



INTRODUCTION AND OVERVIEW:

This lab is designed to demonstrate – without the shadow of any doubt – how and why the spectral estimation works. There are two main parts to this, however, you will need to do little bit of reading to learn a few new tricks.

First, let us review: Random signals cannot be processed in time domain – because, well they are random! However, signals that satisfy the wide sense stationary requirement have certain statistical properties that do not change in time. Furthermore, if the ergodicity assumption is also satisfied, then the parameters can be estimated from a single observation of the random process.

In estimating the spectral content of a random signal, we cannot use standard FT based techniques for two reasons: 1) random signals are really an ensemble of signals, and we would get a different spectrum for each realization; and 2) a random signal often does not satisfy the Dirichlet conditions and hence their DTFT is undefined. Therefore, we “estimate” the spectrum as the DTFT of the autocorrelation function of the truncated signal. This spectrum, in essence, gives us the power spectral density of the random process, which is defined as the amount of power per frequency.

The easiest technique to estimate the PSD is through the periodogram method. According to this method you

- 1) Truncate the signal with a window (rectangular window is default)
- 2) Calculate the DFT of the random signal
- 3) Compute the magnitude square of the DFT
- 4) Divide this by N , the length of the signal
- 5) Divide by 2π or f_s to normalize

Matlab's `periodogram()` function implements exactly this procedure. This technique however, may suffer from the differences (variations) in the signal, particularly when the stationarity assumption is not satisfied. A better, and a more stable method is the Welch's method, where the signal is essentially divided into overlapping blocks, and the periodogram is applied separately to every block. The resulting PSDs are then averaged.

Both of these methods are known as *non-parametric* methods, there are no parameters or models that are assumed. There are also parametric models, where the random process is assumed to be generated from a polynomial (typically all poles) functions. There are several versions in Matlab, study (and implement) at least two such methods (you can use Matlab's build in functions for these methods, such as Burgs or Yule-Walker method)

In this lab, you will implement various such techniques to see how they can be used to estimate the signal's spectrum, particularly in the most hostile conditions, where the noise is substantially heavy.

LAB PROTOCOL:

1. You have been e-mailed the data file **very_noisy_signal.mat** from the class web page. This signal, as the name implies is, well, very noisy! You are to estimate the PSD of this signal using the standard periodogram and Welch methods. More specifically, you are asked to determine what spectral components the signal is probably made of. Bonus points, if you can estimate the uncorrupted time domain signal (before addition of noise). Write an .m file for your solution, which may use built-in Matlab functions.
2. Design a Simulink model to demonstrate the PSD estimation using
 - a. Manual method (steps 1 – 5 described above)
 - b. Standard periodogram (you may try different windows)
 - c. Welch’s method (you may try different windows)
 - d. One (or two) additional parametric techniques.

The inputs to your system should be two (or more) sinusoids at very close frequencies, for example, 200 and 203, sampled at 1kHz, and the noise amplitude should be at least 20 times that of the signal. Using various parameters (and also by playing with the frame size and simulation length), determine the best estimate of the PSD. Which technique appears to be doing better than the others? The block diagram of a “**partial**” model is provided below to get you started. The built-in PSD estimation blocks can be found in the DSP blockset

Make sure to explain the methods, in particular the new ones, in detail, and discuss your results.

PSD estimation using different methods - PARTIAL MODEL
(c) Robi Polikar, 20 April 2006, Rowan University,
Digital Signal Processing ECE 0909351

