Wire-Cell Toolkit Noise Modeling and Generation

Brett Viren

June 24, 2022

Topics

- Present formalism for noise **modeling** and **generation**.
- Understand **spectral interpolation** and **normalization**.
- Describe WCT code implementations with examples and future work.

Note, I follow the notation and formalism of:

- Mathematics Of The Discrete Fourier Transform
  - https://ccrma.stanford.edu/~jos/mdft
- Spectral Audio Signal Processing
  - https://ccrma.stanford.edu/~jos/sasp

Discrete Fourier Transform (DFT)

Frequency spectrum (*fwd*)

$$\omega_k = 2\pi \frac{f_s}{N} k, \;\; f_s \triangleq \frac{1}{T}$$

$$X_k \equiv X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi k}{N}}$$

Time/interval series (*inv*)

$$x_n \equiv x(n) \triangleq x(t = nT)$$

$$x_n = \frac{1}{N} \sum_{n=0}^{N-1} X_k e^{i\frac{2\pi n}{N}}$$

- $n, k \in [0, N-1], \;\; x_n \in \mathbb{R}, \;\; X_k \in \mathbb{C}$
- Asymmetric normalization convention: $\frac{1}{N}$ in the *inv*-DFT.
- Sampling time/frequency: T / f_s (and N) determines binning,
  - Nyquist: $f_n = \frac{f_s}{2}$ largest resolved frequency,
  - Rayleigh: $f_r = \frac{f_s}{N}$ smallest resolved frequency.

Useful squared quantities

Periodogram - normalized power spectrum

$$P_k = \frac{1}{N} |X_k|^2, \ k \in [0, N-1]$$

Parseval's Theorem aka Rayleigh Energy Theorem

$$E = \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 \equiv \sum_{k=0}^{N-1} P_k$$

Mean-squared (*ie*, RMS^2) aka normalized energy

$$\sigma_{rms}^2 \triangleq \frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2 = \frac{E}{N}.$$

Zero padding in time / interpolation in frequency

$$x_n \to x'_n = [x_0, ..., x_{N-1}, 0, ..., 0],\ n \in [0, N'-1],\ N' > N$$

$$X'_k = \mathrm{DFT}_k(x'),\ k \in [0, N'-1]$$

$$P_k \to P'_k = |X'_k|^2/N',\ E \to E' = E,\ \sigma_{rms} \to \sigma'_{rms} = \sqrt{\frac{N}{N'}}\sigma_{rms}$$

- X'_k are **trigonometrically interpolated** from X_k but **not** scaled.
- Energy is constant, but spread over more elements.
- Actually, we want **more** E and keep P and σ_{rms} constant.
  - Can scale up X' by $\sqrt{N'/N}$ to remove bias.
- Same scaling needed after **direct interpolation** in frequency.

Averaging

Given a set of waveforms $\{x^{(m)}\},\ m \in [0, M-1],\ X_k^{(m)} = \mathrm{DFT}_k(x^{(m)})$ we may form simple averages of spectral **amplitude** and **power**,

$$\langle |X_k| \rangle \triangleq \frac{1}{M} \sum_{m=0}^{M-1} |X_k^{(m)}|,$$

$$\langle |X_k|^2 \rangle \triangleq \frac{1}{M} \sum_{m=0}^{M-1} |X_k^{(m)}|^2.$$

Best to choose $M \approx N$ in order to balance **spectral resolution** and **statistical stability**.

Frequency bin noise distribution

We model $X_k \in \mathbb{C}$ as:

- Uniformly distributed phase: $\angle X_k \sim \mathcal{U}(0, 2\pi)$
- Rayleigh distributed amplitude: $|X_k| \sim \mathcal{R}(\sigma_k)$
  - Note: $r \sim \mathcal{R}(\sigma),\ u \sim \mathcal{U}(0,1),\ r = \sigma\sqrt{-2\ln u}$
- Or equivalently via normal distributions:
  - $\mathrm{real}(X_k) \sim \mathcal{N}(0, \sigma_k),\ \mathrm{imag}(X_k) \sim \mathcal{N}(0, \sigma_k)$

The parameter σ_k is the **mode** (not mean) of the Rayleigh distribution.

