Method for Near-Realtime Estimation of Resonant Models

Jeremy L. Wagner
Center for New Music & Audio Technologies
University of California, Berkeley
jeremy.wagner@berkeley.edu

ABSTRACT

We introduce an algorithm for the near-real time automated computation of resonant models for use with the CNMAT Externals resonators~ object for Max/MSP. Unlike prior workflows that focus on precision at the cost of non-real time computation, this method can accurately estimate resonant models in near-real time. Utilizing fractional bin analysis and exponential fitting, analysis of arbitrary input signals is performed with the goal of delivering a fully functional model within fractions of a second of a note onset. We introduce a new Max/MSP object, qrm~, that implements this algorithm and is suitable for analysis and synthesis in live performance contexts. We conclude the paper with an assessment of the model accuracy and discussion of areas for improvement.

1. INTRODUCTION

Experimentation with computer-controlled resonant synthesis has been an active area of inquiry for many decades. [1] The basic idea underlying resonant synthesis is that a bank of IIR filters can be configured in such a way that their combined resonant characteristic approximates that of some sound of interest in the real world. So-called 'resonant models' (comprising amplitude and decay rates for each salient constituent frequency of a sound) when fed to such a bank of IIR filters can be activated with impulses or driven with signals to achieve timbres that are often convincing approximations of real instruments. [2].

In the 1980's and 90's, various approaches were pursued for producing analyses of instrumental sounds and in 1999 the resonators~ Max/MSP object was introduced as part of CNMAT Externals, making resonant modelling synthesis practical for real time performance [3]. Work around resonant modeling persisted through the early 2000's including efforts related to the Sound Description Interchange Format (SDIF)[4].

Throughout this time the production of resonant analyses has been the domain of non-realtime computation. Essentially, due to the complexity of the available analytic algorithms and computation time involved, models tend to be computed ahead of time for later use in performance. This practice has yielded a small number of analysis tools, but as of this writing many of these software applications no longer run on modern computer architectures. Those that are still distributed are notably limited to non-real time analysis [5] and many previously available tools are no longer maintained as software packages.

In the absence of reliable and widely available tools we desire a method for estimating resonance models from arbitrary input signals that (1) functions on modern computer architectures while (2) enabling near-real time, on-the-fly analysis for use in live performance and (3) must produce realistic sounding, usable results. With these design constraints in mind, we have produced a Model Estimation Algorithm implemented as a prototype analysis object for Max/MSP.

2. THE MODEL ESTIMATION ALGORITHM

Our goal is to devise an automated method for extracting a resonant model from a signal vector of arbitrary size that can be implemented in near-real time. We readily acknowledge that the algorithm presented here is optimized for fast execution and, therefore, may suffer some degree of inaccuracy. However, we have taken pains to ensure that this algorithm produces perceptually useful and convincing results.

Presented in the abstract, we consider an audio signal vector under test, S, that contains an onset of some instrumental timbre. For the sake of simplicity, we assume that just a single onset is contained within S and that the signal tends to decrease in amplitude with time.

2.1 Attack Detection

Working under the assumption that the attack of the instrumental timbre produces a peak amplitude within S, the attack detection is simply performed. [1] We extract the sample index of our onset, t, by finding the maximum amplitude within the signal vector under test.

$$t = \arg\max |S| \tag{1}$$

2.2 Resonant Frequency Extraction

For a given resonant model, we seek to extract a timbre and describe its evolution in time as an exponentially decaying process. Our timbre can be described as a vector of sinusoidal components of characteristic frequencies, f_i, and amplitudes, a_i, that decay exponentially according to coefficients b_i.

[1] A more precise attack detection and segmentation method would include taking the Manhattan distance between successive FFT frames to identify attack onsets, but we have found that the peak amplitude suffices for the present purposes while minimizing computation time.

Thus, a resonant model, M, as a function of time, t, approximates the function

$$M(t) \equiv \sum_{i=0}^{n} a_i \exp(b_i t) \cos(2\pi f_i t) \tag{2}$$

up to some number of sinusoids, n, as determined below in Eq. (5).

