Spectral Estimation and Its Application in Electromyography

A Thesis Presented to
The Faculty of the College of Engineering and Technology
Ohio University

In Partial Fulfillment
of the Requirement for the Degree
Master of Science

by
Hussein A. Dia
November 16, 1984
AKNOWLEDGEMENT

I owe special thanks to Dr. K. O. Tanaka for his patience and guidance which helped me finish this thesis.
# TABLE OF CONTENTS

## CHAPTER 1 INTRODUCTION
- 1

## CHAPTER 2 RANDOM PROCESSES AND SPECTRAL ESTIMATION
- 7
  - 2.1 INTRODUCTION
  - 2.2 DEFINITIONS AND REPRESENTATION
  - 2.3 STATISTICAL TIME SERIES ANALYSIS. SPECTRAL ESTIMATION
  - 2.4 HISTORY OF SPECTRAL ESTIMATION
  - 2.5 BASIC PRINCIPLES OF ESTIMATION THEORY. MODELING
  - 2.6 MOVING AVERAGE TIME SERIES SPECTRAL ESTIMATION

## CHAPTER 3 THE FFT AND ITS APPLICATION IN PSD ESTIMATION
- 26
  - 3.1 PREVIEW
  - 3.2 SAMPLING
  - 3.3 DETERMINATION OF THE DFT AND THE IDFT
  - 3.4 THE FFT ALGORITHM
  - 3.5 SPECTRAL ANALYSIS WITH FFT. PERIODOGRAM
  - 3.6 SPECTRUM SMOOTHING. AVERAGED PERIODOGRAM

## CHAPTER 4 AR AND MAXIMUM ENTROPY SPECTRAL ESTIMATION
- 47
  - 4.1 INTRODUCTION
  - 4.2 AR PARAMETER IDENTIFICATION. LINEAR PREDICTION
  - 4.3 LEVINSON-DURBIN'S ALGORITHM. PSD COMPUTATION
  - 4.4 MODEL ORDER SELECTION
  - 4.5 MESE
  - 4.6 CORRESPONDENCE BETWEEN AR AND ME SPECTRAL ESTIMATES

## CHAPTER 5 ARMA SPECTRAL ESTIMATION
- 68
  - 5.1 INTRODUCTION
  - 5.2 ARMA PARAMETER ESTIMATION BY THE METHOD OF ADOPTION A PURE AR MODEL
    - A. ALGORITHM 1
    - B. ALGORITHM 2
CHAPTER 1

INTRODUCTION

In the past two decades attention has been diverting towards digital processing techniques where real-time implementation algorithms are now prevalent in a wide range area of applications, such as: radar, sonar, seismology, speech communications, biomedical engineering, nuclear science data communication, image processing, and many others. As the field of digital signal processing advanced, new theoretical techniques, applications, and hardware implementation were being developed to match the era of modern and sophisticated technology in our everyday life. Nevertheless the procedures used in this field are as old as Newton and Gauss and as new as digital computers and integrated circuits.

There are two major categories which underline the field of signal processing, namely: filtering and estimation. The purpose of filtering is to modify the spectrum of a received signal in some desired manner. On the other hand, estimation is required to estimate
the parameters of a signal masked by noise, an autocorrelation function or a spectral density. The first major contributions to the field of digital signal processing are to be directed to Kaiser (at Bell Laboratories) in the area of digital filter design and synthesis. With the increasing flexibility of digital equipment, attention has been drawn toward digital filtering. This is because of the fact that systems which are used to spectrally shape information are known as filters. Digital systems operate in discrete-time steps. Therefore, when an analog device is replaced by a digital system, the continuous-time input signal has to be converted into a sequence of numbers. This is usually done by sampling the input signal at uniformly spaced time instants. The sequence of samples, then, is called a discrete-time signal. The evolution of a new point of view toward exploring digital processing was further accelerated by the disclosure of an efficient algorithm for computing the Fourier transform, known as the Fast Fourier Transform (FFT). This technique made a frequency domain analysis competitive with the time domain difference equation approach. The parallel between linear time-invariant filtering in the continuous-time and discrete-time cases is readily observed. The theory of Z-transform for the analysis of sampled-data systems, follows closely the Fourier transform theory in the linear time-invariant case. Characteristically, one uses the transformation as a mathematical or physical tool to alter
the problem into one that can be solved.

Kalman's paper (1960) which presented a recursive algorithm (now known as Kalman filter) for estimating the state of a linear dynamical system, was a threshold toward the development of estimation techniques. At a later period, digital signal processing approach became widely applied to the estimation of power spectral densities which is closely related to the estimation of the autocorrelation functions. The contemporary popularity of spectral analysis can be attributed directly to the pioneering works of N. Wiener and A. I. Khinchin. They, independently, developed the fundamental notion of characterizing the second-order statistics of a wide-sense stationary random process by means of the autocorrelation function or power spectrum. In fact these two functions represent a Fourier transform pair.

Spectral analysis brings an important theoretical approach beside the Fourier transform, the statistical analysis of time series. By a statistical time series is meant a signal or a function of time which exhibits random or fluctuating properties. The characteristic feature of a time series is that its future behavior is unpredictable. Statistical properties of the series are described by associating probability distribution functions with any set of time series. A weakly
A stationary stochastic process may be adequately described by its probability distribution such as mean, variance, covariance or autocorrelation and its Fourier transform, the power spectral density. In characterizing these processes the spectral density is often preferred to the autocorrelation function, because a spectral representation may reveal such useful information as hidden periodicities or close spectral peaks.

In many problems such as those where it is required to predict future values of the time series, it is necessary to construct a parametric model for the series. On the other hand, processing of a time series is often accomplished by means of a linear filter. Some powerful parametric models have been used in practice for describing empirical time series such as, the autoregressive model, the moving average model, or the mixed autoregressive-moving average model (ARMA). Although spectral analysis has an important role to play in time series model building, it is in the area of frequency response studies that is most relevant.