- It is key to how we model and generate noise.
- Either of the first two moments estimate σ_k:

$$\langle |X_k| \rangle \approx \sqrt{\frac{\pi}{2}}\sigma_k,\ \langle |X_k|^2 \rangle \approx 2\sigma_k^2$$

White noise special case

- Flat mean spectrum: $\sigma_w \triangleq \sigma_k \; \forall \; k$ with,

$$\langle E \rangle = \frac{1}{N} \sum_{k=0}^{N-1} \langle |X_k|^2 \rangle = 2\sigma_w^2 = N\sigma_{rms}^2.$$

- Autocorrelation related to σ_{rms} at lag $l = 0$ and zero o.w.

$$(x \star x)(l) = N\sigma_{rms}^2 \cdot \delta(l)$$

(Really, these two state the same thing, one in time and one in frequency)

Round trip validation

$$(raw\ waves \rightarrow) spectrum \rightarrow waves \rightarrow spectrum' \rightarrow waves'$$

- Sanity check waveforms.
- Assure distribution of E and σ_{rms} in time are as expected.
- Assure E is same in time and frequency.
- Assure σ_k scales correctly when zero padding.
- Generate x_n from spectra, collect to estimate and recover spectra.

Noise types:

- Flat (white) spectrum and directly generate Gaussian waveforms, both with $\sigma_{rms} = 1$.
- Fictional, shaped spectrum similar to real detector noise, tune to be near $\sigma_{rms} = 1$.

Validation test

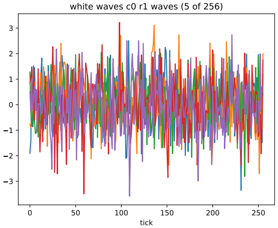
$$(raw\ waves \rightarrow) spectrum \rightarrow waves \rightarrow spectrum' \rightarrow waves'$$

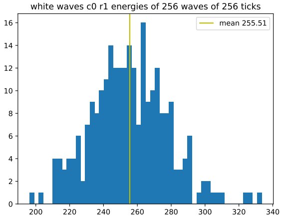
```
$ ./wcb --target=test_noisetools
$ ./build/aux/test_noisetools
$ wirecell-test plot -n noisetools \
    build/aux/test_noisetools.tar \
    aux/docs/test_noisetools.pdf
```

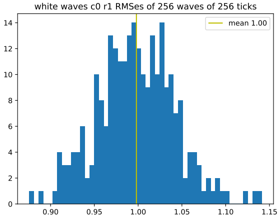
Excerpts from that PDF will are shown next.

- Same set of plots for $spectrum \in (white, gauss, shape)$.
  - "gauss" starts from ("raw") waves, the rest start from a spectrum
- Two "rounds" (labeled **r1, r2**) of $spectrum \rightarrow waves$ are performed.
- Two choices for sizes:
  - Cyclic (**c1**) have $\{x_n\}$ size $N^{(det)} = N^{(fft)} = 256$.
  - Acyclic (**c0**) have $N^{(det)} = 256$ which are zero-padded to use $N^{(fft)} = 512$.

Flat (“white”) spectrum



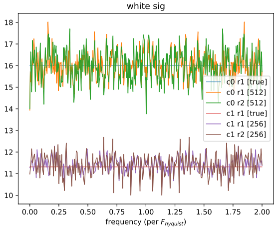


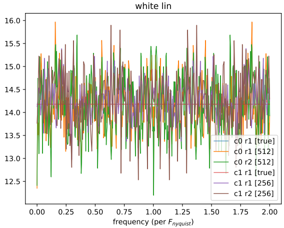


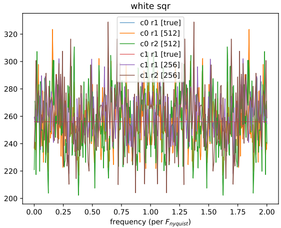
Generated from an exactly flat spectrum of $\sigma_k = \sigma_w = \sqrt{\frac{N}{2}}$, $(\sigma_{rms} = 1.0)$

