EE3TR4 Lab 4

Random Processes

Since we are processing signals on a computer in this lab, all the signals must be discrete time. **The matlab file Lab4.m is posted on Avenue. Please modify this code to do all the numerical experiments.**

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

Numerical Experiment #1: Evaluation of Autocorrelation and Power Spectral Density

In this part, we are given a sample of a random process which corresponds to the output $y(n)$ of an ideal low-pass filter (LPF) driven by white noise. The bandwidth of the filter is 250 Hz. The output data $y(n)$ is available in the file "lab4_num_expt1.mat" on Avenue.

Evaluate the autocorrelation sequence corresponding to *y*, *Ry* and plot it as a function of the lag tau. Use the matlab command "xcorr", in the form "Ry =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 the lag, $\tau = 0$.

Evaluate the corresponding power spectral density (PSD) and plot it as a function of frequency. As explained in the class, the discrete Fourier transform (DFT) of the autocorrelation is the PSD. To compute the PSD, use "fftshift(fft(fftshift(Ry)))" instead of "fft(Ry)". Using the former, the PSD can be easily plotted as a function of the frequency vector. The length of the frequency vector is equal to that of the "tau_vector" which is 2*maxlag+1. The maximum frequency "fmax" is half of the sampling rate (= 1/tstep). Refer to the matlab code "Lab4.m" posted on Avenue.

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(fftshift(fft(fftshift(Ry))))", where abs($\cdot$) evaluates the magnitude of the argument.

Write Up: For this section, (i) calculate the theoretical autocorrelation function of the output, and the corresponding PSD using the methodology discussed in class. Compare (qualitatively) your theoretical results with those from your program, and explain any

discrepancies. Note that the amplitude (i.e. y-axis) of PSD of the matlab could be different from that of the theoretical PSD because of the difference between DFT and Fourier transform.
(ii) change the maxlag from 100 to 200 and then to 500 and observe its impact on the PSD and comment. Include plots of all relevant quantities.
(iii) Estimate the bandwidth of the filter using the autocorrelation plot. Hint: measure the locations of zeros of sinc and compare it with the theoretical calculation.

Note: To measure the locations of zeros, use + (data cursor) on matlab plot.

Numerical Experiment #2: A sinusoid buried in noise

A sinusoidal signal of the form
$x(t) = A\sin(2\pi f_c t + \theta),$
where θ is a random variable which takes any value between 0 and 2π with equal probability, passes through an AWGN channel of unity gain. The AWGN channel adds white noise.

Download the file “lab4_num_expt2.mat”. It contains a signal yt(n) which is the output of the AWGN channel in time domain. Estimate the frequency, f_c of the sinusoid.

Write up: For this section, provide the theoretical calculations (i.e. derivations of analytical expressions for autocorrelation and PSD of the AWGN channel output. Refer to lecture notes). Compare the theoretical autocorrelation function and PSD with those from your program (qualitatively).

(i) Do you observe a peak at the zero lag in the autocorrelation plot? Explain its origin.

(ii) Change the maxlag from 100 to 200 and then to 20000 and observe its impact on the frequency resolution. Estimating the frequency, f_c in the frequency domain and provide a table of measurements for the maxlag of 100, 200 and 20000. Do the frequency estimates converge? Explain the connection between the maxlog and frequency resolution.

(iii) Plot yt as a function of time. It is the sinusoidal signal buried in noise (i.e. x(t) plus white noise). Can you estimate the frequency, f_c directly from yt, without calculating the correlation? Explain.

Bonus Question: Numerical Experiment #3: Delay Estimation

Download the file “lab4_num_expt3.mat”. A Gaussian pulse given by
$x(t) = \exp(-t^2)$
is passed through a channel of transfer function
$H(f) = \exp(-j2\pi fT),$

i.e. it simply delays the input by T seconds and adds white noise $w(t)$. The output of the channel is of the form

$y(t) = \exp[-(t-T)^2] + w(t)$.

The file "lab4_num_expt3.mat" contains the input $x(t)$ as "xt" and the output $y(t)$ as "yt". Plot $x(t)$ vs time (i.e. "tt" defined in Lab4.m) and $y(t)$ vs time. You will find that you cannot estimate the delay by plotting $y(t)$ since the Gaussian pulse is buried in noise. Develop a technique to estimate the delay T and explain why it works. Provide an estimate of T using your technique.

Hint: Find the cross-correlation between $y(t)$ and $x(t)$ using xcorr, i.e.

R_xy (tau) = xcorr(y,x,maxlag)

Note that when the $x(t)$ is delayed by $\tau = T$, the correlation between $x(t)$ and $y(t)$ is maximum.

Write Up: For this section, plot $x(t)$ and $y(t)$ vs time. Explain why your technique works and the straightforward approach of finding the delay by plotting $y(t)$ vs time fails. Provide the plot of cross-correlation between x and y. Can you estimate the delay by the autocorrelation of y (i.e. xcorr(y,y))? Explain.

Bonus question is worth 2 marks, i.e. if you finish Lab4 without bonus question, you can get a max. of 6.25. If you answer bonus question, you can get up to 8.25.

Lab Report Template:

Do the write ups for each section as explained before. Attach the matlab codes.