Sebastian Grubb

Advanced Digital Signal Processing Coursework

Prof. Danilo P. Mandic
TA: Felipe A. Tobar

February 4, 2013

1 Random signals and stochastic processes

Aims:

- To examine the generation of random signals in MATLAB.
- To investigate the effect of a linear system upon a random signal.
- To calculate and understand auto and cross-correlation functions.

In most real-world signal processing applications the measured signals cannot be described by an analytical expression; moreover, by the Wold decomposition theorem [5], a stationary signal can be represented as a sum of its predictable part and its stochastic part. Our focus is on stochastic signals, as these are typically found in real world problems, and the study of their statistical properties.

1.1 Statistical estimation

Generate a 1000-sample vector $\mathbf{x} = [x[1], x[2], \ldots, x[1000]]^T$ where each sample $x[n],\ n = 1, \ldots, 1000$, is a realisation of a uniform random variable $X \sim \mathcal{U}(0,1)$ and display the result with the `plot` function (use the `rand` and the `stem` functions). Observe that despite its stochastic nature, $\mathbf{x}$ exhibits a degree of uniformity due to its time-invariant statistical properties. Such signals are referred to as statistically stationary.
This way, the vector $\mathbf{x}$ can also be seen as a 1000-sample realisation of a **stationary** stochastic process X_n, such that $X_n \sim \mathcal{U}(0,1), \forall k$. In Signal Processing we assume column vectors unless otherwise stated, hence $\mathbf{x}$ has the form $\mathbf{x} = [x[1], x[2], \ldots, x[1000]]^T$.

1. Calculate the *expected value* of X, $m = \mathbb{E}\{X\}$, also known as *theoretical mean*, using your 1000-sample realisation estimate m with the `mean` function in Matlab, and also using the expression [5]

$$\hat{m} = \frac{1}{N}\sum_{n=1}^{N} x[n],$$

   where $\hat{m}$ is the so-called *sample mean* and the circumflex denotes that the quantity is an estimate. Comment on the accuracy of the estimates.

2. Repeat the analysis for the standard deviation: calculate $\sigma = \mathbb{E}\{X - \mathbb{E}\{X\}\}^2$ and also estimate this quantity from $\mathbf{x}$ using both the Matlab function `std`, and the expression[1] [5]

$$\hat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x[n] - \hat{m})^2}. \tag{1}$$

   Accordingly, we refer to $\hat{\sigma}$ as *sample standard deviation*. Comment on the accuracy of the estimates.

3. The bias in the estimation is given by $B = \mathbb{E}\{X\} - m$. Using the `rand` function generate ten 1000-sample realisations of X, denoted by $\mathbf{x}_{1:10}$, and calculate the sample means $\hat{m}_{1:10}$ and standard deviations $\hat{\sigma}_{1:10}$ for each realisation. Plot these estimates and comment on their bias based on how they cluster about their theoretical values. [5]

4. The mean and standard deviation describe second order statistical properties of a random variable, however, to obtain a complete statistical description it is necessary to examine the probability density function (pdf) from which samples are drawn. Approximate the pdf of X by applying the `hist` function to $\mathbf{x}$ and divide the result by the number of samples considered before plotting the pdf. Comment upon the result, in particular, on whether the estimate appears to converge as the number of generated samples increases. [5]

   Note: As mentioned above, the theoretical pdf of X is given by $\mathcal{U}(0,1)$,

5. Repeat (1)-(4) using the `randn` function which generates a zero mean, unit standard deviation Gaussian random variable. [20]

[1] Note that the use of $N-1$, instead of N, in Eq. (1) is known as *Bessel's correction*, and it aims to remove the bias in the estimation of the population variance, and some (but not all) the bias in the estimation of the population standard deviation. This way, Eq. (1) gives an *unbiased* estimator of σ.

1.2 Stochastic processes

A stochastic process is an ordered collection of random variables which usually represents the evolution of some random quantity in time. Although in most real-world problems we take expectations (averages) in the time domain (ergodic processes), where possible, we can also perform averages in the probability space (ensemble). To illustrate this concept, consider the **deterministic** sinusoidal signal $x[n] = \sin(2\pi 5e^{-3}n)$ corrupted by inependent Gaussian noise $\eta[n] \sim \mathcal{N}(0,1)$, represented by $y[n] = x[n] + \eta[n]$.

