Power Spectral Estimation Project

Shibin Zheng

Content

Chapter 1

Introduction

In statistical signal processing, the goal of spectral density estimation (SDE) is to estimate the spectral density (also known as the power spectral density) of a random signal from a sequence of time samples of the signal. Intuitively speaking, the spectral density characterizes the frequency content of the signal. One purpose of estimating the spectral density is to detect any periodicities in the data, by observing peaks at the frequencies corresponding to these periodicities.

Some SDE techniques assume that a signal is composed of a limited (usually small) number of generating frequencies plus noise and seek to find the location and intensity of the generated frequencies. Others make no assumption on the number of components and seek to estimate the whole generating spectrum.

1.1 Overview

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 the method above, BT and Welch estimates belong to classical periodogram, and they directly estimate based on input signal data. The MUSIC method is an eigen-value analysis. The estimators, except classical periodogram and correlogram, model the signals through a system, and by calculate system parameters, the PSD of the signal are obtained.

PSD Setting

For each of the PSD estimation algorithms discussed above, we calculate 4096 PSD points (such that the resolution of PSD is fs/4096) in the interval -fs/2 to fs/2. And we plot all PSD estimates as Power (dB) vs. frequency (Hz).

System Setting

In this report, we conduct the simulation in Matlab 2017a. The generated source codes are compatible with Matlab 2016b. The simulation environment system is win10.

1.2 Test signal

We use two data sets in our project:

Data Set #1: a 128-point real data sequence consisting of unit-amplitude sinusoids at 10Hz, 11Hz, and 25Hz. Sampling frequency fs=64samples/second.

Data Set #2: a data record identical to Data Set #1, but add white noise such that signal-to-noise radio(SNR) is 0 dB.

Signal #1 is

$$x_1(n) = \sin\left(\frac{2\pi * 10}{f_s}(n-1)\right) + \sin\left(\frac{2\pi * 11}{f_s}(n-1)\right) + \sin\left(\frac{2\pi * 25}{f_s}(n-1)\right) \; 1 \le n \le 128$$

x_1 is consisted of unit-amplitude sinusoids of 10Hz, 11Hz, 25Hz. We denote the above case the noise-free case, signal shown as below.

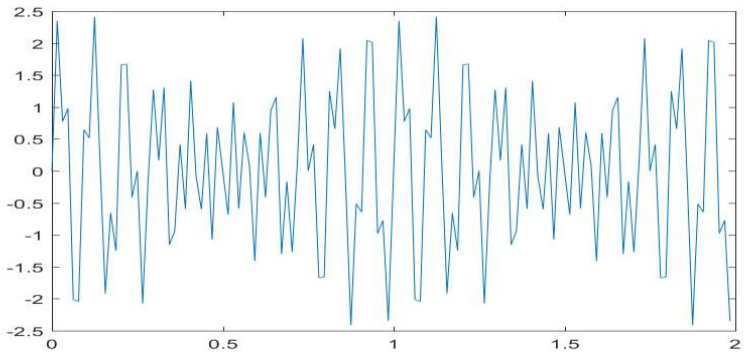


Figure 1.2.1: Signal x_1(n)

Signal #2 is

x2(n) = x1(n) + wn(n), 1 ⩽ n ⩽ 128.

In it, wn(n) is the white noise. The above case the noised case. In this report, we assume x_2(n) is the signal with SNR=0dB. We simply apply the awgn() function to add white noise.

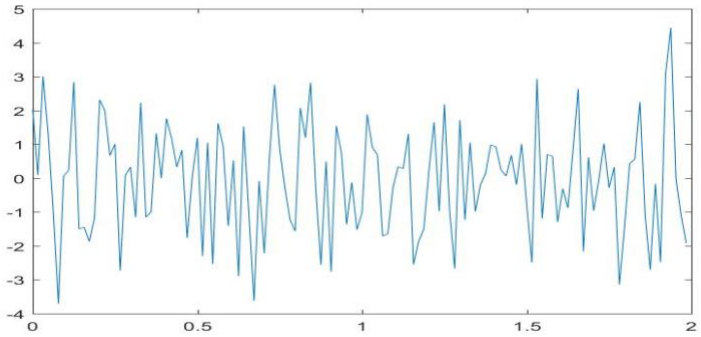


Figure 1.2.1: Signal x_2(n)

Chapter 2

Blackman-Tukey PSD estimates

2.1 Theory backgrounds

It was developed by Blackman and Tukey (1958) and is based on the Wiener-Khinchin theorem, which states that if the Fourier transform of a series x(t) is X(f), and if the autocorrelation function of the series is R, then the Fourier transform of R yields $P_X(f)=|X(f)|^2$ or the power spectrum of $x(t)$. The resulting power-spectrum estimate is called a correlogram. An alternative is direct or windowed FFT of the time series itself, called a periodogram.

$$P_X(f) = |\sum_{k=0}^{M-1} w_k r_k e^{-2\pi i f k}| \tag{2.1}$$

where r_k is the autocorrelation estimate at lag k, M is the maximum lag considered and window length, and w_k is the windowing function. Several window shapes are available in the Toolkit: Bartlett (triangular), Hamming (cosinusoidal), Hanning (slightly different cosinusoidal), and none.
However, there is a trade-off between higher resolution and increasing variance of the spectral estimate.

At the extreme, a single (**M=N**) direct application of FFT to an unwindowed time series results in a periodogram with a theoretical standard deviation of the estimates equal to the estimates at each frequency, regardless of the number of observations in the time series.

Averaging the results from many short data windows throughout the series (or autocorrelation) effectively increases the number of independent samples used in estimation and thereby reduces the estimation variance

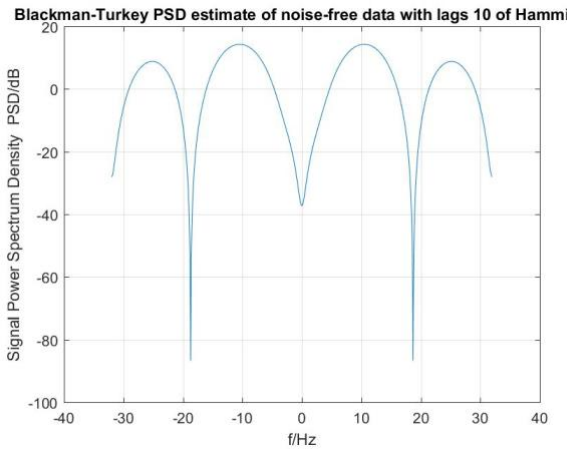
It is shown that the variance of a power spectrum obtained by a windowed correlogram is **2M/3N** of the estimated power at each frequency. Thus a narrower window should be used to smooth the spectrum and reduce the sampling errors on the estimate.

In practice, it is recommended that windows should be no more than **one-fifth to one-tenth** the total number of data points (to obtain desired estimate-variance reductions) and not too much smaller (in order to retain the ability to distinguish between powers at neighboring frequencies and to obtain the desired leakage reductions).

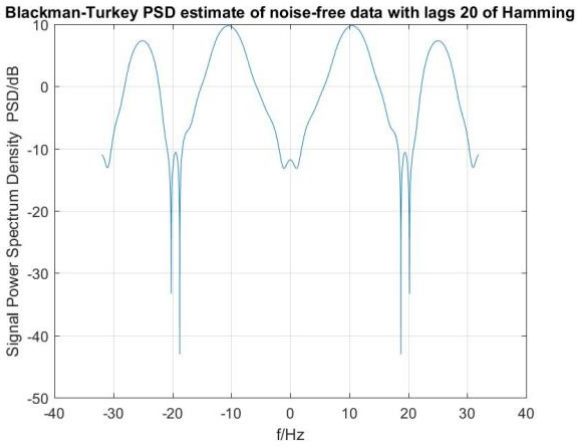
2.2 PSD estimation results

In the following, we form Blackman-Tukey PSD estimates with lags of 10 and 20. Using a Hamming window and unbiased correlation estimates.

We first exam the noise-free case in Figure 2.2.1. By comparing the subfigures, we see large lags will lead to better viewed PSD, however less stable (cannot be seen directly in the figure). Also note that the applied method successfully identifies the power spectrum at 25Hz and 10 -11Hz. However, it fails to distinguish between 10 and 11 Hz



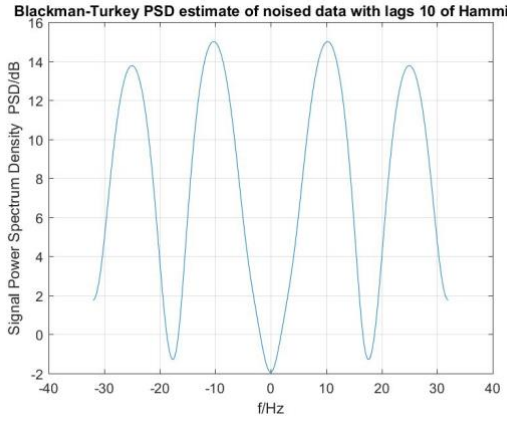
b) Hamming window with lag 10



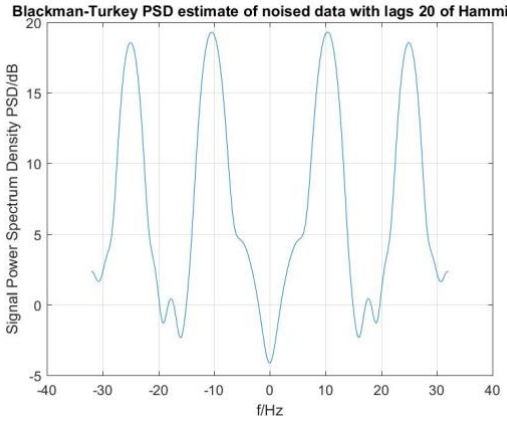
a) Hamming window with lag 20

Figure2.2.1Blackman-Turkey PSD of noise-free signal

We next experiment the noised case in Figure 2.2.2. We see there are 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 10 and 11 Hz.



b) Hamming window with lag 10



a) Hamming window with lag 10

Figure 2.2.2: Blackman-Turkey PSD of noised signal

Chapter 3

Welch Periodogram PSD estimates

3.1 Theory backgrounds

Two modifications to Bartlett's method:

1) the subsequences are allowed to overlap
2) instead of Periodograms, modified Periodograms are averaged

Assuming that successive sequences are offset by D points and that each sequence is L points long, then the ith sequence is

$$x_i(n) = x(n + iD) \quad ; \quad n = 0, 1, \ldots, L-1 \tag{3.1.1}$$

Thus the overlap is L-D points and if K sequences cover the entire N data points then

$$N = L + D(K-1). \tag{3.1.2}$$

Welch's method can be written in terms of the data record as follows

$$\hat{P}_W(e^{j\omega}) = \frac{1}{KLU} \sum_{i=0}^{K-1} \left| \sum_{n=0}^{L-1} w(n)x(n + iD)e^{-jn\omega} \right|^2 \tag{3.1.3}$$

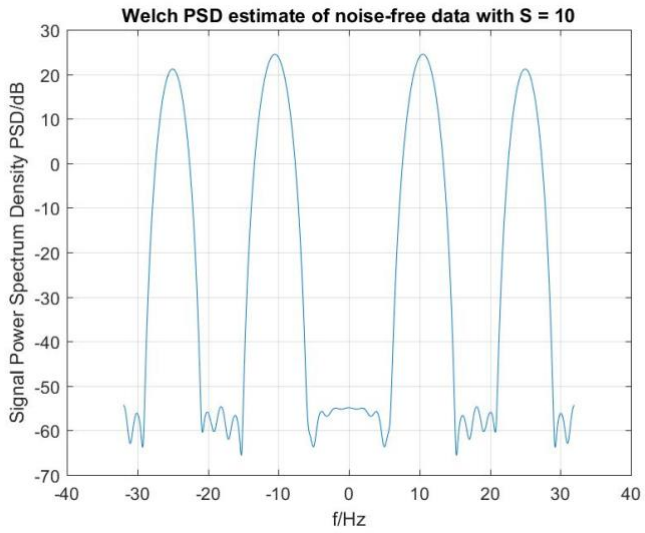
Hence the expected value of Welch's estimate is

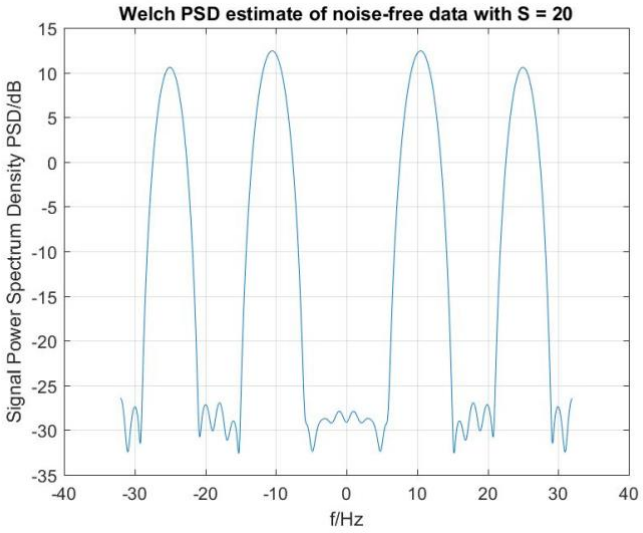
$$\begin{aligned} E\{\hat{P}_W(e^{j\omega})\} &= E\{\hat{P}_M(e^{j\omega})\} \\ &= \frac{1}{2\pi LU} P_x(e^{j\omega}) * |W(e^{j\omega})|^2 \end{aligned} \tag{3.1.4}$$

3.2 PSD estimation results

In the following, we form Welch Periodogram PSD estimates. If N samples are divided into P segments of D samples each, with a shift of S samples between adjacent segments, let S=10 and 20, and let D=32. Use a Hamming window.