- Sane looking waves, recover expected energy and RMS
- Not shown but similar results for:
  - Flat **c1**: cyclic FFT (wrap-around) and **r2**: second round.
  - Directly generating Gaussian $\mathcal{N}(0, 1)$ waves **(c0,c1)**$\otimes$**(r1,r2)**.

Flat (“white”) σ_k, $\langle|X_k|\rangle$, $\langle|X_k|^2\rangle$







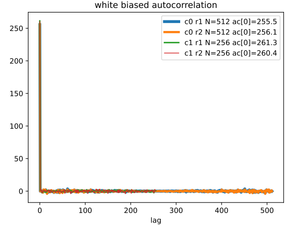
Lines mark expected mean given white noise $\sigma_{rms} = 1$.

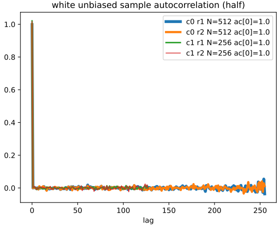
“sig” σ_k normalized to remove interpolation bias.

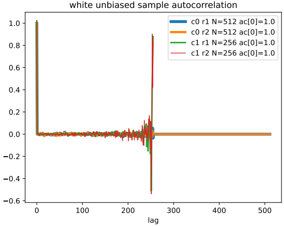
“lin” $\langle|X_k|\rangle$ with interpolation bias.

“sqr” $\langle|X_k|^2\rangle$ also with bias, divide by $N = 256$ to get periodogram.

Flat (“white”) autocorrelation







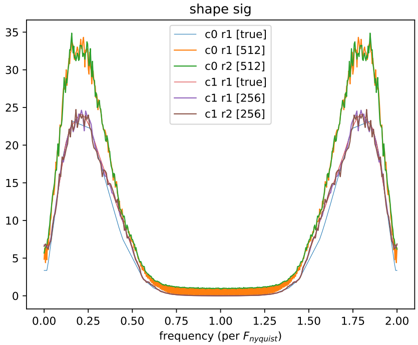
Each shows cyclic/acyclic and first and second rounds.

- Indeed, autocorrelation for $l = 0$ works out correctly (*eg* $\mathrm{bac}[0] \approx N\sigma^2_{rms}$).
- The instability at high lag l is expected in the SAC due to statistical instability divided by a small number for normalization.
  - Note: first SAC plot zoomed to half-range, second if full range.

Fictional spectra

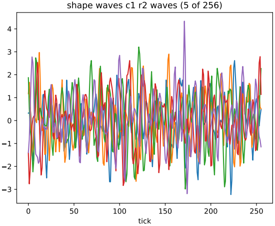
Use analytic Rayleigh distribution as function of frequency to approximate the shape of real noise spectrum and tune normalization so $\sigma_{rms} \approx 1.0$.

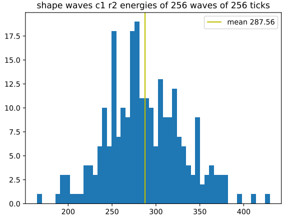
- "true" emulates a "hand digitized", irregularly-sampled spectrum.
  - Random points chosen uniquely for **c0** (acyclic) and **c1** (cyclic)
- Use new `irrterp` irregular interpolation to get regular sampled spectrum.
- Each round of each pair (**c0/c1**) recovers its "true" σ_k spectra.

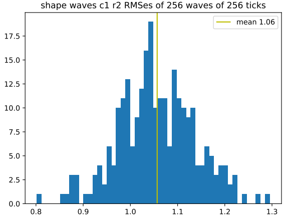


As with white noise, "sig" is the unbiased σ_k spectrum.

