

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 $y[n]$ that is the result of passing white Gaussian noise with variance one ($\sigma_x^2 = 1$) through a system $h[n]$.

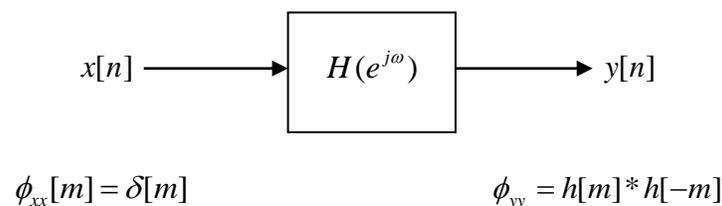


Figure 1: Generating $y[n]$

The goal is to estimate the PDS of $y[n]$, $\Phi_{yy}(e^{j\omega})$. Recall that:

$$\Phi_{yy}(e^{j\omega}) = |H(e^{j\omega})|^2 \Phi_{xx}(e^{j\omega})$$

$$\Phi_{xx}(e^{j\omega}) = \sigma_x^2 = 1$$

$$\Phi_{yy}(e^{j\omega}) = |H(e^{j\omega})|^2$$

Thus estimating $\Phi_{yy}(e^{j\omega})$ is equivalent to estimating $|H(e^{j\omega})|^2$. Your goal is to construct estimates of $|H(e^{j\omega})|^2$ from the samples of the colored noise $y[n]$.

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 $y[n]$. The vector H_{ejw2} is a 512-point DFT

representing samples of the magnitude-squared response $|H(e^{j\omega})|^2$. It is the desired response that you are trying to estimate from $y[n]$. $Hejw2$ 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.

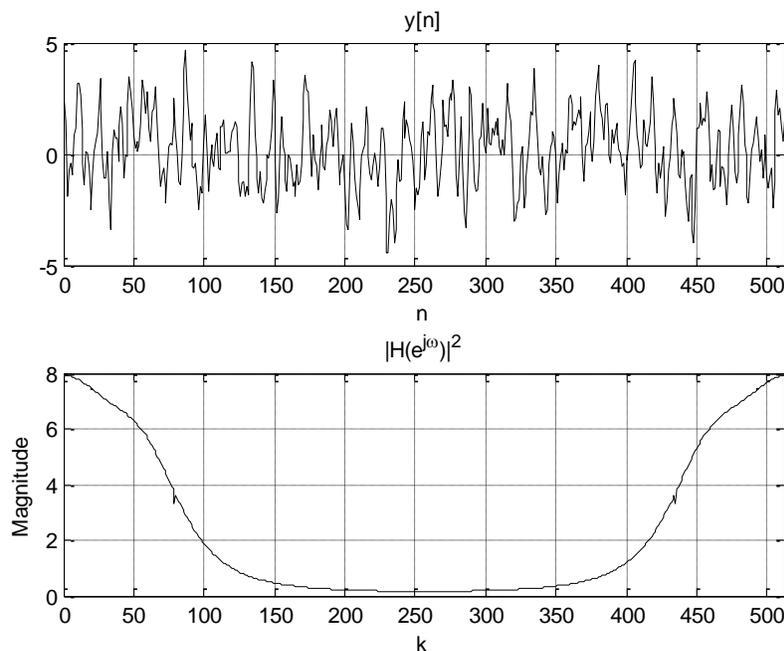


Figure 2: Plots of the data in `pj2data.mat`

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 ω values in the interval $[0, 2\pi)$ (i.e. $\omega_k = 2\pi k / 512$).

In this project you are using the data `y[n]` to construct estimates $|\hat{H}(e^{j\omega_k})|^2$ of $|H(e^{j\omega_k})|^2$, where

$$\omega_k = \frac{2\pi k}{64}.$$

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):

$$\varepsilon = \frac{1}{64} \sum_{k=0}^{63} \left| \left| \hat{H}(e^{j\frac{2\pi k}{64}}) \right|^2 - \left| H(e^{j\frac{2\pi k}{64}}) \right|^2 \right|^2 \quad (1)$$

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

$$\phi_{yy}[m] = E\{y[n+m]y[n]\} \quad (2)$$

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

$$\hat{\phi}_{yy}[m] = \frac{1}{N} \sum_{n=0}^N y[n+m]y[n] \quad (3)$$

MATLAB Notes:

Functions you may find useful in this project include `fft()`, `fftshift()`, `flipplr()`, `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

$$c_{yy} = \text{xcorr}(y, y, 'biased')$$

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

1. Take the first 32 points of $y[n]$ (we define this signal as $y_1[n]$).
2. Plot the autocorrelation of $y_1[n]$ 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

- (a) The deterministic autocorrelation of y_1 is

$$\phi_{y_1 y_1}^d[m] = \frac{1}{N_{y_1}} \sum_{n=0}^{N_{y_1}-1} y_1[n+m]y_1[n] \quad (4)$$

where $N_{y_1} = 32$ is the length of y_1 . Explain why the Fourier transform of this autocorrelation function is a positive, real function.