We first exam the noise free signal case in Figure 3.2.1. We see the side-lobes are depressed significantly (45-80 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 25 Hz and peaks between 10 and 11 Hz, but cannot distinguish between the peaks between 10 and 11 Hz.

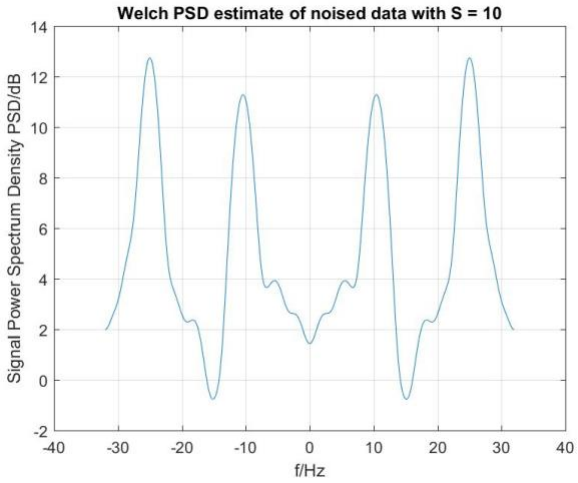


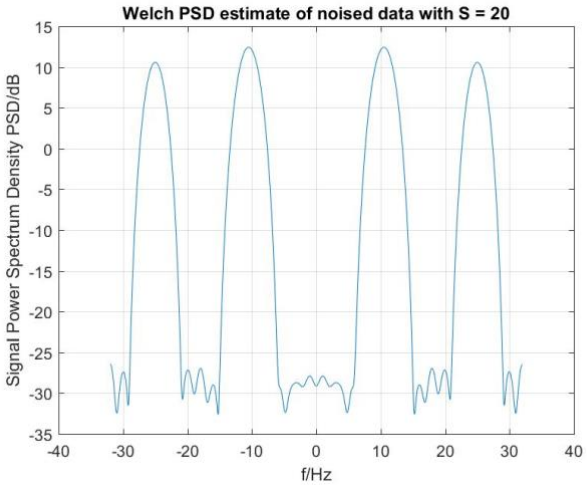


a) a shift of 10 b) a shift of 20

Figure 3.2.1: Welch Periodogram PSD of noise-free signa

We next experiment the noised signal case shown 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 **quite 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 Method

4.1 Theory backgrounds

In this section, we focus on a technique for estimating the **AR** parameters which is called the Yule-Walker Method. For AR signals, **m=0** and **B(z)=1**.

$$r(0) + \sum_{i=1}^{n} a_i r(-i) = \sigma^2 \sum_{j=0}^{0} b_j h_j^* = \sigma^2 \tag{4.1.1}$$

And then write it in matrix format:

$$\begin{bmatrix} r(0) & r(-1) & \dots & r(-n) \\ r(1) & r(0) & & \vdots \\ \vdots & & \ddots & r(-1) \\ r(n) & \dots & & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.1.2}$$

-Rn An r_n

The above equations are called the Yule-Walker equations or Normal YW equations, and form the basis of many AR estimation methods. If $\{r(k)\}_{k=0}^{n}$ were known, we could rewrite the equation:

$$\begin{bmatrix} r(1) \\ \vdots \\ r(n) \end{bmatrix} + \begin{bmatrix} r(0) & \cdots & r(-n+1) \\ \vdots & \ddots & \vdots \\ r(n-1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4.1.3}$$

Or, with obvious definitions,

$$r_n + R_n A_n = 0 \tag{4.1.4}$$

The solution is $A_n = -R_n^{-1} r_n$. Once r_n is found, σ^2 can be obtained from the first row of (4.1.2) or, equivalently, from (4.1.1).

The Yule-Walker method for AR spectral estimation is based directly on (4.2). Given data $\{y(t)\}_{t=1}^{N}$, we first obtain sample covariances $\{\hat{r}(k)\}_{k=0}^{n}$ using the standard biased ACS **estimator**. We insert these ACS estimates in (4.1.2) and solve for $\hat{A}$ and $\hat{\sigma}^2$ as explained above in the known-covariance case.

Note that the covariance matrix in (4.1.2) can be shown to **be positive definite** for any n, and hence the solution to (4.1.2) is unique. When the covariances are replaced by standard biased ACS estimates, the matrix can be shown to be positive definite for

any sample (not necessarily generated by an AR equation) that is not identically equal to zero.

If the **unbiased** autocorrelation estimate is selected, it may result in a **non-positive-definite** autocorrelation matrix.

Unbiased:

$$\hat{r}_{xx}[m] = \begin{cases} \frac{1}{N-m} \sum_{n=0}^{N-m-1} x[n+m]x^*[n], 0 \le m \le N-1 \\ \frac{1}{N+m} \sum_{n=0}^{N-|m|-1} x^*[n+|m|]x[n], -(N-1) \le m < 0 \end{cases}$$

⇒A stable AR filter is not assumed.

The **biased** correlation estimates will always yield **positive semi-definite** autocorrelation matrices.

Biased:

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

Such that a stable AR filter will be assumed

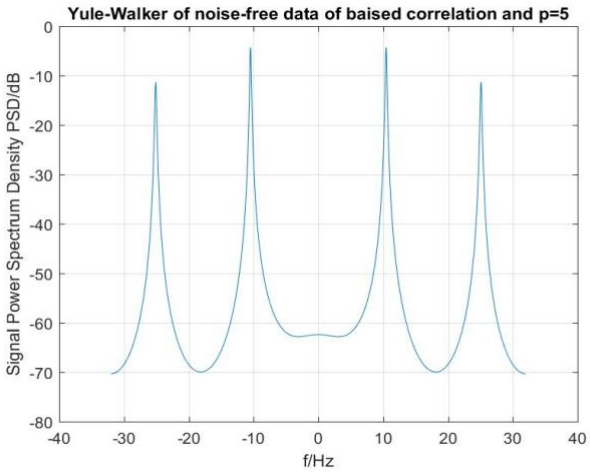
For long data records, the Yule-Walker method can produce reasonable spectral estimates.

For short data records, the methods produce poor resolution spectral estimates relative to other AR methods.

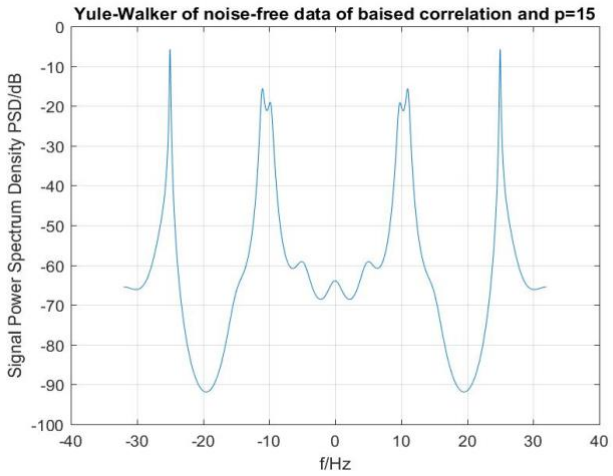
4.2 PSD estimation results

In the following, we form Yule-Walker PSD estimates using model orders of 5, 15, and 30 and applying both biased and unbiased.

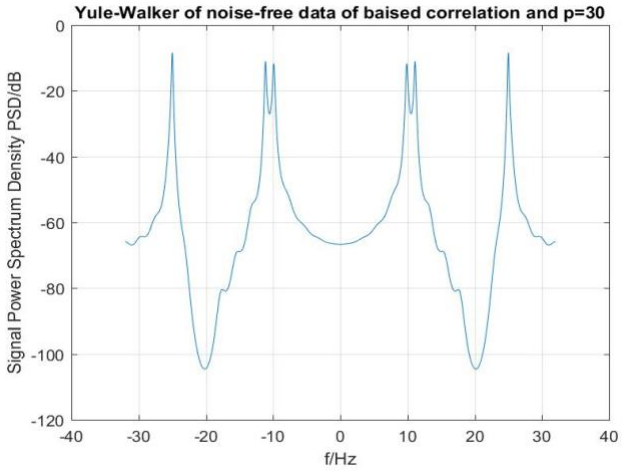
By comparing Figure 4.2.1 and Figure 4.2.2. We observe that **model order 5 is not sufficient** for both signal input to generate PSDs that distinguish the power at 10 Hz and 11 Hz. At order 15, the unbiased estimate successfully finds three peaks (indicate the frequencies of input sinusoids), while the biased estimate fails to do so. And as the order increase, we obtain better PSD that are distinguishing and accurate.



a) model order = 5

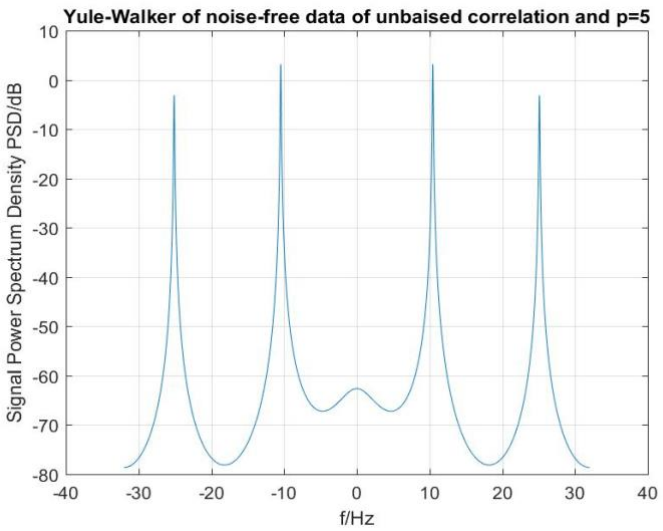


b) model order =15

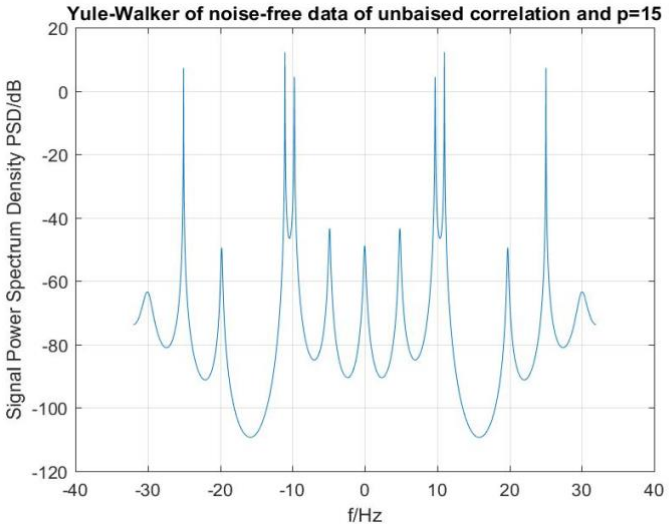


c) model order = 30

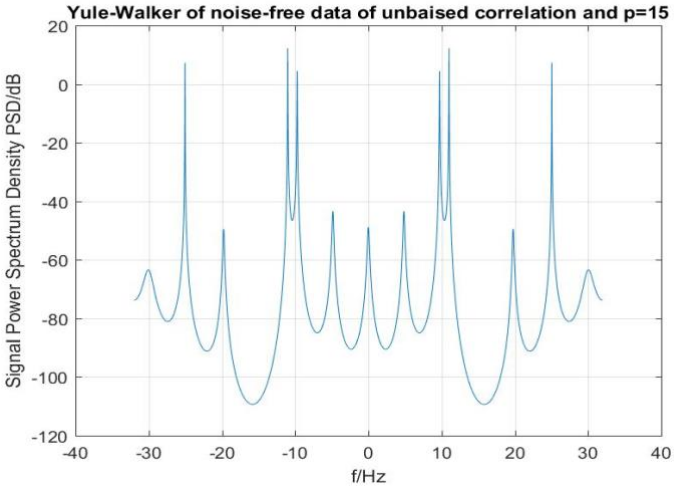
Figure 4.2.1: Yule-Walker PSD estimates of noise-free signal, biased estimate of correlation