Fictional waves

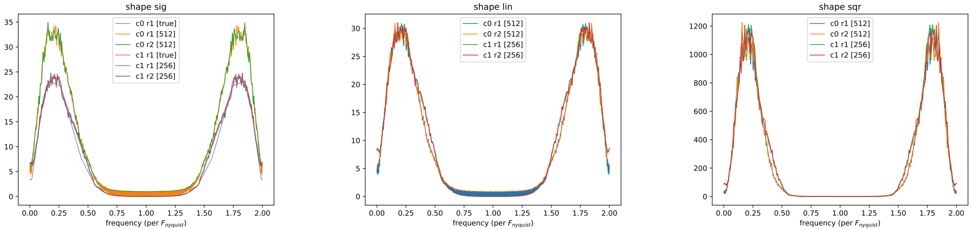






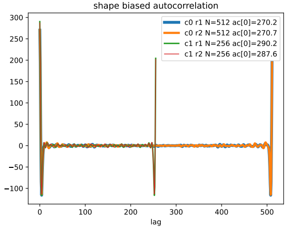
- All $(\mathbf{c0}, \mathbf{c1}) \otimes (\mathbf{r1}, \mathbf{r2})$ give statistically similar energies and RMS's.
- Again, spectrum was tuned so $\sigma_{rms} \approx 1$, expect real world spectra to differ.

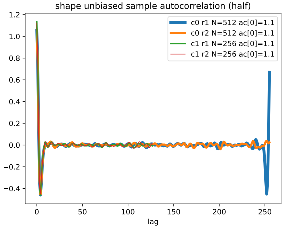
Fictional σ_k, $\langle|X_k|\rangle$, $\langle|X_k|^2\rangle$

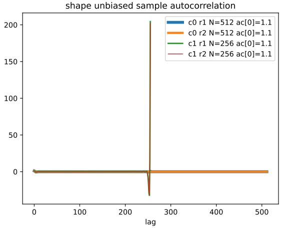


Again, σ_k has interpolation bias removed and $\langle|X_k|\rangle$, $\langle|X_k|^2\rangle$ do not.

Fictional autocorrelation







- As with white noise, show BAC and SAC (half and full range).
- Even BAC has large deviation at high lag $l \approx N/2$.
- How to associate the anti-correlation at small lag with spectral shape?
- Recover expected σ^2_{rms} at $l = 0$.

Collecting noise

- User decides `nsamples`, acyclic choice is $N^{(fft)} = 2^{\lceil \log_2(2*N) \rceil}$
- Autocorrelations are optional as they require extra DFTs.
- Add the $\{x_n^{(det)}\}$ waveforms.
- Retrieve final stats, available are:

`sigmas(), amplitude(), linear(), square(), rms(), periodogram(), bac(), sac(), psd()`

NoiseTools::Collector

```cpp
#include "WireCellAux/NoiseTools.h"
using namespace WireCell::Aux::NoiseTools;

// Eg, traces from IFrame
std::vector<real_vector_t> waves = ...;
size_t nticks = waves[0].size();
size_t nsamples = ...; // user defined
bool do_acs = true;  // off by default

Collector nc(dft, nsamples, do_acs);
for (const auto& wave : waves) {
    nc.add(wave.begin(), wave.end());
}
// Rayleigh sigma_k spectrum
auto sigmas = nc.sigmas();
```

Generating noise

Use $\mathcal{N}/\mathcal{N}$ or $\mathcal{R}(\mathcal{U})/\mathcal{U}$ forms

- Provide a `Fresh` or `Recycled` source of $\mathcal{N}$ or $\mathcal{U}$ distributed randoms.
- Create appropriate, equivalent `Generator{N,U}`

To make waves:

- get σ_k spectrum from `Collector` or file.
- Call `spec()` to get fluctuated σ'_k spectrum and feed to *inv*-DFT.
- Call `wave()` to include the *inv*-DFT to make a wave directly.