By averaging multiple realizations of the process $\mathbf{y} = [y[1], y[2], \ldots, y[N]]^T$, we can obtain improved estimates of the process $\mathbf{x}$, measured as the Signal-to-Noise (SNR) ratio. If M independent realisations of the process $\mathbf{y}$, denoted by $\mathbf{y}_{1:M}$, are considered to compute an ensemble estimate, the variance of such an estimate is given by

$$\sigma_M^2 = \mathbb{E}\left\{\left(\mathbb{E}\{\mathbf{y}\} - \frac{1}{M}\sum_{i=1}^{M}\mathbf{y}_i\right)^2\right\} = \mathbb{E}\left\{\left(\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\eta}_i\right)^2\right\} \tag{2}$$

$$= \frac{1}{M^2}\mathbb{E}\left\{\left(\sum_{i=1}^{M}\sum_{j=1}^{M}\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right)\right\} = \frac{1}{M^2}\left(\sum_{i=1}^{M}\sum_{j=1}^{M}\mathbb{E}\left\{\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right\}\right), \tag{3}$$

every noise sequence $\boldsymbol{\eta}_j$ comprises realisations of zero mean and **uncorrelated** random variables $\eta[n]$, we know that $\mathbb{E}\left\{\boldsymbol{\eta}_i^T\boldsymbol{\eta}_j\right\} = \sigma_\eta^2$ iff $i = j$, and zero otherwise, hence

$$\sigma_M^2 = \frac{1}{M^2}\left(M\sigma_\eta^2\right) = \frac{\sigma_\eta^2}{M}. \tag{4}$$

Therefore, the SNR of an M-member ensemble estimate increases linearly with the number of members of the ensemble and is given by

$$SNR = \frac{\sigma_{\mathbf{y}}^2}{\sigma_M^2} = \frac{\sigma_{\mathbf{y}}^2}{\sigma_\eta^2}M, \quad \text{and in dB: } SNR_{dB} = \log_{10}\left(\frac{\sigma_{\mathbf{y}}^2}{\sigma_\eta^2}M\right)\ [dB] \tag{5}$$

Figure 1 shows a realisation of $\mathbf{y}$, together with ensemble averages for $M = 10, 50, 200, 1000$ and the original deterministic signal $\mathbf{y}$. Additionally, the bottom plot shows the SNR computed from the ensemble averages and its theoretical value in Eq. (5).

We now study three stochastic processes generated by the following codes, which generate an ensemble of M realisations of N samples for each stochastic process.

a)
```
function v=rp1(M,N);
a=0.02;
b=5;
Mc=ones(M,1)*b*sin((1:N)*pi/N);
Ac=a*ones(M,1)*[1:N];
v=(rand(M,N)-0.5).*Mc+Ac;
```

b)
```
function v=rp2(M,N)
Ar=rand(M,1)*ones(1,N);
Mr=rand(M,1)*ones(1,N);
v=(rand(M,N)-0.5).*Mr+Ar;
```

c)
```
function v=rp3(M,N)
a=0.5;
m=3;
v=(rand(M,N)-0.5)*m + a;
```

Execute the above codes (call the functions using the command line) and study difference between the averages in the time and probability spaces, and its relationship with stationarity and ergodicity by following these steps:

1. Compute the ensemble mean and standard deviation for each process and plot them as a function of time. For all the above random processes, use $M = 100$ members of the ensemble each of length $N = 100$. Comment on the stationarity of each process. [10]

2. Generate $M = 4$ realisations of length $N = 1000$ for each process, and calculate the mean and standard deviation for each realisation. Comment on the ergodicity of each process. [10]

3. Write the mathematical description of each of the three stochastic processes. Calculate the theoretical mean and variance for each case and compare the theoretical results with those obtained by sample averaging. [10]

