Stochastic Signal Processing HW 3

Stefanos Baklavas 1115201700093

07 July 2022

Outline

1. Exercise 1
   - Periodogram
   - AR
2. Exercise 2
   - Adaptive filtering
   - LMS Algorithm
3. Exercise 3
   - Noise cancellation with Wiener filter

1. Exercise 1
    - Periodogram
    - AR
2. Exercise 2
3. Exercise 3

Input signal

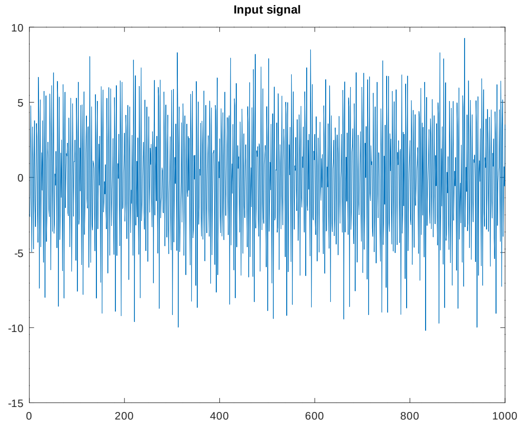

Figure: Input signal from 1.mat

Periodogram

- The power spectrun $S_x(w)$ is used to find the frequency distribution of a signal.
- We use the periodogram in order to estimate $S_x(w)$
- Periodogram is defined as:

$$P_{x,per}(w,N) = \sum_{k=-N+1}^{N-1} r_x(k,N)exp(-jwk)$$

- We use $E[r_x(k,N)] = w(k,N)r_x(k)$ to estimate the cross-correlation factor.
- And so we finally have:

$$E[P_{x,per}(w,N)] = \frac{1}{2\pi}W(w,N) * S_x(w)$$

- $E[P_{x,per}(w,N)]$ is an asymptotically unbiased estimator. Also in our case w[n] is the rectangular window.

Periodogram Results

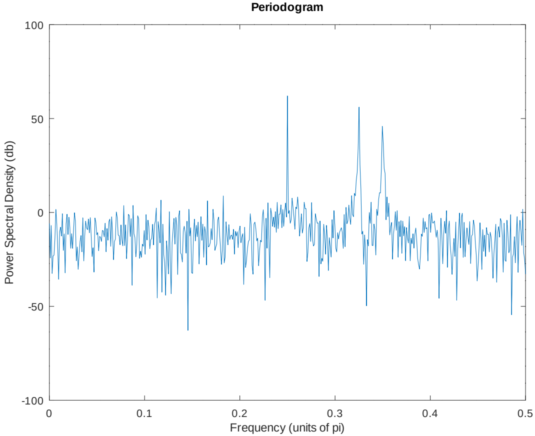

Figure: f1 = 0.25, f2 = 0.32520, f3 = 0.34961

AR

- In this exercise we estimate $S_x(w)$ using Yule-Walker equations for AutoRegressive (AR) processes.
- The form of Yule-Walker equations for AR processes with p poles and q = 0 is:

$$x[n] + \sum_{k=1}^{p} a_p[k]x[n-k] = b[0]u[n] => S_x(w) = \sigma_u^2 \frac{|b[0]|^2}{|A_p(w)|^2}$$

- Where $a_p[k]$ contains the poles of the transfer function H(z), u[n] is the input noise and x[n] is the output .
- We use pyulear() function to estimate the Autoregressive power spectral density of the input signal using Yule-Walker method
- The difference between the two methods is that periodogram estimates $S_x(w)$ using a window and Yule-Walker equations estimete $S_x(w)$ via the transfer function H(z).