a) model order = 5



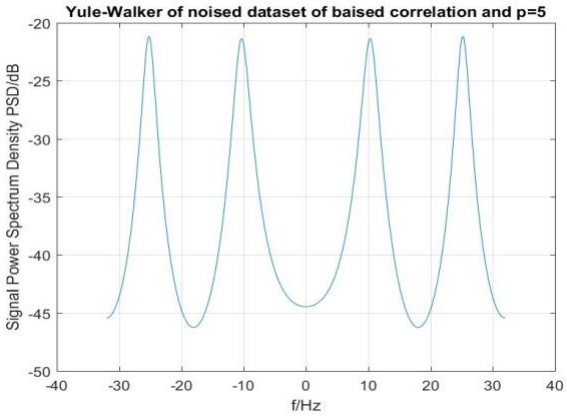
b) model order = 15



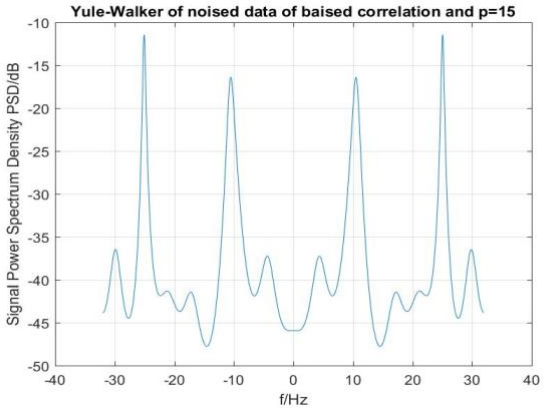
c) model order = 30

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

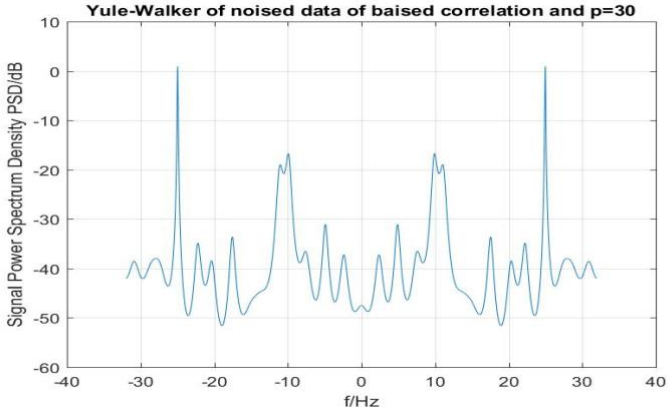
Next, we look at the noise case. At noised case, we note that both biased estimate and unbiased estimate lead to a similar result. And models with order higher than 15 obtain a good result. However, they are both easily influenced by noise, since the peaks at10 and 11 Hz grow less sharp when noise added.



a) model order = 5

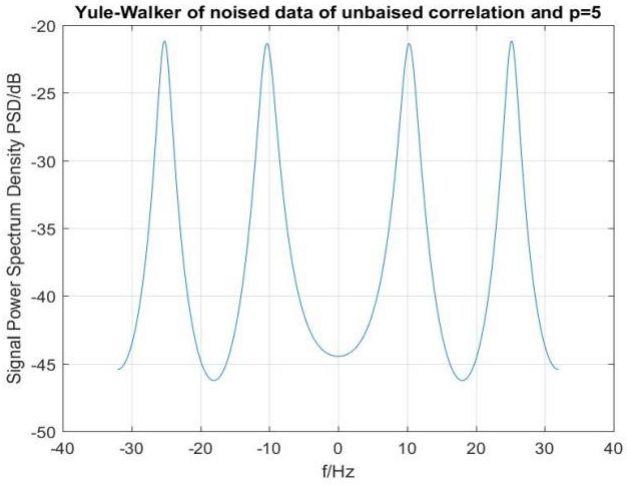


b) model order =15

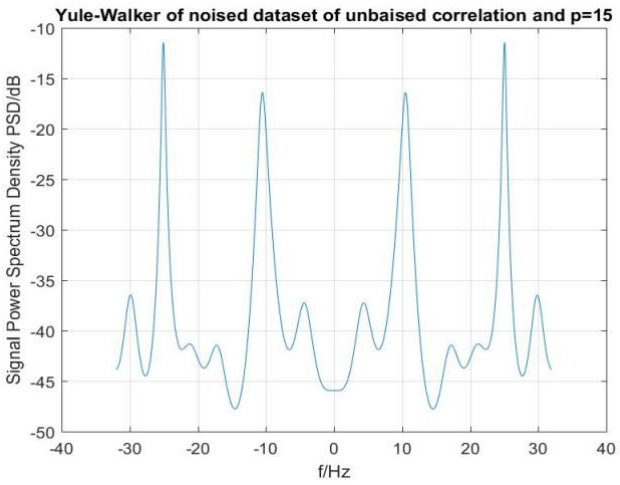


c) model order = 30

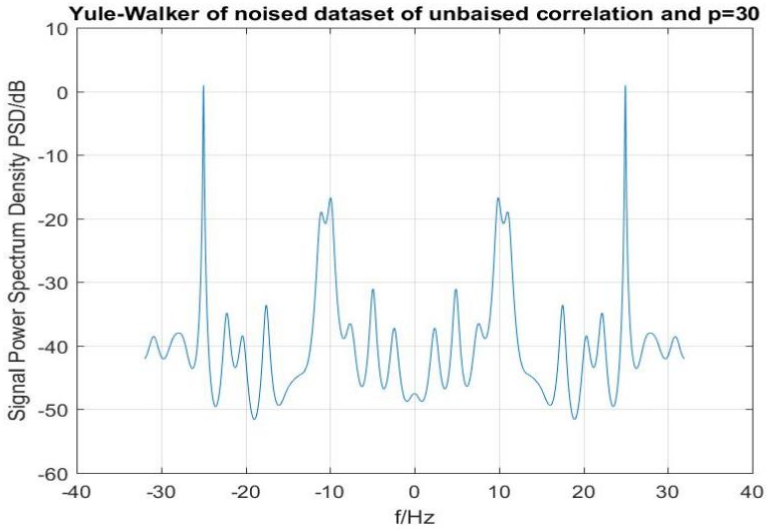
Figure 4.2.3: Yule-Walker PSD estimates of noised signal, biased estimate of correlation



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

And we can see from above,
This estimator would be **influenced by white noise** with limited input data record.
The **unbiased correlation** sequences have a **better** viewed PSD with sharper peaks.
As model **order increases**, we obtain **better** PSD that are distinguishing and accurate

Chapter 5

Burg(Harmonic) PSD estimates

5.1 Theory backgrounds

The thesis developed a method for **AR parameter estimation** that is based **on forward and backward prediction errors**, and on direct estimation of the refection coefficients. In this complement, we develop the Burg estimator and discuss some of its properties. Assume we have data measurements {y(t)} for t = 1, 2, …, N. we define the forward and backward prediction errors for a pth-order model as:

$$\hat{e}_{f,p}(t) = y(t) + \sum_{i=1}^{p} \hat{a}_{p,i} y(t-i), \qquad t = p+1, \ldots, N \tag{5.1.1}$$

$$\hat{e}_{b,p}(t) = y(t-p) + \sum_{i=1}^{p} \hat{a}^*_{p,i} y(t-p+i), \qquad t = p+1, \ldots, N \tag{5.1.2}$$

We use hats to denote estimated quantities, and we explicitly denote the order p in both the prediction error sequences and the AR coefficients. The AR parameters are related to the refection coefficient $\widehat{\boldsymbol{k}}_{\boldsymbol{P}}$ by

$$\hat{a}_{p,i} = \begin{cases} \hat{a}_{p-1,i} + \hat{k}_p \hat{a}^*_{p-1,p-i}, & i = 1, \ldots, p-1 \\ \hat{k}_p, & i = p \end{cases} \tag{5.1.3}$$

Burg's method considers the recursive-in-order estimation of $\hat{k}_P$ given that the AR coefficients for order p -1 have been computed. In particular, Burg's method finds $\hat{k}_P$ to **minimize the arithmetic mean** of the forward and backward prediction error variance estimates:

$$\min_{\hat{k}_p} \frac{1}{2} \left[\hat{\rho}_f(p) + \hat{\rho}_b(p) \right] \tag{5.1.4}$$

Where

$$\hat{\rho}_f(p) = \frac{1}{N-p} \sum_{t=p+1}^{N} |\hat{e}_{f,p}(t)|^2$$

$$\hat{\rho}_b(p) = \frac{1}{N-p} \sum_{t=p+1}^{N} |\hat{e}_{b,p}(t)|^2$$

and where $\left(\hat{a}_{p-1,i}\right)_{\mathrm{i=1}}^{p-1}$ are assumed to be known from the recursion at the previous order.

The prediction errors satisfy the following recursive-in-order expressions

$$\begin{aligned}\hat{e}_{f,p}(t) &= \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1) \\ \hat{e}_{b,p}(t) &= \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t)\end{aligned} \tag{5.1.5}$$

Then

$$\begin{aligned}\hat{e}_{f,p}(t) &= y(t) + \sum_{i=1}^{p-1} \left(\hat{a}_{p-1,i} + \hat{k}_p \hat{a}^*_{p-1,p-i}\right) y(t-i) + \hat{k}_p y(t-p) \\ &= \left[y(t) + \sum_{i=1}^{p-1} \hat{a}_{p-1,i} y(t-i)\right] + \hat{k}_p \left[y(t-p) + \sum_{i=1}^{p-1} \hat{a}^*_{p-1,i} y(t-p+i)\right] \\ &= \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1)\end{aligned} \tag{5.1.6}$$

Similarly,

$$\begin{aligned}\hat{e}_{b,p}(t) &= y(t-p) + \sum_{i=1}^{p-1} [\hat{a}^*_{p-1,i} + \hat{k}_p^* \hat{a}_{p-1,p-i}] y(t-p+i) + \hat{k}_p^* y(t) \\ &= \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t)\end{aligned}$$

We can use the above expressions to develop a recursive-in-order algorithm for estimating the AR coefficients. Note that the quantity to be minimized in (5.1.4) is quadratic in $\hat{k}_P$ since

$$\begin{aligned}\frac{1}{2}[\hat{\rho}_f(p) + \hat{\rho}_b(p)] &= \frac{1}{2(N-p)} \sum_{t=p+1}^{N} \left\{ \left| \hat{e}_{f,p-1}(t) + \hat{k}_p \hat{e}_{b,p-1}(t-1) \right|^2 \right. \\ &\qquad \left. + \left| \hat{e}_{b,p-1}(t-1) + \hat{k}_p^* \hat{e}_{f,p-1}(t) \right|^2 \right\} \\ &= \frac{1}{2(N-p)} \sum_{t=p+1}^{N} \left\{ \left[|\hat{e}_{f,p-1}(t)|^2 + |\hat{e}_{b,p-1}(t-1)|^2 \right] \left[1 + |\hat{k}_p|^2 \right] \right. \\ &\qquad + 2\hat{e}_{f,p-1}(t) \hat{e}^*_{b,p-1}(t-1) \hat{k}_p^* \\ &\qquad \left. + 2\hat{e}^*_{f,p-1}(t) \hat{e}_{b,p-1}(t-1) \hat{k}_p \right\}\end{aligned}$$

we find that the $\hat{k}_P$ that minimizes the above quantity is given by

$$\hat{k}_p = \frac{-2 \sum_{t=p+1}^{N} \hat{e}_{f,p-1}(t) \hat{e}^*_{b,p-1}(t-1)}{\sum_{t=p+1}^{N} \left[|\hat{e}_{f,p-1}(t)|^2 + |\hat{e}_{b,p-1}(t-1)|^2 \right]} \tag{5.1.6}$$

A recursive-in-order algorithm for estimating the AR parameters, called the Burg algorithm, is as follows:

The Burg Algorithm

Step 0. Initialize $\hat{\boldsymbol{e}}_{\boldsymbol{f,0}}(\boldsymbol{t}) = \hat{\boldsymbol{e}}_{\boldsymbol{b,0}}(\boldsymbol{t}) = \boldsymbol{y}(\boldsymbol{t})$.

Step 1. For p = 1, …, n,

(a) Compute $\hat{e}_{f,p-1}(t)$ and $\hat{e}_{b,p-1}(t)$ for t = p + 1, …, N from (5.5).

(b) Compute $\hat{k}_p$ from (5.6).