Extracting the frequencies and amplitudes, of course, is the domain of Fourier Analysis. Naively approaching the task of model estimation, one might consider simply taking a Discrete Fourier Transform (DFT) of the signal under test and using the extracted sinusoidal weights as an approximation for our timbre. However, practical considerations prevent us from using simple DFTs in the present application.

Unlike DFT/FFT analysis, which deconstructs a signal into regularly distributed sinusoids of equally-spaced frequency bins (in the case of the FFT, these bin center frequencies correspond to simple harmonic divisions of the window size) resonant models are very sensitive to frequency accuracy. The model we extract must resonate with the exact frequencies present in the signal under test to serve as a convincing model. The bins of the DFT/FFTs with small window sizes needed for near-realtime analysis are simply too coarsely distributed to provide the needed accuracy.

By way of example, given the constraint that our algorithm is to function in near-real-time, the DFT window must be sized to practical limits. At a sampling rate of 48 kHz and a design constraint of a 30ms execution time and FFT window sizes of 2^n samples, our maximum FFT window size will be 1,024 samples, yielding a frequency bin size of 46.875 Hz. This is far too low a resolution for the present application, particularly regarding lower frequency components of the timbral models we wish to estimate. Accepting this amount of frequency inaccuracy yields very poor resonant models that are only acceptable in upper frequency ranges where the frequency bin size falls below the perceptual limit of 0.06 semitones (6 cents). In the above example, the frequency estimates would be perceptibly inaccurate below ≈13.5 kHz, which is obviously unacceptable for use in musical applications. [2]

The solution that we have implemented in our algorithm is the method of Fractional Bin Analysis by Gaussian Interpolation. To motivate this method, one considers the DFT of a single sinusoid. If the sinusoid exactly matches the frequency (and phase) of a given DFT bin, most of the energy will be represented as an amplitude spike in that DFT bin. (See Fig. 1).

However, provided that our complex sinusoid is of any other frequency than one of the DFT bin center frequencies, the DFT will exhibit significant bin leakage (i.e. measured energy in bins adjacent and surrounding the bin which contains our single sinusoid frequency). The amount of this bin leakage is *maximized* when the sinusoidal frequency is the average of two successive bin center frequencies (see Fig. 2) and is *minimized* when the sinusoid lies on a bin center frequency.

[2] Here, we assume 6 cents as a rough estimate of the lower limit of frequency discrimination for trained musicians. The lower bound on frequency, f, is calculated as $f = \frac{\Delta f}{2^{0.06/12}-1}$, where $\Delta f = 46.875$ Hz.

Moreover, the energy distribution of the DFT bins will be lopsided toward the actual frequency of the sinusoid under test. One can take advantage of this bin leakage to improve the estimate of the exact frequency of the sinusoid that generated the DFT signature. A derivation of this method is beyond the scope of this paper, so the reader is referred to [6] for a complete treatment of the subject.

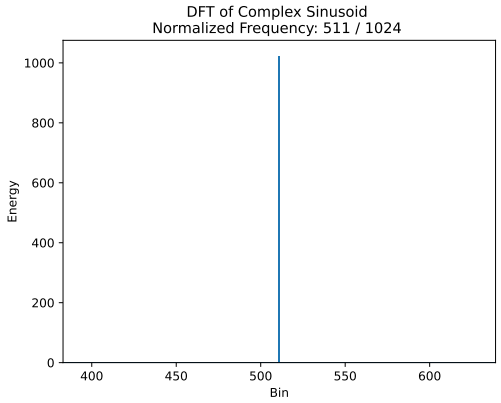

Figure 1. DFT detail of single complex sinusoid with frequency matching that of bin 511.

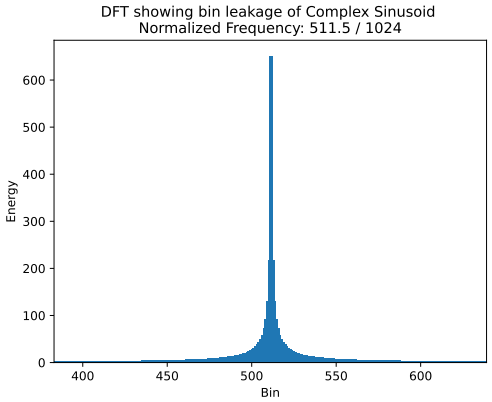

Figure 2. DFT detail showing bin leakage of single complex sinusoid with frequency halfway between those of bin 511 and 512.