Conventional or classical methods of power spectral estimation (Blackmann & Tukey (1958), Bartlett & Welch, later, etc.) were based on autocovariance or autocorrelation functions of the time series. Usually the autocorrelation functions were obtained only from a finite set of data points and they were assumed null beyond the largest lag
(windowing). As a consequence, the power spectral density would not give sufficient information about the signal being studied. The problem of windowing was avoided, lately, when new methods of spectral analysis were introduced. Such methods, which do not require the assumption of periodic extension or that the data outside the available record length is zero, are the maximum entropy method or MEM and the maximum likelihood method or MLM. On the other hand these two methods express a non-parametric nonlinear behavior because they are data dependent. Other modern techniques are also used to build parametric models for power spectral density estimation. This category of models is based upon linear prediction error filter theory and the Levinson-Durbin algorithm for computing the estimated filter coefficients. The most useful models are represented by AR, MA, and ARMA processes.

**ORGANIZATION OF THE THESIS**

In chapter two, random processes and time series analysis along with the principles of spectral estimation theory will be discussed. Chapter three reviews the discrete Fourier transform (DFT) with a detailed discussion of the FFT algorithm and the periodogram power spectral density computation. The AR parametric model and the non-parametric MESE will be discussed in chapter four while the ARMA spectral analysis will be the contents of chapter five. The electromyography and the EMG spectral analysis as a practical application to
the digital signal processing techniques will be presented in chapter six. Finally, numerical computation of the PSD of two sinusoids with white noise, will be analyzed in chapter seven followed by conclusions in chapter eight.
CHAPTER 2

RANDOM PROCESSES

AND

SPECTRAL ESTIMATION

2.1 INTRODUCTION

In the areas of estimation, communication, and control the sources of interference in signal processing are numerous and diverse both in origin and magnitude. In the construction of mathematical models for most systems, these interferences are standardly portrayed as random processes. One of the major statistical tools borrowed by engineers in the modeling of problems in these areas was the adoption of the mean-square error as a criterion of performance. Originally, in dealing with discrete-time systems, the use of this criterion amounted to a direct application of the analysis of variance.

The first models in this area arose from N. Wiener's (1949) [44] ideas on time-series analysis in both the time and frequency domains. At the same time, in order for these rather rudimentary processing schemes to be valid, a number of restrictive assumptions were made on the statistical nature of the signal and noise waveforms, such as
stationarity, independence, and "whiteness". Furthermore, the requirement of linear processing allows considerable flexibility in the distribution properties of the permissible random processes. It imposes restrictions only on the second-order moments of the process. In addition, it provides powerful geometric tools for the analysis of the appropriate models, such as the Hilbert-space environment [6,7]. The establishment of the mean-square error as a performance criterion necessarily restricts the class of permissible random processes \( x(t) \) to include those processes for which this error can be defined, namely the second-order processes (i.e., \( E[x(t)] \), ..., for every \( t \) in the interval of interest). Refer to [7], [11], [54] for more information.

2.2 DEFINITIONS AND REPRESENTATION

There are many important examples of signals which do not have finite energy or are not periodic. Many of the effects encountered in implementing digital signal processing algorithms with finite register length can be represented by additive "noise", which can be conveniently treated as a sequence with finite energy. The key to the mathematical representation of such signals lies in their description in terms of averages; many of the properties of these signals can be summarized in terms of a finite-energy sequence called the autocorrelation or autocovariance, for which the Z- or the Fourier transform often exists. Fourier transform of the autocovariance sequence has a useful interpre-
tation in terms of frequency distribution of the power in the signal.

Random or stochastic signal is considered to be a member of discrete-time signals which is characterized by a set of probability density functions. Formally, a random process is a family of random variables \( \{x_n\} \) or \( \{x(n)\} \); the index is associated with time. This indexing property of random processes makes it possible to be characterized by statistical averages, ensemble averages.

The average or the mean of the process which determines its center of gravity is defined by the following relationship:

\[
m_x = E[x] = \int_{-\infty}^{\infty} p_x(x) \, dx = \mu
\]

(2.1)

where, \( E[.\] \) denotes the expectation operator, and \( p_x(x) \) is the probability density function. If the random variables are quantized, the integral becomes summation over all possible values of the random variable

\[
E[x] = \sum_x x \, p_x(x)
\]

(2.2)

The mean-square value of \( x_n \) is the average of \( x_n^2 \), i.e.

\[
E[x^2] = \sum_x x^2 \, p_x(x)
\]

(2.3)

The variance which is defined as the estimate of the concentration of \( p_x(x) \) near its center of gravity is determined as the mean-square of \( (x_n - m_{x_n}) \):
The mean, mean-square, and variance are simple averages that provide only small amount of information about the process. A more useful average is the **autocorrelation** sequence defined as

\[ \phi_{xx}(n,m) = E[x_n x_m] \]  \hspace{1cm} (2.5)

where the **autocovariance** sequence is defined as:

\[ T_{xx}(n,m) = E[(x_n - m_n)(x_m - m_m)] \]  \hspace{1cm} (2.6)

from which we conclude that for a zero mean process the autocovariance and the autocorrelation sequences are equal. The autocorrelation is a measure of the dependence between values of the random process at different times. In this sense it describes the time variation of a random signal. A measure of the dependence between two different random signals is obtained from the crosscorrelation sequence.