(c) Compute $\hat{a}_{p,i}$ for i= 1, …, p from (5.3).

Then $\widehat{A_n}$= $[\hat{a}_{p,1}, \ldots, \hat{a}_{p,p}]^T$ is the vector of AR coefficient estimates.

Finally, we show that the resulting AR model is stable; this is accomplished by showing that $|\hat{k}_P| \leq 1$ for p = 1 , …, n. To do so, we express $\hat{k}_P$ as

$$\hat{k}_p = \frac{-2c^*d}{c^*c + d^*d} \tag{5.1.7}$$

Where

$$c = [\hat{e}_{b,p-1}(p), \ldots, \hat{e}_{b,p-1}(N-1)]^T$$
$$d = [\hat{e}_{f,p-1}(p+1), \ldots, \hat{e}_{f,p-1}(N)]^T$$

In addition

$$\begin{aligned} 0 \leq \|c - e^{i\alpha}d\|^2 &= c^*c + d^*d - 2\,\mathrm{Re}\{e^{i\alpha}c^*d\} \quad \text{for every } \alpha \in [-\pi, \pi] \\ &\Longrightarrow 2\,\mathrm{Re}\{e^{i\alpha}c^*d\} \leq c^*c + d^*d \quad \text{for every } \alpha \in [-\pi, \pi] \\ &\Longrightarrow 2|c^*d| \leq c^*c + d^*d \implies |\hat{k}_p| \leq 1 \end{aligned}$$

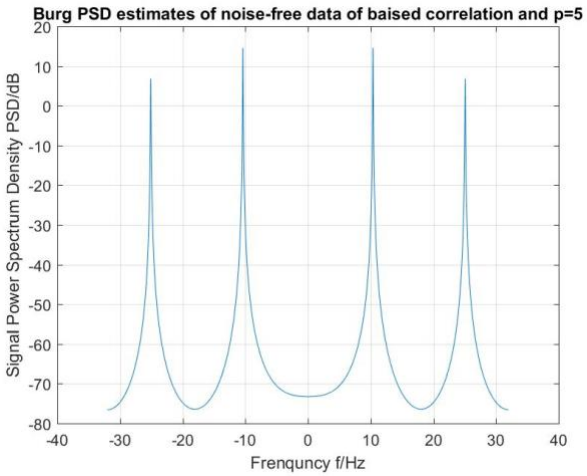
The Burg algorithm is computationally simple, and is amenable to both order-recursive and time recursive solutions. It is identical to the geometric algorithm, except that an alternative estimate for the reflection coefficient, is determined based on a least-squared criterion. At each order P the arithmetic mean of the forward and backward LP error power.

5.2 PSD estimation results

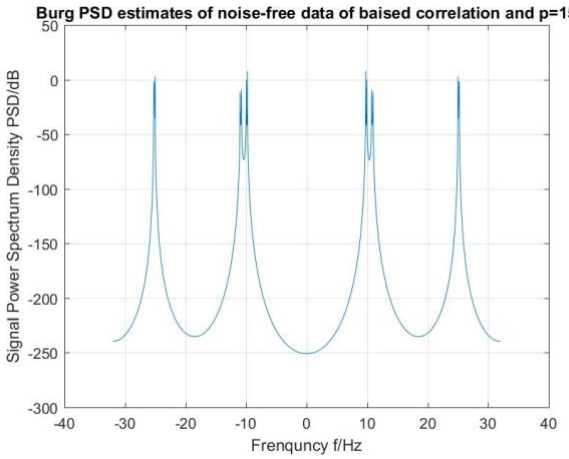
In the following, we form Burg(harmonic) PSD estimates. Use model orders of 5, 15, and 30.

First check the noise free data in Figure 5.2.1. We see at order 15, the applied method has achieved a good estimate, distinguishing the peaks at 10 and 11 Hz. This has never been as in previous chapters. Also, it shows a very good depression on other non-peak frequencies (around 90 dB).

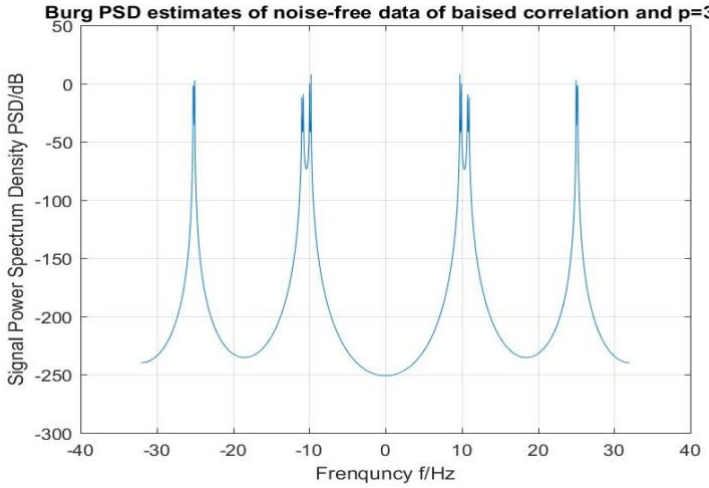
Next, we turn to noised signal in Figure 5.2.2. Even though the applied method fails to find the peaks at 15 and 16 Hz, it still depresses non-peak areas remarkably. As order increases, the false positive peaks appear more often.



a) model order =5

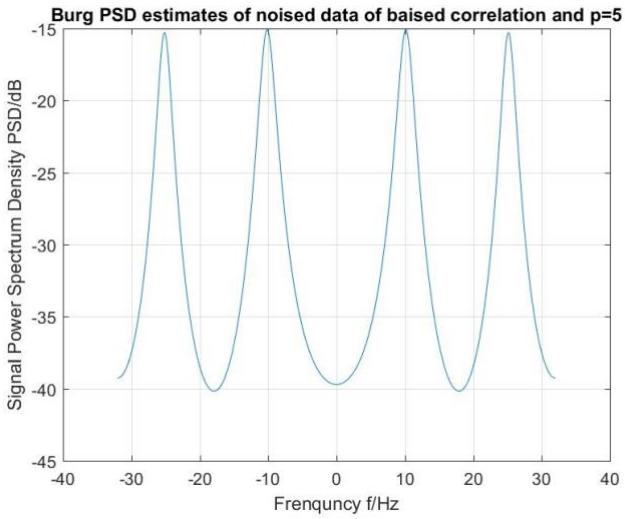


b) model order =15

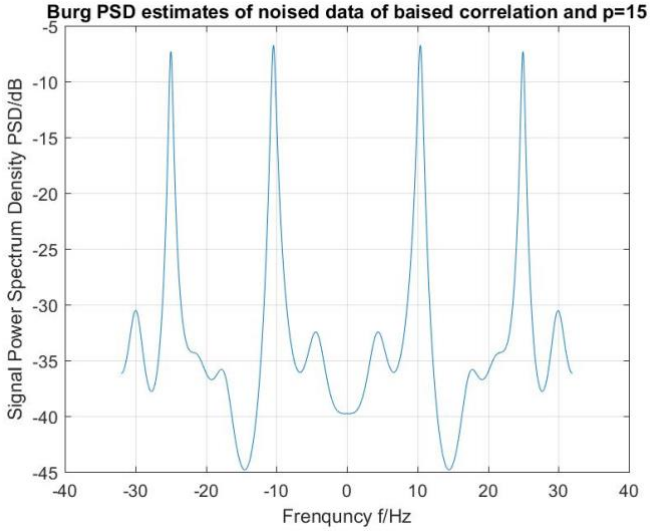


c) model order =30

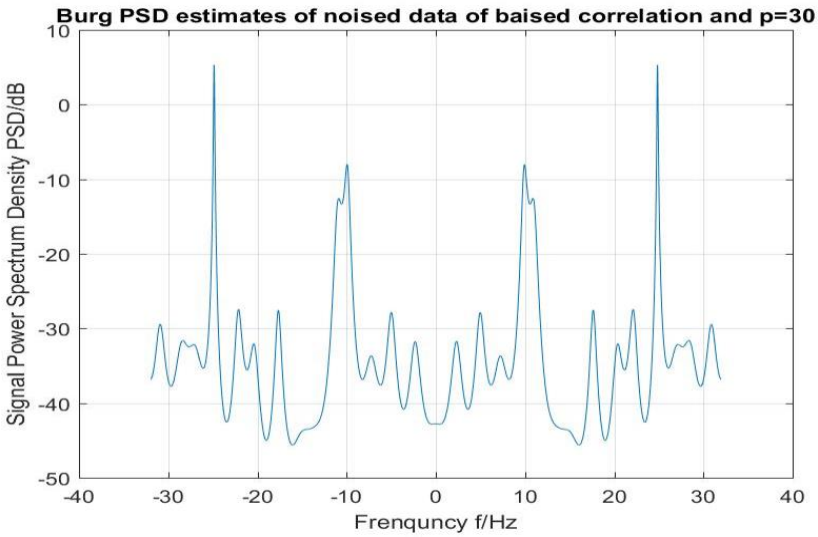
Figure 5.2.1: Burg (Harmonic) PSD estimates of noise-free signal



a) model order = 5



b) model order = 15



c) model order =30

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

As shown above, since **input** data record is **limited**, so that the impact of noise can't be removed.

As model **order increases**, we obtain **better** PSD that are distinguishing and accurate

Chapter 6

Covariance PSD estimates

6.1 Theory background

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

$$\begin{bmatrix} x[p+1] & \dots & x[1] \\ \dots & \dots & \dots \\ x[N-p] & \dots & x[p+1] \\ \dots & \dots & \dots \\ x[N] & \dots & x[N-p] \end{bmatrix} \begin{bmatrix} 1 \\ a[1] \\ \dots \\ a[p] \end{bmatrix} = \begin{bmatrix} e_p^f[p+1] \\ \dots \\ e_p^f[N-p] \\ \dots \\ e_p^f[N] \end{bmatrix} \tag{6.1.2}$$

In it, Tp 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

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

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

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

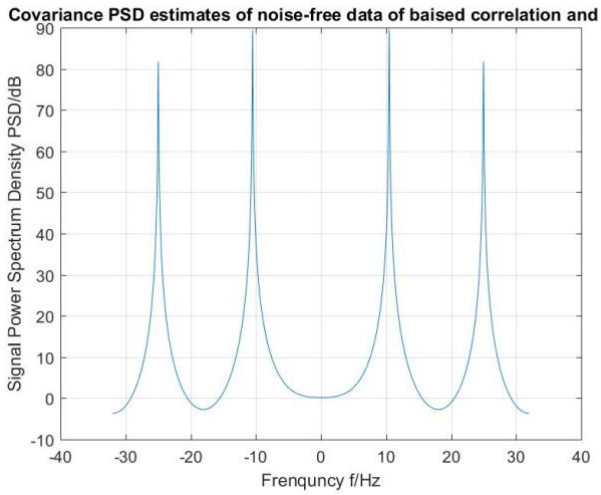
6.2 PSD estimation result

In the following, we form Covariance PSD estimates. Use model orders of 5, 15, and 30.

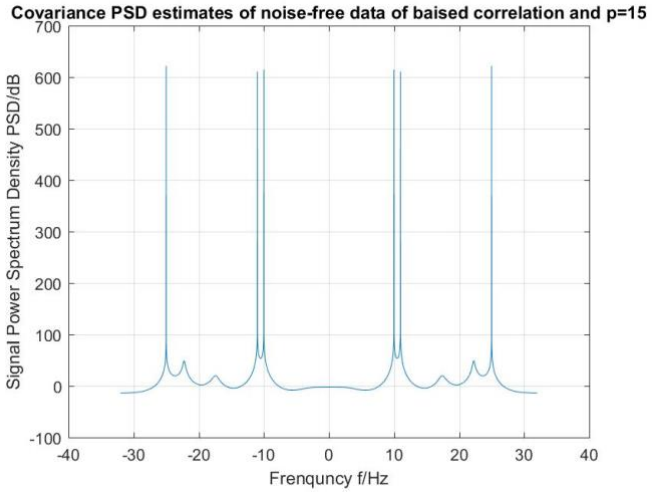
First of all, low order model (p = 5) is not suitable for the tested signals.

