Power Spectral Estimation

Semester Project of EEE 506 Digital Spectral Analysis

Lian Lu
Arizona State University

Due on May 5th, 2015

Contents

Chapter 1

Introduction

Power spectral estimation has been an important topic of the digital signal processing. One interesting application of power spectral estimation is speech recognition, where the estimated power spectral can be used as a preliminary measurement to perform speech bandwidth reduction and further acoustic processing.

In this report, we implement various power spectral estimation methods. By testing them on different data sets (including noise-free signal and noise signal), we compare their performance. Besides, we also tune each methods parameters (e.g., the model order), illustrating the impacts of those parameters.

1.1 Overview of the methods

We summarize the PSD methods used in this report as follows

- Blackman-Tukey PSD estimates
- Welch Periodogram PSD estimates
- Yule-Walker PSD estimates
- Burg (Harmonic) PSD estimates
- Covariance PSD estimates
- Modified Covariance PSD estimates
- Adaptive Least Mean Squares (LMS) PSD estimates
- MUSIC PSD estimates

Among them, the first two belong to the category of classical approach (or direct approach), where they directly deal with the input data; the last one (i.e., the MUSIC PSD) is the eigen-analysis based. The remaining methods are all parametric estimation based approach. In particular, they model the signals by a system, and by figuring out the system parameters, the PSD of the signal can be obtained (see details in the corresponding chapters).

Most of the methods used in this report has tunable settings. For example, we can select biased/unbiased autocorrelation estimates, different model orders and/or different signal subspace dimensions. Therefore, for each method, we also adjust these parameters and see the impacts.

1.2 Test signals

There are two signals we will use to test the PSD estimate method, both are with length 128 time slots. The sampling frequency is $f_s = 64$ *sample/second*.

Signal #1 is

$$x_1(n) = \sin\left(\frac{2\pi \cdot 5}{f_s} \cdot (n-1)\right) + \sin\left(\frac{2\pi \cdot 15}{f_s} \cdot (n-1)\right) + \sin\left(\frac{2\pi \cdot 16}{f_s} \cdot (n-1)\right), 1 \leq n \leq 128.$$

Note that x_1 is consisted of unit-amplitude sinusoids of $5Hz, 15Hz, 16Hz$. We denote the above case the un-noised case, short for under this case, the signal is noise free. The above signal can be plotted in Figure 1.2.1.

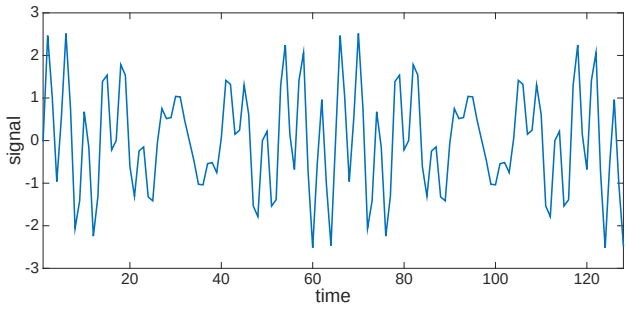


Figure 1.2.1: Signal $x_1(n)$

Signal #2 is

$$x_2(n) = x_1(n) + v(n), 1 \leq n \leq 128.$$

In it, $v(n)$ is the white noise. We denote the above case the noised case. In this report, we assume $x_2(n)$ is the signal with SNR 0db. Because the amplitude of $x_1(n)$ is 3, the $w(n)$ is with variance 3. To avoid randomness impact on the SNR, we generate $x_2(n)$ repeatedly, until the current generated $snr(x_2) \in [0-\epsilon, 0+\epsilon]$. Note that this can be done by using the function $snr(x_1, x_2 - x_1)$, which is provided by Matlab, computing the desired SNR. One sample path of $x_2(n)$ is plotted in Figure 1.2.2.

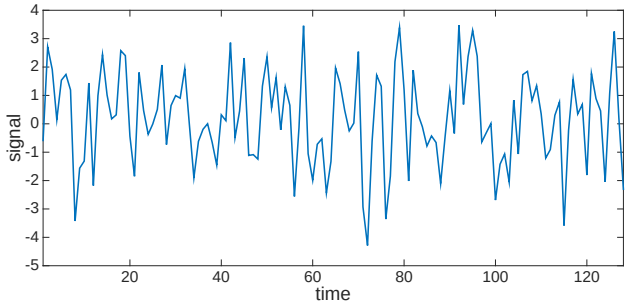


Figure 1.2.2: Signal $x_1(n)$

1.3 PSD Format

We will exam the PSD of the signals x_1 and x_2 in the range of $-\frac{f_s}{2}$ to $\frac{f_s}{2}$. There will be 4096 points in the frequency range. The power density is measured in unit db, while the frequency is measured in unit Hz.

1.4 Simulation System Settings

In this report, we carry out the simulation in Matlab 2014_b. The generated source codes are compatible with Matlab 2012b. The simulation environment system is Apple Mac OS Version 10.10.3. The figures in this report are all EPS format, such that the resolutions are independent of zooming in/out.

Chapter 2

Blackman-Tukey PSD estimates

2.1 Theory backgrounds

Blackman-Tukey PSD estimate method is a correlogram PSD estimate one, which is based on Wiener–Khinchin theorem, stating that *the autocorrelation function of a wide-sense-stationary random process has a spectral decomposition given by the power spectrum of that process.* The true autocorrelation function is typically difficult to obtain, thus we can use a "unbiased" estimate $\hat{r}_{xx}[m]$ by

$$\check{r}_{xx}[m] = \frac{N-|m|}{N}\hat{r}_{xx}[m], \tag{2.1.1}$$

where $\check{r}_{xx}[m]$ is the biased estimate defined as

$$\check{r}_{xx}[m] = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-m-1} x[n+m]x^*[n], & if\ 0 \le m \le N-1 \\ \frac{1}{N}\sum_{n=0}^{N-|m|-1} x[n+|m|]x^*[n], & if\ 0 \le -m \le N-1 \end{cases}. \tag{2.1.2}$$

However, if we use the complete duration of $\hat{r}_{xx}[m]$ from $-(N-1)$ to $(N-1)$, it can be shown the PSD tends to be unstable, due to high statistical variance associated with the high lags of the autocorrelation estimate. To handle this, we add a window function $w[m]$ (e.g., a hamming window), such that we only compute PSD based on lags from $-L$ to L, where L is much less than N. To summarize, we have the formula of Blackman-Tukey PSD estimate

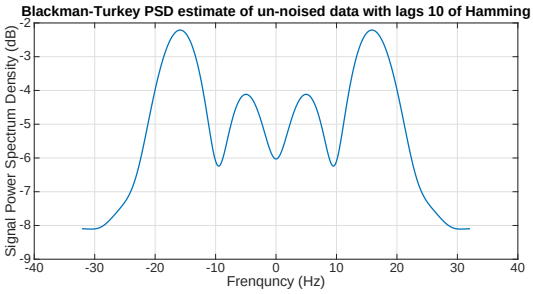
$$\hat{P}_{BT}(f) \quad = \quad T\sum_{m=-L}^{L} w[m]\cdot\hat{r}_{xx}[m]\cdot\exp\left(-j2\pi\cdot fmT\right). \tag{2.1.3}$$

Note that the above PSD estimate is quite general, due to the windowing function $w[m]$.

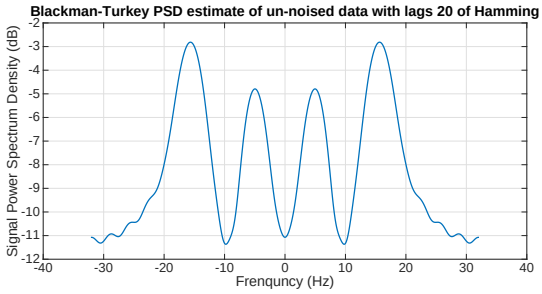
2.2 PSD estimation results

We implement the above algorithm, using a hamming window. Note that we averaged the FFT results by the window size, such that the PSD amplitude is normalized, and fixed at around 0 db. Otherwise, the PSD amplitude varies a lot among different settings and different methods.

We first exam the noise-free case in Figure 2.2.1. By comparing the subfigures, we see large lags will lead to a higher resolution of the PSD, however less stable (cannot be seen directly in the figure). Also note that



(a) Hamming window with lags 10

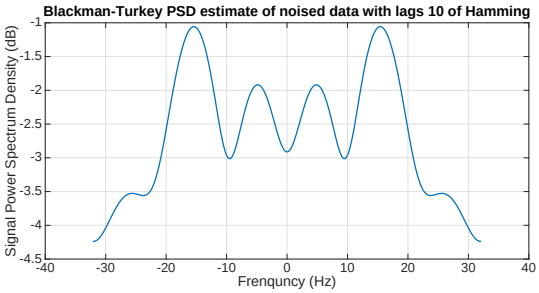


(b) Hamming window with lags 20

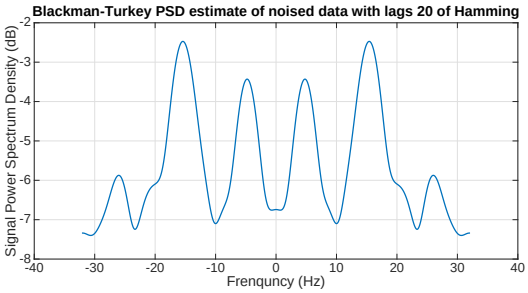
Figure 2.2.1: Blackman-Turkey PSD of noise-free signal

the applied method successfully identifies the power spectrum at $5Hz$ and $15 - 16Hz$. However, it fails to distinguish between 15 and 16 Hz.