**Stationarity** of a random process characterizes the equilibrium condition in which the statistical properties are invariant to a shift of time origin. Thus for a stationary process, first-order averages, such as mean and variance are independent of time, whereas the second-order averages, such as the autocorrelation sequence, are dependent on the time difference \(|m - n|\), i.e. :

\[ \phi_{xx}(n,m) = \phi_{xx}(m - n), \]

or

\[ \phi_{xx}(n,n+r) = \phi_{xx}(r) \]  \hspace{1cm} (2.7)
In many instances we encounter random processes that are not stationary in the strict sense, i.e. their probability distributions are not time-invariant, yet the mean is constant and the autocorrelation function is given by (2.7). Such random processes are said to be stationary in the wide-sense. The ordered set of random variables and its associated probability distribution functions is called a stochastic process. For further information refer to [6], [23], and[54].

2.3 STATISTICAL TIME-SERIES ANALYSIS AND SPECTRAL ESTIMATION

A statistical time-series is a random or non-deterministic function, \( x \), of an independent variable, \( t \), or in the discrete domain is defined as a series of observations \( x(n) \), where \( n=1,2,\ldots,N \) is a time point. Observations are made sequentially through time. Statistical time-series can also be defined as a time-series which exhibits random or fluctuating properties. Given a record of such a series, it is not possible to predict future values exactly. It follows that the time-series can only be described by statistical laws or models.

In order to make the analysis of observed time-series tractable, simplifications have been made like stationarity in wide-sense and the characterization of the process by the lower moments of its probability density function (weak stationarity), such as mean, variance, and autocorrelation. Alternatively the stochastic time-series can be adequately
described by means of a model containing a few parameters which may be estimated from the data [11], [13], [22].

### 2.4 HISTORY OF SPECTRAL ESTIMATION THEORY

In 1929 John Von Neumann put the spectral theory of the atom on a firm mathematical foundation in his spectral representation theorem in Hilbert Space [7]. Meanwhile, Wiener [33] developed the mathematical theory of Brownian movement in 1923, and in 1930 he introduced the generalized harmonic analysis, that is, the spectral representation of a stationary random process. In 1942 Wiener applied his methods to problems of prediction and filtering. His work was interpreted, later, and extended by Norman Levinson [34]. The modern history of spectral estimation has begun with the breakthrough of J. W. Tukey in 1949 [35], which is the statistical counterpart of the breakthrough of Fourier 142 years earlier. A major computational breakthrough occurred with the publication, in 1965, of the fast Fourier transform algorithm by J. S. Cooley & J. W. Tukey[21]. The Cooley-Tukey method made it practical to do signal processing on waveforms in either time or frequency domain, something never practical with continuous systems. With the development of the FFT, the field of spectral analysis grew from obscurity to importance and is now a major discipline.

Further important contributions were the introduction of maximum
entropy spectral analysis (MESE) by J. Burg [36] in 1967, the development of spectral windows by Emmanuel Parzen and others starting in 1950's, the statistical work of Maurice Priestley and his school, hypothesis testing in time series analysis by Peter Whittle [37] starting in 1951, the Box-Jenkins approach by George Box & G. M. Jenkins [29] in 1970, the autoregressive spectral estimation and order determining by E. Parzen and H. Akaike in 1960's [38]. Burg's method of maximum entropy spectral analysis fulfils the requirement which retains all of the estimated lags without modification, and uses a non-zero estimate for the lags not directly estimated. The particular estimation principle used is that the spectral estimate must be the most random or has the maximum entropy of any power spectrum which is consistent with the measured data. This new analysis technique gives a much better resolution for the spectral estimate than is observed by conventional methods with little increase in computing time.

At the turn of the century, Sir Arthur Schuster [30] introduced a numerical method of spectral analysis for empirical time-series. Schuster's method consisted of computing the periodogram \( P(\omega) \)

\[
P(\omega) = \frac{1}{N} \left| x(1)e^{-j\omega} + x(2)e^{-j2\omega} + \ldots \right|^2
\]

(2.8)

By computing the periodogram, the peaks would show the location of the frequencies of the underlying sinusoidal motion. Until the work of Yule
[31] (1871 - 1951), the Schuster method was the only numerical way of empirical spectrum analysis. Many time-series observed in nature yielded a periodogram that was erratic and did not exhibit any dominant peaks. G. U. Yule introduced the concept of a finite parameter model for a stationary random process in his fundamental paper on the investigation of the periodicities in time-series. He introduced the concept of an input-output feedback model. In the case when the stationary process is made up of sinusoidal waves with superimposed white noise, the periodogram is effective in picking out the discrete frequencies of the sinusoids. But a purely nondeterministic stationary process is generalized by the convolution formula (input-output relation).

In 1930 Wiener gave a paper which was standard until the work of Tukey in 1949. Wiener method was intended for very long time-series, as such compute the autocovariance (autocorrelation)

$$\phi_{xx}(k) = \frac{1}{N} \sum_{n} x(n)x(n+k) \tag{2.9}$$

then Fourier transform it to find the spectrum

$$S_x(\omega) = \sum_{k=-p}^{p} \phi_{xx}(k)e^{-j\omega k} \tag{2.10}$$

This is known as Wiener-Khinchin theorem [32]
In characterizing, empirically, a signal in terms of a random process model, it is often necessary to estimate averages of the random process model from a single sample sequence. In order to make the computation of the estimate possible we must base our estimate on a finite segment of the sample sequence \( x(n) \), which is assumed to be a realization of a random process defined by random variables \( x_n \), where \( n = 0,1,2,\ldots,N-1 \). It seems plausible when we consider ergodic processes (random processes for which probability averages are equal to time averages). Then the quantity

\[
\hat{m}_x = \frac{1}{N} \sum_{n=0}^{N-1} x(n)
\]  

might be a sufficiently accurate estimate of \( m_x \) if \( N \) were large enough. If the random sequence has been generated by exciting a discrete linear system with white noise, it may be of interest to estimate the parameters of the linear system. To characterize the process we may wish to estimate such parameters as the mean, variance, and the autocorrelation sequence or the power spectral density knowing that the last two quantities form a Fourier transform pair.

