Spectral Estimation

# Overview

In this project you will explore and compare various parametric and non-parametric techniques for power density spectrum (PDS) estimation. The goal is to approximate the PDS with non-parametric direct methods, such as the periodogram, periodogram averaging, and the non-parametric indirect Blackman-Tukey method. You will also investigate parametric all-pole modeling for PDS estimation. In the process, you will have a chance to sort through many issues involving finite length signals and the DFT.

Suggested Reading: Sections 10.6, 10.7, and 11.0 to 11.5.

You are given a signal  that is the result of passing white Gaussian noise with variance one () through a system.







 

Figure 1: Generating 

The goal is to estimate the PDS of,. Recall that:







Thus estimating  is equivalent to estimating. Your goal is to construct estimates of  from the samples of the colored noise.

The data for this project is contained in the file pj2data.mat.

Copy this file into the local directory where you are working. This file contains two vectors. The vector y is a 512-point vector representing the discrete time signal. The vector Hejw2 is a 512-point DFT representing samples of the magnitude-squared response. It is the desired response that you are trying to estimate from. Hejw2 is given to you as a baseline for error calculation.

You can load the data into your MATLAB environment using the load command. Just cd to the directory where you copied pj2data.mat and at the prompt type:

load pj2data

Your MATLAB environment will now have two vectors y and Hejw2 defined in it. **Warning!** If you had other variables named y and Hejw2 they will be overwritten by the load command.



Figure 2: Plots of the data in pj2data.mat

As a first exercise you may want to plot the data y and Hejw2. They are plotted above for your convenience. Note that the k-indices on Hejw2 represent sampled values in the interval  (i.e.).

In this project you are using the data  to construct estimates  of, where.

Error Criterion: The error criterion is defined as the discrete-squared-error in the frequency domain. An expression for the estimation error is given in (1):



Autocorrelation Estimate: The autocorrelation function of a wide-sense stationary signal $y[n]$ is defined as:



In this project, the autocorrelation estimate provided by the observed data of length  is:



Project Write-Up

This project contains several problems, each of which address some aspect of spectral analysis. The write-up should include a summary of how you went about the project, any relevant plots and, at minimum, address all the questions asked in each of the problems. The total, including your description and plots, should not exceed 14 pages (i.e., 14 sides).

In addition, this project will be graded not only on content and correctness, but also on neatness and style of presentation. Thus, although not a requirement, it is **recommended** that you typeset this report using Latex, Microsoft Word, or some other typesetting program.

You can save your figures off the figure window in MATLAB by clicking export and saving as .eps, .bmp, .tiff, etc. If embedding images into MSWord, be warned that JPEG quality is low. Bitmaps or TIFFs might be better than JPEGs.

## MATLAB Notes:

Functions you may find useful in this project include fft(), fftshift(), fliplr(), conv(), downsample(), xcorr(), toeplitz(), levinson(), freqz(), sum(), abs(), transpose(), ones(), zeros(), triang(). Look at the appropriate help files in MATLAB to see how to use them.

A. Autocorrelation and MATLAB

The function xcorr() in MATLAB calculates the finite autocorrelation between two sequences. Calculate the following sequence

= xcorr(y, y, 'biased')

This is similar to the convolution of two finite sequences  and. For the first part of this exercise, show this fact using the following example:

1. Take the first 32 points of  (we define this signal as).
2. Plot the autocorrelation of  using xcorr().
3. Then plot the autocorrelation using the conv() function.

The two plots you get should be the same, except for a scaling factor.

### Problem A.1

What does xcorr calculate if you replace 'biased' with 'unbiased'?

### Problem A.2

1. The deterministic autocorrelation of  is



where is the length of. Explain why the Fourier transform of this autocorrelation function is a positive, real function.

1. Take the 64-point DFT of  using fft command in MATLAB. Plot the absolute value and the phase of the DFT of. What is the relationship between the 64 point DFT of  and the 64 point DFT of?
2. Use  and generate a signal whose 64 point DFT, which is calculated by fft, is the same as the 64 point DFT of.

