Class Activity #12
Applications of Correlation Functions

Random Processes

Communication Systems performance analysis involves the process of characterizing the interference or noise signal along with the information signal. Both of these waveforms are considered random signals at the receiver. In the case of the information signal, the random fluctuations are what infer the information, while the noise is often a random signal that has its basis in some natural phenomena.

A random process is a time-based *waveform* that is the outcome of some random experiment, much in the same way that a random variable takes on a *value* as the result of an experiment. Depending on the particular value of the random quantity, a different *realization* of the random process is observed. As an example, consider a random variable Θ that is uniformly distributed over $[0,2\pi]$. We can then generate the random process

$$x(t) = \cos(\omega_c t + \Theta).$$

The process $x(t)$ is actually a set of waveforms that have different values in time depending on the random variable Θ (of course, in this simplistic case these waveforms look like phase-delayed versions of one another). This set of waveforms is known as an *ensemble*.

Another common random process example is to think of capturing the voltage signal across a resistor when no other circuit is connected (other than perhaps the oscilloscope probes); all that you see is low-level thermal noise that bounces around some small range. If you repeat this experiment, you will capture another time waveform that may look similar but will have different values in time. Capturing many of these waveforms constitutes a set of realizations or an ensemble of a random process.

We can think of this set of noise waveforms as being plotted on a two-dimensional grid, with time versus random process realization, as shown in Figure 1. We then consider the *x*-axis to be time (*t*) and the *y*-axis to be the realization, denoted as ζ, where ζ is a particular *instantiation* of the random quantity.

Looking at the figure, it can be seen that the set of outcomes for a particular time is itself a random variable. That is, slicing the set of waveforms at any one time (for example, t_1) creates a random variable $x(t_1)$ (technically speaking, $x(t_1;\zeta)$, but we prefer the simpler form). This "sliced" random variable has its own set of statistics that in general is *different* for every choice of *time*. Computing statistics for a random process across time slices is known as *ensemble* averaging since, as the name implies, we are averaging along the ensemble axis.

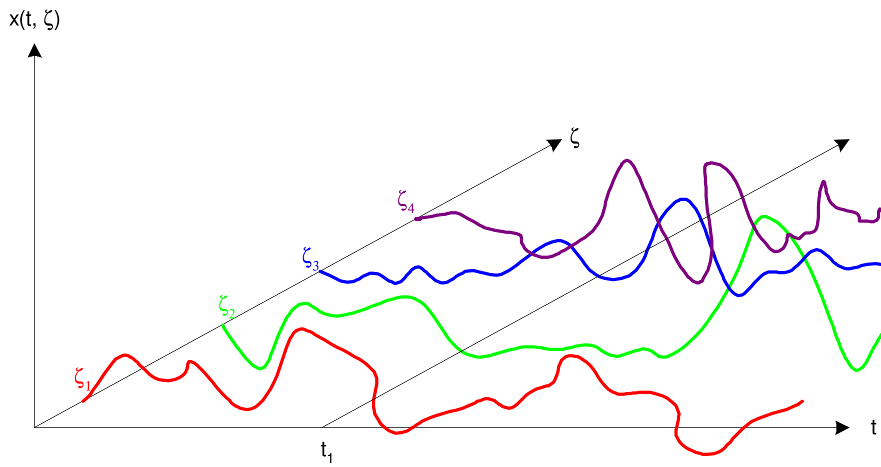

Figure 1: Ensemble Plot of a Random Process

Ensemble Averages and Stationarity

The ensemble statistics computed for a random process are similar for those calculated for a random variable. We are typically interested in the first (mean) and second order correlation/covariance statistics

$$\mu_{x(t_1)} = E[x(t_1)] = \int_{-\infty}^{\infty} x(t_1) f_{X(t_1)}(x(t_1)) dx(t_1)$$

$$R_{xx}(t_1,t_2) = E[x(t_1)x(t_2)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(t_1)x(t_2) f_{X(t_1)X(t_2)}(x(t_1),x(t_2)) dx(t_1) dx(t_2),$$

$$C_{xx}(t_1,t_2) = R_{xx}(t_1,t_2) - \mu_{x(t_1)}\mu_{x(t_2)}.$$

Note that we rarely have the marginal or joint PDFs available; instead, we consider the random process as a mapping function of the random variable. We then apply the Law of the Unconscious Statistician or other methods to evaluate expectation. Typically, we are more interested in the autocorrelation $R_{xx}(t_1,t_2)$ than the autocovariance $C_{xx}(t_1,t_2)$ because one or both of the random processes usually has zero mean, and $R_{xx}(t_1,t_2)$ therefore fully characterizes this aspect of the process. As will be discovered, the autocorrelation $R_{xx}(t_1,t_2)$ can be used to compute the PSD (Power Spectral Density) of the random process; that is, the PSD defines the distribution of the signal power across frequency.