NoiseTools::Generator

```
#include "WireCellAux/RandTools.h"
using namespace WireCell::Aux::randTools;

// Also "Recycled" and also "Normals"
Fresh fu(Uniforms::make_fresh(rng));

// Also GeneratorN with Normals
GeneratorU ng(dft, fu);

// Flucuated sigma spectrum, feed to invDFT()
// auto fsigmas = ng.spec(sigmas);
// Or directly, a fresh noise waveform
auto wave = ng.wave(sigmas);
```

Get σ_k spectrum from `NoiseTools::Collector` or load from file, but don't forget to convert from amplitude (linear or square) to $\sigma_k = \sqrt{\frac{2}{\pi}}\langle|X_k|\rangle = \sqrt{\langle|X_k|^2\rangle/2}$.

New WCT Components

`IncoherentAddNoise`

- Takes one or more `IChannelSpectrum` "models".
- Replaces `AddNoise` but leaves that name as an alias so old configuration still works.
- Uses a `NoiseTools::Generator`.
- Handles conversion from $\langle |X_k| \rangle \to \sigma_k$ (*ie* `IChannelSpectrum` is left as-is, for now?).

`CoherentAddNoise`

- Almost identical to above but generated waveform is added to a group of channels. Could even combine the two if we configure groups-of-single-channel....
- Takes one or more `IGroupSpectrum` models: maps spectrum to group and group to channels.

`GroupNoiseModel`

- Happens to implement both `IChannelSpectrum` and `IGroupSpectrum`.
- For either, reads same file format.
- Still TBD: file and code need to specify normalization information.

`EmpiricalNoiseModel`

- Left as-is for now, but perhaps best to unify it and `GroupNoiseModel`.
  - At least, `GroupNoiseModel` should/will use a similar file format.
  - `GroupNoiseModel` does not support dynamic changes to electronics response.
  - OTOH, `EmpiricalNoiseModel`'s wire-length binning could be handled more generically as a channel "group".

Future WCT Components?

I would like WCT to provide a “standard” method for experiments to produce “proper” WCT noise files. This would require two new components;

`NoiseFinder`

- An `IFrameFilter`
- Accept ADC waveforms
- Convert to Voltage
- Discard signal-like waves
  - *eg* based on *mode* subtraction and outlier-detection
- Output `IFrame` with survivors

`NoiseWriter`

- An `ITerminal` and `IFrameSink`
- Configure with a channel-group map
- Maintain per group `NoiseTools::Collector`’s
- Marshal input to associated channel group’s `Collector`
- On `terminate()` write WCT noise file.

Likely insert a “frame tap” save out the intermediate noise frames for validating.

$\mathcal{FIN}$

(backups)

Signal autocorrelation function of "lag" l

Biased autocorrelation (BAC)

$$(x \star x)(l) \triangleq \sum_{l} x(m)x(m+l)$$

$$\mathrm{DFT}_k(x \star x) = |X_k|^2$$

Unbiased "sample" autocorrelation (SAC)

$\hat{r}(l) \triangleq \frac{(x \star x)(l)}{N-|l|}$ for $|l| < N - 1$ and zero otherwise.

Aside: zero-padding of time sequence

Eg, want FFT for fast **autocorrelation**

$$\hat{r}(l) = \frac{1}{N-l} inv\mathrm{DFT}_l(|\mathrm{DFT}(x)|^2)$$

Zero-padding: FFT requires 2^p, **acyclic** requires $2N$

$$x(n) \rightarrow x_{zp}(n) = [x(0), ..., x(N-1), 0, ..., 0]$$

$$n \in [0, 2N^{(fft)} - 1],\ N^{(fft)} = 2^{\lceil \log_2(2N) \rceil}$$

- N as product of small prime factors may win when $2^p \gg N$.

Zero-padding in time is **interpolation** in frequency

- Results in "trigonometric" type interpolation.
- Normalization unchanged but *inv*-DFT has $\frac{1}{N}$.
  - Will need to take this into considering in some cases.