We next exam the noised case in Figure 2.2.2. We see there is few differences between the noise-free case and noised case. This shows the applied method is robust under noised case. However, as the noise-free case, it still failed to identify there are two power spectrum peaks at 15 and 16 Hz.



(a) Hamming window with lags 10



(b) Hamming window with lags 20

Figure 2.2.2: Blackman-Turkey PSD of noised signal

Chapter 3

Welch Periodogram PSD estimates

3.1 Theory backgrounds

To improve the PSD estimate performance, we always hope to use more data points. Thus, Welch Periodogram PSD estimate is proposed. Basically, we decompose the signal data into several overlapped segments, and we compute the periodogram on each segments. After it, we average the periodogram estimate on each segments and normalize the obtain estimate. By using Welch method, the effects of side-lobes will be decreased, and the resolution is also decreased.

Next, we introduce the Welch method.

Assume that the data record is with length N. We divide the entire data record by segments, where each segment is with length D. The adjacent segments are overlapped with S ($S < D$) points (or there is a S point shift between adjacent segments).

Under the above division, we will obtain P segments, where

$$P = \left\lfloor \frac{N-D}{S+1} \right\rfloor . \tag{3.1.1}$$

The pth segments can be written as

$$x^p[n] = w[n] \cdot x[n + p \cdot S], 0 \leq n \leq D-1, \tag{3.1.2}$$

where $w[n]$ is a optional windowing function (e.g., Hamming window).

Next, on segment p, we have its periodogram PSD

$$\widetilde{P_{xx}^p}(f) = \frac{1}{UDF} \left| X^p(f) \right|^2 ,$$

where

$$X^p(f) \quad = \quad T \sum_{n=0}^{D-1} x^p[n] \exp\left(-j2\pi f n T\right), \tag{3.1.3}$$

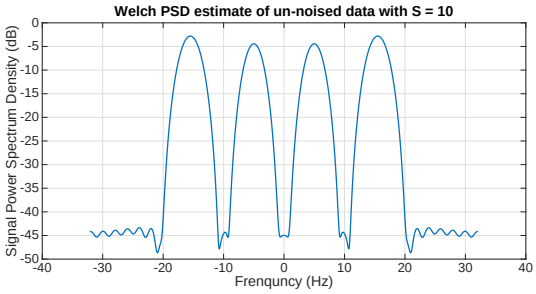
and $U = T \sum_{n=0}^{D-1} w^2[n]$ is the window energy.

At last, we average the PSDs in the following manner

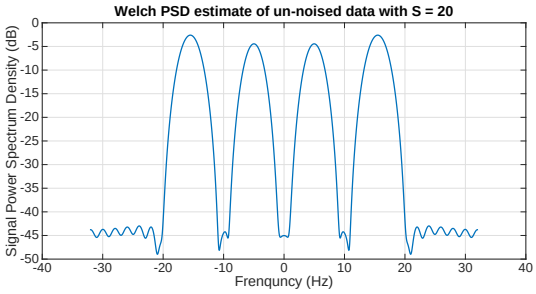
$$\hat{P_w}(f) \quad = \quad \frac{1}{P} \sum_{p=0}^{P-1} \widetilde{P_{xx}^p}(f). \tag{3.1.4}$$

3.2 PSD estimation results

We first exam the noise free signal case in Figure 3.2.1. We see the side-lobes are depressed significantly (45 dB) by using Welch method. For the data used in this report, there is nearly no difference by using shift 10 and 20. Overall, the applied method find the power spectrum peaks at 5 Hz and peaks between 15 and 16 Hz, but cannot distinguish between the peaks between 15 and 16 Hz.



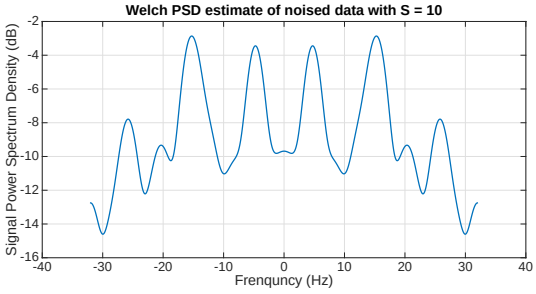
(a) A shift of 10



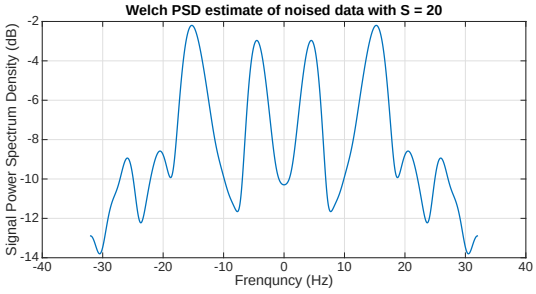
(b) A shift of 20

Figure 3.2.1: Welch Periodogram PSD of noise-free signal

We next exam the noised signal case in Figure 3.2.2. We see there are remarkable differences comparing with the noise-free case, where the "noised peaks" on the PSD estimate. This is reasonable, considering the SNR is pretty low. $S = 20$ depressed such noised peaks better than $S = 10$ case.



(a) A shift of 10



(b) A shift of 20

Figure 3.2.2: Welch Periodogram PSD of noised signal

Chapter 4

Yule-Walker PSD estimates

In this chapter, we begin Yule-Walker PSD estimate, which is the first parameter-based approach in this report.

The parameter-based approach can be explained as follows. Consider a system with transfer function $H(z) = \frac{1}{A(z)}$, where $A(z) = [1\ a_1\ a_2\ ...\ a_p] \cdot [1\ z^{-1}\ z^{-2} ... z^{-p}]^T$. If the driving signal is the white noise with PSD ρ_w, then the output signal PSD will be

$$\frac{T\rho_w}{|A(z)|^2_{z=e^{jw}}}. \tag{4.0.1}$$

Thus, we only need to adjust the parameters $[1\ a_1\ a_2\ ...\ a_p]$, such that the output signal matches the signal that needs the PSD estimates, and we can use (4.0.1) to compute its PSD. The above system is called an autoregressive system, since the current output is only based on p past outputs.

We note that in a more general way, we can use the system function

$$H(z) = \frac{B(z)}{A(z)}, \tag{4.0.2}$$

which makes the system a ARMA system. However, it brings significant computation complexity. Thus, in this report, we focus on the AR system model.

4.1 Theory backgrounds

Yule-Walker PSD estimate is based on AR model. We first build the Yule-Walker equations as follows. Given model order p, the system model is $A(z) = 1 + a_1 z^{-1} + ... + a_p z^{-p}$. Given data $x[n]$, we have

$$x[n] \quad = \quad -\sum_{k=1}^{p} a[k]x[n-k]. \tag{4.1.1}$$

By multiplying $x^*[n-m]$ on each side of the above equations and taking expectations, we have

$$\underbrace{\begin{bmatrix} r_{xx}[0] & r_{xx}[-1] & ... & r_{xx}[-p] \\ r_{xx}[1] & r_{xx}[0] & ... & r_{xx}[-p+1] \\ ... & ... & ... & ... \\ r_{xx}[p] & r_{xx}[p-1] & ... & r_{xx}[0] \end{bmatrix}}_{Matrix\ R} \begin{bmatrix} 1 \\ a[1] \\ ... \\ a[p] \end{bmatrix} = \begin{bmatrix} \rho_w \\ 0 \\ ... \\ 0 \end{bmatrix} \tag{4.1.2}$$

In it, matrix R is both Toeplitz and Hermitian since $r_{xx}[-k] = r_{xx}^*[k]$.

Given R, we can solve the above equation, possibly by using Levinson algorithm, and obtain the $[1\ a_1\ a_2\ ...\ a_p]$ and ρ_w.

