

1 Static noise

This noise source is used to simulate a slowly changing noise source compared to the experiment time (single shot). For every simulation iteration, a noise quantity is added to a Hamiltonian H_s .

$$H = H_0 + H_1(t) + \gamma_s H_s \quad (1)$$

Where γ_s is a random variable¹ from a Gaussian distribution, $\mathcal{N}(\mu, \sigma)$. By default the noise quantity is expressed as the variance² of the Gaussian distribution, σ^2 . The expected unit in the simulation is radians, so the unit of sigma is $[rad^2]$. For every iteration of the simulations a noise quantity of

$$\gamma_s = \mathcal{N}(\mu = 0, \sigma = \sqrt{\sigma^2}) \quad (2)$$

is added. When two noise sources are acting on the same H_s (σ^2), they are counted up as,

$$Var(A + B) = Var(A) + Var(B) \quad (3)$$

When a T_2 is specified, the following conversion is done:

$$\sigma^2 = \frac{2}{T_2^2} \quad (4)$$

Note again units are $[rad^2]$.

2 Spectral noise

Sometimes a user might want to run noise from an arbitrary spectrum, defined by a spectral density $S(\omega)$. In general the user can define spectral densities that are multiplied by a certain noise power (p), e.g.

$$S_{tot}(\omega) = \sum_i p_i S_i(\omega) \quad (5)$$

When running a simulation, for each iteration, noise will be captured from $S_{tot}(\omega)$. The Nyquist criterion tells that one can only sample noise in the following frequency range, $f > \frac{1}{T_{sim}}$ and $f < \frac{1}{2T_{time\ step}}$. This means that we are missing out on all the frequencies outside this range. To compensate for this, the following integral is taken,

$$\sigma_{S(\omega)}^2 = \frac{\pi}{2} \int_{0.1*2\pi}^{\frac{2\pi}{T_{sim}}} S(\omega) d\omega \quad (6)$$

This integral returns the low frequency noise as a static noise component (see section 1).

¹ Is there a good name for this? Now just took γ because it looks pretty, maybe ϵ ?

² Variance is used as this has the same unit as the spectral density function.

As lower cut, we took 0.1Hz as it looked reasonable that most experiments are done in 10 sec. We could also integrate from a frequency of 0, but some distributions might diverge here. Hence we introduce the artificial cap. The high frequency components are neglected, as they should average out in the evolution of the Hamiltonian .

In total we get the following Hamiltonian:

$$H = H_0 + H_1(t) + (\gamma_s + \gamma_d(t))H_n \quad (7)$$

Where γ_s is the contribution due to the static part (low freq components) of the noise and $\gamma_d(t)$ is a time dependent part of the noise. For every iteration of the simulation both γ_s and $\gamma_d(t)$ are updated. In the following we will discuss how $\gamma_d(t)$ is generated³.

Generation of $\gamma_d(t)$

In short, the user specifies for a simulation a spectral density $S_{tot}(\omega)$, with a time with a time T_{sim} and step size of $T_{time\ step}$.

This means we can generate spectral components in the range $F_{fft} \{ \frac{1}{T_{sim}}, \dots, \frac{1}{2T_{time\ step}} \}$ (see below for construction), corresponding to the FFT frequencies needed to generate an array of noise ($\gamma_d(t)$) for simulation time T_{sim} .

The spectral density is a function that describes the variance (or the power) of the noise at a certain frequency f . In that respect we sample random values of noise from a normal distribution for the frequencies F_{fft} ,

$$\begin{aligned} \epsilon_{noise}(F_{fft}) = \mathcal{N}(\mu = 0, \sigma = \sqrt{S_{tot}(F_{fft} * 2\pi)}) + \\ i * \mathcal{N}(\mu = 0, \sigma = \sqrt{S_{tot}(F_{fft} * 2\pi)}) \end{aligned} \quad (8)$$

The imaginary part is added to randomize the phases in the spectrum ⁴.

The next step is to convert the noise from frequency to time space. This is done by simply taking the iFFT.

$$\gamma_d(t) = iFFT(\epsilon_{noise}(F_{fft})) \quad (9)$$

³ Based on, Timmer, J. and Koenig, M. On generating power law noise. Astron. Astrophys. 300, 707-710 (1995).

⁴ I still a bit confused on adding also noise on the imaginary part, is it not cleaner to do, $\epsilon_{noise}(F_{fft}) = \sqrt{S_{tot}(F_{fft} * 2\pi)} * e^{i\theta(F_{fft})}$, where $\theta(F_{fft})$ is a angle generated from an uniform distribution (for every freq in F_{fft}). In this way we keep the original amplitude generated as the one generated by the normal distribution with the given variance and we just add a fully randomized phase?)

Details on generating $\epsilon_{noise}(F_{fft})$ The frequencies to take a Fourier transform are formatted as:

$$F_{fft} = \frac{[0, 1, \dots, n/2 - 1, -n/2, \dots, -1]}{d n} \quad (10)$$

if the number of points (n) is even ($d = T_{time\ step}$) and

$$F_{fft} = \frac{[0, 1, \dots, (n-1)/2, -(n-1)/2, \dots, -1]}{d n} \quad (11)$$

if n is odd.

Since for a iFFT the positive and negative frequencies are related⁵, we sample,

$$\epsilon(\omega_i) = \mathcal{N}(\mu = 0, \sigma = \sqrt{S_{tot}(\omega_i)}) * \frac{1}{T_{sim\ step}} \quad (12)$$

for all frequencies in F_{fft} that are larger than 0. Then we fill in $\epsilon(F_{fft})$ with this $\epsilon(\omega_i)$:

$$\epsilon(F_{fft}) = [0, \epsilon(\omega_0), \dots, \epsilon(\omega_n), \overline{\epsilon(\omega_n)}, \dots, \overline{\epsilon(\omega_0)}] \quad (13)$$

In the same format as specified in (10) and (11). [question](#)⁶

notes:

- For frequency 0, we just assign 0, as it does not take part in the spectrum.
- In case of an even number of points, there is not positive counterpart for the negative frequency, $\frac{-n}{2dn}$. This means when taking the inverse transform one will not have a positive counter part, which will result in imaginary parts in the sequence (unphysical). Therefore this peculiar frequency component is set to 0.

⁵ Amplitudes are each others complex conjugate, e.g. $\epsilon(\omega)e^{i\omega t} + \overline{\epsilon(\omega)}e^{-i\omega t} = Ae^{i\theta}e^{i\omega t} + Ae^{-i\theta}e^{-i\omega t} = 2A\sin(\omega t + \theta)$

⁶ Should we divide $\epsilon_{noise}(F_{fft})$ by 2, as we get 2* times the sin (see prev footnote)?