The estimation of a parameter of the random process is based upon a finite segment of a single sample sequence \( i.e. \), we have \( N \) values
x(n), 0 \leq n \leq N-1, from which to estimate some parameter, \( \alpha \). The estimate \( \hat{\alpha} \) of \( \alpha \) is thus a function of the random variables \( x_n \), i.e.

\[
\hat{\alpha} = F[x(0), x(1), \ldots, x(N-1)]
\]

and therefore \( \hat{\alpha} \) is a random variable. The estimate \( F[.\] \) is good if there is a high probability that its value will be close to \( \alpha \). Generally speaking, it is plausible that for a good estimator the probability density function should be narrow and concentrated around the true value. Two properties of estimators emerge, that are commonly used as a basis for comparison are the bias and the variance. The bias of an estimator is defined as

\[\text{Bias} = \alpha - E[\hat{\alpha}] \tag{2.13}\]

An unbiased estimator is one for which bias = 0. The variance is in effect a measure of the width of the probability density function and is defined by

\[\text{Var}[\hat{\alpha}] = E[(\hat{\alpha} - E[\hat{\alpha}])^2] = \sigma^2 \tag{2.14}\]

A small variance suggests that the probability density is concentrated around its mean value. An estimator is said to be consistent if, as \( N \) becomes larger, the bias and variance both tend to zero.

In many problems, such as those where it is required to predict future values of the series, it is necessary to construct a parametric model for the time series. A stochastic process may be modeled by using
the fact that a time-series in which successive values are highly dependent, can be generated from a series of independent shocks or impulses. We may thus model such process as the output of a linear filter produced by a white noise. Three types of parametric modeling [24,25] are distinguished:

1. A recursive structure with feedback path or autoregressive model (AR), fig. 2.1.

2. A non-recursive structure with feed forward path or moving-average model (MA), fig. 2.2.

3. In order to have adequate flexibility in the modeling of a statistical time series, it is advantageous to include both the AR and MA components thus producing the autoregressive-moving average model (ARMA).

Therefore, a model is assumed in parametric methods of spectral analysis when formulating the problem, and the requirement is to estimate the parameters of the model from observations of the given process for a limited duration of time. In a variety of applications such as found in adaptive filtering, speech processing, spectral estimation, etc., it is desired to estimate statistical characteristics of a wide-sense weak stationary time-series. More often, this required characterization is embodied in the time series autocorrelation sequence as specified by
\[ R_x(n) = E[x(n+m)x(m)] \quad (2.15) \]

The second characterization of the autocorrelation sequence may be given an equivalent frequency domain interpretation. Namely, upon taking the Fourier transform of the autocorrelation sequence, we obtain the associated power spectral density function (PSDF)

\[ S_x(e^{j\omega}) = \sum_{n=-\infty}^{\infty} R_x(n)e^{-j\omega}, \quad -\infty < \omega < \infty \quad (2.16) \]

The autocorrelation sequence can be reconstituted from the PSDF by applying the inverse Fourier transform

\[ R_x(n) = (1/2\pi) \int_{-\pi}^{\pi} S_x(e^{j\omega})e^{-j\omega} d\omega \quad (2.17) \]

It is desired to effect an estimate of the underlying PSDF with this function being based only on a finite set of observations \( x(1), x(2), \ldots, x(N) \) taken at equispaced time intervals \( T \) (sampling rate). Investigators found a solution by postulating a finite parameter model for the PSDF. Employing a rational model, we will be seeking an approximation of (2.16) by a magnitude-squared ratio of polynomials in \( e^{j\omega} \) as given below

\[ S_x(e^{j\omega}) = \frac{|B_q(e^{j\omega})|^2}{|A_p(e^{j\omega})|^2} = \frac{|b_0 + b_1e^{-j\omega} + \ldots + b_qe^{-jq\omega}|^2}{|1 + a_1e^{-j\omega} + \ldots + a_pe^{-jp\omega}|^2} \quad (2.18) \]
Two classes of rational functions are being distinguished, the AR and MA models; but a more general ARMA model (2.18) is receiving more attention [5,17,22].

2.6 MOVING-AVERAGE TIME SERIES ANALYSIS

Consider the time series \( x(n) \), where \( n \) is an integer between 0 and \( N-1 \). The MA time series will be determined by the following expression

\[
x(n) = \sum_{i=0}^{q} b(i) e(n-i)
\]

(2.19)

where \( e(n) \) are uncorrelated \( (0, \sigma^2) \) random variables (normalized white noise) of a finite MA time-series process. Identifying the MA parameters involves a two-step procedure, i) determining the autocorrelation lag estimator \( \hat{R}_x(n) \) using the provided data, and ii) taking the Fourier transform of these estimates. For unbiased estimate, i.e.,

\[
E[\hat{R}_x(n)] = R_x(n)
\]

(2.20a)

we have

\[
\hat{R}_x(n) = 1/(N-n) \sum_{k=1}^{N-n} x(k+n)x(k) \quad -q \leq n \leq q
\]

(2.20b)

On the other hand, using the biased estimate, i.e.,

\[
E[\hat{R}_x(n)] = \frac{1}{N} \hat{R}_x(n)
\]

(2.21a)

\[
\hat{R}_x(n) = \frac{1}{N} \sum_{k=1}^{N-n} x(k+n)x(k) \quad -q \leq n \leq q
\]

(2.21b)
Examples of this case are the Blackmann-Tukey and the periodogram procedures. Considering this approach the spectral density estimate will have the form

\[ S_X(e^{j\omega}) = \sum_{n=-q}^{q} w(n) \hat{R}_X(n)e^{-jn\omega} \]  \hspace{1cm} (2.22)

where \(w(n)\) is a weighting (window) function introduced to achieve side-lobes reduction, often a rectangular window, i.e., \(w(n)=1\), is selected for simplicity.