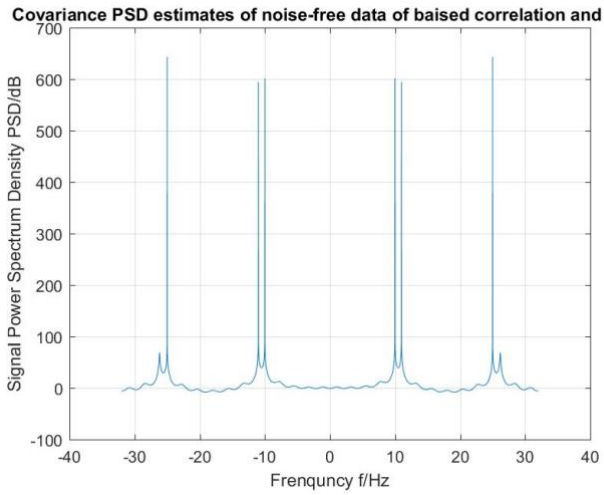
In the noise-free case, we see at model order 15 and 30, the applied method is perfect. It gives the desired PSD with very sharp impulse peaks.

a) model order =5



b) model order = 15

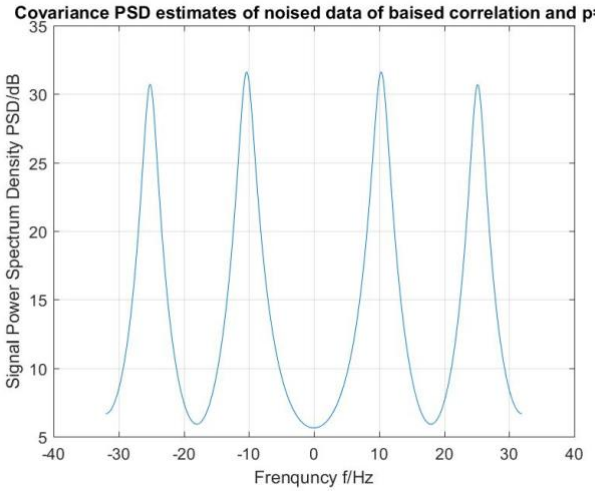




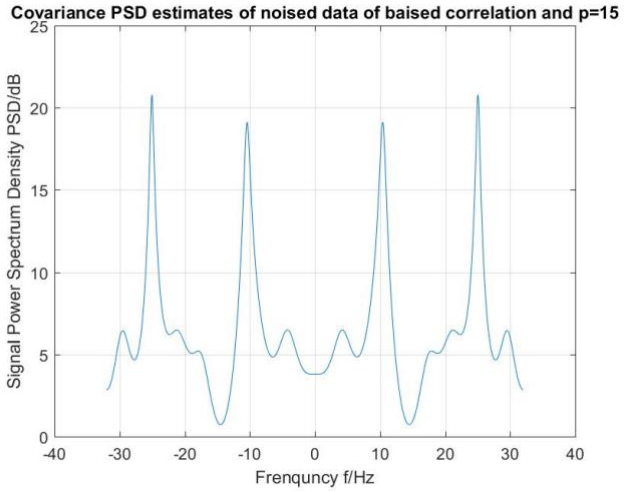
c) model order = 30

Figure 6.2.1: Covariance PSD estimates of noise-free signal

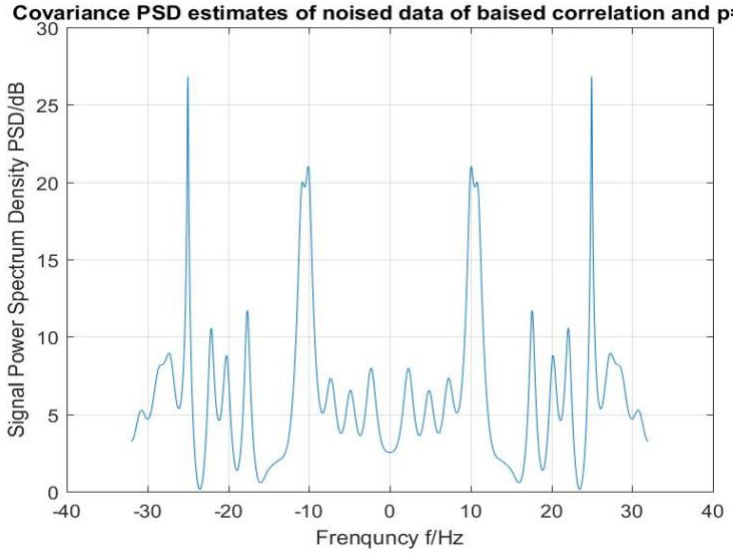
In the noised data case, we see the PSD is impacted by the noises significantly. The sharp peaks disappear, and a similar PSD (comparing with the previous mentioned methods) is achieved. However, the main peaks still exists at 10 Hz and 25 Hz. Notably, we also see a peak split at frequency 10 Hz.



a) model order = 5



b) model order = 15



c) model order = 30

Figure 6.2.2: Covariance PSD estimates of noise signal

As shown above, since **input** data record is **limited**, so that the impact of noise can't be removed.

As model **order increases**, we obtain **better** PSD that are distinguishing and accurate.

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

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

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

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

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

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

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

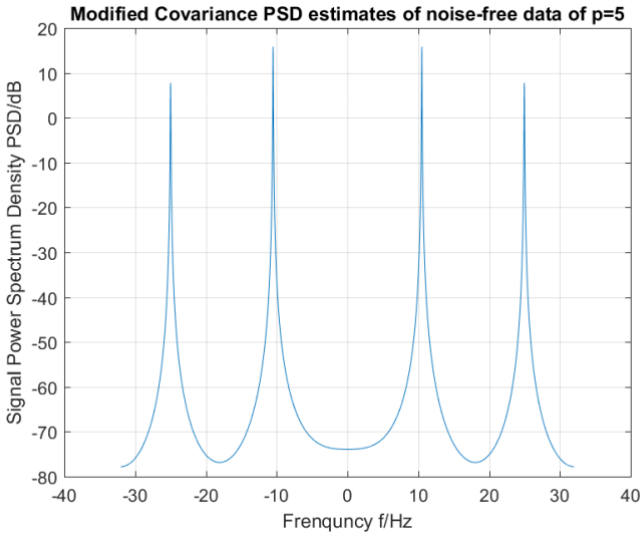
whereR_p can be found as $R_p = \begin{bmatrix} T_p \\ T_p^* J \end{bmatrix}^T \cdot \begin{bmatrix} T_p \\ T_p^* J \end{bmatrix}$. The above method is called modified covariance PSD estimate.

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

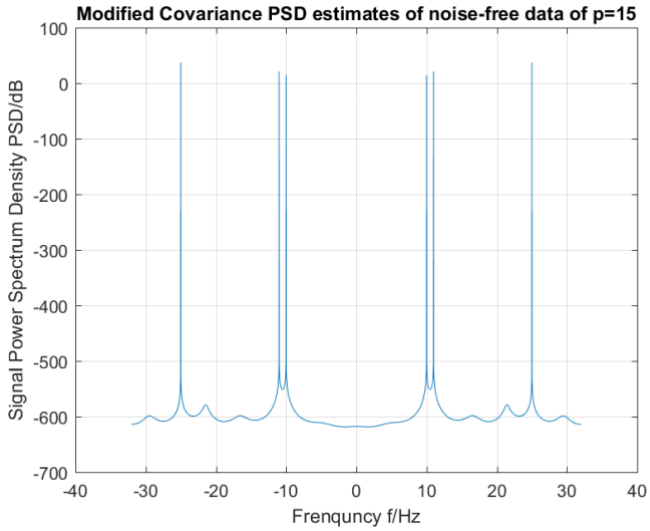
7.2 PSD estimation results

In the following, we form Modified Covariance PSD estimates. Use model orders of 5, 15, and 30.

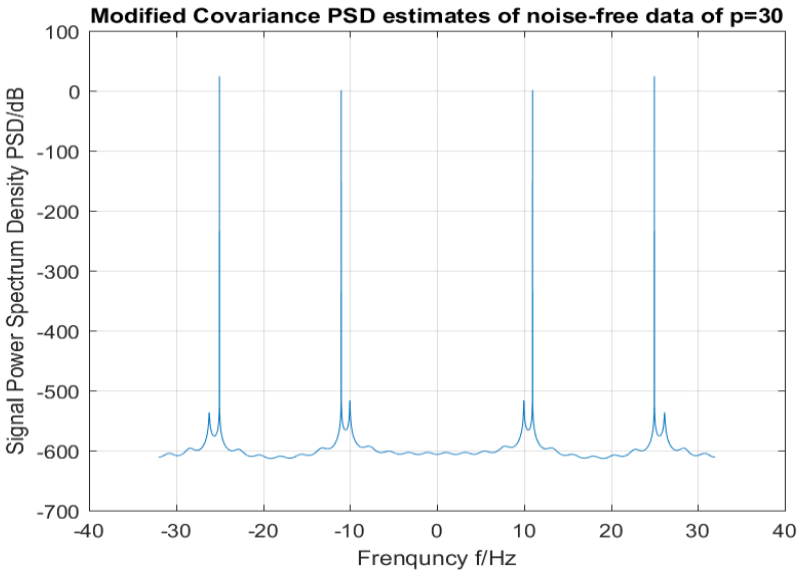
We see in noise free case, the PSD is very similar to that of covariance PSD in previous chapter. We refer readers to that chapter to see details.



a) model order = 5



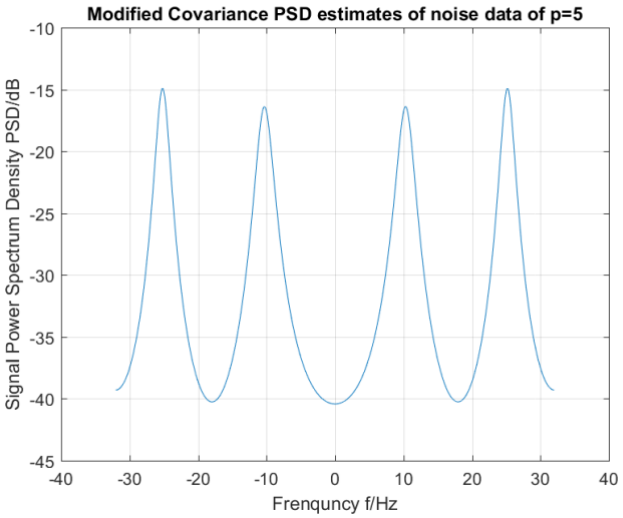
b) model order = 15



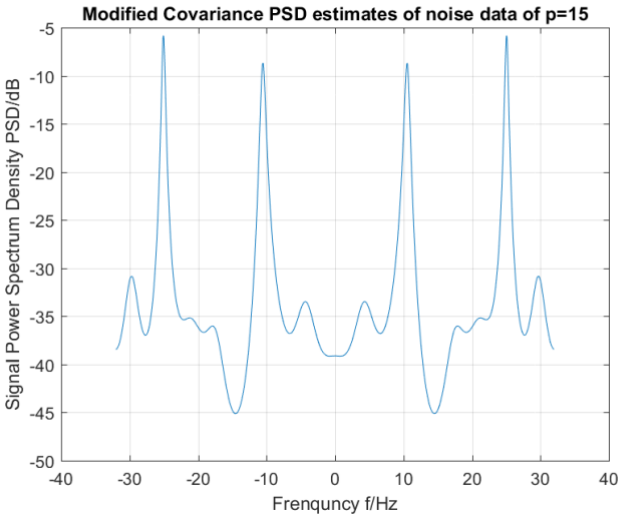
c) model order = 30

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

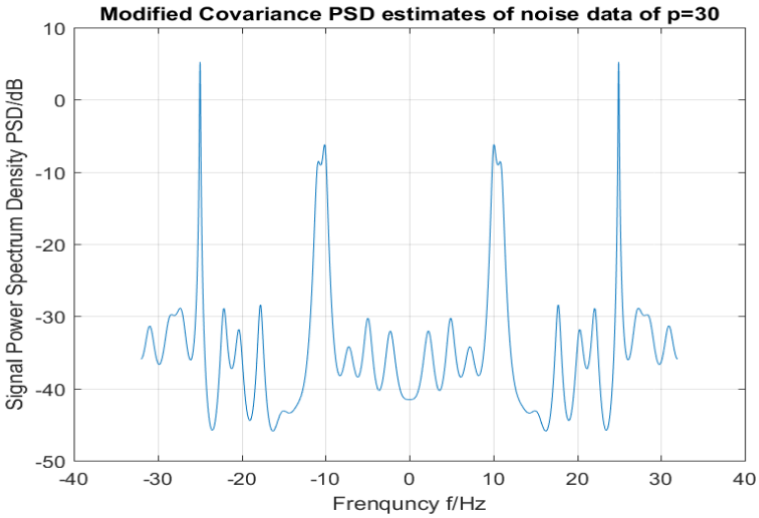
In the noise case, we note that the number of spurious peaks have been reduced, compared with the covariance PSD.



a) model order = 5



b) model order = 15



c) model order = 30

Figure 7.2.2: Modified Covariance PSD estimates of noise signal

As shown above, since **input** data record is **limited**, the correlation of noise is not exactly zero so that noise corrupts the PSD.

