Quantifying the Efficiency of Autoregressive Generators of Power Law Noise

Benjamin N. Taft[1]

Landmark Acoustics LLC, Racine, WI 53405, USA[a]

(Dated: 24 September 2020)

Power law noise, in which the energy at each frequency is related to the power of that frequency, is ubiquitous in nature. Signal detection algorithms, therefore, should often be tested in the context of noise with different "colors", or exponents, of power law noise (PLN). These noise colors range from "red," when the frequency exponent is -2, through "white," when the exponent is 0, to "blue," which has an exponent of 2. There are many algorithms for generating realistic PLN. Generators that use autoregressive (AR) functions are particularly useful because they are reasonably intuitive and can be parameterized to generate the full range of PLN "colors." In principle, the degree, or number of terms, in the AR generator for any color of PLN may be infinite. In practice, there are theoretical caps on the degree for some exponents, and diminishing returns for increasing the degree for others. This article demonstrates quantitative decision rules for choosing the degree for a discrete AR generator of PLN based on the size of the noise vector being generated and the tolerence its user has for slight deviations between the realized power law and the desired exponent.

[a] ben.taft@landmarkacoustics.com

I. INTRODUCTION

Power law noise is a time series where the energy at different frequencies is a function of the frequency raised to a power. True red noise has a power of -2, true white noise has a power of 0, and true violet noise has a power of 2. Pink noise has a power between -2 and 0, exclusive. Blue noise has a power between 0 and 2, exclusive. There are many ways to generate power law noise. A particularly useful one uses an autoregressive model where the weights given to previous values of the time series are a function of the power and a decreasing function of the time lag to the term. Some powers have a fixed number of non-zero terms. White noise, for example, has no non-zero terms, and true red noise has only one. All other powers have a potentially infinite number of terms. The actual number of terms in an autoregressive model is its degree. Obviously, there are diminishing returns to increasing the degree of a model. This project is aimed at finding a rule for choosing how big a model's degree should be. The rule will be quantified according to the difference that one is willing to tolerate between the observed spectral slope and the expected slope of the spectrum of the time series.

There are three crucial parameters that I want to examine:

α**:** The power of the power law. When the spectral energy at a frequency f is $S(f)$ then

$$S(f) \propto f^{\alpha}.$$

N**:** The number of samples in the time series

K**:** The degree of the autoregressive model

It is useful to be able to generate power law noise in a controlled way. Power law noise shows a roughly linear relationship between the logarithms of frequency and amplitude. That is to say, the distribution of energies (P) across frequencies (f) is governed by a parameter (α) such that $P \propto f^{\alpha}, \alpha \in \mathbb{R}$. By analogy with mixtures of light energy, the slope of the relationship between frequency and amplitude in a sample of power law noise is often referred to by color. Red noise has a slope, or α, near -2, pink noise's slope is -1, white's is 0, blue's is +1, and violet's is 2. Natural sounds and speech often occur in an environment of noise with an energy distribution somewhere between pink and white. That is, there is more energy in the lower frequencies of these environments' background noise. Room acoustics have pink power law noise because of reverberation (Summers, 2013). Aquatic acoustic environments are filled with pink noise from both shipping (Helble *et al.*, 2012) and natural processes. Algorithms for detecting features of speech or animal sounds should therefore be examined in the presence of power law noise (Skowronski and Fenton, 2008).

Feature extractors should have well-understood responses to the kinds of power law noise that they are likely to encounter in the wild (Beckham and Austin, 1975). One way of doing this would be to mix test sounds with natural recordings of power law noise. This has the disadvantage of depending on the availability and quality of noise recordings. Any such recordings also necessarily conflate the frequency content of the background noise with the frequency response of the recording system. It can also be difficult to predict how well an algorithm that has been trained to work in specific recordings of background noise will generalize to different real-world situations. Using artificially generated noise can solve these problems because the user can dictate an algorithm's parameters to match the full range of cases expected in the real world.

Published algorithms that generate discrete-time power law noise seem to fall into two approaches. One approach, is essentially a finite impulse response (FIR) filter (Kasdin and Walter, 1992). It uses two discrete Fourier Transforms to manipulate a batch of samples in the frequency domain (Zhivorimov, 2018). That is,

$$\mathbf{x} = \mathcal{F}^{-1}\left\{\mathbf{h}\,\mathcal{F}\{\mathbf{n}\}\right\}, \tag{1}$$