Next, the remaining problem is to determine the matrix R. We directly plug in the autocorrelation estimates by $\check{r}_{xx}[m]$ or $\hat{r}_{xx}[m]$,

$$\check{r}_{xx}[m] = \frac{N - |m|}{N}\hat{r}_{xx}[m], \tag{4.1.3}$$

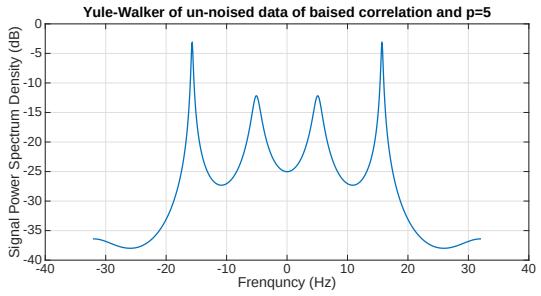
where $\check{r}_{xx}[m]$ is the biased estimate defined as

$$\check{r}_{xx}[m] = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-m-1} x[n+m]x^*[n], & if\ 0 \le m \le N-1 \\ \frac{1}{N}\sum_{n=0}^{N-|m|-1} x[n+|m|]x^*[n], & if\ 0 \le -m \le N-1 \end{cases}. \tag{4.1.4}$$

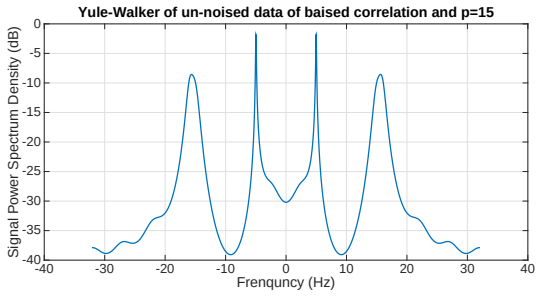
Note that if we use a biased estimate $\check{r}_{xx}[m]$, a stable AR filter will be obtained, while it is not true in general for unbiased estimate $\hat{r}_{xx}[m]$.

4.2 PSD estimation results

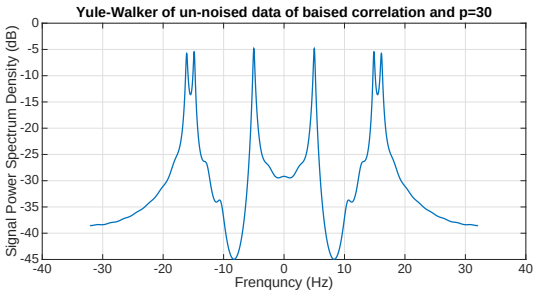
We exam the performance of Yule-Walker method in this section. We first check the impact of the biased/unbiased correlation estimate. By comparing Figure 4.2.1 and Figure 4.2.2. We see that model order 5 is not sufficient for both to indentify the power at 15 Hz and 16 Hz, because they are so near to each other. At order 15, the unbiased estimate sucessfully finds the above mentioned two peaks, while the biased estimate fails to do so. At order 30, biased estimate achieves a good PSD result, while unbiased estimate gets a messed up one.



(a) Model order = 5

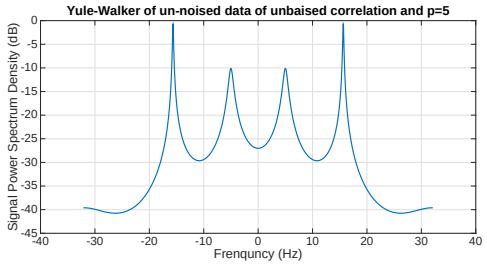


(b) Model order = 15

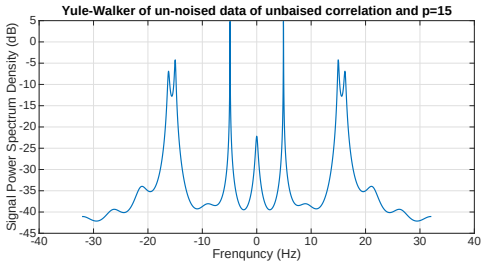


(c) Model order = 30

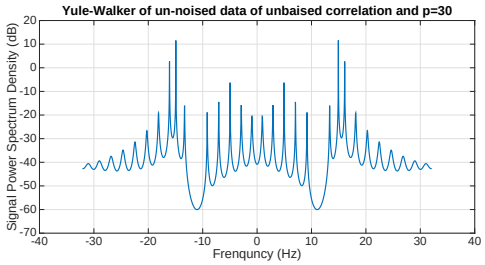
Figure 4.2.1: Yule-Walker PSD estimates of noise-free signal, biased estimate of correlation



(a) Model order = 5



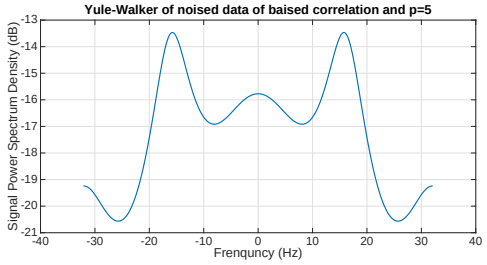
(b) Model order = 15



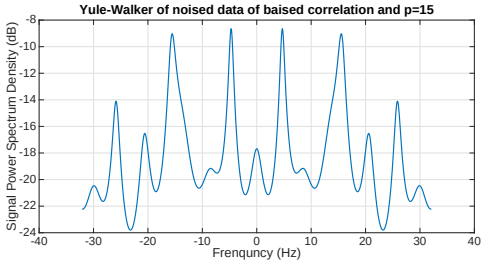
(c) Model order = 30

Figure 4.2.2: Yule-Walker PSD estimates of noise-free signal, unbiased estimate of correlation

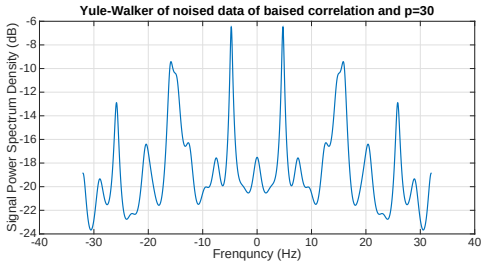
Next, we look at the noise case. At noised case, we note that both biased estimate and unbiased estimate lead to a similar result. And model order higher than 15 performs a good result. However, they are both not very good against noise, since they fail to identify 15 and 16 Hz peaks.



(a) Model order = 5

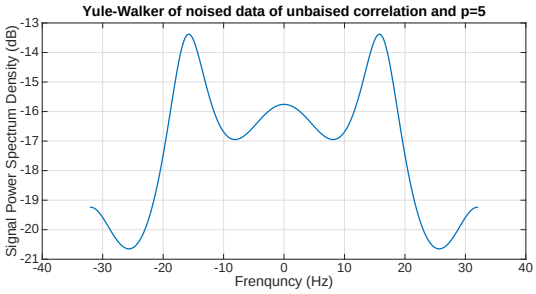


(b) Model order = 15

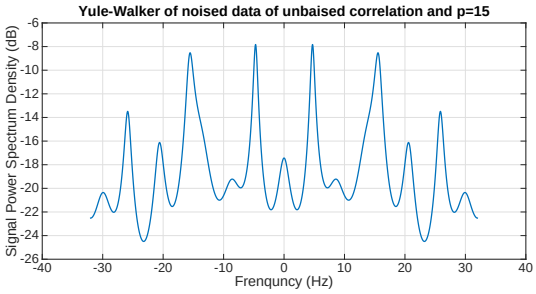


(c) Model order = 30

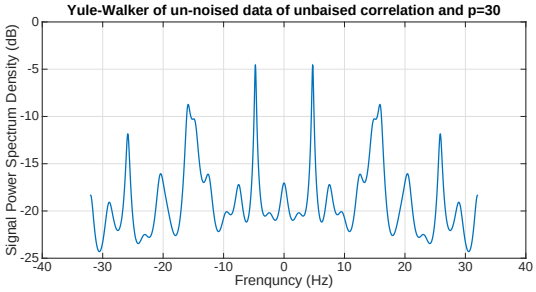
Figure 4.2.3: Yule-Walker PSD estimates of noised signal, biased estimate of correlation



(a) Model order = 5



(b) Model order = 15



(c) Model order = 30

Figure 4.2.4: Yule-Walker PSD estimates of noised signal, unbiased estimate of correlation

Chapter 5

Burg (Harmonic) PSD estimates

5.1 Theory backgrounds

Burg PSD method is a *reflection coefficient estimate* method. Such methods build on the observation that the AR model can be completed captured by the reflection coefficients K_p, where

$$K_p = a_p[p]. \tag{5.1.1}$$

This means we only need to find all reflection coefficients $a_q[q]$, $q = 1, 2, ..., p$.