As model **order increases**, we obtain **better** PSD that are distinguishing and accurate. But As shown in Figure7.2.1 model with order 15 has better result than model with order 30. This may be caused by matrix computation in Matlab, and there are data loss every computation step.

Chapter 8

Adaptive least mean squares PSD estimates

8.1 Theory backgrounds

In previous chapters, we consider block algorithms, which mean complete data records are limited and starting from 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 $\bar{a}_{p,N}$,

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

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 $\bar{x}^T_{p-1}[\mathrm{N}]$ 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

In the following, we form adaptive least mean squares PSD estimates. Use model orders of 5, 15, and 30. Try three different adaptive step sizes.

We will try the step sizes 0.001, 0.01 and 0.015. 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 25 Hz components.

Next, we focus on each case with different step size, The *step size 0.01and 0.015 tend to have sharper peaks* than those of step size 0.001. And in a particular case shown in Figure 8.2.3 the *step size 0.01 with model order 30* gives the best PSD estimate.

Comparing between *different model order* with same step size in both noised and un-noised cases, from Figure 8.2.1 to Figure 8.2.3 and Figure 8.2.4 to Figure 8.2.6, as *model order goes up, PSD estimate perform a better job*.

Comparing between two cases with identical step size and model order, the noise would still corrupt the estimates.

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 25 Hz and 10 Hz. Other step size settings lead to a bad PSD, since they all missed the peaks at 25 Hz.

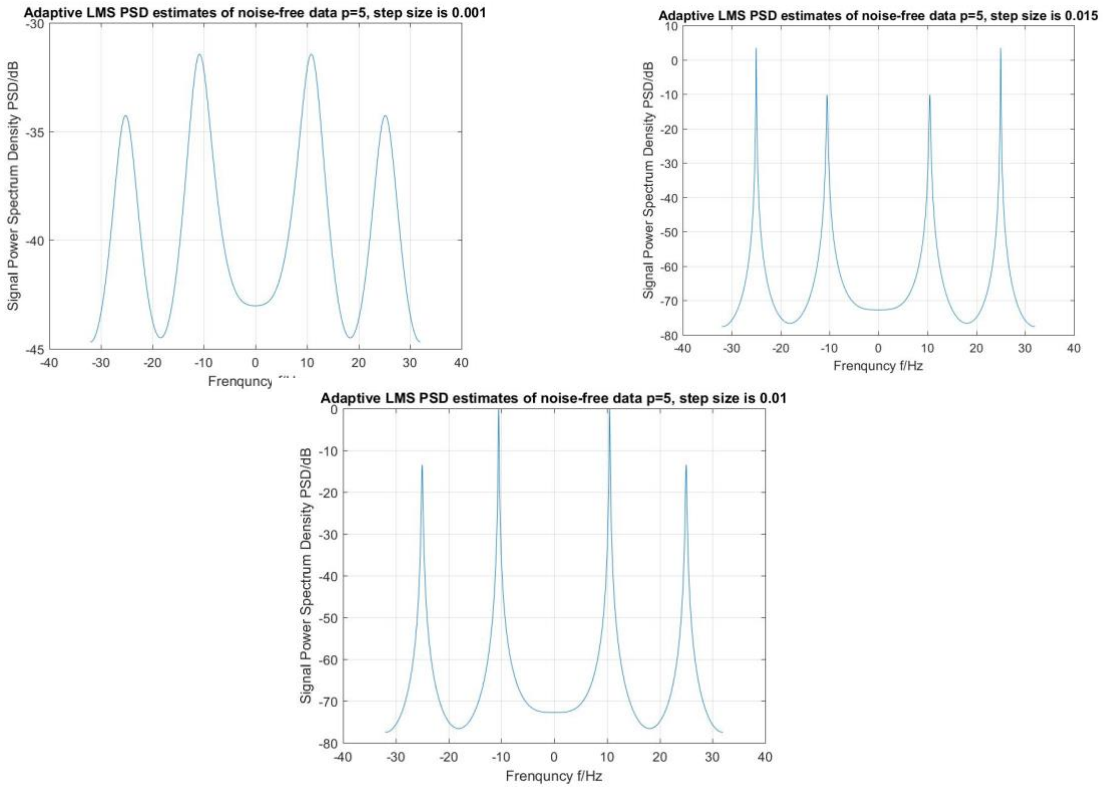


Figure 8.2.1: Adaptive LMS PSD estimates of noise-free signal, model order = 5

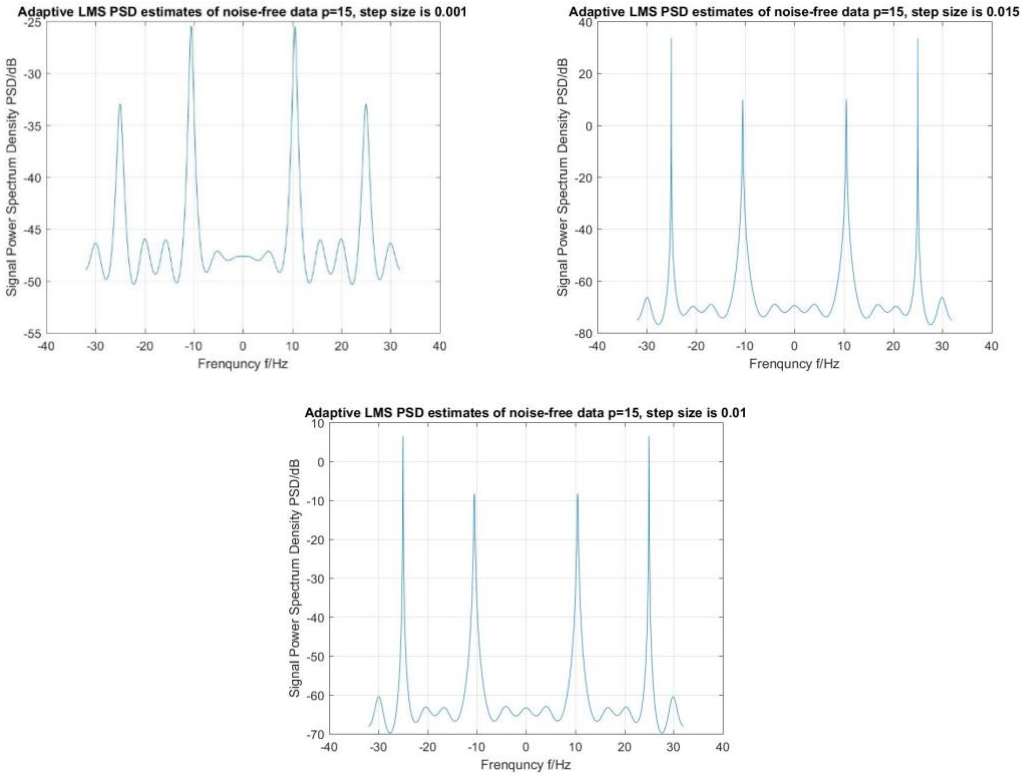


Figure 8.2.2: Adaptive LMS PSD estimates of noise-free signal, model order = 15

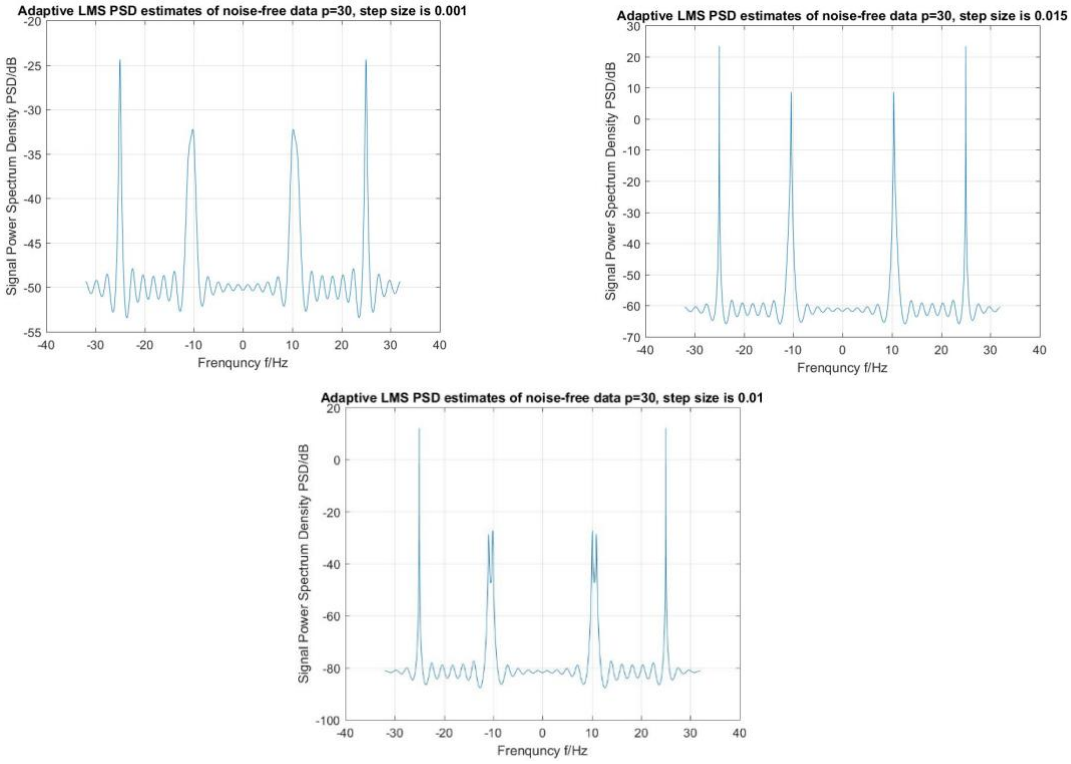


Figure 8.2.3: Adaptive LMS PSD estimates of noise-free signal, model order = 30

As shown above the model order 30 with step size 0.01 perform a nice PSD estimate.

