Chapter 12
Spectral Analysis of Signals

The process of determining the frequency contents of a continuous-time signal in the discrete-time domain is known as spectral analysis. Most of the phenomena that occur in nature can be characterized statistically by random processes. Hence, the main objective of spectral analysis is the determination of the power spectrum density (PSD) of a random process. The power is the Fourier transform of the autocorrelation sequence of a stationary random process. The PSD is a function that plays a fundamental role in the analysis of stationary random processes in that it quantifies the distribution of the total power as a function of frequency. The power spectrum also plays an important role in detection, tracking, and classification of periodic or narrowband processes buried in noise. Other applications of spectrum estimation include harmonic analysis and prediction, time series extrapolation and interpolation, spectral smoothing, bandwidth compression, beam forming, and direction finding. The estimation of the PSD is based on a set of observed data samples from the process. Estimating the power spectrum is equivalent to estimating the autocorrelation. This chapter deals with the nonparametric methods, parametric methods, and subspace methods for power spectrum estimation. Further, the spectrogram computation of non-stationary signals using STFT is also briefly discussed in this chapter.

12.1 Nonparametric Methods for Power Spectrum Estimation

Classical spectrum estimators do not assume any specific parametric model for the PSD. They are based solely on the estimate of the autocorrelation sequence of the random process from the observed data and hence work in all possible situations, although they do not provide high resolution. In practice, one cannot obtain

K. D. Rao and M. N. S. Swamy, *Digital Signal Processing*,
https://doi.org/10.1007/978-981-10-8081-4_12

unlimited data record due to constraints on the data collection process or due to the necessity that the data must be WSS over that particular duration.

When the method for PSD estimation is not based on any assumptions about the generation of the observed samples other than wide-sense stationary, then it is termed a nonparametric estimator.

12.1.1 Periodogram

The periodogram was introduced in [1] searching for hidden periodicities while studying sunspot data. There are two distinct methods to compute the periodogram. One approach is the indirect method. In this approach, first we determine the autocorrelation sequence $r(k)$ from the data sequence *x(n)* for $-(N-1) \leq k \leq (N-1)$ and then take the DTFT, i.e.,

$$\hat{P}_{\text{PER}}(f) = \sum_{k=-N+1}^{N-1} \hat{r}[k]\mathrm{e}^{-j2\pi fk}. \tag{12.1}$$

It is more convenient to write the periodogram directly in terms of the observed samples $x[n]$. It is then defined as

$$\hat{P}_{\text{PER}}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j2\pi fn}\right|^2 = \frac{1}{N}|X(f)|^2 \tag{12.2}$$

where $X(f)$ is the Fourier transform of the sequence *x(n)*. Thus, the periodogram is proportional to the squared magnitude of the DTFT of the observed data. In practice, the periodogram is calculated by applying the FFT, which computes it at a discrete set of frequencies.

$$D_f = \left\{f_k : f_k = \frac{k}{N}, \quad k = 0, 1, 2, \ldots, (N-1)\right\}$$

The periodogram is then expressed by

$$\hat{P}_{\text{PER}}(f_k) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j2\pi kn/N}\right|^2 \quad f_k \in D_f. \tag{12.3}$$

To allow for finer frequency spacing in the computed periodogram, we define a zero-padded sequence according to

$$x'[n] = \begin{cases} x[n], & n = 0, 1, \ldots, N-1 \\ 0, & n = N, N+1, \ldots, N' \end{cases}. \tag{12.4}$$

Then we specify the new set of frequencies $D'_f = \{f_k : f_k = k/N, k \in \{0, 1, 2, \ldots, (N-1)\}\}$, and obtain

$$\hat{P}_{\text{PER}}(f_k) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-j2\pi kn/N'}\right|^2 \quad f_k \in D'_f. \tag{12.5}$$

A general property of good estimators is that they yield better estimates when the number of observed data samples increases. Theoretically, if the number of data samples tends to infinity, the estimates should converge to the true values of the estimated parameters. So, in the case of a PSD estimator, as we get more and more data samples, it is desirable that the estimated PSD tends to the true value of the PSD. In other words, if for finite number of data samples the estimator is biased, the bias should tend to zero as $N \to \infty$ as should the variance of the estimate. If this is indeed the case, the estimator is called consistent. Although the periodogram is asymptotically unbiased, it can be shown that it is not a consistent estimator. For example, if $\{\tilde{X}[n]\}$ is real zero mean white Gaussian noise, which is a process whose random variables are independent, Gaussian, and identically distributed with variance σ^2, the variance of $\hat{P}_{\text{PER}}(f)$ is equal to σ^4 regardless of the length N of the observed data sequence. The performance of the periodogram does not improve as N gets larger because as N increases, so does the number of parameters that are estimated, $P\ (f_0)$, $P\ (f_1)$, …, $P\ (f_{N-1})$. In general, the variance of the periodogram at any given frequency is

$$\begin{aligned} \operatorname{Var}\left[\widehat{P}_{\text{PER}}(f)\right] &= \operatorname{Cov}\left[\widehat{P}_{\text{PER}}(f), \widehat{P}_{\text{PER}}(f)\right] \\ &= P^2(f)\left[1 + \left(\frac{\sin 2\pi Nf}{N \sin 2\pi f}\right)^2\right] \end{aligned} \tag{12.6a}$$

For frequencies not near 0 or 1/2, the above equation reduces to

$$\operatorname{Var}(\hat{P}_{\text{PER}}(f)) \cong P^2(f) \tag{12.6b}$$

where $P^2(f)$ is the periodogram spectral estimation based on the definition of PSD.

Example 12.1 Consider a random signal composed of two sinusoidal components of frequencies 120 and 280 Hz corrupted with Gaussian distributed random noise. Evaluate its power spectrum using periodogram. Assume sampling frequency F_s = 1024 Hz.

Solution The following MATLAB program can be used to evaluate the power spectrum of the considered signal using Bartlett's method.

%Program 12.1

```
Power spectrum estimation using the periodogram
clear;clc;
N = 512;%total number of samples
k = 0:N-1;
f1 = 120;
f2 = 280;
F_T = 1024;%sampling frequency in Hz
T = 1/F_T;
x = sin(2*pi*f1*k*T) + sin(2*pi*f2*k*T)+2*randn(size(k));%vector of length N
%containing input samples
[pxx,f] = psd(x,length(x),F_T);
plot(f,pxx);grid;
xlabel('Frequency(Hz)');ylabel('Power spectrum');
```

The power spectrum obtained from the above MATLAB program is shown in Fig. 12.1.

12.1.2 Bartlett Method

In the Bartlett method [2], the observed data is segmented into K non-overlapping segments and the periodogram of each segment is computed. Finally, the average of periodogram of all the segments is evaluated. The Bartlett estimator has a variance that is smaller than the variance of the periodogram.

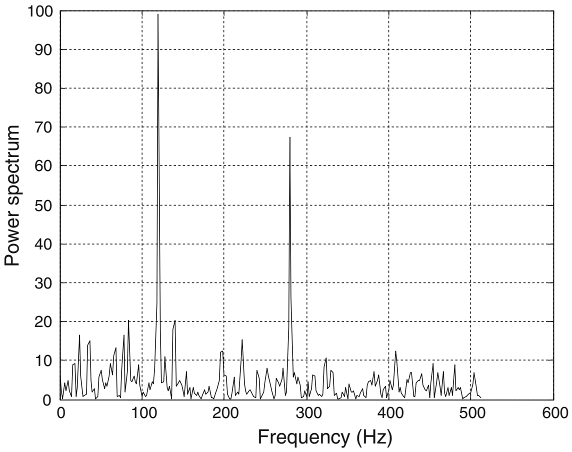


Fig. 12.1 Power spectrum estimation using the periodogram

Consider a length N sequence $x(n)$. Then, $x(n)$ can be segmented into K subsequence, each subsequence having a length L. If the ith subsequence is denoted by $x_i(n)$, $0 \leq i < K$, then the ith subsequence can be obtained from the sequence $x(n)$ as

$$x_i(n) = x(iL + n), \quad 0 \leq n \leq L - 1$$

and its periodogram is given by

$$\widehat{P}_i(f) = \frac{1}{L}\left|\sum_{n=0}^{L-1} x_i(n)\mathrm{e}^{-j2\pi fn}\right|^2 \quad i = 0, 1, \ldots, K - 1 \tag{12.7}$$

Then the Bartlett spectrum estimator is

$$\widehat{P}_{\mathrm{B}}(f) = \frac{1}{K}\sum_{i=1}^{K} \widehat{P}_i(f) \tag{12.8}$$

$$\mathrm{Var}\left(\widehat{P}_{\mathrm{B}}(f)\right) = \frac{1}{K}P^2(f). \tag{12.9}$$

The variance of the Barlett estimator can be related to the variance of the periodogram as follows.