where $\mathbf{x}$ is a vector of power law noise, $\mathcal{F}\{\cdot\}$ is the discrete Fourier transform, $\mathbf{h}$ is a vector that describes the desired distribution of energy in the output noise, and $\mathbf{n}$ is a vector of random values from the standard normal distribution. An FIR approach therefore requires approximately $O(N \log N)$ calculations to generate N samples of power law noise. This makes the FIR approach extremely efficient (Zhivorimov, 2018). However, it produces its noise in batches of fixed length. If you do not know how many samples you need beforehand, or if N is extremely large, the FIR approach may not be practical. Another approach, uses autoregression (AR) models to generate new noise samples as a function of preceding samples (Pachet *et al.*, 2015). That is,

$$x_i = n_i + \sum_{k=1}^{K} a_k x_{i-k}, \tag{2}$$

where $\mathbf{a}$ is a vector of coefficients from an AR generator of degree K. Thus an AR generator of degree K requires approximately $O(NK)$ calculations to generate N samples of power law noise. Consequently, the AR approach takes more computations than the FIR approach when $K > \log N$. However, the AR may still be better-suited to online applications because it generates power law noise one sample at a time, rather than all at once.

One AR generator of discrete power law noise is described by Kasdin and Walter (1992), and calculates each term as a ratio of rising factorials of $\frac{\alpha}{2}$ and k:

$$a_k = \frac{(\frac{\alpha}{2})_k}{k!} = \frac{\prod_{i=0}^{k-1} \frac{\alpha}{2} + i}{k!} = \frac{\Gamma(\frac{\alpha}{2} + k)}{\Gamma(\frac{\alpha}{2})k!} = \frac{1}{k\mathrm{B}(\frac{\alpha}{2}, k)}. \tag{3}$$

Kasdin (1995) points out a convenient recursive way to generate the generator's terms:

$$a_k = \prod_{i=1}^{k} \frac{\frac{\alpha}{2} + i - 1}{i}. \tag{4}$$

There should be diminishing returns to adding more terms to an AR generator, but the rate at which the returns decrease should depend on the value of α. The minimum efficient degree of an AR generator depends on the exponent of the power law noise being generated because different exponents correspond to generators that make use of different amounts of past information. Brownian noise, when $\alpha = -2$, only requires a degree of one because $a_1 = -1$, and a_k is zero otherwise. This makes sense because Brownian noise is made by adding a new sample of white noise to a running total of previous samples. Similarly, white noise, when $\alpha = 0$, is a zero-degree generator because $a_1 = 0$. This also makes sense because white noise doesn't use any previous information, just a new sample of Gaussian noise. However, for violet noise, when $\alpha = 2$, there is no obvious cutoff because $a_k = 1$ for all k. This might correspond to noise from an infinite feedback loop These observations suggest that the minimum effective degrees of the generator should be a function, $\hat{K}(\alpha)$, defined on $-2 \leq \alpha < 2$. Furthermore, the function should be continuous, pass through $(-2, 1)$ and

$(0,0)$, have a local maximum somewhere in $-2 < \alpha < 0$, a local minimum at $\alpha = 0$, and then increase towards a vertical asymptote at $\alpha = 2$.

Kasdin demonstrated a recursive formula for creating a discrete AR generator of power law noise (Kasdin and Walter, 1992). The number of terms in the generator, K, is its degree. The nth term of a discrete autoregressive generator for noise with a power law of α is

$$h\left(\alpha, n\right) = \frac{\left(\frac{\alpha}{2}\right)_n}{n!} = \frac{\prod_{k=0}^{n-1}\left(\frac{\alpha}{2} - k\right)}{n!} = \frac{\Gamma\left(\frac{\alpha}{2} + n\right)}{\Gamma\left(\frac{\alpha}{2}\right) n\Gamma\left(n\right)} = \frac{1}{n\mathrm{B}\left(\frac{\alpha}{2}, n\right)}. \tag{5}$$

There are diminishing returns to increasing the degree of the generator, and this paper looks for a way to compute the smallest appropriate degree for any particular application of the generator. To quantify 'smallest appropriate degree,' I will compare the expected log-log slope of the noise's spectrum, α, to the actual slope, m. The closer the match between α and m, the better the generator, so the quantity we are trying to minimize is $|\alpha - m|$, or, equivalently, $\sqrt{(\alpha - m)^2}$.

We control four things when we generate power law noise: the exponent of the power law, α, the degree of the generator, K, the length of the time series output, N, and the amount of error that we are willing to tolerate, ϵ. For the applications that I am considering, α, N, and ϵ will be driven by the question under investigation. By minimizing K within the context of those constraints, we can minimize our use of time and/or computational resources.