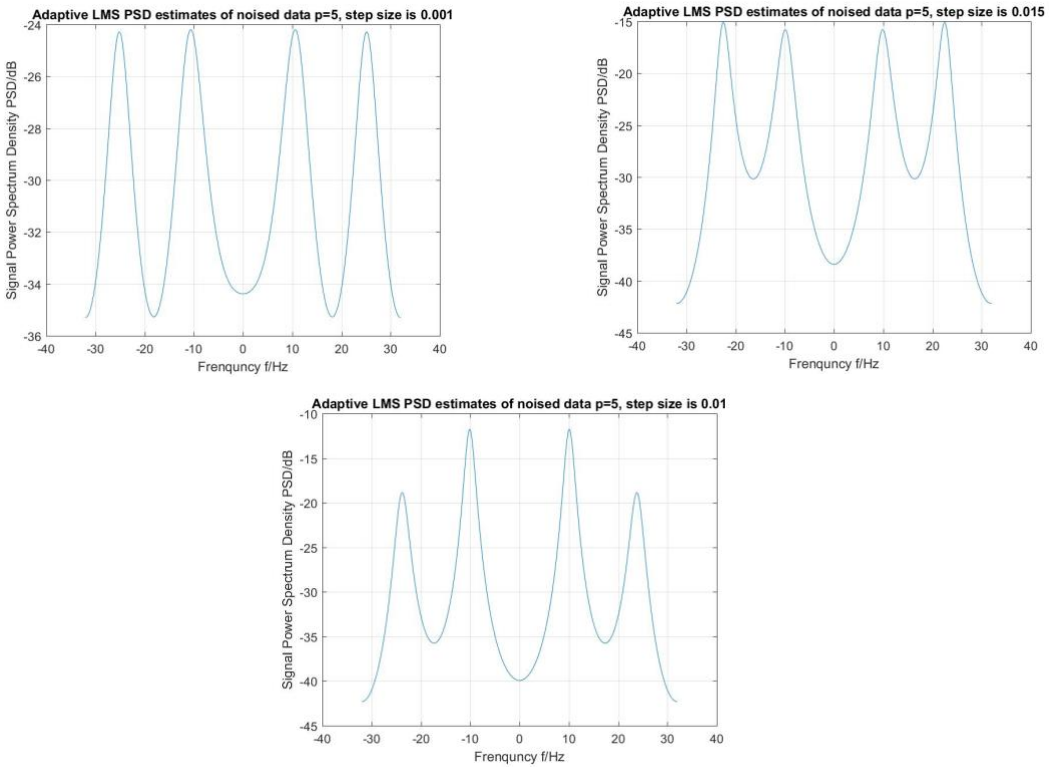


Figure 8.2.4: Adaptive LMS PSD estimates of noised signal, model order = 5

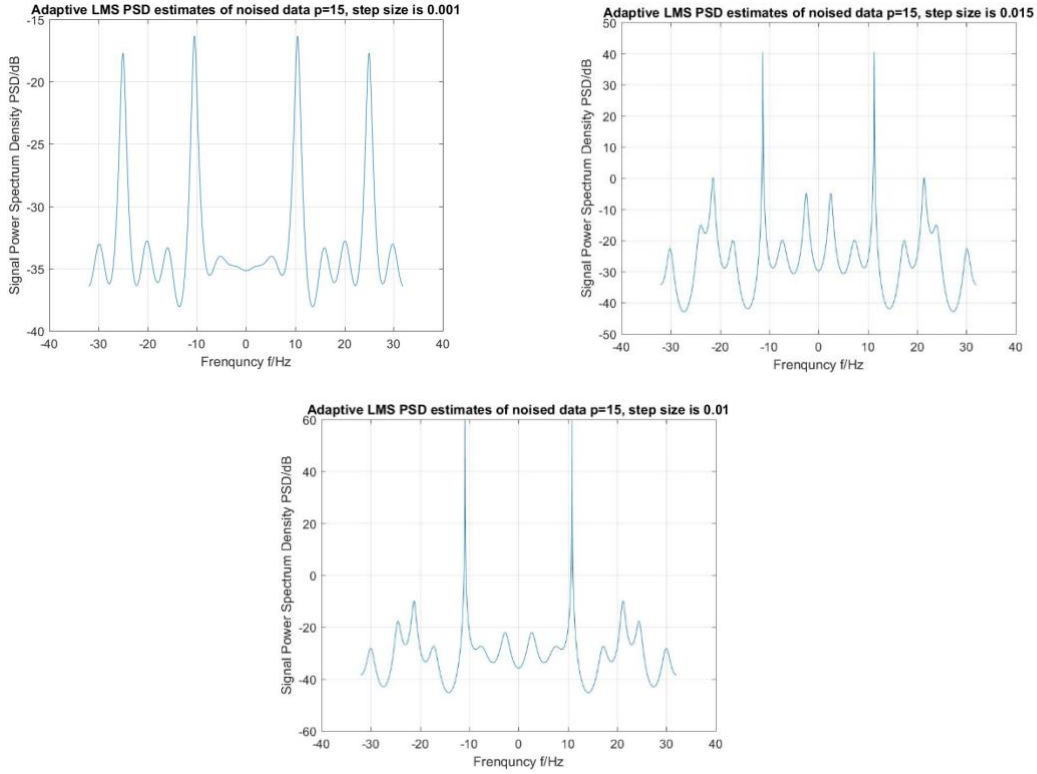


Figure 8.2.5: Adaptive LMS PSD estimates of noised signal, model order = 15

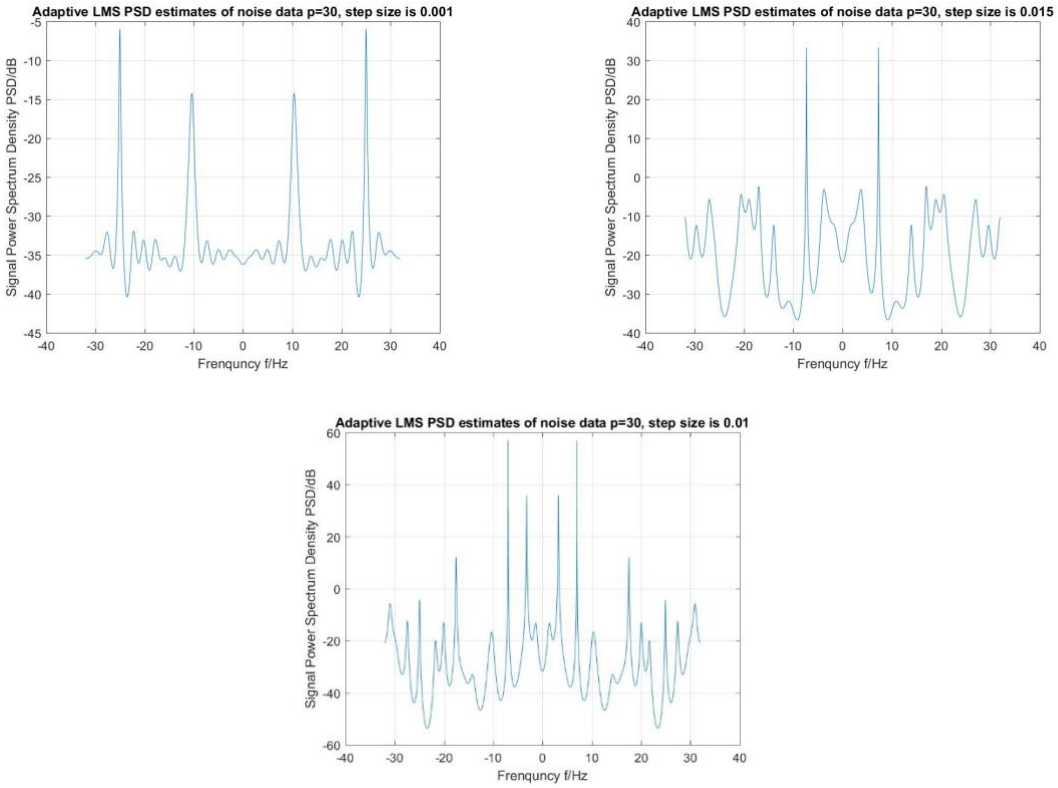


Figure 8.2.6: Adaptive LMS PSD estimates of noised signal, model order = 30

As a conclusion for this section, it is known the step size has a **trade-off** between **convergence rate and the accuracy of the AR parameter estimates**. The convergence time can't be shown in figures above but the accuracy depends on model order, step size and noise, and it is necessary to peak an appropriate step size and model order.

Chapter 9

MUSIC PSD estimates

9.1 Theory backgrounds

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

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

In the above equation, $\sum_{i=1}^{M}(\lambda_i + \mathrm{p}_w)\bar{v}_i\bar{v}_i$ represents the signal subspace, and $\sum_{i=M+1}^{p+1} \mathrm{p}_w \bar{v}_i\bar{v}_i$ represents the noise subspace. We note that the above equation is called Eigen decomposition, where $\lambda_i + \mathrm{p}_w$ and p_w is the eigen values of Rp 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 p_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

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

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

Thus, we can get a frequency estimator function

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

Let α_k = 1, we have the MUSIC method.

9.2 PSD estimation results

In the following, we form MUSIC PSD estimates. Use model orders of 5, 15, and 30. Try different values foe the number of signal vectors

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	14,10,7
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 =**7** 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. We can observe that the MUSIC method's performance is dependent on the selection of the signal subspace size.