In particular, we have the Levinson recrusion as

$$a_p[n] = a_{p-1}[n] + K_p a^*_{p-1}[p-n], 1 \leq n \leq p-1 \tag{5.1.2}$$

and

$$\rho_p = \rho_{p-1}(1 - |K_p|^2). \tag{5.1.3}$$

From the above, we have the recursive relationship between the prediction errors

$$e^f_p[n] = e^f_{p-1}[n] + K_p e^b_{p-1}[n-1] \tag{5.1.4}$$

$$e^b_p[n] = e^b_{p-1}[n-1] + K^*_p e^f_{p-1}[n] \tag{5.1.5}$$

The way of computing K_p is the main difference among different reflection coefficient estimates. In Burg method (also called Maximum entropy method), a harmonic estimate of K_p is adopted as follows

$$K_p = \frac{-2\sum_{n=p+1}^{N} e^f_{p-1}[n] \cdot e^{b*}_{p-1}[n-1]}{\sum_{n=p+1}^{N} \left|e^f_{p-1}[n]\right|^2 + \sum_{n=p+1}^{N} \left|e^b_{p-1}[n-1]\right|^2}. \tag{5.1.6}$$

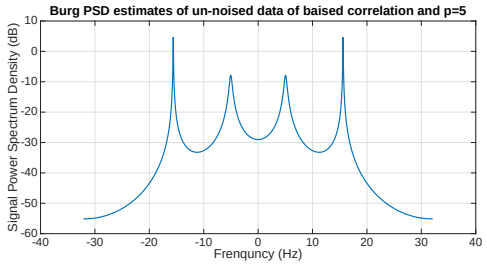
The above is called the harmonic mean of the forward and backward prediction errors. Further, it is obtained by minimizing the arithmetic mean of the forward and backward linear prediction errors ρ^{fb}_p.

5.2 PSD estimation results

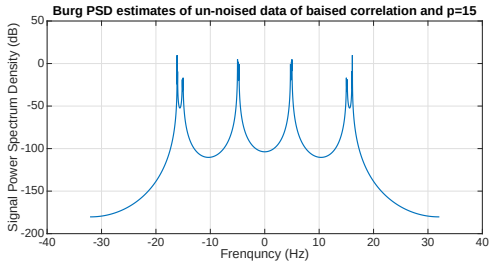
We implement the burg method. First check the noise free data in Figure 5.2.1. We see at order 15, the applied method has achieved a pretty good estimate, finding the peaks at 15 and 16 Hz. This has never

been as in previous chapters. Also, it shows a very good depression on other non-peak frequencies (around 100 dB).

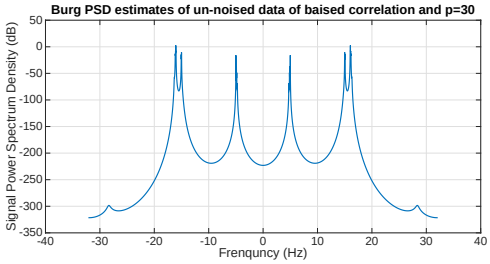
Next, we turn to noised signal in Figure 5.2.2. Even though the applied method fails to find the peaks at 15 and 16 Hz, it still depresses non-peak areas remarkably. As order increases, the false positive peaks appear more often.



(a) Model order = 5

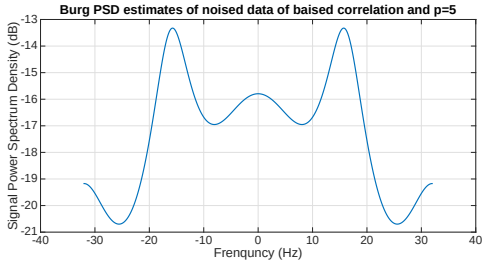


(b) Model order = 15

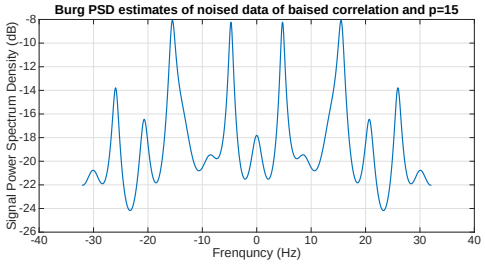


(c) Model order = 30

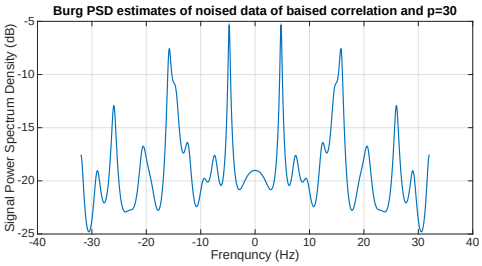
Figure 5.2.1: Burg (Harmonic) PSD estimates of noise-free signal



(a) Model order = 5



(b) Model order = 15



(c) Model order = 30

Figure 5.2.2: Burg (Harmonic) PSD estimates of noise signal

Chapter 6

Covariance PSD estimates

6.1 Theory backgrounds

The covariance PSD estimate is the non-window case of the least square error prediction method. In this track, we treat the AR model as a prediction filter. Therefore, the forward linear prediction error can be written as

$$e_p^f[n] = x[n] + \sum_{k=1}^{p} a_p^f[k]x[n-k]. \tag{6.1.1}$$

If we only use the available data records (meaning we do not assume the data unavailable is 0), then $e_p^f[n]$ can be computed from $p+1$ to N. We summarize the forward linear prediction error in a matrix form

$$\underbrace{\begin{bmatrix} x[p+1] & ... & x[1] \\ ... & ... & ... \\ x[N-p] & ... & x[p+1] \\ ... & ... & ... \\ x[N] & ... & x[N-p] \end{bmatrix}}_{Matrix\ T_p} \begin{bmatrix} 1 \\ a[1] \\ ... \\ a[p] \end{bmatrix} = \begin{bmatrix} e_p^f[p+1] \\ ... \\ e_p^f[N-p] \\ ... \\ e_p^f[N] \end{bmatrix}. \tag{6.1.2}$$

In it, T_p is a Toeplitz matrix, consisting of the available data records.

We can show that by minimizing the expectation of the linear forward prediction error, we can simply solve the matrix equation

$$(T_p^H T_p) \cdot \begin{bmatrix} 1 \\ a[1] \\ ... \\ a[p] \end{bmatrix} = \begin{bmatrix} \rho_p^f \\ \vec{0_p} \end{bmatrix}. \tag{6.1.3}$$

In the matrix $T_p^H T_p$, each element can be computed as

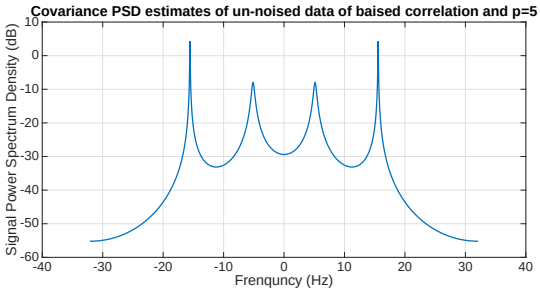
$$r_p[i,j] \quad = \quad \sum_{n=p+1}^{N} x^*[n-i] \cdot x[n-j], for\ 0 \le i,j \le p. \tag{6.1.4}$$

6.2 PSD estimation results

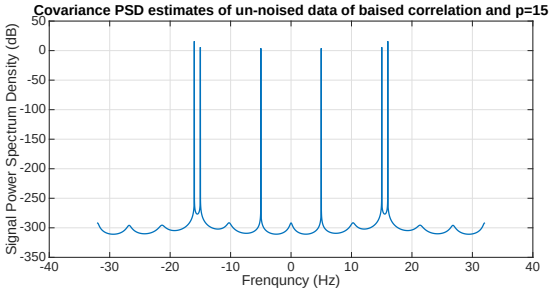
We implement the covariance methods, and obtain the following results.

First of all, low order model ($p = 5$) is not suitable for the tested signals.

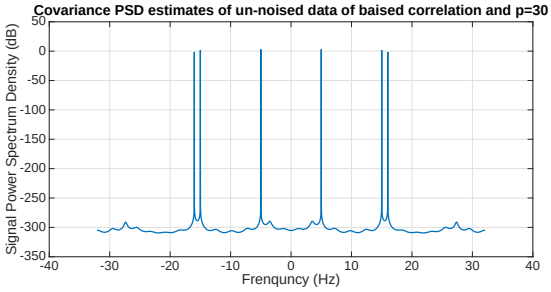
In the noise-free case, we see at model order 15 and 30, the applied method is perfect. It gives the desired PSD with very sharp impulse peaks. This is because covariance method is good at quantify the PSD of sinusoid functions.



