EE640 STOCHASTIC SYSTEMS
SPRING 2008
COMPUTER PROJECT 1A

PART A: SYNTHESIS(4-8-08, 3-24-2010)

A.1 GENERATING GAUSSIAN NOISE FROM UNIFORM NOISE

Let Nx=My=128:

1. Uniform pseudo-random numbers. Generate 6 random images, each with a different seed. The images are all My x Nx where each element is uniformly distributed between 0 and 1. Each element is independent from the others. Let N = My x Nx. Mathematically refer to the images as N x 1 vectors in lexicographical form as:
$$\underline{u}_1, \underline{u}_2, \underline{u}_3, \underline{u}_4, \underline{u}_5, \underline{u}_6 \qquad \text{(A-1)}$$

2. Prove the parametric transformation equations for converting from a uniform distribution to a gaussian distribution are correct.
$$\underline{g}_i[2n+1] = \sqrt{-2\ln \underline{u}_i[2n+1]}\cos 2\pi \underline{u}_i[2n+2] \qquad \text{(A-2)}$$
$$\underline{g}_i[2n+2] = \sqrt{-2\ln \underline{u}_i[2n+1]}\sin 2\pi \underline{u}_i[2n+2] \qquad \text{(A-3)}$$
where n=0,1,2, …., (N/2 -1).

3. Generate six My x Nx gaussian random images from the associated images in part A.1.1. Use the transformation developed in A.1.2. Generate them with a 0 mean and unity variance and store as you did in A.1.1. Refer to them in lexicographical form as
$$\underline{g}_1, \underline{g}_2, \underline{g}_3, \underline{g}_4, \underline{g}_5, \underline{g}_6 \qquad \text{(A-5)}$$

4. Linear combinations of r.v.s. Generate five My x Nx images such that
$$\underline{s}_1 = \underline{u}_1 + \underline{u}_2 \qquad \underline{s}_4 = \underline{u}_1 + \underline{u}_2 + \underline{u}_3 + \underline{u}_4 + \underline{u}_5$$
$$\underline{s}_2 = \underline{u}_1 + \underline{u}_2 + \underline{u}_3 \qquad \underline{s}_5 = \underline{u}_1 + \underline{u}_2 + \ldots \underline{u}_6 \qquad \text{(A-6)}$$
$$\underline{s}_3 = \underline{u}_1 + \underline{u}_2 + \underline{u}_3 + \underline{u}_4$$

Display your results in a way that is suitable for visualizing them.
In the next project 1B, you will analyze your synthetic data.

A.2 GENERATING CONTROL NOISE FROM DETERMINISTIC DATA

There are 23 target and clutter images which are saved in target.zip and clutter.zip, respectively. The size of each image is 128 x 128 pixels. Choose one target and one clutter image from the two classes. Fig. 1 shows an example of 5 target training and Fig. 2 shows 5 examples of clutter images.

Figure 1: Five samples of the 23 target images.

Figure 2: Five samples of the 23 clutter images.

1. Find the PSD envelope of your target and clutter images by taking the 2-D Discrete Fourier Transform, then select the magnitude of the resulting spectra and refer to it has Ht for the target and Hc for the clutter image. We want to generate noise with an equivalent PSD.
2. Generate two independent My x Nx uniform noise images, At and Ac, with element ranges between 0 and 2π. Combine these with Ht and Hc such that the magnitude of each element is the same as Ht and Hc but the phase angle of the complex elements is determined by the two random images At and Ac, respectively. Inverse the result back to space domain and display the noise images.
3. Generate two independent My x Nx Gaussian noise images, gt and gc with element values having zero mean and unit variance. 2-D DFT these images to get Gt and Gc spectra. Elementwise multiply these spectra by Ht and Hc, respectively. Then inverse DFT the result back to the space domain and display results.
4. Are the noise images pairs, in task 2 and 3, statistically equivalent? Show why or why not, mathematically.