The Bartlett estimator variance is reduced by a factor of K compared to the variance of the periodogram. However, the reduction in the variance is achieved at the cost of decrease in resolution. Thus, this estimator allows for a trade-off between resolution and variance.

The following example illustrates the computation of the power spectrum of a random signal using the Bartlett method.

Example 12.2 Consider the random signal of Example 12.1 and evaluate its power spectrum using Bartlett's method.

Solution The following MATLAB program can be used to evaluate the power spectrum of the considered signal using Bartlett's method.

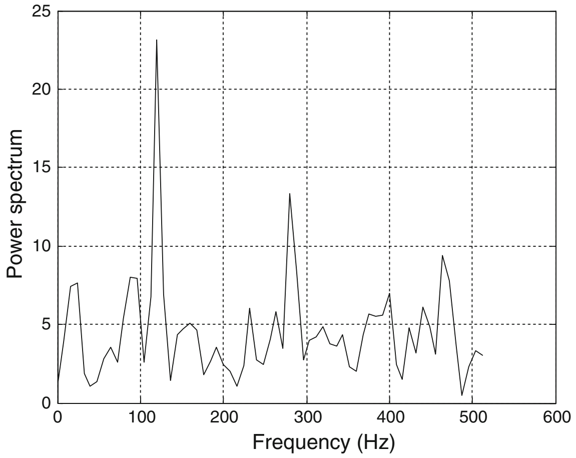


Fig. 12.2 Power spectrum estimation using Bartlett's method

%Program 12.2

```
Power spectrum estimation using Bartlett's method
clear;clc;
N = 512;%total number of samples
k = 0 : N-1;
f1 = 120;
f2 = 280;
FT = 1024;%sampling frequency in Hz
T = 1/FT;
x = sin(2*pi*f1*k*T) + sin(2*pi*f2*k*T)+2*randn(size(k));%vector of length N
%containing input samples
L = 128;%length of subsequence
[pxx,f] = psd(x,L,FT);
plot(f,pxx);grid;
xlabel('Frequency(Hz)');ylabel('Power spectrum');
```

The power spectrum obtained from the above MATLAB program is shown in Fig. 12.2.

12.1.2.1 Welch Method

The Welch method [3] is another estimator that exploits the periodogram. It is based on the same idea as the Bartlett's approach of splitting the data into segments and finding the average of their periodogram. The difference is that the segments are overlapped, and the data within a segment is windowed. If a sequence *x(n)* of length *N* is segmented into *K* subsequences, each subsequence having a length *L* with an overlapping of *D* samples between the adjacent subsequences, then

$$N = L + D(K - 1) \tag{12.10}$$

where N is the total number of observed samples and K the total number of subsequences. Note that if there is no overlap, $K = N/L$, and if there is 50% overlap, $K = 2\ N/L - 1$.

The ith subsequence is defined by

$$x_i(n) = x(n + iD), \quad 0 \le n \le L - 1; 0 \le i \le K - 1, \tag{12.11}$$

and its periodogram is given by

$$\hat{P}_i(f) = \frac{1}{L}\left|\sum_{n=0}^{L-1} w(n)x_i(n)\mathrm{e}^{-j2\pi fn}\right|^2. \tag{12.12}$$

Here $\hat{P}_i(f)$ is the modified periodogram of the data because the samples $x(n)$ are weighted by a non-rectangular window $w(n)$; the Welch spectrum estimate is then given by

$$\widehat{P}_{\mathrm{Wel}}(f) = \frac{1}{KC}\sum_{i=1}^{K} \widehat{P}_i(f) \tag{12.13}$$

where C is the normalization factor for power in the window function given by

$$C = \frac{1}{K}\sum_{n=0}^{K-1} w^2(n)$$

Welch has shown that the variance of the estimator is

$$\mathrm{Var}\left(\widehat{P}_{\mathrm{Wel}}(f)\right) \approx \frac{1}{K}P^2(f) \quad \text{for no overlapping} \tag{12.14a}$$

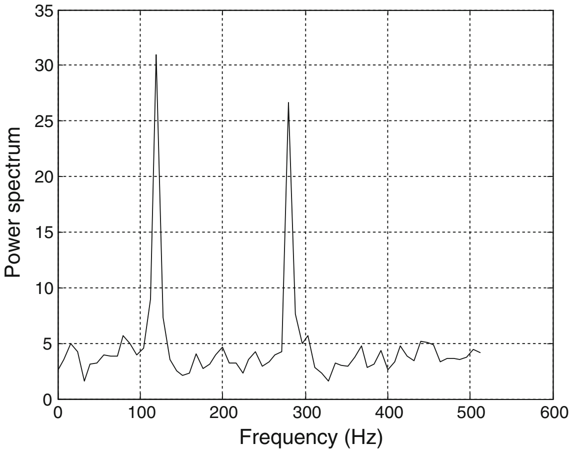
$$\approx \frac{9}{8K}P^2(f) \quad \text{for 50\% overlapping and Bartlett window.} \tag{12.14b}$$

By allowing overlap of subsequences, more number of subsequences can be formed than in the case of Bartlett's method. Consequently, the periodogram evaluated using the Welch's method will have less variance than the periodogram evaluated using the Bartlett method.

Example 12.3 Consider the random signal of Example 12.1 and evaluate its power spectrum using Welch's method with 50% overlapping and Hamming window.

Solution The following MATLAB program can be used to evaluate the power spectrum of the considered signal using Welch's method.

Fig. 12.3 Power spectrum estimation using Welch's method



%Program 12.3

```
Power spectrum estimation using Welch's method
clear;clc;
N = 512;%total number of samples
k = 0 : N-1;
f1 = 120;
f2 = 280;
FT = 1024;%sampling frequency in Hz
T = 1/FT;
x = sin(2*pi*f1*k*T) + sin(2*pi*f2*k*T)+2*randn(size(k));%vector of length N
%containing input samples
L = 128;%length of subsequence
window = hamming(L);% window type
overlap = L/2;%number of overlapping samples(50%overlapping)
[pxx,f] = psd(x,L,FT,window,overlap);
plot(f,pxx);grid;
xlabel('Frequency(Hz)');ylabel('Power spectrum');
```

The power spectrum obtained from the above MATLAB program is shown in Fig. 12.3.

12.1.2.2 Blackman–Tukey Method

In this method, autocorrelation of the observed data sequence *x(n)* is computed first. Next, the autocorrelation is windowed and then the Fourier transform is applied on it to obtain the power spectrum. Hence, the power spectrum using the Blackman–Tukey method [4] is given by

$$\widehat{P}(f) = \sum_{k=-(N-1)}^{N-1} w(k)\widehat{r}(k)\mathrm{e}^{-j2\pi fk} \tag{12.15}$$

where the window $w(k)$ is real nonnegative, symmetric, and non-increasing with $|k|$, that is,

$$\text{(i)} \quad 0 \le w(k) \le w(0) = 1 \tag{12.16a}$$

$$\text{(ii)} \quad w(-k) = w(k) \tag{12.16b}$$

$$\text{(iii)} \quad w(k) = 0.M < |k|. \quad M \le N - 1. \tag{12.16c}$$

It should be noted that the symmetry property of $w(k)$ ensures that the spectrum is real. It is obvious that the autocorrelation with smaller lags will be estimated more accurately than the ones with lags close to N because of the different number of terms that are used. Therefore, the large variance of the periodogram can be ascribed to the large weight given to the poor autocorrelation estimates used in its evaluation. Blackman and Tukey proposed to weight the autocorrelation sequence so that the autocorrelations with higher lags are weighted less. The bias, the variance, and the resolution of the Blackman–Tukey method depend on the applied window. For example, if the window is triangular (Bartlett),

$$w_{\mathrm{B}}[k] = \begin{cases} \frac{M-|k|}{M}, & |k| \le M \\ 0, & \text{otherwise} \end{cases} \tag{12.17}$$

and if $N \gg M \gg 1$, the variance of the Blackman–Tukey estimator is

$$\mathrm{Var}(\hat{P}_{\mathrm{BT}}(f)) \cong \frac{2M}{3N} P^2(f) \tag{12.18}$$

where $P(f)$ is the true spectrum of the process. Compared to Eqs. (12.6a) and (12.6b) it is clear that the variance of this estimator may be significantly smaller than the variance of the periodogram. However, as M decreases, so does the resolution of the Blackman–Tukey estimator.