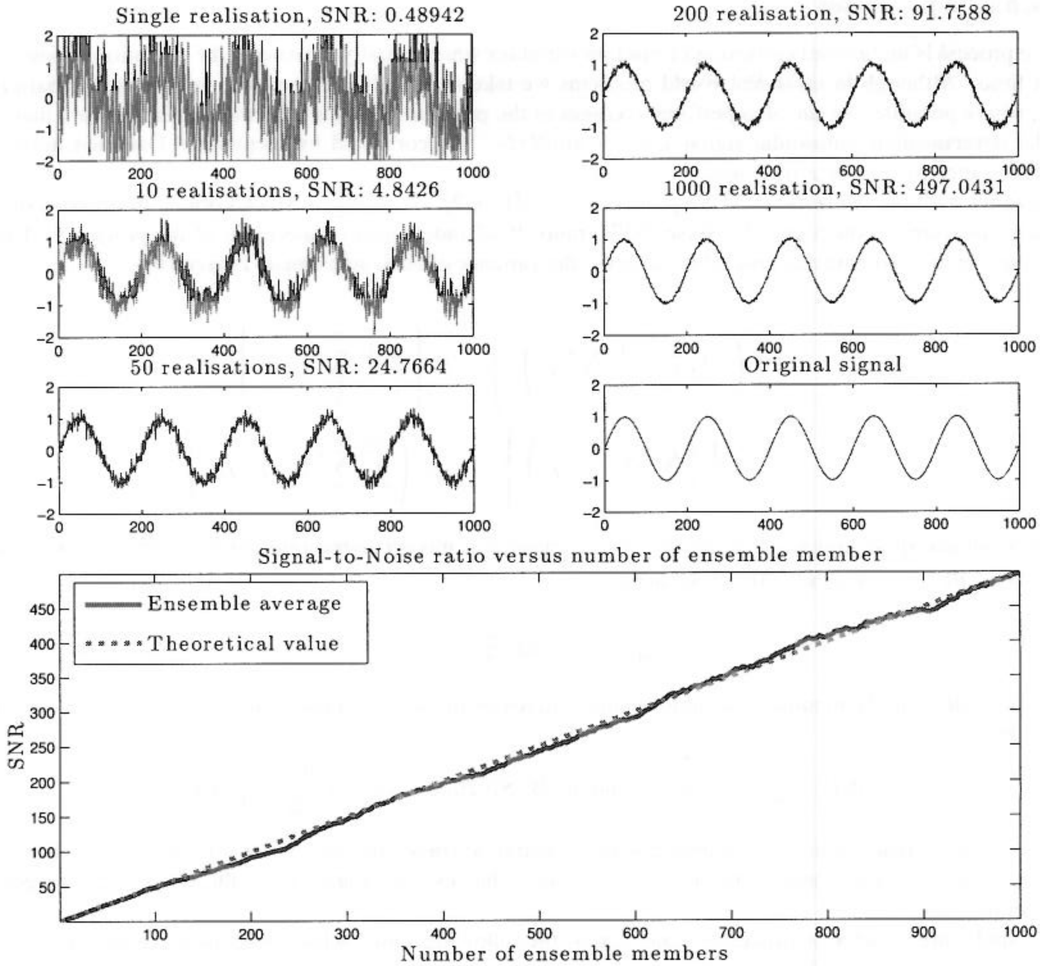

Figure 1: Ensemble estimates of a deterministic sequence corrupted by zero-mean, uncorrelated white noise.

1.3 Estimation of probability distributions

Stochastic processes can be uniquely represented by their probability density functions (pdf), since the pdf comprises all the statistical information about a random variable. When the process at hand is stationary, the statistical properties are time-invariant, and the processes is uniquely defined by a time-invariant pdf. Furthermore, for ergodic processes such distributions can be estimated via time averages. Implement a pdf estimator and test it on the three stochastic processes studied in Part 1.2.

1. Write an m-file named `pdf` using the function `hist`. Test your code for the case of a stationary process with a Gaussian pdf, `v=randn(1,N)`. The length N should be at least 100. [10]

2. For those processes in Part 2.2 (a,b,c) that are stationary and ergodic, run your Matlab code to approximate the pdf for $N = 100, 1000, 10000$. Compare the estimated pdf's with the theoretical pdf using the Matlab `subplot` function and comment on the results as N increases. [10]

3. Is it possible to estimate the pdf of a nonstationary process with your function `pdf`? Comment on the difficulties encountered for the case when the signal mean of an $N = 1000$ long signal `rp1` changes from 0 to 1 at sample point $N = 500$. [10]