I expect the error for a particular parameterization of the generator, $E_\alpha(N, K)$ to asymptotically decrease as N and K increase. A useful equation for describing that relationship is

$$E(\alpha) = B_0(\alpha) + \frac{B_1(\alpha)}{\sqrt{N}} + \frac{B_2(\alpha)\sqrt{N}}{K} \tag{6}$$

The key insight is **Degree Ratio**, which is $\frac{K}{\sqrt{N}}$. This has really nice properties. First, its limit is straightforward:

$$\lim_{K\to\infty} E(\alpha) = B_0(\alpha) + \frac{B_1(\alpha)}{\sqrt{N}} \tag{7}$$

which means we can use epsilon-delta to look for a degree that is arbitrarily close to this limit:

$$\lim_{\delta\to 0} \frac{B_2(\alpha)\sqrt{N}}{K+\delta} < \epsilon \tag{8}$$

which simplifies to:

$$B_2(\alpha)\sqrt{N} < \epsilon(K+\delta) \tag{9}$$

$$B_2(\alpha)\sqrt{N} < \epsilon K + \epsilon\delta \tag{10}$$

$$B_2(\alpha)\sqrt{N} < \epsilon K \tag{11}$$

$$\epsilon K > B_2(\alpha)\sqrt{N} \tag{12}$$

$$K > \frac{B_2(\alpha)\sqrt{N}}{\epsilon} \tag{13}$$

II. METHODS

A. Simulations

I used a custom Python module to simulate multiple realizations of the power law generator in which I varied the exponent of the power law (α), the length of the output series (N), and the degree of the autoregression generator (K). The values of α ranged from -2 to 2, with an interval of 0.02. As described above, I was testing the hypothesis that the exponent (α) and degree ratio ($K/\sqrt{N}$) of an AR generator are the key quantities for determining its optimal degree. For that reason, the output size and degree of each simulation were not independent. I wanted to test degree ratios between 1/20 and 20. This meant that the smallest N was near 400, the smallest number for which a degree ratio of 1/20 is at least one. The largest N was near 5000, which I chose arbitrarily as the largest output size that I would deal with. The actual values of N that I used were the integers in the range [400, 5000] that were composite multiples of only 2 and 3. The actual output sizes consisted of twenty-seven values between 432 ($2^4 3^3$) and 4608 ($2^9 3^2$). The degrees thus ranged from 1 to 416 when $N = 432$ and from 1 to 1358 when $N = 4608$. I used twenty iterations for each combination of exponent, output size, and degree for a total of 14,853,900 simulations.

In each simulation, I computed the frequency spectrum of each series, and then found the slope (called m), of the best-fit line through its log-log representation. I calculated each simluation's error as the absolute value of the difference between the its observed power law and its α value.

$$E = \left| m - \alpha \right| . \tag{14}$$

The final data set contained the values Exponent (α), Size (N), Degree (K), Slope (m), and Slope Error (E) for each simulation.

B. Model Fitting

The final goal of this study is to find a formula for calculating an efficient degree for an autoregressive generator of power law noise. The inputs to this formula are the models exponent (α), its size (N), and the amount of error that the user is willing to tolerate (ϵ).

Using the simulations described above, the first step toward finding this formula is to find an expression that predicts the absolute difference between the desired power law (α) and the observed power law (m), or the generator's slope error (E). I fit a generalized linear model of E for each value of α in the simulations, using the same formula each time:

$$E \sim B_0 + B_1 \frac{1}{N} + B_2 \frac{\sqrt{N}}{K} + e \tag{15}$$

The models each contained four terms: an intercept term, an inverse size term, an inverse degree ratio term, and an error term. Using the `glm` method in `R`, I treated the error term as Gamma distributed, with a log linking function (R Core Team, 2020). Based on the algebra above, the key to estimating the most efficient degree of a generator is the inverse degree ratio term.

The second step toward finding a formula for an efficient degree is to create a model that estimates the inversed degree ratio term as a function of a generators exponent. There previous step yielded 201 estimates for $B_2(\alpha)$, just one for each exponent value. I used `R`'s `lm` function to fit polynomial regressions of the form

$$B_2(\alpha) \sim \sum_{i=0}^{20} \beta_i \alpha^i \tag{16}$$

I used `R`'s `step` methods to find the most parsimonious combination of coefficents. I fit two separate polynomials, one for pink noise ($\alpha <= 0$), and one for blue noise ($\alpha >= 0$).

III. RESULTS

DISCUSSION

A time series is called power law noise when its future values have some randomness to them but also depend on past values from the time series. A signal is power law noise with exponent α when the energy of its frequency spectrum, $S(f)$, is proportional to f^α at each frequency f. On a log-log plot, a best-fit line through such a spectrum has a slope of α.

The excellent thing about this is that it comes back to degree ratio:

$$\frac{K}{\sqrt{N}} = \frac{B_2(\alpha)}{\epsilon} \tag{17}$$

I can estimate the coefficients of equation 6, $\mathbf{b}_\alpha$, with an empirical approach.