In general, we have

\[ S_{MA}(e^{j\omega}) = \sum_{k=0}^{q} b(k) e^{-jk\omega} \]  \hspace{1cm} (2.23)

which is a special rational model of (2.18) where the denominator is set equal to 1. \(S_{MA}(e^{j\omega})\) is said to be a MA model of order \(q\). It could be shown that [25] the Blackman-Tukey spectral estimate (2.22) is a special case of the more general model (2.23).

**Estimating the MA parameters** [22], [24]

Consider the MA(q) time-series given by (2.19), then, the response's autocorrelation sequence is given by

\[ R_X(n) = \begin{cases} \sum_{k=0}^{q} b(k) b(k-n), & \text{for } -q \leq n \leq q \\ 0 & \text{elsewhere} \end{cases} \]  \hspace{1cm} (2.24)
as a sequence of finite length \((2q+1)\). As a special case of (2.16) where, \(-q < n < q\), the spectral density function associated with the autocorrelation sequence (2.24) is given by

\[
S_x(z) = \sum_{n=-q}^{q} R_x(n) z^{-n}
\]  \hspace{1cm} (2.25)

Substituting the value of \(R_x(n)\) yields

\[
S_x(z) = \sum_{k=0}^{q} b(k) z^{-k} \cdot \sum_{m=0}^{q} \bar{b}(m) z^{m}
\]  \hspace{1cm} (2.26)

Because of the Toeplitz nature of the autocorrelation matrix, i.e.,

\[
R_x(n) = R_x(-n)
\]  \hspace{1cm} (2.27)

which implies that the finite-power series \(S_x(z)\) has complex conjugate symmetrical coefficients and consequently its zeros will occur in reciprocal pairs. It is, therefore, possible to factor the power spectral density function as follows

\[
= \alpha^2 \prod_{k=1}^{q} (1 - z_k z^{-1})(1 - z_k z)
\]  \hspace{1cm} (2.28)

where \(\alpha\) is a real scalar.

Comparing (2.26) and (2.28) it is apparent that

\[
\sum_{k=0}^{q} b(k) z^{-k} = \alpha \prod_{k=1}^{q} (1 - z_k z^{-1})
\]  \hspace{1cm} (2.29)

The required \(b(k)\) parameter identification is achieved by carrying out the right-hand-side multiplication in (2.29) and then equating the
coefficients of equal powers of $z^{-k}$.

A flow chart which analyzes the computational operation of an MA process, Blackman-Tukey (BT) procedure, is found in fig. 2.3

$$S_x(e^{j\omega}) = \sum_{n=-q}^{q} R_x(n)e^{-jn\omega} -\pi < \omega < \pi$$

$$= R_x(0) + \sum_{n=-q}^{-1} R_x(n)e^{-jn\omega} + \sum_{n=1}^{q} R_x(n)e^{jn\omega} \quad (2.30)$$

$$= R_x(0) + 2\sum_{n=1}^{q} R_x(n)\cos(n\omega) -\pi < \omega < \pi$$
Fig 2.1 Autoregressive process generated as filtering a white noise with feedback paths (Recursive filter).
Fig. 2.2 Moving-average process generation as a white noise filtered nonrecursively with feed forward paths.
Flow chart for finding the spectrum of a MA process by the BT method (PrOg. #3).
In continuous-time system theory the Laplace transform can be considered as a generalization of the Fourier transform. In a similar manner it is possible to generalize the Fourier transform for the discrete-time signal and systems resulting in what is commonly referred to as the Z-transform \[6,23\]. In general, the Z-transform of a given signal of infinite duration is given by

\[F(z) = \sum_{n=-\infty}^{\infty} f(n)z^{-n}\] (3.1.1)

for \(z\) defined at various points of the unit circle. The magnitude of \(F(z)\) will be the frequency content or spectrum of the given segment of the signal. Such analysis is useful in fields where frequency content of a signal provides information that enables us to draw conclusions about the origin of the signal. In polar form

\[z = re^{j\omega}\] (3.1.2)

Then,

\[F(re^{j\omega}) = \sum_{n=-\infty}^{\infty} f(n)(re^{j\omega})^{-n}\] (3.1.3)
Thus, according to equation (3.1.4) the Z-transform of \( f(n) \) can be interpreted as the Fourier transform of \( f(n) \) multiplied by an exponential sequence. For \( r=1 \), the Z-transform is equal to the Fourier transform of the sequence. For the special case in which the sequence to be represented is of finite duration, \( N \) samples of a digital signal in the form

\[
f(0), f(1), \ldots, f(N-1)
\]

i.e. has only a finite number of non-zero values, it is possible to develop an alternative Fourier representation, referred to as the discrete Fourier transform or DFT which is defined as a Fourier representation of a finite-length sequence which is itself a sequence rather than a continuous function. The DFT corresponds to samples equally spaced in frequency of the Fourier transform of the signal. Another alternative definition to the DFT is given as follows: the rules for finding the frequency response samples from the pulse response and vice-versa are called the discrete Fourier transform and the inverse Fourier transform, DFT and IDFT, respectively. [4], [6], [8].