Example 12.4 Consider the random signal of Example 12.1 and evaluate its power spectrum using Blackman–Tukey method.

Solution The following MATLAB program can be used to evaluate the power spectrum of the considered signal using Blackman–Tukey method.

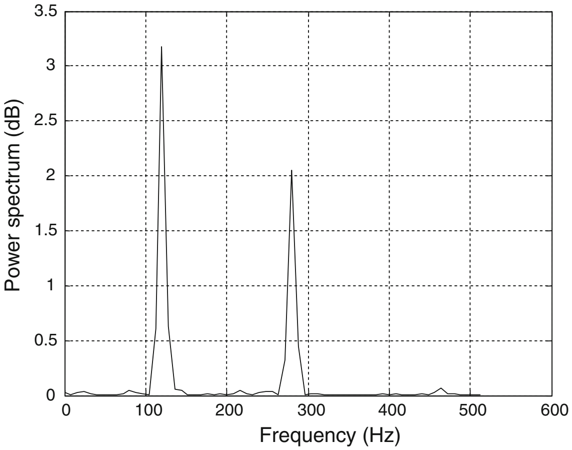


Fig. 12.4 Power spectrum estimation using Blackman–Tukey method

%Program 12.4

```
Power spectrum estimation Blackman–Tukey method
clear;clc;
N = 512;%total number of samples
k = 0 : N-1;
f1 = 120;
f2 = 280;
FT = 1024;%sampling frequency in Hz
T = 1/FT;
x = sin(2*pi*f1*k*T)+sin(2*pi*f2*k*T)+2*randn(size(k));%vector of length N
%containing input samples
r = f_corr(x,x,0,0);% evaluates correlation of input samples
L = 128;%length of window
window = Bartlett(L);% window type
[pxx,f] = psd(r,L,FT,window);
plot(f,pxx);grid
xlabel('Frequency(Hz)');ylabel('Power spectrum(dB)');
```

The power spectrum obtained from the above MATLAB program is shown in Fig. 12.4.

12.1.3 Performance Comparison of the Nonparametric Methods

The performance of a PSD estimator is evaluated by quality factor. The quality factor is defined as the ratio of the squared mean of the PSD to the variance of the PSD given by

Table 12.1 Comparison of performance of classical methods

Classical method	Quality factor
Periodogram	1
Bartlett	1.11 N Δf
Welch (50% overlap)	1.39 N Δf
Blackman–Tukey	2.34 N Δf

$$Q_P = \frac{\mathrm{var}(\hat{p}(f))}{E^2(\hat{p}(f))}. \tag{12.19}$$

Another important metric for comparison is the resolution of the PSD estimators. It corresponds to the ability of the estimator to provide the fine details of the PSD of the random process. For example, if the PSD of the random process has two peaks at frequencies f_1 and f_2, then the resolution of the estimator would be measured by the minimum separation of f_1 and f_2 for which the estimator still reproduces two peaks at f_1 and f_2. It has been shown in [5] for triangular window that the quality factors of the classical methods are as shown in Table 12.1.

From the above table, it can be observed that the quality factor is dependent on the product of the data length N and the frequency resolution Δf. For a desired quality factor, the frequency resolution can be increased or decreased by varying the data length N.

12.2 Parametric or Model-Based Methods for Power Spectrum Estimation

The classical methods require long data records to obtain the necessary frequency resolution. They suffer from spectral leakage effects, which often mask weak signals that are present in the data which occur due to windowing. For short data lengths, the spectral leakage limits frequency resolution.

In this section, we deal with power spectrum estimation methods in which extrapolation is possible if we have a priori information on how data is generated. In such a case, a model for the signal generation can be constructed with a number of parameters that can be estimated from the observed data. Then, from the estimated model parameters, we can compute the power density spectrum.

Due to modeling approach, we can eliminate the window function and the assumption that autocorrelation sequence is zero outside the window. Hence, these have better frequency resolutions and avoid problem of leakage. This is especially true in applications where short data records are available due to time variant or transient phenomena.

The parametric methods considered in this section are based on modeling the data sequence $y(n)$ as the output of a linear system characterized by a rational system function of the form

$$\boldsymbol{H}(\boldsymbol{Z}) = \frac{\boldsymbol{B}(\boldsymbol{Z})}{\boldsymbol{A}(\boldsymbol{Z})} = \frac{\sum_{k=0}^{q} \boldsymbol{b}_k \boldsymbol{z}^{-k}}{\boldsymbol{1} + \sum_{k=1}^{p} \boldsymbol{a}_k \boldsymbol{z}^{-k}}. \tag{12.20}$$

For the linear system with rational system function $\boldsymbol{H}(\boldsymbol{Z})$, the output $\boldsymbol{y}(\boldsymbol{n})$ is related to input $\boldsymbol{w}(\boldsymbol{n})$ and the corresponding difference equation is

$$\boldsymbol{y}(\boldsymbol{n}) + \sum_{k=1}^{p} \boldsymbol{a}_k \boldsymbol{y}(\boldsymbol{n} - \boldsymbol{k}) = \sum_{k=0}^{q} \boldsymbol{b}_k \boldsymbol{w}(\boldsymbol{n} - \boldsymbol{k}) \tag{12.21}$$

where $\{\boldsymbol{b}_k\}$ and $\{\boldsymbol{a}_k\}$ are the filter coefficients that determine the location of the zeros and poles of $\boldsymbol{H}(\boldsymbol{Z})$, respectively.

Parametric spectral estimation is a three-step process as follows

Step 1 Select the model
Step 2 Estimate the model parameters from the observed/measured data or the correlation sequence which is estimated from the data
Step 3 Obtain the spectral estimate with the help of the estimated model parameters.

In power spectrum estimation, the input sequence is not observable. However, if the observed data is considered as a stationary random process, then the input can also be assumed as a stationary random process.

Autoregressive Moving Average (ARMA) Model
An ARMA model of order (p, q) is described by Eq. (12.21). Let $P_w(f)$ be the power spectral density of the input sequence, $P_y(f)$ be the power spectral density of the output sequence, and $H(f)$ be the frequency response of the linear system, then

$$P_y(f) = |H(f)|^2 P_w(f) \tag{12.22}$$

where $H(f)$ is the frequency response of the model.

If the sequence $\omega(n)$ is a zero mean white noise process of variance σ_ω^2, the autocorrelation sequence is

$$r_{yy}(m) = \sigma_\omega^2 \delta(m). \tag{12.23}$$

The power spectral density of the input sequence $w(n)$ is

$$P_w(f) = \sigma_\omega^2. \tag{12.24}$$

Hence, the power spectral density of the output sequence $y(n)$ is

$$\begin{aligned} P_y(f) &= |H(f)|^2 P_w(f) \\ &= \sigma_\omega^2 |H(f)|^2 \\ &= \sigma_\omega^2 \left| \frac{B(f)}{A(f)} \right|^2. \end{aligned} \tag{12.25}$$

Autoregressive (AR) Model
If $q = 0$, $b_0 = 1$, and $b_k = 0$ for $1 \leq k \leq q$ in Eq. (12.21), then

$$H(Z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{12.26}$$

with the corresponding difference equation

$$y(n) + \sum_{k=1}^{p} a_k y(n-k) = w(n) \tag{12.27}$$

which characterizes an AR model of order p. It is represented as AR (p).