Aside: white noise is fully uncorrelated

Sampled autocorrelation

$$\hat{r}(l = 0) \approx \sigma^2 \triangleq \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2, \; \hat{r}(l \neq 0) \approx 0$$

- This becomes an equality as $N \to \infty$.
- Will use $\hat{r}(0) \approx \sigma^2$ to validate noise code.

Noise modeling and generating procedure

1. Select a set of *detected waveforms* rich in noise (no signal).
   - Convert from units of ADC to Volts,
   - $\Rightarrow x^{(det)}(n),\ n \in [0, N^{(det)} - 1]$.
2. Partition full set into subsets of "like" waveforms,
   - eg, coherent groups, similar wire lengths.
3. Collect *fwd*-DFT statistics averaged over each subset:
   - $\langle |X_k| \rangle$ *spectral amplitude*,
   - $\langle |X_k|^2 \rangle$ *spectral power*,
   - $k \in [0, N^{(fft)} - 1]$
4. Sample and fluctuate $\langle |X_k| \rangle$ and apply *inv*-DFT to produce *simulated noise waveforms*,
   - $\Rightarrow x^{(sim)}(n),\ n \in [0, N^{(sim)} - 1]$.

Must take care of the fact $N^{(det)} \neq N^{(fft)} \neq N^{(sim)}$!

Welch's (aka *periodogram*) method for estimating spectra

Simple average over M DFTs of waveforms of size N

$$\langle|X_k|\rangle \triangleq \frac{1}{M}\sum_{m=1}^{M}|X_k^{(m)}|,\ k \in [0, N-1] \text{ and } \textit{etc} \text{ for } \langle|X_k|^2\rangle$$

Chosing M and N

- Larger N gives better **spectral resolution**,
- Larger M gives better **statistical stability**,
- Choose $M \approx N$ gives **balanced optimization**.

Special case for white noise

May *repartition* the waveforms to achieve balanced optimization

$$N' = M' = \sqrt{M * N}$$

Noise waveforms from non-flat spectrum must be kept whole.

Generating waveforms

Average Rayleigh mode spectrum

$$\sigma_k = \sqrt{\frac{2}{\pi}} \langle |X_k| \rangle, \; k \in [0, N^{(fft)} - 1]$$

Sample from Rayleigh $\mathcal{R}$ and uniform $\mathcal{U}$ distributions

$$|X_k| \sim \mathcal{R}(\sigma), \; \angle(X_k) \sim \mathcal{U}(0, 2\pi)$$

Or, real and imaginary parts from Gaussian $\mathcal{N}$

$$real(X_k) \sim \mathcal{N}(0, \sigma), \; imag(X_k) \sim \mathcal{N}(0, \sigma)$$

Generate waveform from the complex, X_k's

$$invDFT_n([X_0, ..., X_{N^{(fft)}-1}]), \; n \in [0, N^{(sim)} - 1] \to x^{(sim)}(n)$$

- Need only generate $k \in [0, N^{(fft)}/2]$ and apply Hermitian-symmetry.

$N^{(det)} \neq N^{(fft)} \neq N^{(sim)}$

Reminder of Parseval's theorem:

$$E = \sum_{n=0}^{N-1} |x_n|^2 = \frac{1}{N} \sum_{k=0}^{N-1} \langle |X_k|^2 \rangle$$

When we interpolate in frequency, say $N \to N' > N$

- Subsequent *inv*-DFT makes more time samples, thus more energy.
- Interpolation holds normalization constant.
- But, the *inv*-DFT divides by $1/N'$, reducing energy.
- To conserve energy, we must **interpolate and scale**:

$$X_k \to X'_k = \sqrt{\frac{N'}{N}} X_k, \; N \to N'$$

Equivalently, this preserves RMS in time.