(a) Model order = 5



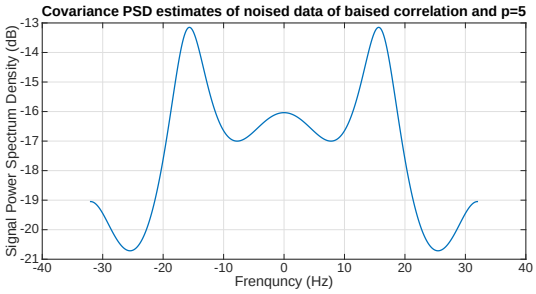
(b) Model order = 15



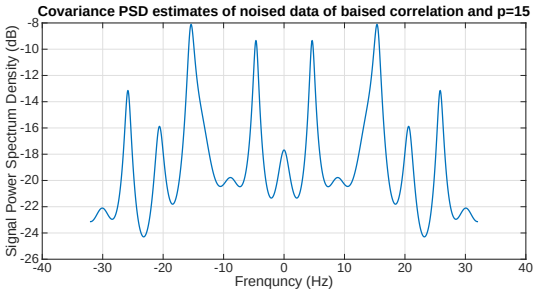
(c) Model order = 30

Figure 6.2.1: Covariance PSD estimates of noise-free signal

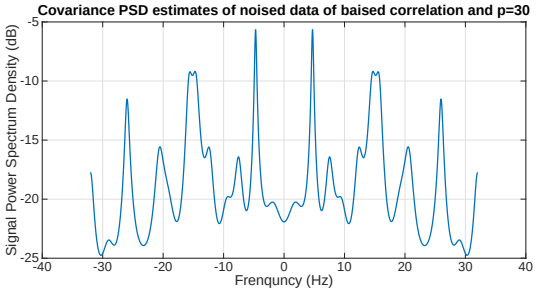
In the noised data case, we see the PSD is impacted by the noises significantly. The sharp peaks disappear, and a similar PSD (comparing with the previous mentioned methods) is achieved. However, the main peaks still exists at 5 Hz and 15 Hz. Notably, we also see a peak split at frequency 15 Hz.



(a) Model order = 5



(b) Model order = 15



(c) Model order = 30

Figure 6.2.2: Covariance PSD estimates of noise signal

Chapter 7

Modified Covariance PSD estimates

7.1 Theory backgrounds

In the previous chapter, we study the covariance method, which is based on the forward linear prediction error. In fact, we also can define the backward prediction error as

$$e_p^b[n] = x[n-p] + \sum_{k=1}^{p} a_p^b[k] \cdot x[n-p+k]. \tag{7.1.1}$$

If we define the both forward and backward linear prediction error vector

$$\overrightarrow{e_p} = \begin{bmatrix} \overrightarrow{e_p^f} \\ \overrightarrow{e_p^b} \end{bmatrix},$$

then we can get a matrix equation as

$$\overrightarrow{e_p} = \begin{bmatrix} T_p \\ T_p^* J \end{bmatrix} \cdot \begin{bmatrix} 1 \\ a[1] \\ ... \\ a[p] \end{bmatrix},$$

where T_p is defined as before and J is a reflection matrix with the form $\begin{bmatrix} & & 1 \\ & ... & \\ 1 & & \end{bmatrix}$.

Similarly to the covariance method, we minimize the errors expectation $E\left[|\overrightarrow{e_p}|^2\right]$, and it can be shown to be equivalent to solve the equation

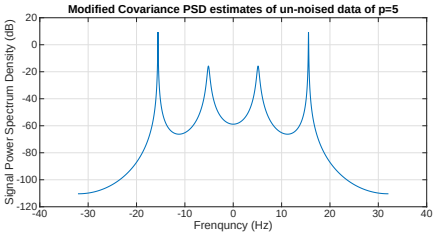
$$R_p \cdot \begin{bmatrix} 1 \\ a[1] \\ ... \\ a[p] \end{bmatrix} = \begin{bmatrix} \rho_p^{fb} \\ \overrightarrow{0_p} \end{bmatrix}, \tag{7.1.2}$$

where R_p can be found as $R_p = \begin{bmatrix} T_p \\ T_p^* J \end{bmatrix}^T \cdot \begin{bmatrix} T_p \\ T_p^* J \end{bmatrix}$. The above method is called modified covariance PSD estimate.

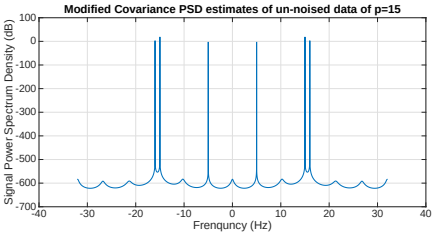
By using this method, the spurious peaks effect can be reduced.

7.2 PSD estimation results

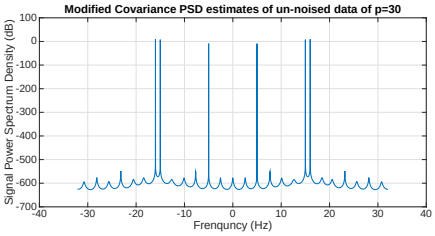
We see in noise free case, the PSD is very similar to that of covariance PSD in previous chapter. We refer readers to that chapter to see details.



(a) Model order = 5



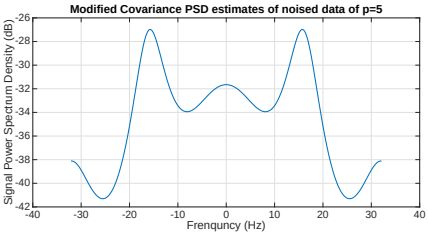
(b) Model order = 15



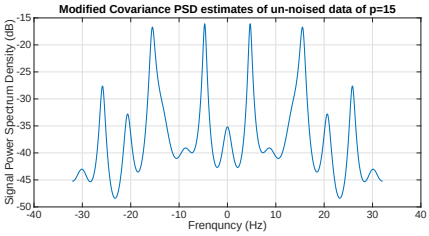
(c) Model order = 30

Figure 7.2.1: Modified Covariance PSD estimates of noise-free signal

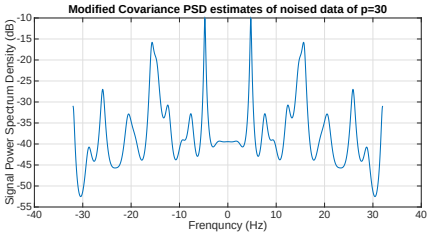
In the noise case, we note that the number of spurious peaks have been reduced, compared with the covariance PSD.



(a) Model order = 5



(b) Model order = 15



(c) Model order = 30

Figure 7.2.2: Modified Covariance PSD estimates of noise signal

Chapter 8

Adaptive Least Mean Squares (LMS) PSD estimates

8.1 Theory backgrounds

In previous chapters, we consider block algorithms, which mean complete data records are arrived at the beginning. However, if the data record is long, such block algorithms may not be suitable, due to long data collection period.

In this chapter, we consider the sequential algorithm, where data come in one by one. Each time a new data arrives, we can carry out PSD estimate. Such sequential algorithm can adapt the signal dynamics. We generalize the least mean square method to an adaptive one.

By the gradient method, we need to update the parameter vector in the following way: at time $N+1$, we have the parameter estimation at N as $\tilde{\bar{a}}_{p,N}$,

$$\tilde{\bar{a}}_{p,N+1} = \tilde{\bar{a}}_{p,N} - \mu \nabla E\left[\left|e^f_{p,N}[N+1]\right|^2\right], \tag{8.1.1}$$

where $e^f_{p,N}[N+1]$ is the filter residual

$$e^f_{p,N}[N+1] \quad = \quad x[N+1] + \bar{x}^T_{p-1}[N] \cdot \tilde{\bar{a}}_{p,N}, \tag{8.1.2}$$

and $\bar{x}^T_{p-1}[N]$ is defined as

$$\bar{x}^T_{p-1}[N] = [x[N], x[N-1], ..., x[N-p+1]]^T. \tag{8.1.3}$$

In practice, (8.1.1) can be replaced by its instantaneous estimate

$$\tilde{\bar{a}}_{p,N+1} = \tilde{\bar{a}}_{p,N} - e^f_{p,N}[N+1] \cdot 2\mu \cdot \bar{x}^*_{p-1}[N]. \tag{8.1.4}$$

To ensure the convergence (i.e., convergence to the block algorithm of LMS) of the method, a sufficient condition is

$$\mu \leq \frac{1}{tr(R)}. \tag{8.1.5}$$

In this report, we assume the parameters a is initialized as zero. Note that a good initial guess may lead to a good performance or fast convergence.

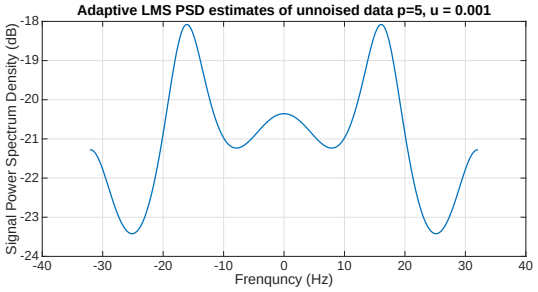
8.2 PSD estimation results

We will try the step sizes 0.001, 0.005 and 0.01. We plot the results in Figure 8.2.1 to Figure 8.2.6.

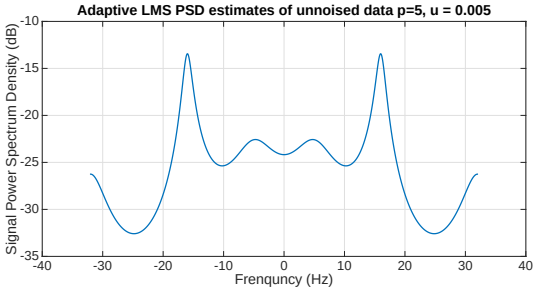
Overall, similar to the covariance methods in Chapter 6.2, model order 5 is not sufficient to build a good PSD estimate. In particular, the 5 Hz is failed to be found out. This is because order 5 model only relies on past 5 records, which is not long enough to capture the 5 Hz components.