### 3.2 Sampling Theorem

The sampling theorem is an important concept, for it allows us to replace continuous band-limited signal by a discrete sequence of
Sampled-Data signals have defined values only at certain instants of time. Since it is not possible to feed continuous data into a digital computer, any signal or data input must be represented as a set of numerical values (see fig. 1). In order to discuss the spectrum of a sampled-data signal, it is necessary to evaluate that of the unit Dirac function given by

\[ G(j\omega) = \int_{-\infty}^{\infty} \delta(t) \exp(-j\omega t) \, dt \quad (3.2.1) \]

Therefore sampled-data signal is in effect composed of a set of weighted Dirac pulses each of which contains energy distributed over a very wide band of frequencies. [2, 6, 14]

Uniform sampling theorem pertains to the specifications of a given signal by its samples at uniform intervals. This condition implies that the Fourier transform of \( f(t) \) is zero beyond a certain frequency \( \omega_m \) rad./sec. (\( f_m \) Hz), then the complete information about \( f(t) \) is contained in its uniformly spaced samples at a distance less than \( (1/2f_m) \) sec. The function \( f(t) \) is sampled once every \( T \) seconds, where \( T < 1/2f_m \), or at a rate greater than or equal to \( 2f_m \) samples per second. An ideal uniform sampler would measure the exact values of \( f(t) \) at sampling instants \( nT \) to provide the sequence \( f(nT) \) (see fig. 2). Let
Fig. 3.1 Sampling of an analog signal

f_0, f_1, ..., f_{N-1} are successive samples of f(t)

T: Period of sampling; T \leq 1/(2f_m)

Fig. 3.2 Uniform Sampling
the sampling pulse train \( s(t) \) be a sequence of Dirac function:

\[
s(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT) \quad (3.2.2)
\]

then the sampled signal will be

\[
f_s(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t-nT)
\]

\[
= \sum_{n=-\infty}^{\infty} f(t) \delta(t-nT) \quad (3.2.3)
\]

where

\[
\delta(t-nT) = \begin{cases} 
1, & \text{at } t = nT \\
0 & \text{otherwise}
\end{cases} 
\quad (3.2.4)
\]

Therefore

\[
f_s(t) = \sum_{n=-\infty}^{\infty} f(nT) \delta(t-nT) \quad (3.2.5)
\]

Sampling could also be performed by modulation, from which we conclude that sampling introduces new spectral components that are translations of the baseband transform \( F(\omega) \). The sampled signal will have the following form

\[
f_s(t) = \sum_{n=-\infty}^{\infty} c_n f(t) \exp(jn\omega_s t) \quad (3.2.6)
\]

where \( c_n \) designates the Fourier series coefficients of \( s(t) \).

Taking Fourier transform of (3.2.6), we get
\[ F_s(\omega) = \sum_{n=-\infty}^{\infty} c_n F(\omega - n\omega_s) \quad (3.2.7) \]

### 3.3 DETERMINATION OF THE DFT AND IDFT

In case where a sequence to be represented is periodic (of finite duration), the discrete Fourier series [19,23] fits best. Consider the periodic sequence \( f(n) \) with a period of \( N \) samples so that

\[ f(n) = f(n+kN) \quad (3.3.1) \]

for any integer value of \( k \). It is possible to represent \( f(n) \) in terms of complex exponential sequence with fundamental frequency \( \frac{2\pi}{N} \). There are only \( N \) distinct complex exponentials having a frequency that is an integer submultiple of the fundamental frequency, i.e.

\[ \omega_k = k \frac{2\pi}{N} \quad (3.3.2) \]

such that

\[ e^{j\omega n} = \exp(jk(2\pi/N)n) \quad (3.3.3) \]

Thus the Fourier series representation of the periodic sequence \( f(n) \) need only contain \( N \) of these complex exponentials such that

\[ f(n) = \sum_{k=0}^{N-1} F(k) \exp(j(2\pi/N)kn) \quad (3.3.4) \]

where \( F(k) \) represents the amplitude of the sinusoid at frequency \( \omega_k \).

For convenience we include a multiplying factor \( 1/N \) such that:

\[ f(n) = (1/N) \sum_{k=0}^{N-1} F(k) \exp(j(2\pi/N)kn) \quad (3.3.5) \]
F(k) is obtained from the periodic sequence f(n) as follows:

Multiplying both sides of (3.3.5) by \( \exp(-j(2\pi/N)kn) \) and summing over \( n \) will result in:

\[
\sum_{n=0}^{N-1} f(n) \exp(-j(2\pi/N)nm) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) \exp(j(2\pi/N)n(k-m)) \tag{3.3.6}
\]

Interchanging the order of the summation at the RHS of (3.3.6) and using the relation:

\[
\sum_{n=0}^{N-1} \exp(-j(2\pi/N)n(k-m)) = \begin{cases} 
N & \text{if } k=m \\
0 & \text{otherwise}
\end{cases} \tag{3.3.7}
\]

we obtain:

\[
\sum_{n=0}^{N-1} f(n) \exp(-j(2\pi/N)nm) = \sum_{k=0}^{N-1} F(k) \delta(k-m) \tag{3.3.8}
\]

or

\[
F(k) = \sum_{n=0}^{N-1} f(n) \exp(-j(2\pi/N)nk) \tag{3.3.9}
\]

where \( k = 0, 1, \ldots, N-1 \)

Note that \( F(k) \) is also periodic with a period \( N \), i.e.

\[
F(i) = F(i+N), \quad i = 0, 1, \ldots, N-1 \tag{3.3.10}
\]

We could assign the following notations to \( f(n) \) and \( F(k) \)

\[
F(k) = \text{DFT}\{f(n)\}
\]
\[ f(n) = \text{IDFT}\{F(k)\} \] (3.3.11)

for \( n, k = 0, 1, \ldots, N-1. \)

### 3.4 THE FAST FOURIER TRANSFORM ALGORITHM

The frequency domain approach to digital filtering was considered to be impractical because computation was voluminous, approximately proportional to \( N^2 \), for processing a \( N \)-point sequence. The computational algorithms that exploit both the symmetry and the periodicity of \( \exp(j(2\pi/N)kn) \) can be employed in achieving significant reductions of the computation. The possibility of greatly reduced computation was generally overlooked until about 1965 when Cooley & Tukey published a paper representing an algorithm for finding the DFT. This algorithm is mostly applicable when \( N \) is a composite number.