We can leverage this bin leakage to improve the estimates provided by DFT/FFT analyses with smaller window sizes. To begin, given our input signal, S, indexed by variable n, the DFT is defined as

$$X_k = \sum_{n=0}^{N-1} S_n e^{\frac{-2\pi i k n}{N}} \tag{3}$$

We need to extract the spectral peaks. For this application we are interested in only those most salient spectral peaks, p_i, so we first identify the DFT bin with the greatest energy as p_0

$$p_0 = \max |X_k| \tag{4}$$

We then identify a set of peak DFT bins, $p\{k_0, ..., k_n\}$, within a certain amplitude range, α, below p_0. (In our implementation, α is determined as a user-specified decibel value relative to the amplitude of p_0, but here, for simplicity, we can regard it as a simple amplitude value)

$$p\{k_0, ..., k_n\} \ni |X_k| \geq |p_0| - \alpha \tag{5}$$

For each of the identified peak bin indexes $k_n \ni p$, we perform a fractional bin analysis by Gaussian Interpolation to extract a refined frequency estimate for the sinusoidal component that generates that DFT signature. The result is an array, δ, whose elements represent correction factors of the bin center frequencies of p, defined in [6] as

$$\delta_k = \frac{\ln\left(\left|\frac{X_{k+1}}{X_{k-1}}\right|\right)}{2\ln\left(\frac{\left|X_k^2\right|}{\left|X_{k+1}\right|\left|X_{k-1}\right|}\right)} \tag{6}$$

The elements of δ are calculated for each of the spectral peaks in p and are used to generate refinements – so called 'fractional-bin' numbers. That is, for a bin index, k, and a correction factor, δ_k, the corrected bin center frequency that represents the likely frequency of our sinusoidal component is represented by $\Delta f(k + \delta_k)$, where Δf is the bin width of the DFT. As such, corrected frequency estimates for our peaks, p, are given by

$$f_i = (p_i + \delta_i)\frac{f_s}{N} \tag{7}$$

where f_s represents the sampling rate and N is the size of our DFT window. The elements of f represent the refined frequency estimates for sinusoidal constituents of our timbre.

2.3 Initial Amplitude Estimation

In the development of our model estimator, it has not proven necessary to refine the initial amplitude estimates of the note onset to the same degree of accuracy performed for the frequency estimates. As such, our implementation uses the amplitudes of the raw DFT bins. These amplitudes will always be less than actual sinusoidal amplitudes, which is a benefit in conferring stability on the resulting resonant model. Furthermore, the exponential fitting in the subsequent step helps to average out any error in these values, so the additional precision is not worth the computational cost. Future improvements to the algorithm may revisit this question, but for now, the amplitude values of the raw DFT bins are deemed sufficient for the present purpose. Thus, our amplitudes are taken as

$$a_i \approx X_i, i \ni p \tag{8}$$

2.4 Decay Rate Estimation

For each frequency in our resonant model, we need to estimate the decay characteristic of that frequency over time. Here, we assume that the spectral peaks in our signals under test will be strictly decreasing in amplitude over the signal's time evolution. This condition, while actually false, is roughly the case for a large class of musical timbres (plucked/struck strings, staccato wind instrument attacks, etc.) [3] Whatever the resonant system we are trying to model, we are interested in the way that the amplitude of each of the constituent resonant frequencies falls off over time. Since many resonant systems exhibit exponential decay of their resonant frequencies, we assume the decay to be exponential—at least in relation to a single impulse. Indeed, CNMAT's resonators~ object is built around this same assumption. We can describe this decay mathematically by

$$a_t = a_0 \exp(-Bt) \tag{9}$$

where a_t is the amplitude of a given frequency at time, t, the initial amplitude of that frequency, a_0, is the amplitude measured at the onset of the sound. Thus, we are interested in calculating the decay rate, B, for each of the constituent frequencies in our resonant model estimate. To do so, we need to measure the amplitude of the identified peak frequencies, f_i, of our signal attack at successive moments in time and fit an exponential curve to those values.