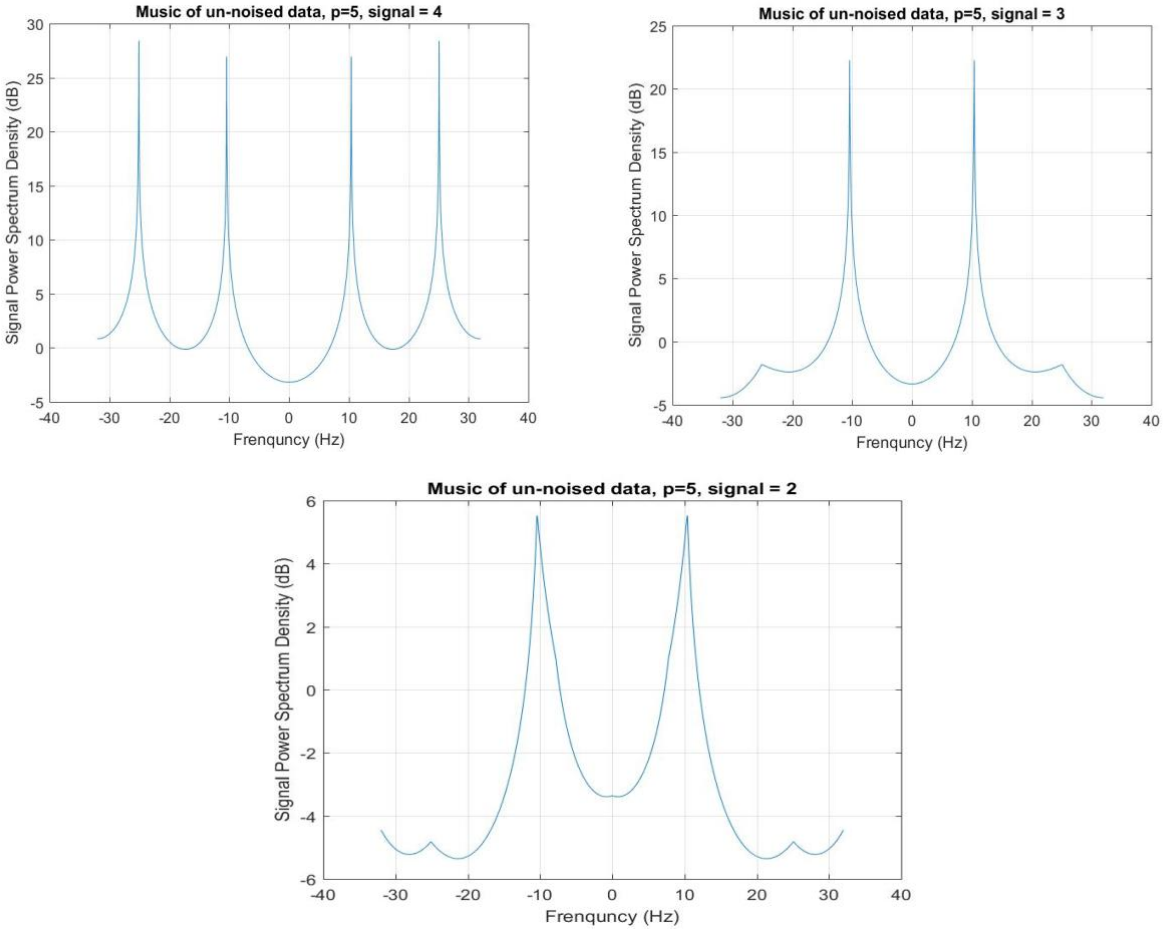


Figure 9.2.1: MUSIC PSD estimates of noise-free signal, model order = 5

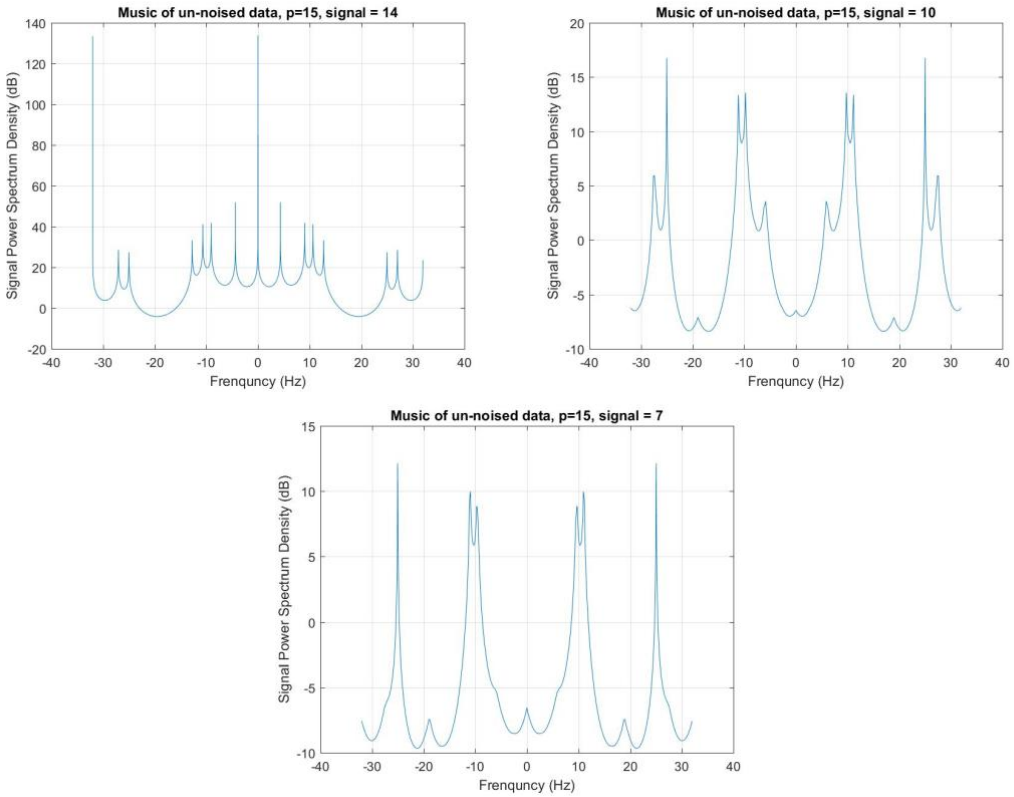


Figure 9.2.2: MUSIC PSD estimates of noise-free signal, model order = 15

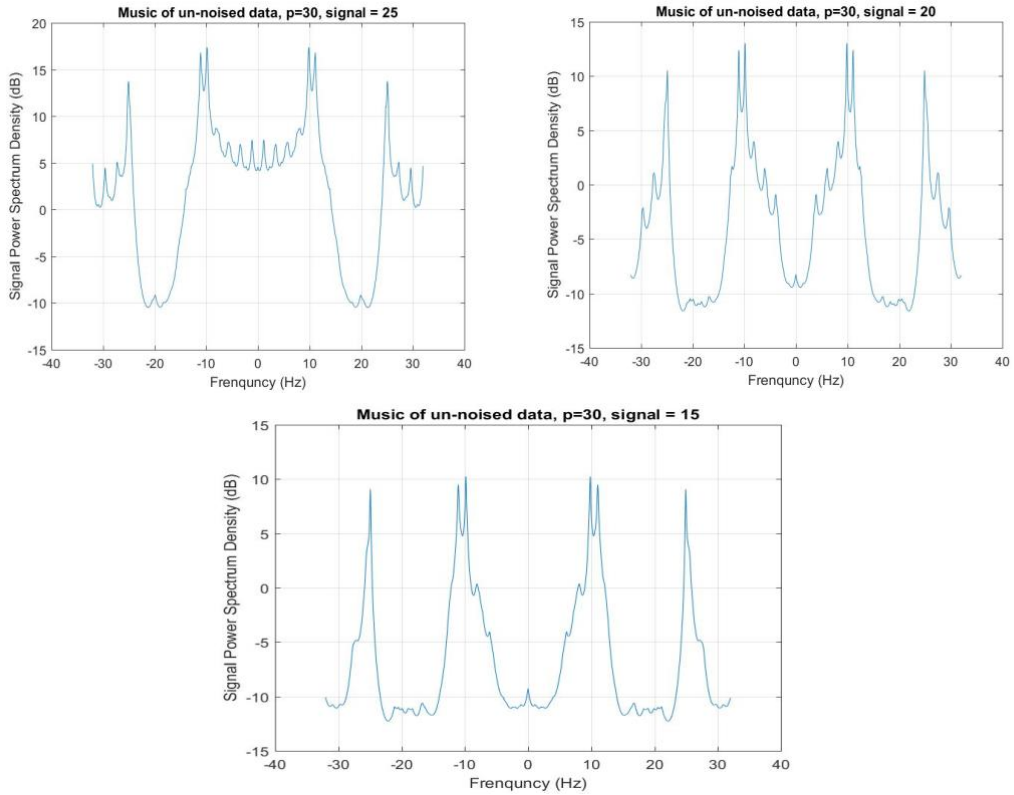


Figure 9.2.3: MUSIC PSD estimates of noise-free signal, model order = 30

Next, we exam the noised case in Figure 9.2.4 to Figure 9.2.6. We see in model order 15, signal subspace= 7 case, it gets a line spiting at 10 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.

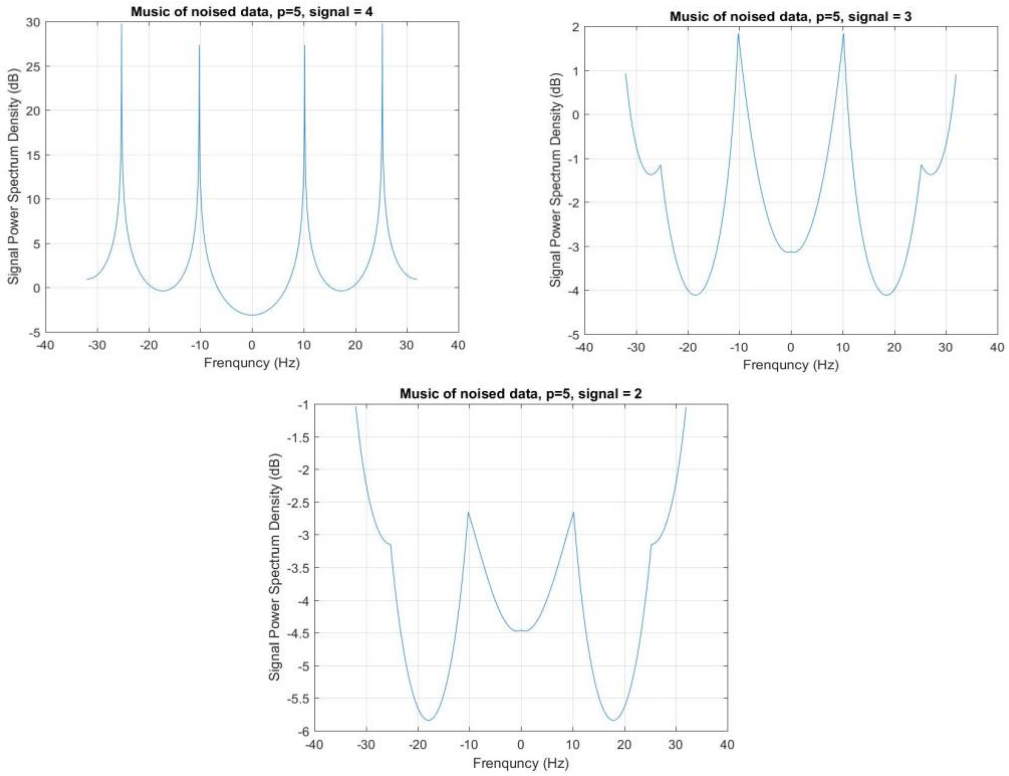
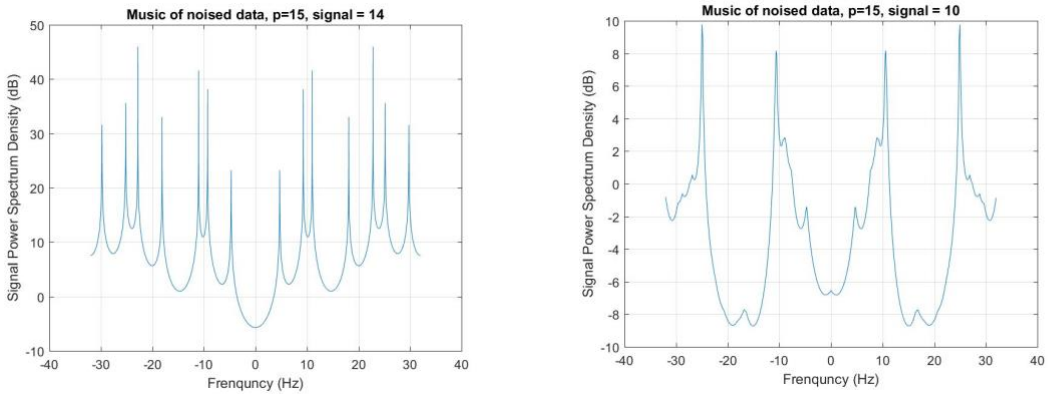


Figure 9.2.4: MUSIC PSD estimates of noised signal, model order = 5



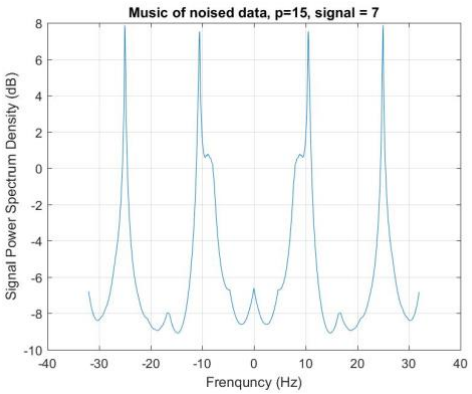


Figure 9.2.5: MUSIC PSD estimates of noised signal, model = 15

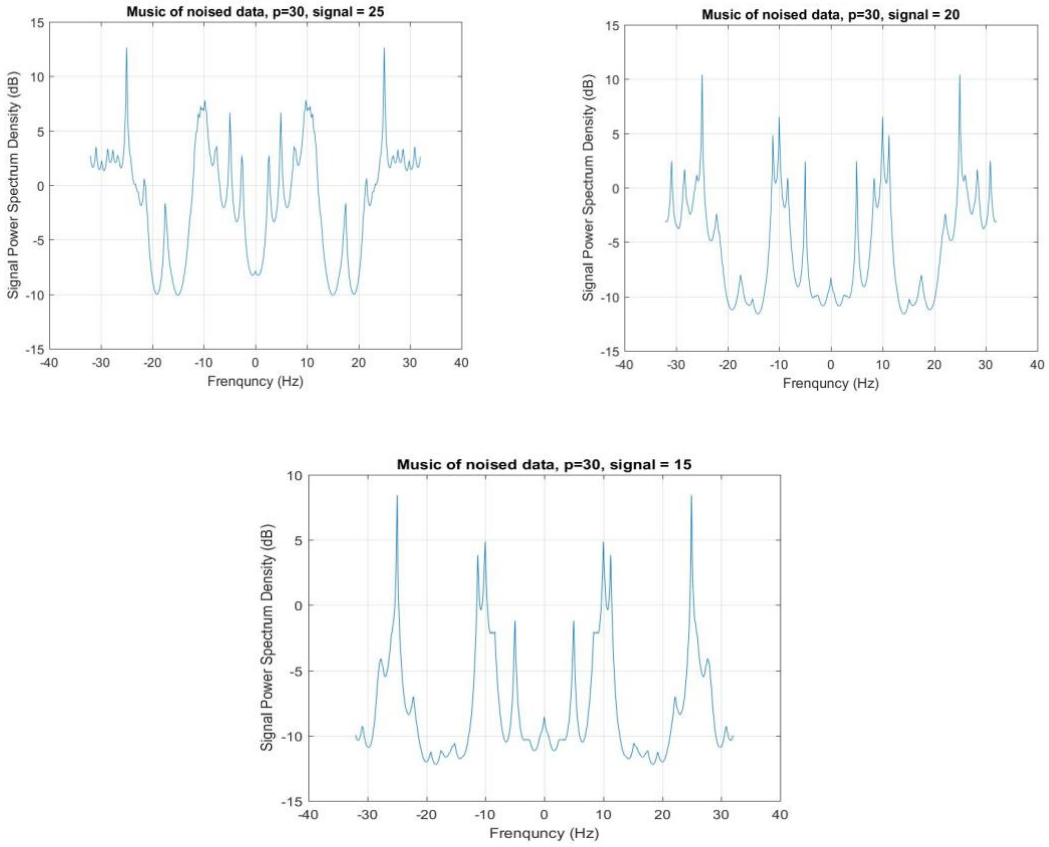


Figure 9.2.6: MUSIC PSD estimates of noised signal, model order = 30

As a conclusion for this section, increasing model order does not directly help with better performance on PSD estimates, but the performance depends **both on model order and signal subspaces size.** Noise, in this section by comparing two cases, seems to add **more sharp peaks** to the PSD estimates. But the main frequency components are **distinguished** in both noised and noise free cases when subspace is appropriate.

Chapter 10

Conclusion

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

For classic approaches, including Blackman-Tukey and Welch periodogram PSD, Welch Periodogram PSD is better than blackman-tukey, because it *resolves better peaks* as shown in the Figures, which is cause by *less variance* calculated. However, compared to other algorithms, they have the worst resolution, even if the appropriate window function is selected. The advantage of this classic approach is that they rely *directly on signal data* and *do not depend on the model* (and the corresponding model sequence) as a priori. Furthermore, it may require *more computation and less resolution*. Basically, this classic approach is *not efficient* and will not be the first choice if we can use other methods.

Next, in the *parameters based* approach, including the Yule-Walker PSD, Burg PSD, Covariance PSD and Mofified Covariance PSD, they all have better resolution if the appropriate model order is selected.

In the noise free case: given the correct model order, we see the *least squares* method (i.e. Covariance PSD and Modified Covariance PSD) give very sharp peaks in PSD estimate, perfect performance under the circumstance of no noise. The berg algorithm also found a near peak, but the peak is not clear.

To remove the impact of noise, Modified the covariance PSD is superior to the other, it attenuate frequencies except signal frequencies sharply in PSD estimate .

In general, the Modified Covariance PSD is preferred in *parameters based* methods. At the same time, the burg's algorithm is a good choice *less resolution* can be achieved.

Along another track, the order method was tested and the LMS algorithm was adapted. This method is required when we need to do *real-time* processing. It converges to the LMS method when selecting the *correct sequence and step* length.

Finally, if *good signal subspace size and model order* are selected, the method MUSIC performs well based on the method. For MUSIC, the choice of *signal subspace size* is *more important* than the order of the models. Although MUSIC gives a good performance, this method may not be as easy to implement as the parameter approach.