EE3TR4 Lab 3

Random Processes

Jim Reilly x22895
reillyj@mcmaster.ca

Due: Last day of classes, Wed Apr. 8, 2015.

This lab is a computer assignment. You are not required to attend any lab session at all for this exercise. However, if you need help, there will be a TA available in the lab room during the regularly scheduled lab sessions. Their contact information is available in the course outline on the website.

If you need help with matlab, there are books available on reserve in Thode. A good reference on the subject is "Mastering Matlab", available at the bookstore. There is also the matlab tutorial on the course website.

Since we are processing signals on a computer in this lab, all the signals must be discrete-time. Recall that if the signal is properly sampled, then there are no frequency components present in the signal greater than $f_s/2$, where f_s is the sampling frequency. It is common practice to normalize the frequency scale when representing spectra of discrete-time signals relative to f_s. Thus, normalized frequency scales for discrete-time signals go from -1/2 to +½ Hz. These values represent ±fs/2 Hz. respectively.

In this lab, we assume all signals are wide-sense stationary, ergodic, and zero mean.

1. Evaluation of Autocorrelation and Power Spectral Density

In this part, we construct a sample of a white random process and put it through a filter whose impulse response is a discrete-time decaying exponential given by $h(n) = 0.8^n$. We then evaluate the autocorrelation and power spectral density of the output signal and compare to what we expect theoretically. Here is the procedure:

```
% get long white random sequence w
w=randn(40000,1);

% generate impulse response h(n)
decay=0.8;
t=0:30;
h=exp(log(decay)*t);   % this is equivalent to h(n) = 0.8^n
```

convolve the input (w) with the impulse response h to get the output y (use the matlab function "conv").

Evaluate the autocorrelation sequence corresponding to y and plot. Use the matlab command "xcorr", in the form "xcorr(y,y,maxlag)", where "maxlag is the maximum autocorrelation lag you choose. The default value is the length of y, which is very long, resulting in a very compressed autocorrelation plot. If you use a value, say of 100 or less for maxlag, then the plot expands and you can see clearly the features of the autocorrelation function around tau = 0.

Evaluate the corresponding power spectral density and plot.

The PSD may be evaluated using the matlab "fft" function. The output from this function needs some interpretation however. As mentioned above, the frequency range for a discrete-time signal is from -1/2 to +1/2 Hz (normalized). The input to the fft function is always a vector (array). Let it be of length N. Then, the fft output is also a vector of length N, representing the Fourier transform of the input at N frequency points which are uniformly spaced over the range [-1/2 to +1/2]. The ordering of the frequency values is a bit surprising at first though, because the first half (corresponding to indeces 1, …,N/2) of the elements of the output vector correspond to the frequency range [0 ½] (i.e., positive frequencies), while the second half (indeces N/2+1:N) correspond to the frequency range [1/2 1], or equivalently, [-1/2 0] (negative frequencies). This ordering of output values is different from what you expect. The interpretation of the output takes a bit of getting used to at first.

Because of the way we use the fft function, the output will be complex, even though a power spectral density function is supposed to be pure real. This has to do with the way the fft input is set up. We can't go in to the details here, but they will become clear in EE4TL4. To get around this problem, we simply use "abs(fft(Ry))", where Ry is the autocorrelation function of interest.

Write Up: for this section, calculate the theoretical autocorrelation function of the output, and the corresponding PSD. Compare your theoretical results with those from your program, and explain any discrepancies. Also, try using shorter lengths (e.g. 5000 or 2000) instead of the length 40000 as indicated above, for the input sequence. Explain your results. Include plots of all relevant quantities.

2. Estimation of a Filter Response

We are given the output *y* of a filter whose impulse response *h* is unknown, except for the fact it is pure real. The filter input *x* is white noise with zero mean. Estimate the impulse response of the filter. Can we determine the phase response of the filter? In view of this issue, is your estimate of the impulse response unique?

Download the file “ydata.mat” from the course website into some appropriate directory on your computer. Set the matlab working directory to the one which you downloaded to. Type “load ydata” in the matlab command window. The variable *y* will then contain the desired output sequence from the filter.

Write Up: For this section, carefully explain how you estimated the impulse response. Include plots of relevant quantities to support your argument.

3. Sinusoids in noise

Download the file “part3.mat”. It contains a signal *x* consisting of a sinusoid buried in zero-mean noise. Estimate the amplitude and frequency of the sinusoid. Assume the sampling frequency of *x* is 1 KHz (i.e., the time between samples is 1 msec.)

Writeup: For this section, explain the method you used to estimate the sinusoid’s frequency. Explain why the method is effective. Is it possible to estimate the phase of the sinusoid? Why? Include plots of relevant quantities.