A.3 DIGITAL STOCHASTIC SIGNAL IMAGES

1. Generate two image vectors, a pseudo-random binary (i.e., bipolar) sequence and pseudo-random intensity sequence such that:

$$\underline{b}_{binary}[n] = \begin{cases} 1 & for \quad \underline{u}_1[n] \geq 0.5 \\ -1 & for \quad \underline{u}_1[n] < 0.5 \end{cases} \tag{A-7}$$

and

$$\underline{s}_{intensity}[n] = \left(\underline{g}_1[n]\right)^2 \tag{A-8}$$

where n=1,2,...N. The size of the images in 2-D is the same as in A.2.

Display these images so it is easy to visualize the difference in the noise types.

APPENDIX

If you encode noise into the frequency domain and require that the inverse fft yields all real values in the "space" or "time" domain, then there must be certain symmetry conditions met in the frequency domain. To demonstrate this, create an 8x8 matrix, A, of random noise. Then fft it to the frequency domain as B and evaluate the symmetry within the real and imaginary components.

A =

0.9501	0.8214	0.9355	0.1389	0.4451	0.8381	0.3046	0.3784
0.2311	0.4447	0.9169	0.2028	0.9318	0.0196	0.1897	0.8600
0.6068	0.6154	0.4103	0.1987	0.4660	0.6813	0.1934	0.8537
0.4860	0.7919	0.8936	0.6038	0.4186	0.3795	0.6822	0.5936
0.8913	0.9218	0.0579	0.2722	0.8462	0.8318	0.3028	0.4966
0.7621	0.7382	0.3529	0.1988	0.5252	0.5028	0.5417	0.8998
0.4565	0.1763	0.8132	0.0153	0.2026	0.7095	0.1509	0.8216
0.0185	0.4057	0.0099	0.7468	0.6721	0.4289	0.6979	0.6449

B=fft2(A);

real(B)=

33.5960	2.5077	1.4570	-2.7182	-0.8693	-2.7182	1.4570	2.5077
-1.1868	-2.5328	-1.0442	-0.3453	0.8548	-0.5695	-0.9498	0.5963
2.0612	-0.2416	1.2408	2.2483	0.8064	-1.4776	1.4948	0.0927
1.5700	0.1783	-1.1027	3.3849	0.9106	-0.3712	-1.7902	3.3390
0.0120	1.3609	1.9355	2.6287	-0.6061	2.6287	1.9355	1.3609
1.5700	3.3390	-1.7902	-0.3712	0.9106	3.3849	-1.1027	0.1783
2.0612	0.0927	1.4948	-1.4776	0.8064	2.2483	1.2408	-0.2416
-1.1868	0.5963	-0.9498	-0.5695	0.8548	-0.3453	-1.0442	-2.5328

Note that the for a column index n=1,2,…N and a row index m=1,2,…M, that if m=1 and n>N/2 then real(B(m,n))=real(B(m,n1)) where n1=2+N-n. The same is true for the column index such that if n=1 and m>M/2 then m1=2+M-m, so that real(B(m,n))=real(B(m1,n)).

imag(B)=

0	0.5449	-1.3813	3.1988	0	-3.1988	1.3813	-0.5449
-1.0333	-1.0398	0.6684	1.8551	-0.7279	-0.2027	-0.8158	0.4521
0.1559	-2.0416	-2.9714	-0.5650	-1.3007	2.7827	0.6965	-0.6664
0.3264	-0.9061	0.4175	1.0483	-1.8740	-0.7414	2.5313	0.7601
0	0.3909	-3.4593	2.7975	0	-2.7975	3.4593	-0.3909
-0.3264	-0.7601	-2.5313	0.7414	1.8740	-1.0483	-0.4175	0.9061
-0.1559	0.6664	-0.6965	-2.7827	1.3007	0.5650	2.9714	2.0416
1.0333	-0.4521	0.8158	0.2027	0.7279	-1.8551	-0.6684	1.0398

Note that the for a column index n=1,2,…N and a row index m=1,2,…M, that if m=1 and n>N/2 then imag(B(m,n)) = -imag(B(m,n1)) where n1=2+N-n. The same is true for the column index such that if n=1 and m>M/2 then m1=2+M-m, so that imag(B(m,n)) = -imag(B(m1,n)).