Moving Average (MA) Model
If $a_k = 0$ for $1 \leq k \leq p$ in Eq. (12.21), then

$$H(Z) = B(Z) = \sum_{k=0}^{q} b_k z^{-k} \tag{12.28}$$

with the corresponding difference equation

$$y(n) = \sum_{k=0}^{q} b_k w(n-k) \tag{12.29}$$

which characterizes a MA model of order q. It is represented as MA (q).

The AR model is the most widely used model in practice since the AR model is well suited to characterize spectrum with narrow peaks and also provides very simple linear equations for the AR model parameters. As the MA model requires more number of model parameters to represent a narrow spectrum, it is not often used for spectral estimation. The ARMA model with less number of parameters provides a more efficient representation.

12.2.1 Relationships Between the Autocorrelation and the Model Parameters

The parameters in AR(p), MA(q), and ARMA(p,q) models are related to the autocorrelation sequence $r_{yy}(m)$.

This relationship can be obtained by multiplying the difference Eq. (12.21) by $y^*(n-m)$ and taking the expected value on both sides. Then

$$\begin{aligned} E[y(n)\cdot y^*(n-m)] = &-\sum_{k=1}^{p} a_k E[y(n-k)\cdot y^*(n-m)] \\ &+\sum_{k=0}^{q} b_k E[w(n-k)\cdot y^*(n-m)] \end{aligned} \tag{12.30a}$$

$$r_{yy}(m) = -\sum_{k=1}^{p} a_k r_{yy}(m-k) + \sum_{k=0}^{q} b_k \gamma_{wy}(m-k). \tag{12.30b}$$

$\gamma_{\omega x}(m)$ is the cross-correlation between $w(n)$ and $y(n)$.

The cross-correlation $\gamma_{\omega x}(m)$ is related to the filter impulse response h as

$$\begin{aligned} \gamma_{\omega x}(m) &= E[x^*(n)w(n+m)] \\ &= E\left[\sum_{k=0}^{\infty} h(k)w^*(n-k)w(n+m)\right] \\ &= \sum_{k=0}^{\infty} h(k)E[w^*(n-k)w(n+m)] \\ &= \sigma_{\omega}^2 h(-m) \end{aligned} \tag{12.31}$$

$$E[w^*(n)w(n+m)] = \sigma_w^2 \delta(m) \tag{12.32a}$$

$$\gamma_{wx}(m) = \begin{cases} 0,\, m>0 \\ \sigma_{\omega}^2 h(-m),\, m\leq 0 \end{cases}. \tag{12.32b}$$

By setting $q = 0$ in Eq. (12.30a), an AR model can be adopted. Then, the model parameters can be related to the autocorrelation sequence as

$$r_{yy}(m) = \begin{cases} -\sum_{k=1}^{p} a_k r_{yy}(m-k), m>0 \\ -\sum_{k=1}^{p} a_k r_{yy}(m-k) + \sigma_w^2, m=0 \\ r_{yy}^*(-m), m<0 \end{cases}. \tag{12.33}$$

The above equation can be written in matrix form as

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(-1) & \dots & r_{yy}(-p+1) \\ r_{yy}(1) & r_{yy}(0) & \dots & r_{yy}(-p+2) \\ \dots & \dots & \dots & \dots \\ r_{yy}(p-1) & r_{yy}(p-2) & \dots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \cdot \\ \cdot \\ r_{yy}(p) \end{bmatrix} \tag{12.34}$$

From Eq. (12.33), we can obtain the variance

$$\sigma_w^2 = r_{yy}(0) + \sum_{k=1}^{p} a_k r_{yy}(-k) \tag{12.35}$$

Combining Eqs. (12.34) and (12.35), we get

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(-1) & \ldots & r_{yy}(-p+1) \\ r_{yy}(1) & r_{yy}(0) & \ldots & r_{yy}(-p+2) \\ \ldots & \ldots & \ldots & \ldots \\ r_{yy}(p-1) & r_{yy}(p-2) & \ldots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \vdots \\ r_{yy}(p) \end{bmatrix} \tag{12.36}$$

which is known as the Yule–Walker equation.

The correlation matrix is a Toeplitz non-singular and can be solved with Levinson–Durbin algorithm for obtaining the inverse matrix.

12.2.2 Power Spectrum Estimation Based on AR Model via Yule–Walker Method

Since the autocorrelation sequence actual values are not known a priori, their estimates are to be computed from the data sequence using

$$\widehat{r}_{yy}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} y^*(n) y(n+m) \quad m \geq 0 \tag{12.37}$$

These autocorrelation estimates and the AR model parameter estimates are used in Eq. (12.36) in place of their true values, and then the equation is solved using the Levinson–Durbin algorithm to estimate the AR model parameters. Then, the power density spectrum estimate is computed using

$$P_{\text{Yul}}(f) = \frac{\widehat{E}_p^2}{\left|1 + \sum_{k=1}^{p} \hat{a}_k \mathrm{e}^{-j2\pi f k}\right|^2} \tag{12.38}$$

where $\hat{a}_k$ are AR parameter estimates and

$$\widehat{E}_p = r_{yy}(0) \prod_{k=1}^{p} \left[1 - |\hat{a}_k|^2\right] \tag{12.39}$$

is the estimated minimum mean squared value for the pth order predictor.

The following example illustrates the power spectrum estimation based on AR model via Yule–Walker method.

Example 12.5 Consider a fourth-order AR process characterized by

$$\begin{aligned} y(n) &= 2.7607y(n-1) - 3.8106y(n-2) + 2.6535y(n-3) - 0.9238y(n-4) \\ &= w(n) \end{aligned}$$

where $w(n)$ is a zero mean, unit variance, white noise process.

Estimate the power spectrum of the AR process using the Yule–Walker method.

Solution The MATLAB function *pyulear(X,m,F_T)* gives the power spectrum of a discrete-time signal X using the Yule–Walker method. *m* being the order of the autoregressive (AR) model used to produce the PSD. F_T is the sampling frequency. A prediction filter with two zeros at $z_1 = 0.9804e^{j0.22\pi}$; $z_2 = 0.9804e^{j0.28\pi}$ gives the fourth-order AR model parameters. The two zeros are close to the unit circle; hence, the power spectrum will have two sharp peaks at the normalized frequencies 0.22π and 0.28π rad/sample.

The following MATLAB Program 11.5 is used to obtain the power spectrum using the Yule–Walker method.

%Program 12.5

```
Power spectrum estimation via Yule–Walker method
clear;clc;
F_T = 1024;
randn(‘state’,1);
w = randn(200,1);
y = filter(1,[1-2.7607 3.8106 -2.6535 0.9238],w);
pyulear(y,4,F_T);
```

The power spectrum obtained from the above program based on 200 samples is shown in Fig. 12.5.

Due to lack of sufficient resolution, the two peaks corresponding to the frequencies 0.22π and 0.28π are not seen. The resolution can be improved by increasing the data length.

When the above program is run with 1000 data samples, the power spectrum estimate obtained is shown in Fig. 12.6 in which we can see clearly the two peaks corresponding to the frequencies 0.22π and 0.28π.

12.2.3 Power Spectrum Estimation Based on AR Model via Burg Method

The Burg method [6] can be used the estimation of the AR model parameters by minimizing the forward and backward errors in the linear predictors. Here we consider the problem of linearly predicting the value of a stationary random process either forward in time (or) backward in time.