Let us define the time difference as $\tau = t_2 - t_1$, such that we can equivalently rewrite the correlation function as

$$R_{xx}(t_1,t_2) = R_{xx}(t_1,t_1+\tau).$$

Written in this form, it is easy to see that the correlation function imparts information about the frequency in variation of the random process. If $R_{xx}(t_1,t_1+\tau)$ changes slowly with time difference τ, we can say that the process $x(t)$ is slowly varying; conversely, if

$R_{xx}(t_1, t_1+\tau)$ changes rapidly with time, $x(t)$ is quickly varying. In the slowly varying case, the random process has overall lower frequency content and a higher degree of sample-to-sample correlation.

A Strict Sense Stationary (SSS) random process is one for which none of the ensemble statistics depend on time, and therefore SSS is a very restrictive and uncommon situation. A much more common situation is that of a Wide Sense Stationary (WSS) random process, in which the first and second order statistics do not depend on absolute time. A WSS process has an expected value that does not depend on time at all and an autocorrelation function that only depends on the relative time difference. That is, for any choice of times t_1 and t_2,

$$\mu_{x(t_1)} = E[x(t_1)] = \mu_{x(t_2)} = E[x(t_2)] \equiv \mu_x ,$$

$$R_{xx}(t_1, t_2) = E[x(t_1)x(t_2)] = R_{xx}(t_2 - t_1) = R_{xx}(\tau) = E[x(t)x(t+\tau)].$$

Thus, considering Figure 1 for a WSS process, it does not matter what time t_1 you choose; if you average over ζ, you will compute the same average value μ. Furthermore, the averaging (correlation) over two time choices only depends on the *difference* of those times, not their absolute values.

Finally, note that (considering WSS processes)

$$\sigma_x^2 = R_{xx}(0) - \mu_x^2 .$$

That is, the variance of the process (at any time) is simply the autocorrelation evaluated "at zero lag" (i.e., time difference $\tau = 0$) minus the mean squared.

Time Averages and Ergodicity

An ergodic random process is one for which time averaging gives identical results to ensemble averaging. Note that ergodicity is more restrictive than a WSS requirement, with ergodic processes being a subset of WSS processes. For an ergodic process, only ***one*** realization of the process is required to obtain the desired statistical information. Denoting this single realization as $x(t)$, we calculate the sample mean as

$$\hat{\mu}_x = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)\,dt .$$

If the sample mean is the same as the WSS ensemble mean, i.e., if

$$\hat{\mu}_x = \mu_x ,$$

then the random process $x(t)$ is ergodic in the mean. Furthermore, let us define the sample correlation function as

$$\hat{R}_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)\,dt .$$

Similarly, if $\hat{R}_{xx}(\tau) = R_{xx}(\tau)$, the random process $x(t)$ is ergodic in the second order statistics.

An important property of the autocorrelation is that the function's peak occurs at zero lag, i.e.,

$$R_{xx}(0) \geq |R_{xx}(\tau)|.$$

The cross-correlation of jointly WSS random processes $x(t)$ and $y(t)$ is defined as

$$R_{yx}(\tau) = E[y(t+\tau)x(t)].$$

Note that $R_{yx}(\tau) = R_{xy}(-\tau)$. If the processes are jointly ergodic, we have that

$$R_{yx}(\tau) = \lim_{T\to\infty}\left\{\frac{1}{2T}\int_{-T}^{T} y(t+\tau)x(t)dt\right\}.$$

White Processes

A white process is a zero-mean WSS process for which the autocorrelation function is of the form (this function is only nonzero for $\tau = 0$):

$$R_{xx}(\tau) = \sigma_x^2\delta(\tau).$$

In this case, the autocorrelation function form indicates that the process has zero correlation across time and is only correlated when compared with itself (recall: $\tau = 0$ is no time shift at all). This type of process has a very high degree of randomness since even adjacent times have no correlation to each other. The process is denoted as "white" since its PSD can be shown to be flat over all frequency (an analogy to white light which is flat across the visible spectrum). A white process yields an impulse for its autocorrelation function with a flat PSD across the entire frequency band.

Measuring Round-Trip Delay

In this application, the cross-correlation function is used to estimate the round-trip propagation delay of an electromagnetic wave reflected off of a distant object. Using this result, the distance to the object can be estimated (assuming some propagation speed – the speed of light if the environment were a vacuum – the distance is simply the propagation delay multiplied by the speed, i.e., $d = c \times t$).

Let us assume that we generate a random process $x(t)$ at our transmitter that we then aim towards the object of interest. A portion of the transmitted energy will reflect back to our receiver from which we will measure a signal

$$y(t) = \alpha x(t-2\Delta) + n(t),$$

where $0 < \alpha << 1$ is the loss constant due to imperfect reflection and propagation loss, Δ is the one-way propagation delay and $n(t)$ is an interference or noise signal that inadvertently corrupts our received signal. We make the usual assumption that $x(t)$ and $n(t)$ are uncorrelated and that $n(t)$ is a zero-mean process.