Next, we first exam the noise free signal in Figure 8.2.2 to Figure 8.2.3. We see LMS shows a very similar result with the covariance methods, despite the adaptive one is with less sharp peaks. This is reasonable, because the LSM is a adaptive algorithm, which may need a longer time to converge. Current data length 128 may not be long enough. For example, see Figure 8.2.3. Typically, smaller step size will lead to a better PSD estimate. This is true by comparing step size 0.005 and 0.01. However, step size 0.001 is not as good as step size 0.005, because its estimte parameters may not converge at that time.

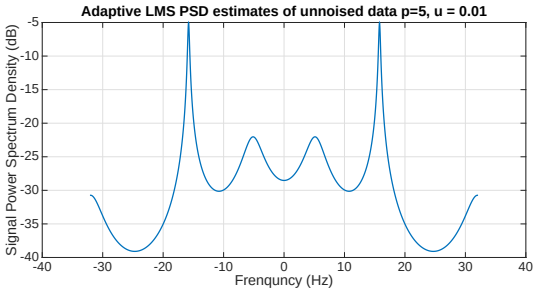
Now, look at Figure 8.2.4 to Figure 8.2.6, where noised signals are tested. We see for model order 15 and 30, step size 0.001, both show a good PSD estimate, finding the peaks at 5 Hz and 15 Hz. Other step size settings lead to a bad PSD, since they all missed the peaks at 5 Hz.



(a) Step size = 0.001

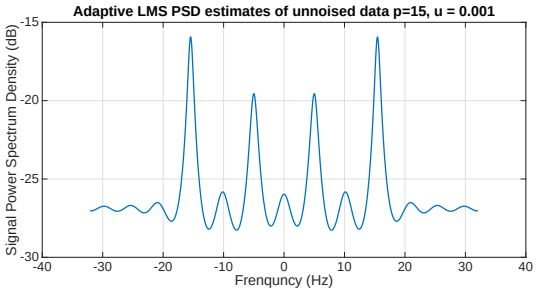


(b) Step size = 0.005

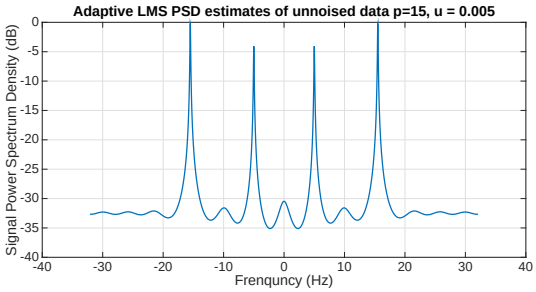


(c) Step size = 0.01

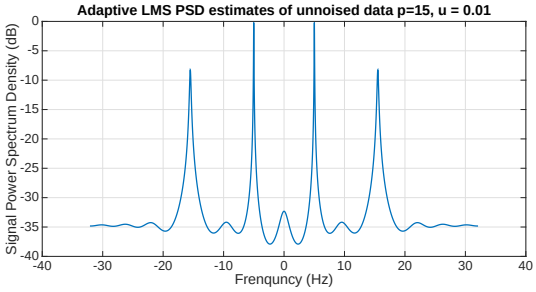
Figure 8.2.1: Adaptive LMS PSD estimates of noise-free signal, model order = 5



(a) Step Size = 0.001

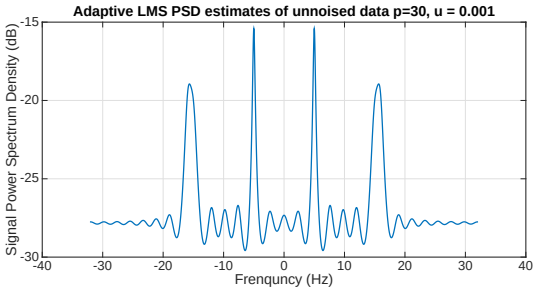


(b) Step Size = 0.005

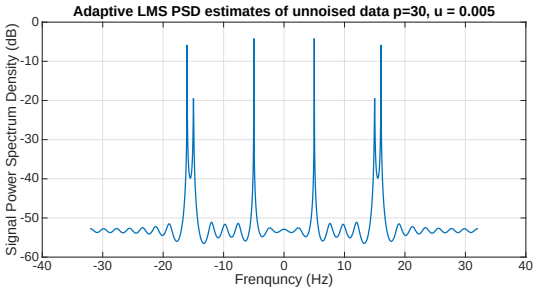


(c) Step Size = 0.01

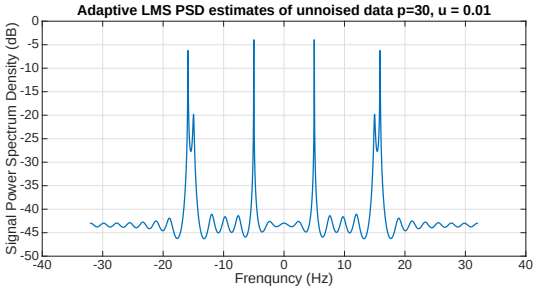
Figure 8.2.2: Adaptive LMS PSD estimates of noise-free signal, model order = 15



(a) Step Size = 0.001

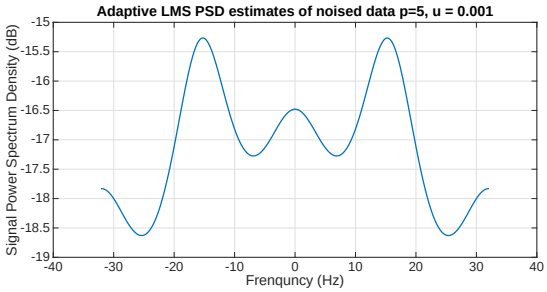


(b) Step Size = 0.005

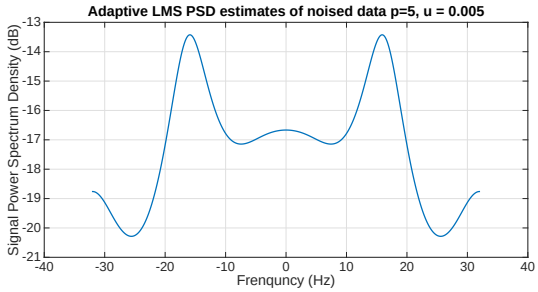


(c) Step Size = 0.01

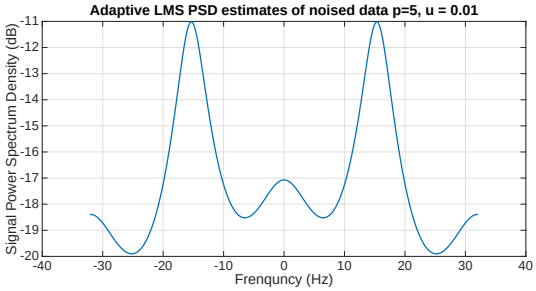
Figure 8.2.3: Adaptive LMS PSD estimates of noise-free signal, model order = 30



(a) Step Size = 0.001

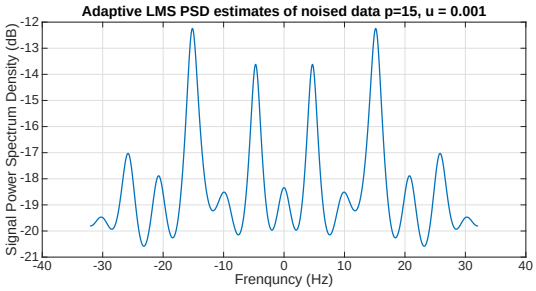


(b) Step Size = 0.005

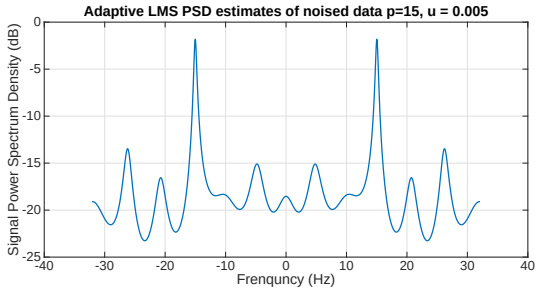


(c) Step Size = 0.01

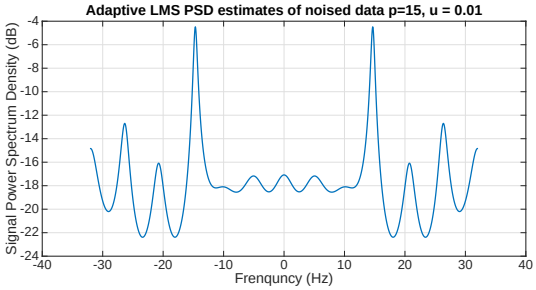
Figure 8.2.4: Adaptive LMS PSD estimates of noised signal, model order = 5