Given our array of peak DFT bins, p, we can perform a series of DFT's at successive moments in time, t, to derive amplitude values for each of the spectral peak estimates identified earlier.

$$a(p_k, t) = |X_{k,t}| \tag{10}$$

For each frequency, f_i, in our resonant model, we can then use these values, $a(p_k, t)$, to estimate an initial amplitudes, A, and decay rates, B using an exponential fitting. [4]

$$A_i = \exp\left[\frac{\sum_{j=1}^{n}\ln a_{i,j}\sum_{j=1}^{n}t_j^2 - \sum_{j=1}^{n}t_j\sum_{j=1}^{n}t_j\ln a_{i,j}}{n\sum_{j=1}^{n}t_j^2 - (\sum_{j=1}^{n}t_j)^2}\right] \tag{11}$$

$$B_i = f_s\left[\frac{n\sum_{j=1}^{n}\ln t_j a_{i,j} - \sum_{j=1}^{n}t_j\sum_{j=1}^{n}\ln a_{i,j}}{n\sum_{j=1}^{n}t_j^2 - (\sum_{j=1}^{n}t_j)^2}\right] \tag{12}$$

where n is the number of time values (number of DFTs offset by t_n). Times t_n are indexes in the signal, S, relative to the signal onset, and $a_{i,j}$ are the amplitudes of frequency f_i at sample onsets $t_0 \ldots t_j$, and f_s is the sampling rate.

In our implementation, we have found that sampling the waveform at just $n = 5$ points along its decay produces excellent results, provided that we must implement a few safeguards and refinements to Eqs. (11) & (12). With such a small number of sample values and in the limits of floating-point arithmetic it is quite common for this exponential fitting to generate values for A and B that tend

[3] Notable, important exceptions include piano notes, which (as discussed in Sec. 4) exhibit more complex time evolutions and cannot be considered to decay purely exponentially for all partials. For this reason, in implementation, some corrections must be made to avoid decay-rate values that would cause the model to become unstable.

[4] Eqs. (11) & (12) adapted from [7]

to infinity, which would break our model. Two strategies have been employed to avoid these errors. First, the amplitude values $a_{i,j}$ are incremented by an infinitesimal ϵ to avoid the possibility of the $\ln a_{i,j}$ terms producing $-\infty$. Second, we disproportionally weight the initial amplitude values (those identified by Eq.(8)) to ensure the fitted exponential curve gives an accurate representation of the onset amplitude for each of the frequencies. The resulting exponential decays provide an excellent approximation of the decay characteristic of the target sound.

2.5 Compiling Model Parameters

To complete the resonant model, M, we compile the frequency, amplitude, decay-rate triples into a vector that can be passed directly to the CNMAT Externals resonators~ object.

$$M \equiv \{f_0, A_0, B_0, ..., f_m, A_m, B_m\} \tag{13}$$

This vector of values configures the bank of resonant filters in resonators~ to behave in a manner similar to the resonant characteristics of the target timbre represented by S.

3. IMPLEMENTATION

The Model Estimation Algorithm as described thus far was first prototyped in Max in 2021, utilizing pfft~ analysis and list processing in the ODOT expression language. This prototype is now ported to a new Max object, written in C and based on the max-sdk is called qrm~. In both cases, the algorithm is designed to interact with signals recorded into named buffer~ objects. Since the intended use-case is to only occasionally generate resonant models on an as-needed basis, it seemed pragmatic to program the object as a pure Max (non-MSP) object. This simplifies the complexity of the implementation while limiting computational load.