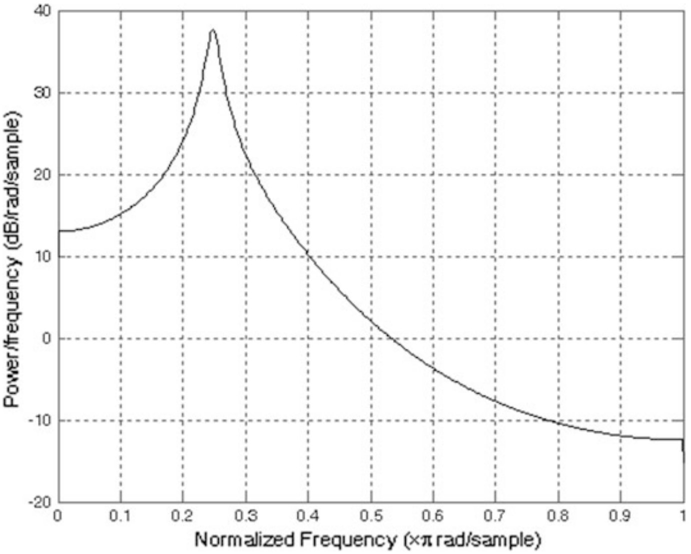


Fig. 12.5 Power spectral density estimate using Yule–Walker method based on 200 samples

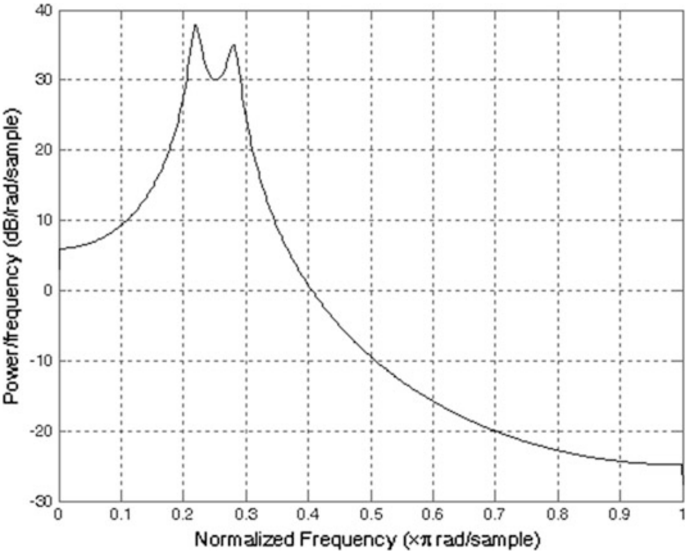


Fig. 12.6 Power spectral density estimate using Yule–Walker method based on 1000 samples

Forward Linear Prediction

Here in this case, from the past values of a random process, a future value of the process can be predicted. So, consider one-step forward linear prediction as depicted in Fig. 12.7, for which the predicted value of $y(n)$ can be written as

$$\widehat{y}(n) = -\sum_{k=1}^{p} a_p(k)y(n-k) \tag{12.40}$$

where $\left\{-a_p(k)\right\}$ are the prediction coefficients of the predictor of order p.

The forward prediction error is the difference between the value $y(n)$ and the predicted value $(\widehat{y}(n))$ of $y(n)$ and can be expressed as

$$\begin{aligned} e_p^f(n) &= y(n) - \widehat{y}(n) \\ &= y(n) + \sum_{k=1}^{p} a_p(k)y(n-k). \end{aligned} \tag{12.41}$$

Backward Linear prediction

In the backward linear prediction, the value $y(n-p)$ of a stationary random process can be predicted from the data sequence $y(n), y(n-1), \ldots, y(n-p+1)$ of the process. For one-step backward linear prediction of order p, the predicted value of $y(n-p)$ can be written as

$$\widehat{y}(n-p) = -\sum_{k=1}^{p} a_p^*(p-k)y(n+k-p). \tag{12.42}$$

The difference between $y(n-p)$ and estimate $\dddot{y}(n-p)$ is the backward prediction error which can be written as denoted as

$$e_p^b(n) = y(n-p) + \sum_{k=0}^{p-1} a_p^*(k)y(n+k-p). \tag{12.43}$$

For lattice filter realization of the predictor, a p-stage lattice filter is described by the following set of order-recursive equation

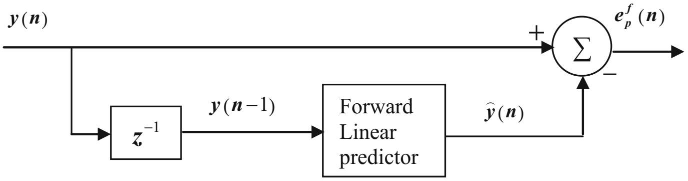


Fig. 12.7 One-step forward linear predictor

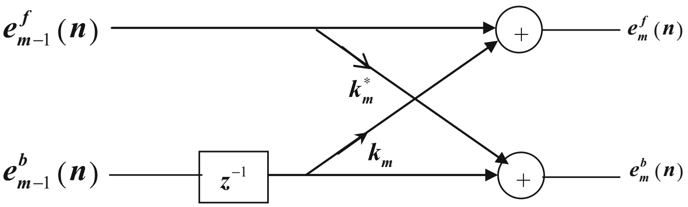


Fig. 12.8 A typical stage of a lattice filter

$$e^f_0(n) = e^b_0(n) = x(n) \tag{12.44a}$$

$$e^f_m(n) = e^f_{m-1}(n) + K_m e^b_{m-1}(n-1) \quad m = 1, 2, \ldots, p \tag{12.44b}$$

$$e^b_m = K^*_m e^f_{m-1}(n) + e^b_{m-1}(n-1) \quad m = 1, 2, \ldots, p \tag{12.44c}$$

where K_m is the mth reflection coefficient in the lattice filter.

A typical stage of a lattice filter is shown in Fig. 12.8.

From the forward and backward prediction errors, the least squares error for given data $y(n), n = 0, 1, \ldots, N-1$, can be expressed as

$$\varepsilon_m = \sum_{n=m}^{N-1} \left(\left| e^f_m(n) \right|^2 + \left| e^b_m(n) \right|^2 \right). \tag{12.45}$$

Now, the error is to be minimized with respect to predictor coefficients satisfying the following Levinson–Durbin recursion

$$a_m(k) = a_{m-1}(k) + K_m a^*_{m-1}(m-k), \quad 1 \le k \le m-1; 1 \le m \le p. \tag{12.46}$$

where $K_m = a_m(m)$ is the mth reflection coefficient in the lattice filter of the predictor.

Minimization of ε_m with respect to the reflection coefficient K_m yields

$$\widehat{K}_m = \frac{-\sum_{n=m}^{N-1} e^f_{m-1}(n) \left(e^b_m(n) \right)^*}{\frac{1}{2} \widehat{E}_m} \tag{12.47}$$

where $\widehat{E}_m$ is the total squared error which is an estimate of $\widehat{E}^f_{m-1} + \widehat{E}^b_{m-1}$, $\widehat{E}^f_{m-1}$ and $\widehat{E}^b_{m-1}$ being the least squares estimates of the forward and backward errors given by

$$\widehat{E}_{m-1}^{f} = \sum_{n=m}^{N-1} \left| e_{m-1}^{f}(n) \right|^2 \qquad (12.48)$$

$$\widehat{E}_{m-1}^{b} = \sum_{n=m}^{N-1} \left| e_{m-1}^{b}(n) \right|^2. \qquad (12.49)$$

The estimate $\widehat{E}_m$ can be computed by using the following recursion

$$\widehat{E}_m = \left(1 - \left|\widehat{K}_m\right|^2\right)\widehat{E}_{m-1}\left|e_m^f(m-2)\right|^2 - \left|e_m^b(m-2)\right|^2 \qquad (12.50)$$

The Burg method computes the reflection coefficients $\widehat{K}_m$ using Eqs. (12.47) and (12.50), and AR parameters are estimated by using Levinson–Durbin algorithm. Then, the power spectrum can be estimated as

$$P_{\mathrm{BUR}}(f) = \frac{\widehat{E}_m^2}{\left|1 + \sum_{k=1}^{p} \widehat{a}_k \mathrm{e}^{-j2\pi fk}\right|^2}. \qquad (12.51)$$

The following example illustrates the power spectrum estimation using the Burg method.

Example 12.6 Consider the AR process as given in Example 12.5. Evaluate its power spectrum based on 200 samples using the Burg method.

Solution The MATLAB Program 12.5 can be used by replacing pyulear(y,4,Fs) by pburg(y,4,Fs) to compute power spectrum using the Burg method. Thus, the PSD obtained based on 200 samples using the Burg method is shown in Fig. 12.9.

The two peaks corresponding to the frequencies 0.22π and 0.28π are clearly seen from Fig. 12.9. Using the Burg method based on 200 samples, whereas it is not as shown in Fig. 12.5 using Yule–Walker method for the same number of samples.