Steps to prepare mean spectral amplitude

1. Zero-pad time sequence $N^{(det)} \rightarrow N^{(fft)}$,
2. Apply *fwd*-DFT to form mean spectral amplitude contribution,
3. Scale amplitude by $\sqrt{\frac{N^{(fft)}}{N^{(det)}}}$.

Steps to generation of waveforms

1. Interpolate mean amplitude $N^{(fft)} \rightarrow N'^{(fft)} \geq N^{(sim)}$,
2. Scale amplitude by $\sqrt{\frac{N'^{(fft)}}{N^{(fft)}}}$ (and by $\sqrt{2/\pi}$, convert $\mu \rightarrow \sigma$),
3. Apply *inv*-DFT to get time series,
4. Truncate time series to $N'^{(fft)} \rightarrow N^{(sim)}$.

Integral Downsampling

In time, sum sequential L samples to get new size M,

$$x_n \to x'_m = \sum_{n=m}^{m+L-1} x_n,\ m \in [0, M-1],\ N = LM$$

In frequency, produces **aliasing** (sum L jumps of size M)

$$\sigma'_m = \sum_{l=0}^{L-1} \sigma_{(m+lM)},\ m \in [0, M-1]$$

Reduces both N and the Nyquist frequency by $1/L$.
The sum of size L means same energy spread over factor L fewer samples so must normalize linear spectra by $\sqrt{1/L}$.

Non-integral downsampling

$$N \rightarrow N' \triangleq LM,\ L = \lceil \frac{N}{M} \rceil$$

Then interpolate spectrum to N', with $\sqrt{N'/N}$ scaling and apply integral downsampling for total saling $\sqrt{N'/NL}$

Reduce sample period with fixed N

$$T \to T' = rT, f_n \to f'_n = f_n/r,\ r < 1$$

This interpolation in time is equivalent to extrapolating the spectrum in frequency. Extrapolation requires some model.

- constant extrapolation from spectral value at f_n is reasonable when the spectrum there is dominated by white noise.
- zero-pad the spectrum above f_n may be applicable when the original signals are nominally zero at f_n but statistical fluctuation on the mean spectrum failed to achieve exactly zero.
  - (Maybe a sign that the hardware antialiasing filters and/or original sampling rate were not well chosen?)

General resampling

Have

$$\sigma_{1,n},\ n \in [0, N_1 - 1],\ f_1^{(r)} = 1/N_1 T_1, f_1^{(n)} = 1/2T_1$$

Want:

$$\sigma_{2,n},\ n \in [0, N_2 - 1],\ f_2^{(r)} = 1/N_2 T_2, f_2^{(n)} = 1/2T_2$$

Relative sizes of N, M and T, T' give potentially 4 combinations.

Interpolate $N_1 \rightarrow N_1' = N_2 \frac{f_1^{(n)}}{f_2^{(n)}}$ so $f_1^{(r)} \rightarrow {f'}_1^{(r)} = f_2^{(r)}$ (ie, same binning)

- gain $\sqrt{N_1'/N_1}$ normalization

Calculate $L \triangleq \lceil f_1^{(n)}/f_2^{(n)} \rceil$ and extrapolate $N_1' \rightarrow N_1'' = LN_2$.

- gain $\sqrt{N_1''/N_1'}$ if zero pad, but no gain if extrapolate non-zero constant.

If $f_1^{(n)} \leq f_2^{(n)}$ return extrapolated spectrum (N_1'').
Else, perform aliasing with L on N_1''.

- gain $\sqrt{1/L}$

General resampling with larger period.

$$T_2 > T_1,\ R_{21} = T_2/T_1 > 1,\ f_2^{(n)} < f_1^{(n)}$$

The input bin index $n' \triangleq \frac{N_1}{2R_{21}}$ is approximately at $f_2^{(n)}$.

Interpolate so $n' \rightarrow n'' = \frac{N_2}{2} \triangleq \frac{N_1''}{2R_{21}},\ N_1 \rightarrow N_1'' = N_2 R_{21}$

If $N_1'' > N_2$ we may alias by pretending same periods.