The qrm~ object accepts a number of parameters. It is instantiated with an argument specifying the buffer~ object it will address (also settable with a **set** message). The **set_thresh** message allows the user to specify the threshold amplitude (in decibels) relative to the peak partial amplitude. Partials below this amplitude are rejected. In essence, this allows the user to tune the amount of detail to include in the model. Lastly, the **fft_size** message allows the user to specify FFT window sizes up to 65,536 to tune the analysis to better represent lower frequency material. Despite this large window size, since the object is not tied to the MSP scheduler, the analysis of even the largest available windows is typically executed in a fraction of a millisecond.

When the object receives a single integer value on input it will produce the fractional bin spectrum analyzed at that integer index within the target buffer. The output of the object is therefore a suitable input for the CNMAT Externals sinusoids~ object. Since sinusoids~ can be reconfigured in real time, this feature essentially allows the user to scrub around a target buffer by index, thereby allowing the easy implementation of accurate time-stretching, spectral freezes and related effects.

When the object receives a 2-element list of integers it interprets these as the limits of a signal window and performs the full model estimation algorithm with n=5 FFTs for the decay rate estimation spaced evenly between the derived attack onset and the end of the window. This causes the object to produce the full resonant model of that buffer region as a vector of frequency/amplitude/decay-rate triples, suitable for direct input to resonators~. Passed to a resonators~ object and activated with an impulse or signal, the model can then be performed live.

4. ASSESSMENT & CRITIQUE

In initial testing of the algorithm implementation, results have been surprisingly favorable. The algorithm is perfectly capable of producing models that, on first listen, are nearly indistinguishable from the signals that generate them. The reader is encouraged to download and audition the examples at [8].

As an illustration we have prepared the following spectral plots of a plucked banjo note pitched at A#3 and the resonant model based on our analysis performed with a window size of 2,048 samples.

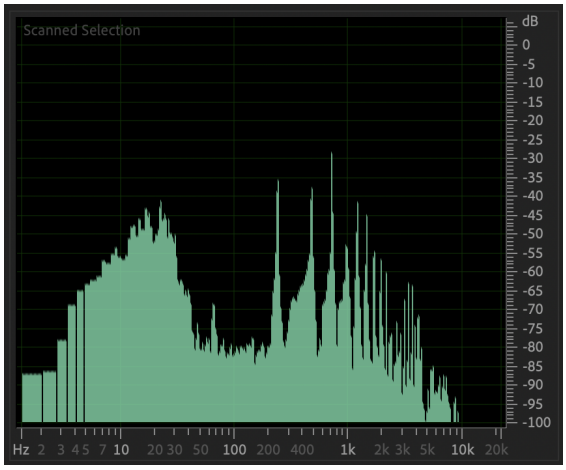

Figure 3. Spectrum of banjo A#3. Analysis window size: 65,536. Performed in Adobe Audition

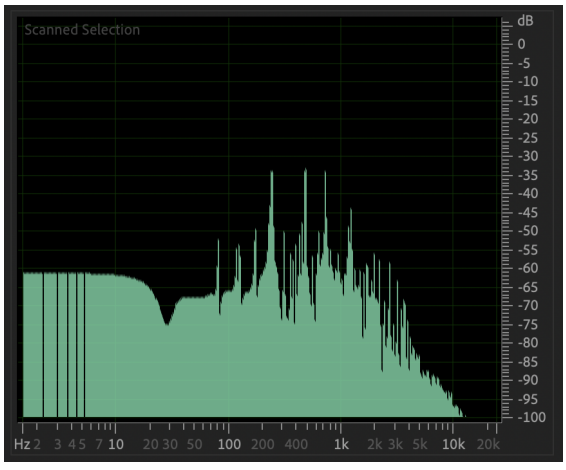

Figure 4. Spectrum of resonators configured with an analysis of the same banjo note A#3 and activated by a Dirac impulse. Analysis window size: 65,536. Performed in Adobe Audition

Obviously, there are detectable differences between these two spectra, but the similarities are equally compelling. The first 3 spectral peaks of the original sound (see Fig. 3; 233 Hz, 464 Hz, & 701 Hz) are very closely approximated by those of the resonant model (see Fig. 4; 234 Hz, 464 Hz & 698 Hz). The peak at 16 Hz in the target sound (a visible modulation in the source sample) is notably less pronounced as a definite peak in the spectrum of resonant model, but energy in that subsonic octave is nevertheless represented. Qualitatively, the resonant model emphasizes certain non-harmonic frequencies which lead to a notably more metallic timbre. Nevertheless, the connection between the original sound and the resonant model is very convincing in the author's opinion.