The main advantages of the Burg method are high-frequency resolution, stable AR model, and computational efficiency. The drawbacks of the Burg method are spectral line splitting at high SNRs and spurious spikes for high-order models.

12.2.4 Selection of Model Order

Generally, model order is unknown a priori. If the guess for the model order is too low, it will result in highly smoothed spectral estimate and the high-order model increases resolution but low-level spurious peaks will appear in the spectrum. The two methods suggested by Akaike [7, 8] for model order selection are:

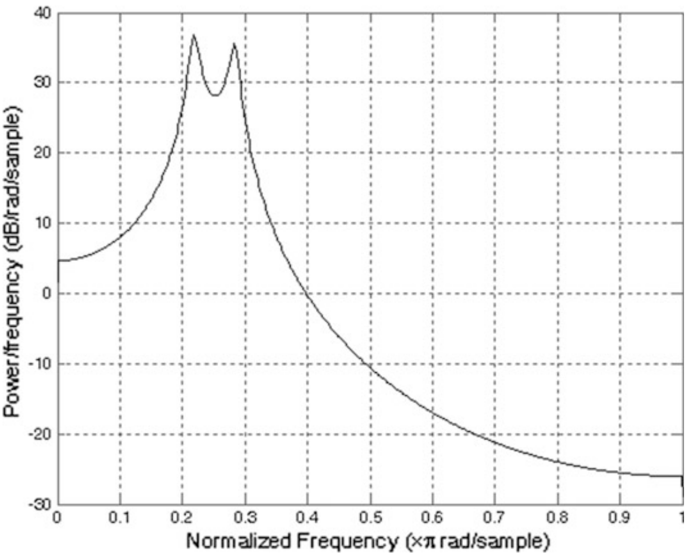


Fig. 12.9 Power spectral density estimate via Burg method based on 200 samples

1. Akaike forward prediction error (FPE) criterion states that

$$\mathbf{FPE}(p) = \sigma_{w_p}^2 \left(\frac{N+p+1}{N-p-1} \right) \tag{12.52}$$

should be minimum. Here, N stands for the number of data samples, p is the order of the filter, and $\sigma_{w_p}^2$ is the white noise variance estimate.

2. Akaike forward prediction error (FPE) criterion states that

$$\mathbf{FPE}(p) = N\ \ln(\sigma_{w_p}^2) + 2p \tag{12.53}$$

should be minimum.

12.2.5 Power Spectrum Estimation Based on MA Model

By setting $p = 0$ in Eq. (11.30a) and letting $h(k) = b(k), 1 \le k \le q$, a MA model can be adopted. Then, the model parameters can be related to the autocorrelation sequence as

$$r_{yy}(m) = \begin{cases} \sigma_\omega^2 \sum_{k=0}^{q} b_k b_{k+m}; & 0 \le m \le q \\ 0; & m > q \\ r_{yy}^*(-m); & m < 0 \end{cases}. \tag{12.54}$$

Then, the power spectrum estimate based on MA model is

$$P_{\mathbf{MA}}(f) = \sum_{m=-q}^{q} r_{yy}(m)\, \mathrm{e}^{-j2\pi fm}. \tag{12.55}$$

Equation (12.55) can be written in terms of MA model parameter estimates $\left(\widehat{b}_k\right)$, and the white noise variance estimate $(\widehat{\sigma}_w^2)$ can be written as

$$P_{MA}(f) = \widehat{\sigma}_w^2 \left| 1 + \sum_{k=1}^{q} \widehat{b}_k\, \mathrm{e}^{-j2\pi fk} \right|^2. \tag{12.56}$$

12.2.6 Power Spectrum Estimation Based on ARMA Model

ARMA model is used to estimate the spectrum with less parameters. This model is mostly used when data is corrupted by noise.

The AR parameters are estimated first, independent of the MA parameters, by using the Yule–Walker method or the Burg method. The MA parameters are estimated assuming that the AR parameters are known.

Then, the ARMA power spectral estimate is

$$P_{\mathrm{ARMA}}(f) = \widehat{\sigma}_w^2 \left| \frac{1 + \sum_{k=1}^{q} \widehat{b}_k \mathrm{e}^{-j2\pi fk}}{1 + \sum_{k=1}^{p} \widehat{a}_k \mathrm{e}^{-j2\pi fk}} \right|^2. \tag{12.57}$$

12.3 Subspace Methods for Power Spectrum Estimation

The subspace methods do not assume any parametric model for power spectrum estimation. They are based solely on the estimate of the autocorrelation sequence of the random process from the observed data. In this section, we briefly discuss three subspace methods, namely, Pisarenko harmonic decomposition, MUSIC, and eigenvector method.

12.3.1 Pisarenko Harmonic Decomposition Method

Consider a process y consisting of m sinusoids with additive white noise. The autocorrelation values of the process y can be written in matrix form as

$$\begin{bmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \vdots \\ r_{yy}(m) \end{bmatrix} = \begin{bmatrix} e^{j2\pi f_1} & e^{j2\pi f_2} & e^{j2\pi f_3} & \ldots & e^{j2\pi f_m} \\ e^{j4\pi f_1} & e^{j4\pi f_2} & e^{j4\pi f_3} & \ldots & e^{j4\pi f_m} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ e^{j2\pi m f_1} & e^{j2\pi m f_2} & e^{j2\pi m f_3} & \ldots & e^{j2\pi m f_m} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_m \end{bmatrix} \tag{12.58}$$

where P_i is the average power in the ith sinusoid.

If the frequencies f_i, $1 \leq i \leq m$, are known, then from the known autocorrelation values $r_{yy}(1)$ to $r_{yy}(m)$, the sinusoidal powers can be determined from the above equation.

The stepwise procedure for the Pisarenko harmonic decomposition method is as follows.

Step 1: Estimate the autocorrelation vector from the observed data.
Step 2: Find the minimum eigenvalue and the corresponding eigenvector (v_{m+1})
Step 3 : Find the roots of the following polynomial

$$\sum_{k=0}^{M} v_{m+1}(k+1)z^{-k} \tag{12.59}$$

where v_{m+1} is the eigenvector. The roots lie on the unit circle at angles $2\pi f_i$ for $1 \leq i \leq M$,M is dimension of eigenvector,
Step 4 : Solve Eq. (12.58.) for sinusoidal powers (P_i).

12.3.2 Multiple Signal Classification (MUSIC) Method

The MUSIC estimates the power spectrum from a signal or a correlation matrix using Schmidt's eigen space analysis method [9]. The method estimates the signal's frequency content by performing eigen space analysis of the signal's correlation matrix. In particular, this method is applicable to signals that are the sum of sinusoids with additive white Gaussian noise and more, in general, to narrowband signals. To develop this, first let us consider the 'weighted' spectral estimate

$$P(f) = S^H(f)\left(\sum_{k=m+1}^{M} c_k V_k V_k^H\right)S(f) = \sum_{k=m+1}^{M} c_k\left|S^H(f)V_k\right|^2 \tag{12.60}$$

where m is the dimension of the signal subspace, V_k, $k = m+1,\ldots,M$ are the eigen vectors in the noise subspace, c_k are a set of positive weights, and

$$S(f) = \left[1, \mathrm{e}^{j2\pi f}, \mathrm{e}^{j4\pi f}, \ldots, \mathrm{e}^{j2\pi(m-1)f}\right] \text{ is complex sinusoidal vector.}$$

It may be noted that at $f = f_i$, $S(f_i) = S_i$, such that at any one of the p sinusoidal frequency components of the signal we have,

$$P(f_i) = 0 \quad i = 1, 2, \ldots, m. \tag{12.61}$$

This indicates that

$$\frac{1}{P(f)} = \frac{1}{\sum_{k=m+1}^{M} c_k\left|S^H(f)V_k\right|^2} \tag{12.62}$$

is infinite at $f = f_i$. But, in practice due to the estimation errors, $\frac{1}{P(f)}$ is finite with very sharp peaks at all sinusoidal frequencies providing a way for estimating the sinusoidal frequencies.