To estimate Δ from our data, we compute the cross-correlation

$$R_{yx}(\tau) = E[y(t+\tau)x(t)] = E[(\alpha x(t+\tau-2\Delta) + n(t+\tau))x(t)],$$

for which it is easy to show that

$$R_{yx}(\tau) = \alpha R_{xx}(\tau - 2\Delta).$$

Since we know that $R_{xx}(\tau)$ peaks at an argument of zero-lag, it must be true that $R_{yx}(\tau)$ peaks at a lag of $\tau = 2\Delta$. Hence, by computing $R_{yx}(\tau)$ and finding the time at which this function reaches its maximum value will indicate the propagation delay to the object.

Sinusoid Buried in Noise

The autocorrelation can be used to find the frequency of a sinusoid buried in noise. One property of the autocorrelation function is that if the random process is periodic, the autocorrelation of that random process will also be periodic with the same period.

Consider a sinusoid of random phase buried in noise, where we write

$$x(t) = s(t) + n(t),$$

with $s(t) = A\cos(\omega_o t + \Theta)$ and where we have $\Theta \sim U(-\pi, \pi)$. Under the assumption that $s(t)$ and $n(t)$ are uncorrelated and that $n(t)$ is a zero-mean white-noise process of variance σ_n^2, it is straightforward to show that

$$R_{xx}(\tau) = R_{ss}(\tau) + R_{nn}(\tau) = \frac{A^2}{2}\cos(\omega_o \tau) + \sigma_n^2 \delta(\tau).$$

That is, the only contribution to the autocorrelation by the noise occurs at zero lag. We can ignore the noise contribution for the purposes of determining ω_o. Furthermore, since a periodic random process yields a periodic autocorrelation of the same period, inspection of the autocorrelation will indicate the period uncorrupted by the noise.

Assignment:

1. Use MATLAB to simulate the round-trip delay estimation problem. Assume your system is ergodic.
   - Choose a quasi-random signal for $x(t)$. One suggestion: let $x(t)$ be a random pulse train; that is, a series of pulses that have a value of ± 1, where the selection of a positive or negative pulse for transmission is determined randomly. Note: you could condition on a call to `rand` to send either a positive or negative pulse; however, use pulses of at least a several samples in duration.
   - Simulate the round trip transmission and reception of your signal as reflected by an object. Add a randomly generated integer delay to your received pulse along

with flat attenuation (your gain constant α to achieve your intended SNR). Add white zero-mean noise using the MATLAB `randn` function. Compute your SNR as

$$SNR_{dB} = 10 \times \log_{10}(SNR) = 10 \times \log_{10}\left(\sigma_x^2 / \sigma_n^2\right).$$

- Compute the cross-correlation of your received signal with the transmitted signal (you may use the MATLAB function `xcorr` or you can write your own). One thing to keep in mind regarding the MATLAB function `xcorr`; this function computes the two-sided cross-correlation when only positive lags are needed.
- Find the peak of your cross-correlation function to estimate the round-trip delay and compare to the actual delay.
- Attempt to qualify the performance of your round-trip estimator. You may want to plot estimated delay versus actual delay for a range of SNR. Does accuracy depend on the amount of delay? Is there an SNR for which your ability to estimate delay is no longer accurate? How does your data record length (i.e., $2T$) affect your results? Why?

2. Use MATLAB to simulate the frequency estimation of the sinusoid in noise application assuming your process is ergodic.
   - Pick a suitable frequency and generate your signal (with a random phase). Recall that you need to make sure your waveform is adequately represented (i.e., sampled fine enough) and that there is a sufficient number of cycles present for your autocorrelation calculation. Add the zero-mean white noise using `randn` to create the desired SNR (you want to test with poor SNRs such that the sinusoidal wave is buried in the noise). Compute the autocorrelation function (you may again use `xcorr` or your own code for this purpose; in this case, to generate an autocorrelation, there is only a single argument to `xcorr`). Either via visual inspection of the autocorrelation function or some algorithm you have devised, estimate the frequency of the original signal.
   - Attempt to qualify the performance of your frequency estimator for a range of frequencies. You may want to plot estimated frequency versus actual frequency for a range of SNRs. Does accuracy depend on the frequency? Is there an SNR for which your ability to estimate frequency is no longer accurate? How does your data record length affect your results? Why?

Turn in (due at the final exam):

- Plots showing estimated versus actual quantities for steps (1) and (2) as you vary parameters.
- Discussion of the performance of your algorithms for steps (1) and (2).
- Overall m-file script to run the lab and code for any custom functions you wrote.