The design constraints and methods that speed computation of these models do have their shortcomings. One of the trade-offs/limitations of the fractional bin analysis is that the model output is limited to, at most, one sinusoidal component per DFT bin. As such, timbres with closely spaced spectral components—particularly in low frequency ranges—still require large DFT sizes to yield the desired spectral density. This shortcoming is illustrated in Fig. 4. Note the comparative lack of detail below 100 Hz, which, with a window size of just 2,048 samples is represented by, at most, 5 frequencies in our model. This shortcoming can be partially ameliorated by choosing a larger analysis window size.

There are also limitations on the types of timbres that can be represented by this type of analysis. As early as 1986 Potard, Baisnée & Barriere discuss "resonances [for which] the amplitude evolution ... cannot be considered as exponential" [1]. Indeed, one need only consider the time evolution of a piano note spectrum to concede that the time evolution of a single frequency isn't strictly decreasing. Rather, energy from higher frequency vibratory modes is redistributed with time to lower frequency modes and this yields complex envelopes for lower partials. With such complex behaviors, no exponential fitting is valid and the time evolution of the resulting resonance model will not provide realistic results.

Lastly, though this implementation can very precisely model the resonant decay characteristic of a sound, the models it generates still need to be activated appropriately. Experimentation with different driving functions for these models will produce varying degrees of realism depending on the type of sound being modeled. That is, a model can sound convincing or synthetic based entirely on the function with which it is activated. [5] This implies opportunities for future work in modeling onsets and also opens up a space of play for the composer, who is free to explore a range of activation functions as a compositional parameter. These limitations and opportunities invite the further development of resonant model synthesis in combination with other methods and strategies.

5. CONCLUSION

Resonant modeling synthesis remains a rich space for exploration, even at this late stage in its history and, in this author's opinion, it is a technique that deserves a revival. At the present moment there is a need to revive some of the tools and capabilities that once existed in this space, but have fallen out of service due to un-serviced technical debt. The work described in this paper represents an attempt to re-invigorate interest in a rather dormant field. Yet, it is a topic that, we humbly suggest, deserves reconsideration.

[5] The reader is directed to [8] for illustrative examples.

Acknowledgments

The author wishes to thank Alois Cerbu and Luke Dzwonczyk for their review and helpful comments on original drafts of this paper, Reviewer #3, whose detailed critique of this paper has improved it, and to Adrian Freed and David Wessel without whose prior work this project would not be possible.

6. REFERENCES

[1] Y. Potard, P. Baisnée and J.-B. Barriere, "Experimenting with Models of Resonance Produced by a New Technique for the Analysis of Impulsive Sounds," in International Computer Music Conference, 1986.

[2] IRCAM, "Diphone Studio [Computer Software]," Paris, 2020.

[3] T. Jehan, A. Freed and R. Dudas, "Musical Applications of New Filter Extensions to Max/MSP," in International Computer Music Conference, Berkeley, 1999.

[4] M. Wright, R. Dudas, S. Khoury, R. Wang and D. Zicarelli, "Supporting the Sound Description Interchange Format in the Max/MSP Environment," in ICMC Proceedings 1999, 1999.

[5] J. Lochard and F. Cornu, "MaxModRes 0.21 [Computer Software].," IRCAM, Paris, 2022.

[6] M. Gasior., "Improving FFT Frequency Measurement Resolution by Parabolic and Gaussian Spectrum Interpolation.," in AIP Conference Proceedings 732, Geneva, Sw, 2004.

[7] E. W. Weisstein, ""Least Squares Fitting – Exponential."," 9 February 2025. [Online]. Available: https://mathworld.wolfram.com/LeastSquaresFittingExponential.html.

[8] J. Wagner, "qrm∼", 21 Apr 2025. [Online]. Available: https://github.com/CNMAT/qrm_tilde