(a) Step Size = 0.001

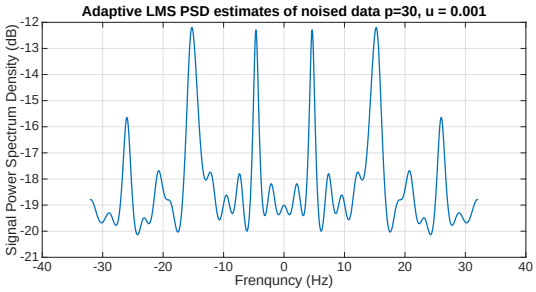


(b) Step Size = 0.005

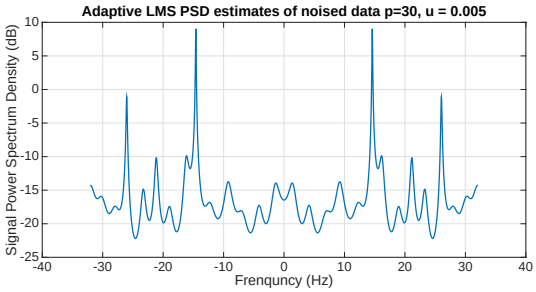


(c) Step Size = 0.01

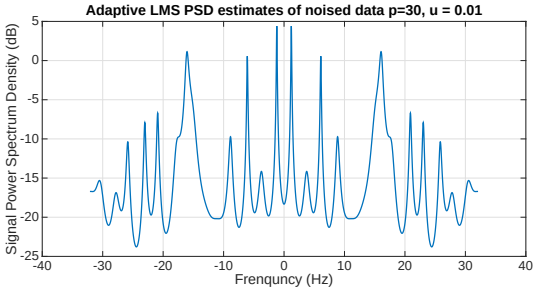
Figure 8.2.5: Adaptive LMS PSD estimates of noised signal, model order = 15



(a) Step Size = 0.001



(b) Step Size = 0.005



(c) Step Size = 0.01

Figure 8.2.6: Adaptive LMS PSD estimates of noised signal, model order = 30

Chapter 9

MUSIC PSD estimates

9.1 Theory backgrounds

In this study, we study a new new approach based on Eigen analysis. Consider the correlation matrix R_p, it is proven that we can decompose it in the following manner

$$R_p = \sum_{i=1}^{M} (\lambda_i + \rho_w) \bar{v_i} \bar{v_i}^T + \sum_{i=M+1}^{p+1} \rho_w \bar{v_i} \bar{v_i}^T. \tag{9.1.1}$$

In the above equation, $\sum_{i=1}^{M} (\lambda_i + \rho_w) \bar{v_i} \bar{v_i}$ represents the signal subspace, and $\sum_{i=M+1}^{p+1} \rho_w \bar{v_i} \bar{v_i}$ represents the noise subspace. We note that the above equation is called Eigen decomposition, where $\lambda_i + \rho_w$ and ρ_w is the eigen values of R_p and $\bar{v_i}$ is the corresponding eigen vector.

We have the following observations: i) the noise subspace is with small eigen value, comparing with signal subspace; ii) the eigen value fixed at ρ_w for noise subspace; iii) the eigen values are orthogonal to each other for eigen vectors in noise subspace and signal subspace.

Suppose we are given the signal subspace eig-vector number M. We consider a combination of the noise subspace eigen vectors $\sum_{k=m+1}^{p+1} \alpha_k \bar{v_k}$. It should also orthogonal to the vectors from signal subspace. We consider a vector denoted by

$$\bar{e}(f) \quad = \quad \begin{bmatrix} 1 \\ \exp(j2\pi fT) \\ ... \\ \exp(j2\pi pfT) \end{bmatrix}. \tag{9.1.2}$$

$\bar{e}(f) \cdot \sum_{k=m+1}^{p+1} \alpha_k \bar{v_k} = 0$ whenever f is from the signal subspace.

Thus, we can get a frequency estimator function

$$\frac{1}{\sum_{k=M+1}^{p+1} \alpha_k \left\| \bar{e}^H(f) \cdot \bar{v}_k \right\|^2}. \tag{9.1.3}$$

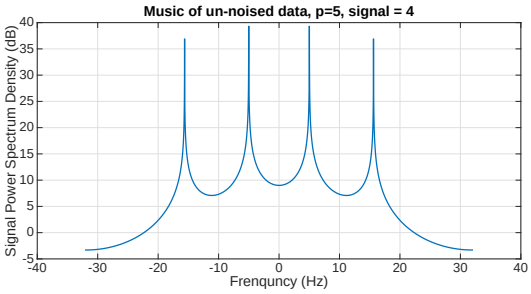
Let $\alpha_k = 1$, we have the MUSIC method.

9.2 PSD estimation results

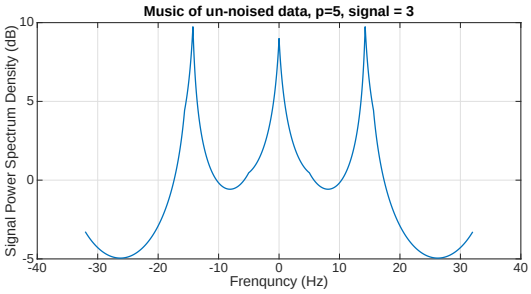
We test MUSIC performance in this section. We will check different model orders and the corresponding signal subspace vector numbers, according to the following table.

Model Order	Signal Subspace Vector Number
5	4,3,2
15	14,11,9
30	25,20,15

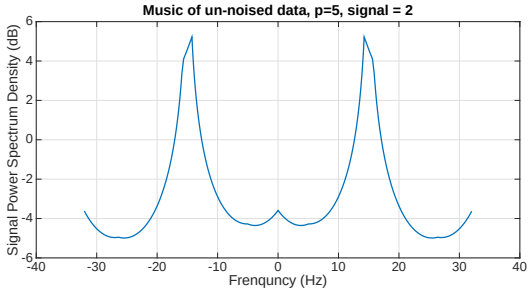
We first exam the noise free case in Figure 9.2.1 to Figure 9.2.3. We can see model order 5 is not sufficient to model the signal. For the model order 15, the signal subspace size = 9 achieves the best performance; for model order 30, the signal subspace size = 15 achieves the best performance. Both model orders can obtain a relatively good PSD estimate. From this part, we see the MUSIC method's performance is largely dependent on the selection of the signal subspace size.



(a) Signal Subspace = 4

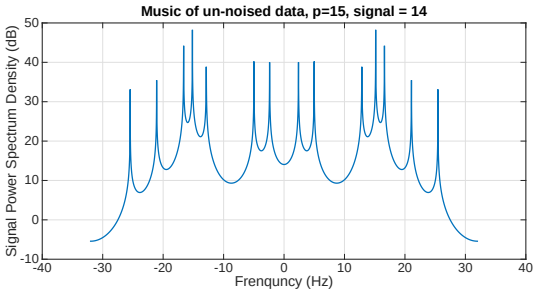


(b) Signal Subspace = 3

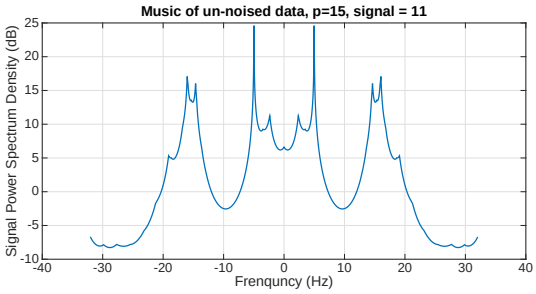


(c) Signal Subspace = 2

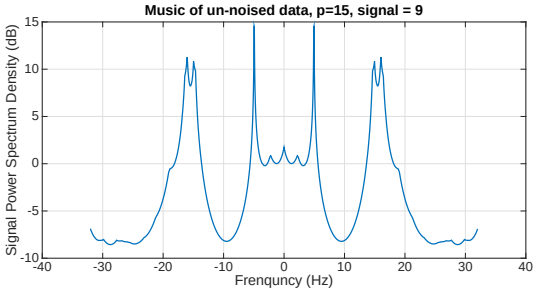
Figure 9.2.1: MUSIC PSD estimates of noise-free signal, model order = 5



