Subject: Random numbers with predefined spectral density
Posted by hahn on Mon, 21 Aug 1995 07:00:00 GMT

Hi,

I want to calculate random numbers with other than white noise
spectrum. The noise spectrum (amplitude vs. frequency) is
given and I need random numbers that adhere to this spectrum.

Actually, the spectrum is given by a FIR filter of order 12:
 coeff =  [ 1.48, 1.64, 0.927, ..., 3.1e-3]

How to proceed from here ?

Norbert Hahn

---

Subject: Re: Random numbers with predefined spectral density
Posted by steinhh on Tue, 22 Aug 1995 07:00:00 GMT

In article <41ab22$jp5@rs18.hrz.th-darmstadt.de>, hahn@hrz.th-darmstadt.de (Norbert Hahn)
writes:
|>
|> I want to calculate random numbers with other than white noise
|> spectrum. The noise spectrum (amplitude vs. frequency) is
|> given and I need random numbers that adhere to this spectrum.
|>
|> Actually, the spectrum is given by a FIR filter of order 12:
|>  coeff =  [ 1.48, 1.64, 0.927, ..., 3.1e-3]
|>
|> How to proceed from here ?
|>
|> Norbert Hahn

Simply produce white noise, and filter it with your desired noise
spectrum (beware of the difference between *amplitude* and the
*power* spectrum).

Given  N = Noise amplitude vs. frequency, an array with NN elements,
you simply use:

    NOISE = fft( fft(randomn(seed,NN),-1)*N , 1)

You should also include some renormalization of the total power
in your noise, of course.

---

## Subject: Re: Random numbers with predefined spectral density
Posted by David Ritscher on Thu, 24 Aug 1995 07:00:00 GMT
View Forum Message <> Reply to Message

> I want to calculate random numbers with other than white noise
> spectrum. The noise spectrum (amplitude vs. frequency) is
> given and I need random numbers that adhere to this spectrum.

> Actually, the spectrum is given by a FIR filter of order 12:
>   coeff = [ 1.48, 1.64, 0.927, ..., 3.1e-3]


Just make a white noise sequence and filter it, using these same
coefficients, as per the following example:

```
coeff = [ 1.48, 1.64, 0.927, 0, 0, 3.1e-3]
n_samples = 10000

n_coef = n_elements(coeff)
n_samples = 10000 + 2*n_coef

white = randomn(seed, n_samples)
colored = convol(white, coeff)   ;filter the white noise source
; throw away the first and last values, some of which were not filtered:
colored = colored(n_coef:n_samples-1-n_coef)

; note - although the IDL/PVWave CONVOL funtction is not really a
; convolution (i.e., (convol(a, reverse(b)) really performs the convolution
; of a and b), since this order doesn't change the spectrum of the result,
; this consideration can be ignored.
```


--

David Ritscher

Raum 47.2.279
Zentralinstitut fuer Biomedizinische Technik (ZIBMT)
Universitaet Ulm
D-89069 ULM
Germany

Tel:  ++49 (731) 502 5313
Fax:   ++49 (731) 502 5343
          or   5317

internet:   david.ritscher@zibmt.uni-ulm.de