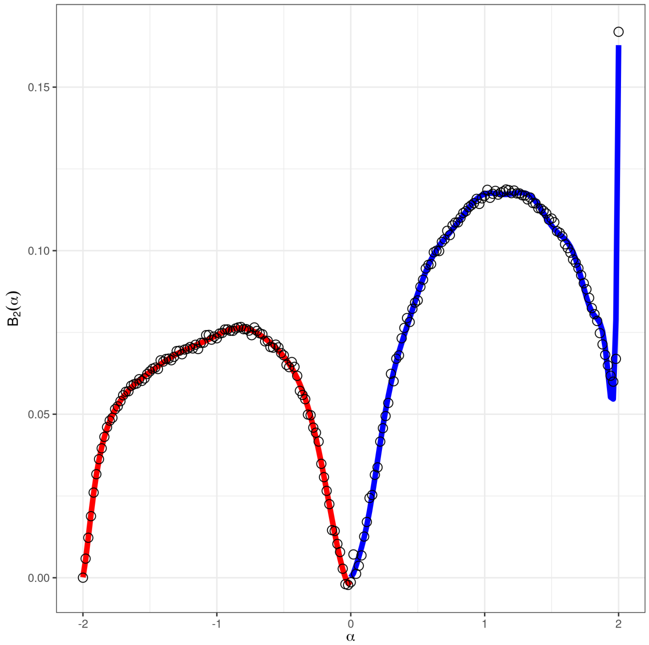

FIG. 1. The coefficients of the inverse Degree Ratio ($\sqrt{N}/K$) terms from the models that relate the Slope Error of simulated PLN to the size and degree of the generator.

TABLE I. The coefficients to the polynomial fit between a "pink" noise generator's exponent ($\alpha <= 0$) and the Degree Ratio term in the model of that generator's Slope Error.

Exponent	Coefficient
Intercept	-0.00227314914613336
2	1.63031943883604
3	4.98482144802658
4	4.97536531055866
7	4.21645358084128
10	20.0002829702428
11	36.311281476006
12	27.0101728835473
13	8.52755459477657
15	-0.531541800533245
17	0.031440622365535
20	0.000291974320416611

ACKNOWLEDGMENTS

This research was supported by Landmark Acoustics LLC.

TABLE II. The coefficients to the polynomial fit between a “blue noise generator’s exponent ($\alpha > 0$) and the Degree Ratio term in the model of that generator’s Slope Error.

Exponent	Coefficient
Intercept	9.4196455046796e-05
2	4.41910305885241
3	-79.5386727156783
4	818.799633843592
5	-4841.87374846092
6	17902.4307608028
7	-44065.3564402454
8	74941.2141298672
9	-89603.5421646984
10	75260.5594029616
11	-43364.831286192
12	16030.1475015981
13	-3098.55090428394
15	100.084618948057
17	-3.86836146896404
20	0.0236495550796521

Beckham, L. W., and Austin, M. E. (**1975**). “Invariant gauss–gauss detection in colored noise,” The Journal of the Acoustical Society of America **57**(3), 747–749.

Helble, T. A., Ierley, G. R., D’Spain, G. L., Roch, M. A., and Hildebrand, J. A. (**2012**). “A generalized power-law detection algorithm for humpback whale vocalizations,” The Journal of the Acoustical Society of America **131**(4), 2682 – 2699.

Kasdin, N. J. (**1995**). “Discrete simulation of colored noise and stochastic processes and 1/f/sup/spl alpha//power law noise generation,” Proceedings of the IEEE **83**(5), 802–827.

Kasdin, N. J., and Walter, T. (**1992**). “Discrete simulation of power law noise (for oscillator stability evaluation),” in *Proceedings of the 1992 IEEE Frequency Control Symposium*, pp. 274–283.

Pachet, F., Roy, P., Papadoupoulos, A., and Sakellariou, J. (**2015**). “Generating $1/f$ noise sequences as constraint satisfaction: the Voss constraint,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2482–2488.

R Core Team (**2020**). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Skowronski, M. D., and Fenton, M. B. (**2008**). “Model-based detection of synthetic bat echolocation calls using an energy threshold detector for initialization,” The Journal of the Acoustical Society of America **123**(5), 2643, https://doi.org/10.1121/1.2896752.

Summers, J. E. (**2013**). “Information transfer in auditoria and room-acoustical quality,” The Journal of the Acoustical Society of America **133**(4), 242–248, https://doi.org/10.1121/1.4793579, doi: 10.1121/1.4793579.

Zhivorimov, H. (**2018**). “A method for colored noise generation,” Romanian Journal of Acoustics and Vibration **15**(1), 14–19.