Choosing $c_k = 1$ for all k, the MUSIC frequency estimator [10] is written as

$$P_{\mathbf{MUSIC}}(f) = \frac{1}{\sum_{k=m+1}^{M}\left|S^H(f)V_k\right|^2} \tag{12.63}$$

The peaks of $P_{\mathbf{MUSIC}}(f)$ are the estimates of the sinusoidal frequencies, and the powers of the sinusoids can be estimated by solving Eq. (12.58). The following example illustrates the estimation of power spectrum using the MUSIC method.

Example 12.7 Consider a random signal generated by the following equation

$$x(n) = \sin\left(\frac{2\pi f_1 n}{F_\mathrm{s}}\right) + 2\sin\left(\frac{2\pi f_2 n}{F_\mathrm{s}}\right) + 0.1w(n)$$

where the frequencies f_1 and f_2 are 220 and 332 Hz, respectively, the sampling frequency F_s is 2048 Hz and $w(n)$ is a zero mean, unit variance, white noise process. Estimate power spectrum of the sequence $\{x(n), 0 \le n \le 1023\}$.

Solution The MATLAB function *pmusic(X,m,'whole')* gives the power spectrum of a discrete-time signal X using the MUSIC method, m being the number of complex sinusoids in the signal X. If X is an autocorrelation data matrix of discrete-time signal x, the function *corrmtx* can be used to generate data matrices. The signal vector *x* consists of two real sinusoidal components. In this case, the dimension of the signal subspace is 4 because each real sinusoid is the sum of two complex exponentials.

The following MATLAB Program 12.6 is used to obtain the power spectrum using the MUSIC method.

Program 12.6

Power spectrum estimation using the MUSIC method

```
clear;clc;
randn('state',0);
N = 1024;%total number of samples
k = 0 : N-1;
f1 = 280;
f2 = 332;
FT = 2048;%sampling frequency in Hz
T = 1/FT;
x = sin(2*pi*f1*k*T) + 2*sin(2*pi*f2*k*T)+0.1*randn(size(k));%input vector of length N
X = corrmtx(x,12);%estimates (N+12) by (12+1) rectangular autocorrelation matrix
pmusic(X,4,'whole'); %estimates power spectrum of x containing two sinusoids
```

The power spectrum obtained from the above program is shown in Fig. 12.10.

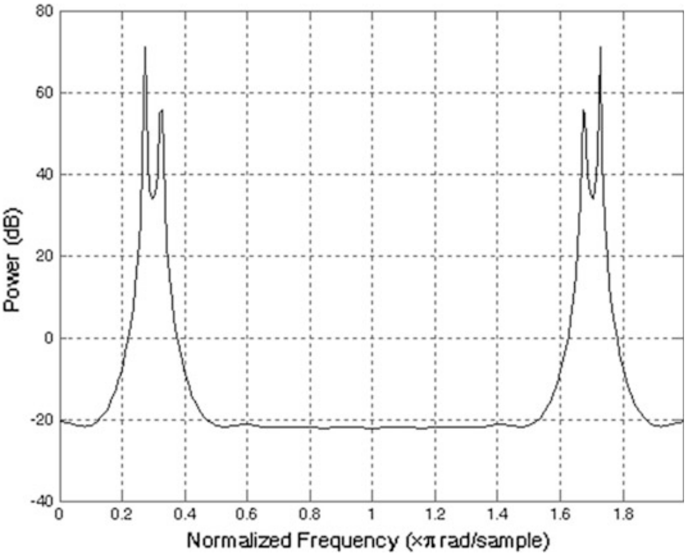


Fig. 12.10 Power spectral density estimate using MUSIC method

12.3.3 Eigenvector Method

The eigenvector method is a weighted version of the MUSIC method. Selecting $c_k = \frac{1}{\lambda_k}$ in Eq. (12.62) for all k, the eigenvector method produces a frequency estimator given by Johnson [11].

$$P_{\text{eig}}(f) = \frac{1}{\left(\sum_{k=m+1}^{M} \left(\frac{1}{\lambda_k}\right) \left|V_k^H S(f)\right|^2\right)} \tag{12.64}$$

where M is the dimension of the eigenvectors and V_k is the kth eigenvector of the autocorrelation matrix of the observed data sequence. The integer m is the dimension of the signal subspace, so the eigenvectors V_k used in the sum correspond to the smallest eigenvalues λ_k of the autocorrelation matrix. The eigenvectors used in the PSD estimate span the noise subspace. The power spectrum estimation using the eigenvector method is illustrated through the following example.

Example 12.8 Consider the random signal generated in the Example 12.7 and estimate its power spectrum using the eigenvector method.

Solution The MATLAB function *peig(X,m,'whole')* estimates the power spectrum of a discrete-time signal X, m being the number of complex sinusoids in the signal X. If X is an autocorrelation data matrix of discrete-time signal x, the function *corrmtx* can be used to generate data matrices. The signal vector *x* consists two real sinusoidal components. In this case, the dimension of the signal subspace is 4 because each real sinusoid is the sum of two complex exponentials.

The following MATLAB Program 12.7 is used to obtain the power spectrum using the eigenvector method.

Program 12.7

Power spectrum estimation using the eigenvector method

```
clear;clc;
randn('state',0);
N = 1024;%total number of samples
k = 0 : N-1;
f1 = 280;
f2 = 332;
FT = 2048;%sampling frequency in Hz
T = 1/FT;
x = sin(2*pi*f1*k*T)+2*sin(2*pi*f2*k*T)+0.1*randn(size(k));%input vector of length N
X = corrmtx(x,12);%estimates (N+12) by (12+1) rectangular autocorrelation matrix
peig(X,4,'whole');%estimates power spectrum of x containing two sinusoids
```

The power spectrum produced from the above program is shown in Fig. 12.11.

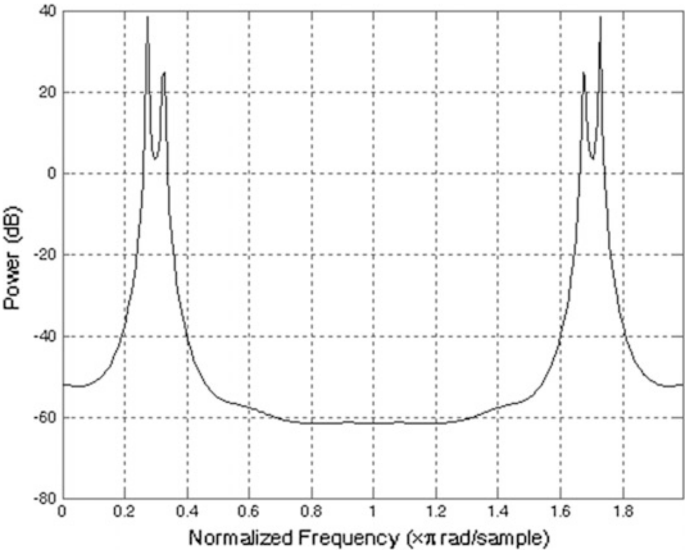


Fig. 12.11 Power spectral density estimate using eigenvector method

12.4 Spectral Analysis of Non-stationary Signals

A signal with time-varying parameters, for example, a speech signal, is called a non-stationary signal. The spectrogram which shows how the spectral density of a signal varies with time is a basic tool for spectral analysis of non-stationary signals. Spectrograms are usually generated using the short-time Fourier transform (STFT) using digitally sampled data. To compute the STFT, a sliding window which usually is allowed to overlap in time is used to divide the signal into several blocks of data. Then, an N-point FFT is applied to each block of data to obtain the frequency contents of each block.

The window length affects the time resolution and the frequency resolution of the STFT. A narrow window results in a fine time resolution but a coarse frequency resolution, whereas a wide window results in a fine frequency resolution but a coarse time resolution. A narrow window is to be used to provide wideband spectrogram for signals having widely varying spectral parameters. A wide window is preferred to have narrowband spectrogram. The following example illustrates the computation of the spectrogram of a speech signal.

Example 12.9 Consider a speech signal 'To take good care of yourself' from the sound file 'goodcare.wav' (available in the CD accompanying the book). Compute the spectrogram of the speech signal using Hamming window of lengths 256 samples and 512 samples with an overlap of 50 samples.