### Problem A.3

The PDS is the Fourier transform of the autocorrelation function defined in . The Fourier transform of  is an estimate of the PDS, which is the 64-point periodogram of  using a rectangular window. The periodogram can be calculated directly from  without first convolving it with itself: Take a 64-point DFT of  and then find its magnitude squared. This will give you a 64-point DFT representing.

For the last part of this exercise, plot three figures:

1. The absolute value of the 64-point DFT of , the deterministic autocorrelation sequence.
2. The magnitude squared of the 64-point DFT of.
3. The magnitude squared of the 64-point DFT of the first 64 points of.

What is the relationship between the three signals in A.3.a, A.3.b and A.3.c ?

# B. Nonparametric PDS Estimation

### Problem B.1

Estimate the PDS of  using only the first 32-points of the data. Do this by taking the 64-point periodogram. Plot the 64 DFT points of the desired frequency response, , and your estimate of the PDS at  on the same plot. Calculate the estimation error as given in .

### Problem B.2

Estimate the PDS using all 512 points. Do this by taking the 1024-point periodogram of. Similar to Problem B.1, plot the 64 DFT points of the desired frequency response, , and your estimate of the PDS at those frequencies on the same plot. Calculate the estimation error as given in .

### Problem B.3

Estimate the PDS using periodogram averaging. Write a script to find the 64-point periodogram of every 32 non-overlapping samples and average the results (i.e. there will be 512/32 = 16 periodograms to average). The expression for the periodogram averaging is:



Similar to Problem B.1, plot the 64 DFT points of the desired frequency response and your estimate on the same plot. Comment on any differences between this estimate and the previous two. Calculate the estimation error as given in .

### Problem B.4

In this problem you will use the indirect Blackman-Tukey method to estimate the PDS of. The Blackman-Tukey method requires you to first estimate the autocorrelation of . Use the following three steps to generate your Blackman-Tukey estimate:

1. Estimate the autocorrelation of  using and.
2. Truncate your autocorrelation estimate by using only  for  (i.e. Do not use the estimate of autocorrelation for.)
3. Take the 64-point DFT of the truncated autocorrelation function.

Similar to Problem B.1, plot 64 DFT points of the desired frequency response and your estimate on the same plot. Calculate the estimation error as given in .

### Problem B.5

Format the errors you found in Problems B.1-B.4 in a table and discuss your results.

1. Which method performed the best?
2. Explain why the Blackman-Tukey method does so well without any averaging.
3. Multiply your estimated autocorrelation obtained in part B.4.(2) with, a triangular window of size  (nonzero for) with unit center tap. Take a -point DFT of this new estimate. Calculate the estimation error as given in . Discuss why the performance is different than your result for problem (B.4).

# C. Parametric PDS Estimation

Estimate the PDS of  using all-pole modeling. Here the stable impulse response  in Figure 1 is modeled with a class of stable filters with the following structure



.

The goal is to fit an all-pole model to the data and use that model to estimate.

### Problem C.1

Formulate the Yule-Walker equations for 2nd through 7th order all-pole models, and solve for the coefficients to estimate. Use the values of  as estimated in Problem B.4. In a table, list the coefficients and the prediction error you found for. For which order is the prediction error minimum? (hint: check out the levinson command in MATLAB)

Draw the lattice implementation for.

### Problem C.2

In addition to finding the denominator coefficients, you must also find the gain factor. In the Yule-Walker estimation method, the minimum mean-squared value for  in the th-order predictor is given by



Plot the estimation error as defined in as a function of. You may find the command freqz() helpful. Comment on any interesting features of this error curve. What does this imply about the system function? What is your best estimate of the number of poles in? Compare the prediction error to the estimation error.

### Problem C.3

1. Another method to find the gain factor  is to use the Yule-Walker equations. In this case, the estimate of is given by:



Show how this equation is obtained. Use  and find the estimation error in . Compare s with the estimates s in Problem C.2.

1. Another estimator is defined based on the prediction error signal,. The estimator  is



Explain how this estimator is obtained. Repeat C.3.a for this estimator.

### Problem C.4

MATLAB contains functions called

1. periodogram(),
2. pwelch(),
3. pyulear().

View the help files for these functions and explore their use in the context of Problems B.1, B.2, B.3 and C. Show how these commands can provide the same results which you obtained for these relevant problems . **Do not use these functions to find answers in the previous problems in the project**.