- (b) Take the 64-point DFT of $c_{y_1 y_1}[m]$ using `fft` command in MATLAB. Plot the absolute value and the phase of the DFT of $c_{y_1 y_1}[m]$. What is the relationship between the 64 point DFT of $c_{y_1 y_1}[m]$ and the 64 point DFT of $\phi_{y_1 y_1}^d[m]$?
- (c) Use $c_{y_1 y_1}[m]$ and generate a signal whose 64 point DFT, which is calculated by `fft`, is the same as the 64 point DFT of $\phi_{y_1 y_1}^d[m]$.

Problem A.3

The PDS is the Fourier transform of the autocorrelation function defined in (2). The Fourier transform of $\phi_{y_1 y_1}^d[m]$ is an estimate of the PDS, which is the 64-point periodogram of $y[n]$ using a rectangular window. The periodogram can be calculated directly from $y_1[n]$ without first convolving it with itself: Take a 64-point DFT of $y_1[n]$ and then find its magnitude squared. This will give you a 64-point DFT representing $|Y_1(e^{j\omega_k})|^2$.

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

- (a) The absolute value of the 64-point DFT of $\phi_{y_1 y_1}^d[m]$, the deterministic autocorrelation sequence.
- (b) The magnitude squared of the 64-point DFT of $y_1[n]$.
- (c) The magnitude squared of the 64-point DFT of the first 64 points of $y[n]$.

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 $y[n]$ 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, $|H(e^{j\omega})|^2$, and your estimate of the PDS at $\omega_k = \frac{2\pi k}{64}$ on the same plot. Calculate the estimation error as given in (1).

Problem B.2

Estimate the PDS using all 512 points. Do this by taking the 1024-point periodogram of $y[n]$. Similar to Problem B.1, plot the 64 DFT points of the desired frequency response, $|H(e^{j\omega})|^2$, and your estimate of the PDS at those frequencies on the same plot. Calculate the estimation error as given in (1).

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:

$$|\hat{H}(e^{j\omega})|^2 = \frac{1}{16} \sum_{i=1}^{16} |Y_i(e^{j\omega})|^2$$

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 (1).

Problem B.4

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

- (1) Estimate the autocorrelation of $y[n]$ using (3) and $N = 512$.
- (2) Truncate your autocorrelation estimate by using only $\hat{\phi}_{yy}[m]$ for $0 \leq |m| \leq 15$ (i.e. Do not use the estimate of autocorrelation for $|m| > 15$.)
- (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 (1).

Problem B.5

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

- (a) Which method performed the best?
- (b) Explain why the Blackman-Tukey method does so well without any averaging.
- (c) Multiply your estimated autocorrelation obtained in part B.4.(2) with $w[n]$, a triangular window of size 31 (nonzero for $-15 \leq m \leq 15$) with unit center tap. Take a 64-point DFT of this new estimate. Calculate the estimation error as given in (1). Discuss why the performance is different than your result for problem (B.4).

C. Parametric PDS Estimation

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

$$\hat{H}^{(p)}(e^{j\omega}) = \frac{A}{1 + \sum_{k=1}^p a_k^{(p)} e^{-j\omega k}} \quad (5)$$

The goal is to fit an all-pole model to the data and use that model to estimate $\Phi_{yy}(e^{j\omega})$.

Problem C.1

Formulate the Yule-Walker equations for 2nd through 7th order all-pole models, and solve for the coefficients to estimate $\Phi_{yy}(e^{j\omega})$. Use the values of $\hat{\phi}_{yy}[m]$ as estimated in Problem B.4. In a table, list the coefficients and the prediction error you found for $p = 2, \dots, 7$. For which order is the prediction error minimum? (hint: check out the `levinson` command in MATLAB)

Draw the lattice implementation for $p = 3$.

Problem C.2

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

$$\hat{A}_1^{(p)} = \sqrt{\hat{\phi}_{yy}[0] \prod_{k=1}^p (1 - |a_k^{(k)}|^2)}$$

Plot the estimation error as defined in (1) as a function of p . You may find the command `freqz()` helpful. Comment on any interesting features of this error curve. What does this imply about the system function $H(e^{j\omega})$? What is your best estimate of the number of poles in $H(e^{j\omega})$? Compare the prediction error to the estimation error.

Problem C.3

- (a) Another method to find the gain factor A is to use the Yule-Walker equations. In this case, A_2 the estimate of A is given by:

$$\hat{A}_2^{(p)} = \sqrt{\hat{\phi}_{yy}[0] + \sum_{k=1}^p a_k^{(p)} \hat{\phi}_{yy}[k]} \quad (6)$$

Show how this equation is obtained. Use $\hat{A}_2^{(p)}$ and find the estimation error in (1). Compare $\hat{A}_2^{(p)}$ s with the estimates $\hat{A}_1^{(p)}$ s in Problem C.2.

- (b) Another estimator is defined based on the prediction error signal, $\hat{w}_p[n]$. The estimator \hat{A}_3 is

$$\hat{A}_3^{(p)} = \sqrt{\frac{1}{N} \sum_{n=1}^N \|\hat{w}_p[n]\|^2} \quad (7)$$

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

Problem C.4

MATLAB contains functions called

- (a) `periodogram()`,
- (b) `pwelch()`,
- (c) `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.**