Solution The STFT of a non-stationary signal x can be computed by using the MATLAB file *specgram(x,wl,Fs, window,noverlap)*

where wl stands for window length, and noverlap is the number of overlap samples.

Program12.8

Spectrogram of a speech signal

```
[x,F_T] = wavread('goodcare.wav');
i = 1:length(x)
figure(1),plot(x)
xlabel('Time index i');ylabel('Amplitude');
figure(2), specgram(x,256, F_T,hamming(256),50)
```

The speech signal 'To take god care of yourself' is shown in Fig. 12.12.

The spectrograms of the speech signal for window lengths of 256 and 512 samples are shown in Fig. 12.13a, b respectively.

From the above spectrograms, the trade-off between frequency resolution and time resolution is evident.

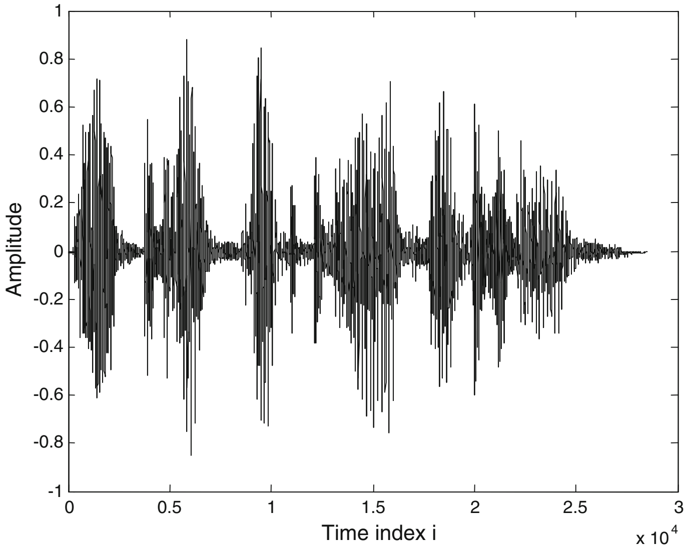


Fig. 12.12 A speech signal

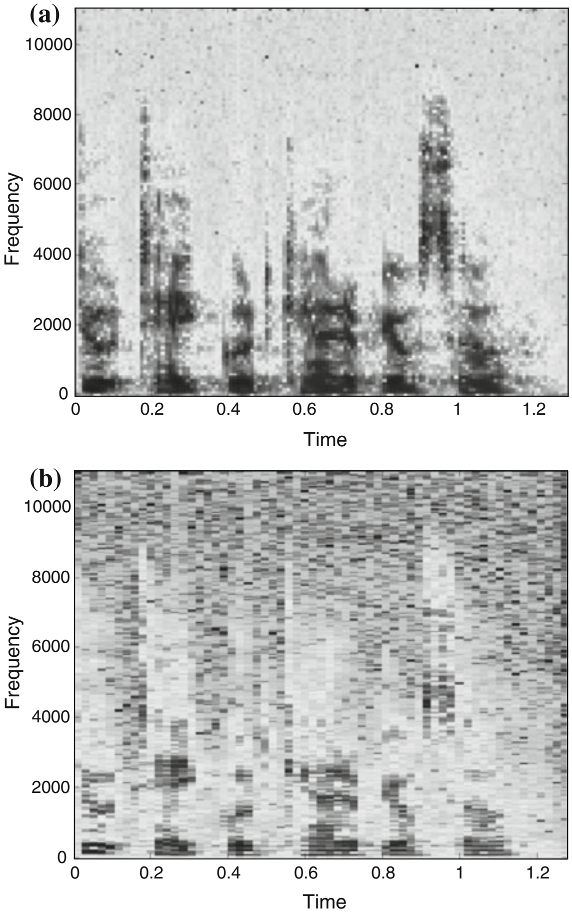


Fig. 12.13 **a** Spectrogram with window length 256, overlap = 50 and **b** spectrogram with window length 512, overlap = 50

12.4.1 MATLAB Exercises

1. Consider a random signal of length 1024 composed of two sinusoidal components of frequencies 180 and 320 Hz with sampling frequency of F_T = 2048 Hz corrupted with zero mean, unit variance, white noise process. Evaluate its power spectrum using Bartlett's method with subsequence lengths of each 256 samples.
2. Consider the following signal of length N = 1024 with sampling frequency of F_s = 2048 Hz corrupted with zero mean, unit variance, white noise process.

$$x(i) = \sin\left(\frac{800\pi i}{F_T}\right)\cos\left(\frac{800\pi i}{F_T}\right) + w(i), \quad 0 \le i < N$$

where $w(i)$ is zero mean Gaussian white noise.

Evaluate its power spectrum using Welch's method with subsequence lengths of each 256 samples using Blackman window for overlaps of 64 and 128 samples, respectively.

3. Consider the following signal of length $N = 1024$ with sampling frequency of $F_s = 2048$ Hz corrupted with zero mean, unit variance, white noise process.

$$x(i) = \sin^2\left(\frac{400\pi i}{F_T}\right)\cos^2\left(\frac{200\pi i}{F_T}\right) + w(i), \quad 0 \le i < N$$

where $w(i)$ is zero mean unit variance, white noise process.

Evaluate its power spectrum using Blackman–Tukey method with window length of 256 samples.

4. Consider a fourth-order AR process characterized by

$$y(n) + a_1 y(n-1) + a_2 y(n-2) + a_3 y(n-3) + a_4 y(n-4) = w(n)$$

where $w(n)$ is a zero mean, unit variance, white noise process. The parameters $\{a_1, a_2, a_3, a_4\}$ are chosen such that the prediction error filter

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}$$

has zeros at the locations.

$$0.98e^{\pm j0.15\pi} \text{ and } 0.98e^{\pm j0.35\pi}$$

Estimate the power spectrum of the AR process using the Yule–Walker method and Burg method based on 200 samples. Comment on the results.

5. Consider the following signal of length $N = 1024$ with sampling frequency of $F_s = 2048$ Hz corrupted with zero mean, unit variance, white noise process.

$$x(n) = \sum_{i=1}^{3} e^{j2\pi f_i n} + w(n), \quad 0 \le n < N$$

where $w(i)$ is zero mean unit variance, white noise process, and the frequencies are Hz $f_1 = 256\,\text{Hz}, f_2 = 338\,\text{Hz}$, and $f_3 = 338\,\text{Hz}$.

Evaluate its power spectrum using the MUSIC method and the eigenvector method. Comment on the results.

6. Consider a speech signal from the sound file 'speech.wav' (available in the CD accompanying the book) and compute its spectrogram for different window lengths with and without overlap. Comment on the results.

References

1. A. Schuster, On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. Terr. Magn. Atmos. Electr. **3**, 13–41 (1898)
2. M.S. Bartlett, Smoothing *periodograms* from time series with continuous spectra. Nature **161**, 686–687 (1948)
3. P.D. Welch, The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. IEEE Trans. Audio Electroacoust. **15**, 70–83 (1967)
4. R.B. Blackman, J.W. Tukey, *The Measurement of Power Spectra from the Point of View of Communication Engineering* (Dover Publications, USA, 1958)
5. J.G. Proakis, D.G. Manolakis, *Digital Signal Processing Principles, Algorithms and Applications* (Printice-Hall, India, 2004)
6. J.P. Burg, A new Analysis technique for time series data. Paper presented at the NATO Advanced Study Institute on Signal Processing, Enschede, 1968
7. H. Akaike, Power spectrum estimation through autoregressive model fitting. Ann. Inst. Stat. Math. **21**, 407–420 (1969)
8. H. Akaike, A new look at the statistical model identification. IEEE Trans. Autom. Control **19** (6), 716–723 (1974)
9. S.L. Marple, in *Digital Spectral Analysis*. (Prentice-Hall, Englewood Cliffs, NJ, 1987), pp. 373–378, 686–687
10. R.O. Schmidt, Multiple emitter location and signal parameter estimation. IEEE Trans. Antennas Propag. **AP-34**, 276–280 (1986)
11. D.H. Johnson, The application of spectral estimation methods to bearing estimation problems. Proc. IEEE **70**(9), 126–139 (1982)