(a) Signal Subspace = 14

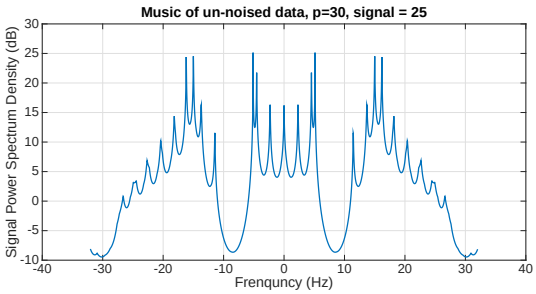


(b) Signal Subspace = 11

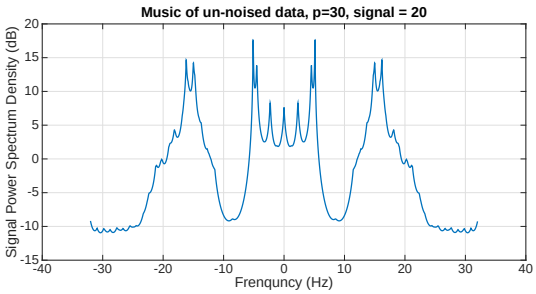


(c) Signal Subspace = 9

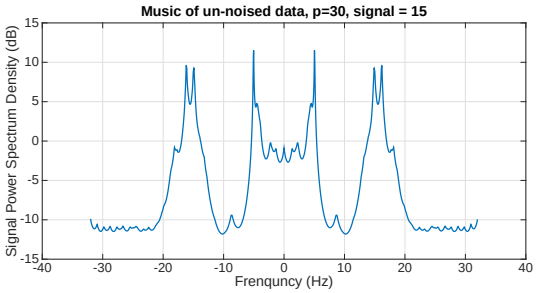
Figure 9.2.2: MUSIC PSD estimates of noise-free signal, model order = 15



(a) Signal Subspace = 25



(b) Signal Subspace = 20

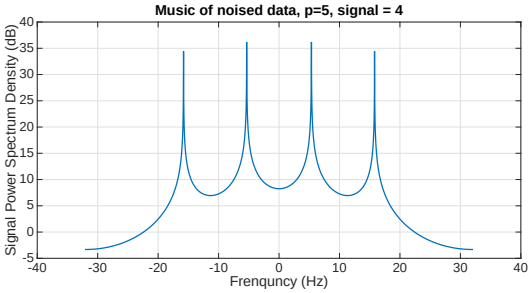


(c) Signal Subspace = 15

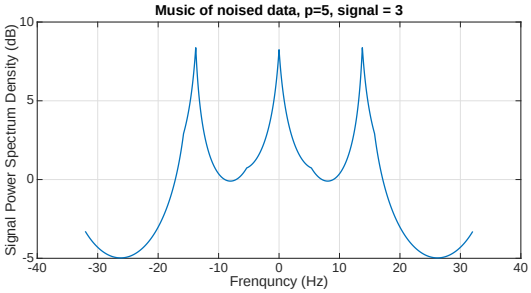
Figure 9.2.3: MUSIC PSD estimates of noise-free signal, model order = 30

Next, we exam the noised case in Figure 9.2.4 to Figure 9.2.6. We see in model order 15, signal subspace = 9 case, it gets a line spiting at 15 Hz, which is desired. Similar PSDs are achieved in model order 30, but

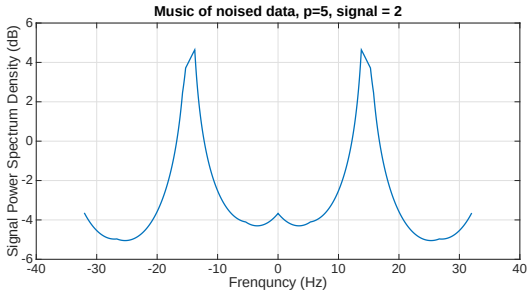
with more false positive peaks. This shows MUSIC method can get the PSD of the signal to some degree, even at the presence of noises.



(a) Signal Subspace = 4

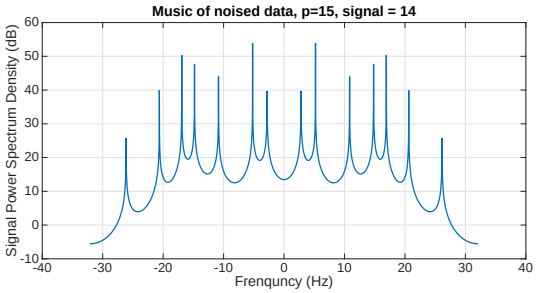


(b) Signal Subspace = 3

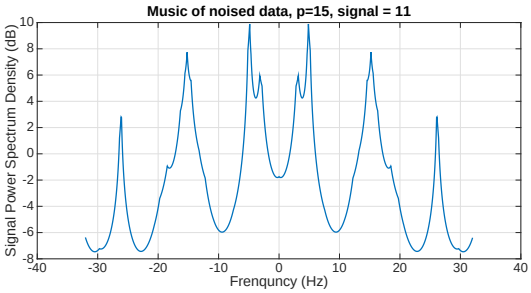


(c) Signal Subspace = 2

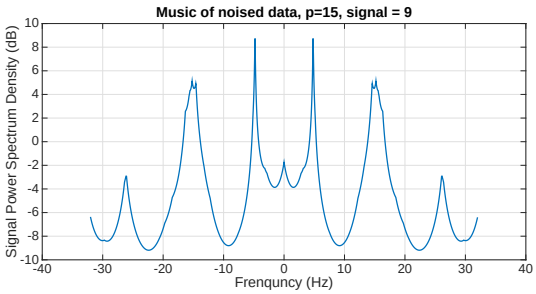
Figure 9.2.4: MUSIC PSD estimates of noised signal, model order = 5



(a) Signal Subspace = 14

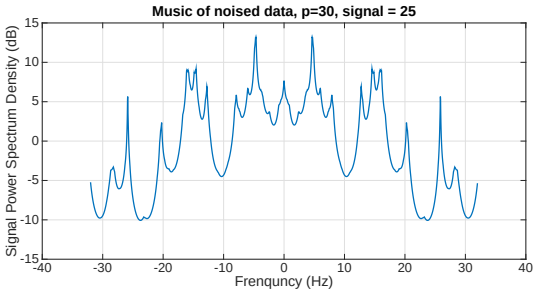


(b) Signal Subspace = 11

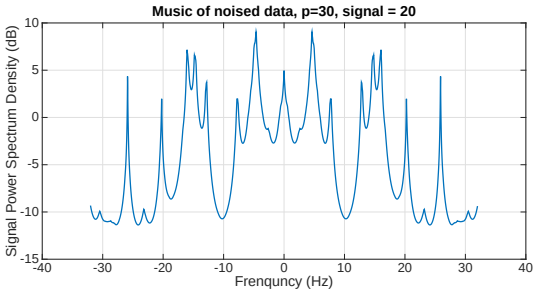


(c) Signal Subspace = 9

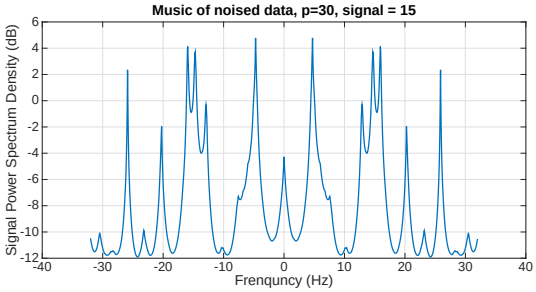
Figure 9.2.5: MUSIC PSD estimates of noised signal, model = 15



(a) Signal Subspace = 25



(b) Signal Subspace = 20



(c) Signal Subspace = 15

Figure 9.2.6: MUSIC PSD estimates of noised signal, model order = 30

Chapter 10

Conclusions

In this chapter, we compare the methods we used in this report, and summarize the reports.

For the classical methods, including Blackman-Tukey and Welch Periodogram PSD estimates, Welch Periodogram PSD outperforms Blackman-Tukey, since it depresses the valley more remarkably. However, they both have worst resolutions comparing to the other algorithms, even a proper window function is selected. The advantage of such classical methods is they do not rely on models (and the corresponding model order) as a priori. Basically speaking, such classical methods will not be preferred if other methods can be used.

Next, in the parameter-based approach, including Yule-Walker PSD, Burg PSD, Covariance PSD and Modified Covariance PSD, they all have better resolution, if a proper model order is selected.

- In the noise free case: Given the proper model order, we see the least square methods (Covariance PSD and Modified Covariance PSD) give very sharp peaks at their PSD estimates, illustrating perfect performance in the noise free case. However, note that the test data is consisting of sinusoid functions, and more general signals need to be tested to justify this opinion. The burg algorithm also finds the near peaks, but with less sharp peaks.

- In the noised case: Modified Covariance PSD outperforms others, since it depresses the noise frequencies most.

Overall, in the parameter-based approach, the Modified Covariance PSD is preferred. At the same time, burg's algorithm is also a good choice, if less resolution requirement is wanted.

Along another track, a sequential method, adaptive LMS algorithm, is tested. This method needs to be used when we need to do "online" processing. When proper model order and step size are selected, it converges to the "offline version" of the LMS method.

At last, the eigen-based approach, MUSIC, performs good if a good signal subspace size and model order is selected. It seems for MUSIC, the signal subspace size selection is more important than the model order. Despite the good performance given by MUSIC, the eigen method may not be as easily implemented as parameter methods, since it is requires to design algorithm settings in a subtle way.