AR Results

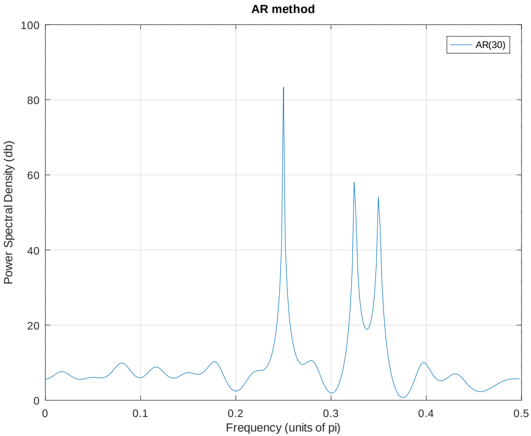

Figure: f1 = 0.25, f2 = 0.32522, f3 = 0.34961

1. Exercise 1
2. Exercise 2
    - Adaptive filtering
    - LMS Algorithm
3. Exercise 3

Adaptive Filtering

Adaptive filtering for non WSS processes is a method where:

- Given an input signal x[n] = d[n] + u[n]
- And a filter w[n]
- We try to adjust w[n] coefficients such as the output d'[n] = w[n]*x[n] is the initial denoised signal d[n]

We use adaptive filtering instead of a stationary filter because in this exercise we have non WSS processes.

LMS Algorithm

The asset of the LMS Algotithm is its simplicity in the renewal of its coefficients.

- Updating kth coefficient

$$w_{n+1}[k] = w_n[k] + e[n] * x[k - n]$$

- Where e[n] = d[n] − y[n]. d[n] is the desired output and y[n] is the output we get for each iteration

LMS Algorithm

The Algorithm as implemented for 2.mat file:

Parameters:
p = filter order
μ = step
x = input vector from 2.mat
y = output vector from 2.mat
Initialization:
w[0] = $zeros(p)^T$
Computation:
$x[n] = [x[n], x[n-1], ..., x[n-p+1]]^T$
$e[n] = y[n] - w[n] * x[n]$
$w[n+1] = w[n] + (\mu e^* x[n])^T$

---

where $0 < \mu < \frac{2}{(p+1)r_x(0)}$

Exercise 2 Results 1/3

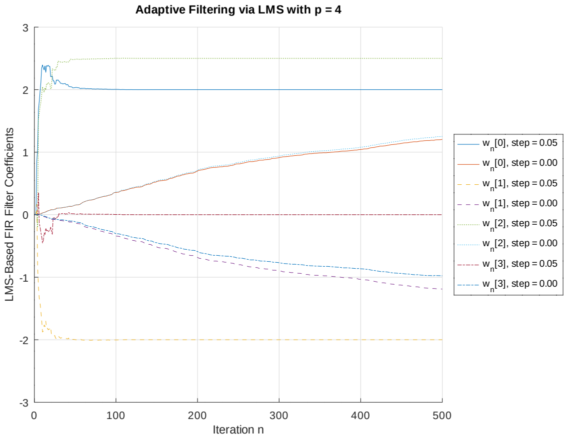

Figure:

Exercise 2 Results 2/3

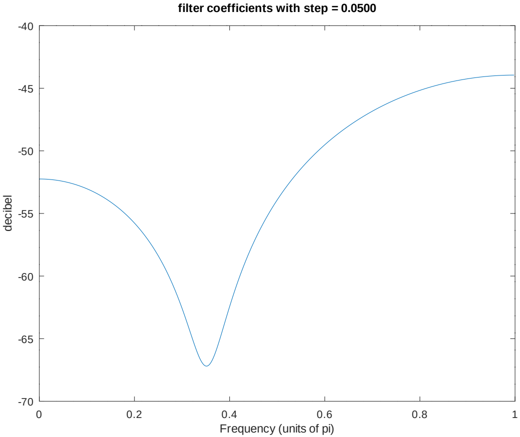

Figure: m1 = 0.05

Exercise 2 Results 3/3

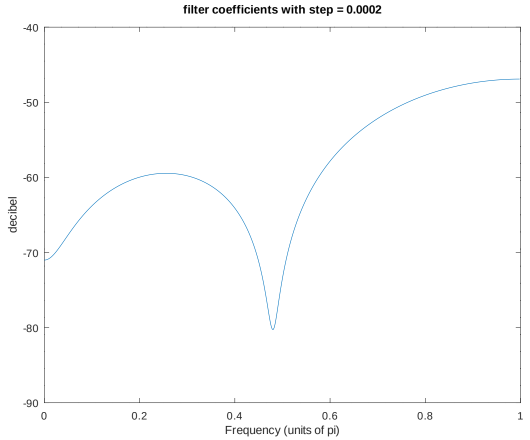

Figure: m2 = 0.0002 according to lmax

1 Exercise 1

2 Exercise 2

3 Exercise 3

Noise cancellation with Wiener filter

Block Diagram

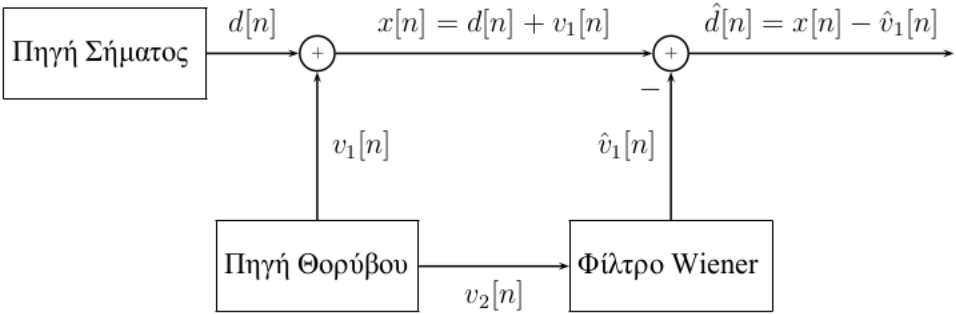

Figure: Noise cancellation with Wiener filter

Wiener-Hopf equations

- The Wiener-Hopf equations for this case are:

$$R_{u2}w = r_{u1u2}$$

- And because $r_{u1u2} = r_{xu2}$ the final form of Wiener-Hopf equations is:

$$R_{u2}w = r_{xu2}$$

Resulsts 1/2

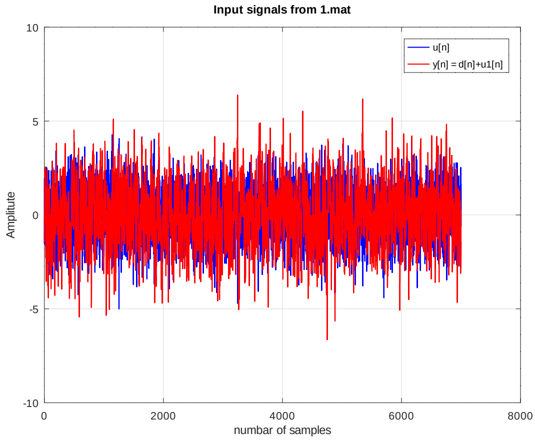

Figure: Input signals from 3.mat

Resulsts 2/2

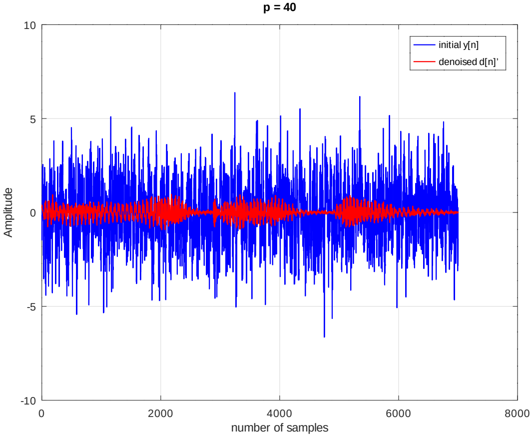

Figure: Noise cancellation with Wiener filter

The End !

Thank you for your attention!