responses Using Subband ESPRIT," in *Proceedings of the International Conference on Digital Audio Effects*, 2018.

[7] Roy, R. and Kailath, T., "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, 37(7), pp. 984–995, 1989.

[8] Badeau, R., Boyer, R., and David, B., "EDS parametric modeling and tracking of audio signals," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, pp. 139–144, 2002.

[9] Maestre, E., Scavone, G. P., and Smith, J. O., "Modeling of a violin input admittance by direct positioning of second-order resonators," *The Journal of the Acoustical Society of America*, 130(4), pp. 2364–2364, 2011.

[10] Maestre, E., Abel, J. S., Smith, J. O., and Scavone, G. P., "Constrained pole optimization for modal reverberation," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, 2017.

[11] via YouTube, A. B., "Spring Impulse from video Fender Deluxe Reverb II spring tank vs Logidy Epsi," `https://www.youtube.com/watch?v=SpSMStt6Vh8`, verified August 30, 2020.

[12] Roos), S. P. E., "Samplicity's Bricasti M7 Impulse Response Library," `http://www.samplicity.com/bricasti-m7-impulse-responses/`, accessed April 1, 2019; readers may need to use the archive.org Wayback Machine or a cache.

[13] Chaigne, A., Touzé, C., and Thomas, O., "Nonlinear vibrations and chaos in gongs and cymbals," *Acoustical science and technology*, 26(5), pp. 403–409, 2005.