Suppose \( N = r_1 \cdot r_2 \cdots \cdot r_q \), where \( r_i \) are a set of factors of \( N \). The FFT algorithm requires a number of operations proportional to \( N \cdot (\sum r_i) \). An important special case is when \( r_1 = r_2 = \ldots = r_q = 2 \), i.e.

\[ r_1 = 2 \log_2(N) \]

Many different algorithms have been derived, each with its own advantages and disadvantages in terms of programming and hardware implementation. An algorithm called decimation in time will be discussed here, for the case of \( N \) is a power of 2 which is the most
convenient and practical choice and in the same time it could be used easily in digital computers. Decimation in frequency, another alternate choice could also be implemented.

**DECIMATION IN TIME ALGORITHM**

Decimation is a term used in mathematics for the purpose of recording a sequence according to some prescribed rule; is defined here as decomposing a sequence into successively smaller subsequences. Therefore the increase in efficiency can be achieved by decomposing the DFT computation into successively smaller DFT computations. The DFT of a finite sequence $x(n)$ for $0 \leq n < N-1$ is given by

$$ X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)nk} \quad (3.4.1) $$

where

$$ X(k) = \sum_{n=0}^{N-1} x(n).W_N^{nk} \quad (3.4.2) $$

where

$$ W_N = e^{-j2\pi/N} \quad (3.4.3) $$

The idea behind FFT is to break the original N-point sequence into two shorter sequences, the DFT's of which can be combined to give the DFT of the original N-point sequence. In this case it would require an
order of \((N/2)^2 = N^2/2\) complex multiplications and additions, a saving of a factor of 2. The process above can be iterated so as to reduce the computation of an \((N/2)\)-point DFT to two \((N/4)\)-point DFT's, and thereby effect another saving of a factor of 2.

Define two \((N/2)\)-point sequences \(x_1(n)\) and \(x_2(n)\) as the even and odd parts of \(x(n)\), respectively. Then the \(N\)-point DFT can be written as

\[
X(k) = \sum_{n(even)}^{nk} x(n)W_N^n + \sum_{n(odd)} x(n)W_N
\]

\[\text{(3.4.4)}\]

\[
X(k) = \sum_{n=0}^{(N/2)-1} x(2n)W_N^{2n} + \sum_{n=0}^{(N/2)-1} x(2n+1)W_N^{2(n+1)}
\]

\[\text{(3.4.5)}\]

but

\[
\frac{2}{N} W_N^n = (e^{j2\pi/N})^2 = e^{j2\pi/(N/2)} = W_{N/2}
\]

\[\text{(3.4.6)}\]

which means:

\[
X(k) = \sum_{n=0}^{(N/2)-1} x_1(n)W_N^n + W_N^{nk} \sum_{n=0}^{(N/2)-1} x_2(n)W_{N/2}^{nk}
\]

\[\text{(3.4.7)}\]

\[
X(k+N/2) = X_1(k) - W_N^k X_2(k), \quad k=0, 1, \ldots, N-1
\]

\[\text{(3.4.8)}\]

\[
X(k+N/2) = X_1(k) + W_N^k X_2(k), \quad k=0, 1, \ldots, N-1
\]

\[\text{(3.4.9)}\]

where \(X_1(k)\) and \(X_2(k)\) are the \(N/2\)-point DFT's of \(x_1(n)\) and \(x_2(n)\), respectively. The second step is to compute the \(N/2\)-point \(X_1(k)\) and
\( X_2(k) \) by combining \( N/4 \)-point DFT, i.e.

\[
X_1(k) = A(k) + \frac{N}{2} W_k B(k), \quad k=0,1,\ldots,N-1 \tag{3.4.10}
\]

where \( A(k) \) and \( B(k) \) are the \( N/4 \)-point DFT of the even and odd members of \( x_1(n) \) respectively.

The process above of reducing an \( L \)-point (\( L \) is a number raised to the power of 2) to \( (L/2) \)-point DFT's can be continued until we are left with two-point DFT to evaluate. At each stage there are \( N/2 \) butterflies (see fig. 3.3) the importance of their structure to the FFT flow graph is that only one additional memory location is required to transform an \( N \)-point sequence stored in the same array into which the output sequence is stored. This is the so-called in-place algorithm. In efficient FFT programs, sequences are usually shuffled (permuted) into bit-reversed order by working down the array, successively interchanging \( f(n) \) and \( f(i) \), where \( i \) is the bit-reversed value of \( n \). The interchange of points that have already been accomplished is prevented by comparing \( n \) and \( i \) (Table 1).

\( W_k \) can be computed, alternatively, using:

\[
W^k = \cos\left(\frac{2\pi}{N}k\right) - j\sin\left(\frac{2\pi}{N}k\right) \tag{3.4.11}
\]

or

\[
W_N^k = W_N^{k-L} \quad W_N^L \tag{3.4.12}
\]
The basic operation of the decimation in-time algorithm is the so-called "butterfly", which is represented by the following flow graph.

\[
X = A + BW_N^k
\]

\[
Y = A - BW_N^k
\]

Fig. 3.3
Table 1

<table>
<thead>
<tr>
<th>n</th>
<th>binary</th>
<th>bit-reversed</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>111</td>
<td>7</td>
</tr>
</tbody>
</table>

(Example of bit-reversed values for N = 8)

Comparison of computation volume between the direct method and the FFT algorithm is given in the table below:

Table 2

<table>
<thead>
<tr>
<th>N</th>
<th>N^2</th>
<th>Nlog_2 N</th>
<th>N^2/Nlog_2 N</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>8</td>
<td>2.0</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>24</td>
<td>2.7</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>64</td>
<td>4.0</td>
</tr>
<tr>
<td>32</td>
<td>1024</td>
<td>160</td>
<td>6.4</td>
</tr>
<tr>
<td>64</td>
<td>4096</td>
<td>384</td>
<td>10.7</td>
</tr>
<tr>
<td>128</td>
<td>16384</td>
<td>896</td>
<td>18.3</td>
</tr>
<tr>
<td>256</td>
<td>65536</td>
<td>1024</td>
<td>32.0</td>
</tr>
<tr>
<td>512</td>
<td>262144</td>
<td>4608</td>
<td>56.9</td>
</tr>
<tr>
<td>1024</td>
<td>1048576</td>
<td>10240</td>
<td>102.4</td>
</tr>
</tbody>
</table>

Computations by direct method are proportional to N^2 while using FFT algorithm, computations will be proportional to Nlog_2 N.
with an initial condition to start every stage of the FFT algorithm, where the powers of \( W_N \) are equally spaced.

### 3.5 SPECTRAL ANALYSIS USING FFT

In a large percentage of applications, the problem of measuring the spectrum corresponds to finding the Z-transform of a finite record of the sequence at a large number of points equally spaced around the unit circle. These measurements correspond to measuring the DFT of the finite sequence implemented by FFT.

Given a wide-sense weak stationary discrete-time random process \( x(n) \) with the autocorrelation function \( R_{xx}(m) \), where \( 0 \leq n < N-1, 0 < m < M, \) and \( M < N-1 \). The observed sequence is

\[
x_N(n) = w(n)x(n)
\]

where \( w(n) \) is a window function.

A widely used method is the Blackman-Tukey [15,35] approach which involves taking the Fourier transform of a set of windowed autocorrelation estimates of the time-series. The autocorrelation function of the process can be obtained from

\[
R_{xx}(m) = E[x(n+m)x(n)] \tag{3.5.2}
\]

and an estimator \( \hat{R}_{xx}(m) \), of \( R_{xx}(m) \), is found to be (2.20) or (2.21) which is called the sample autocorrelation function. Based on the
available autocorrelation lag estimates $\hat{R}_{xx}(m)$ the unbiased estimate
can be replaced by the biased one, since the later has less mean-square
error. Consider the Fourier transform of the biased autocorrelation
function

$$I_N(\omega) = \sum_{m=-N+1}^{N-1} \hat{R}_{xx}(m) \exp(-j\omega n) \quad (3.5.3)$$

The Fourier transform of the finite-length data sequence $x(n)$ is:

$$X(\exp(j\omega)) = \sum_{n=0}^{N-1} x(n) \exp(-j\omega n) \quad (3.5.4)$$

then,

$$I_N(\omega) = 1/N |X(e^{j\omega})|^2 \quad (3.5.5)$$

The advent of spectrum analysis based on Fourier analysis can be
 traced to Schuster [30] who made a Fourier series to fit the variation
in sun-spot numbers in an attempt to find hidden periodicities in the
measured data. The spectrum was named the periodogram. The periodogram
spectral estimate is obtained as the magnitude-squared of the output
values from an FFT performed on the data set. Consider the periodogram
defined as

$$\hat{P}_{per}(f) = 1/N\Delta t |\Delta t x(n)\exp(-j2\pi fn\Delta t)|^2 \quad (3.5.6)$$

Use of the FFT will permit the evaluation of $\hat{P}_{per}$ at the discrete set
of $N$ equally spaced frequency, $f_m = m\Delta f$, $m=0,1,.....N-1$, and $\Delta f=1/N\Delta t$
which gives
Using the periodogram, it is the peak in the power spectral density plot rather than the area under plot that is equal to the power of the assumed signal. The periodogram is computationally efficient using the FFT but it has the disadvantage of limited resolution due to the finite data sequence available which is viewed as being obtained by windowing an infinite-length sample sequence. The unmeasured data, in this case, will be assumed zero. This multiplication of the actual series by a window function means the overall transform is the convolution of the desired transform with the window function. The convolution will spread the power into adjacent frequency regions, a phenomenon termed leakage. Thus sidelobes of adjacent frequency cells add in a constructive or destructive manner to the main lobe of a response in another frequency cell of the spectrum affecting the estimate of the power in that cell. Leakage effects due to windowing can be reduced by the selection of windows with nonuniform weighting [19,23]. Using a triangular window (Bartlet) we get

$$E[I_N(\omega)] = \sum_{m=-N+1}^{N-1} \frac{(N-m)}{N} R_{XX}(m) \exp(-j\omega n) \quad (3.5.8)$$

while a rectangular window produces
3.6 AVERAGED PERIODOGRAM

Smoothing the Fourier transform of the estimate of the autocorrelation does produce a good estimate of the power spectrum. A standard approach to reducing the variance of the estimates is to average them over a number of independent estimates. This could be done by averaging the periodograms performed over the same set of the given data making use of the zero padding technique. This technique will also take care of the drawback resulted from the FFT algorithm that provides a sampled version of the periodogram in which the frequency samples are separated by $2\pi/N$ radians. For many applications of interest, this sampling may be too coarse in that the detailed continuous frequency behavior of the periodogram may be somewhat obscured through the sampling process. In this approach a data sequence $x(n)$, $0 \leq n < N-1$, is divided into $K$ segments of $M$ samples each so that

$$N = KM$$

and

$$x^{(i)}(n) = x(n+iM-M)$$

where $0 \leq n < M-1$, and $1 \leq i \leq K$

The concept of zero padding is justified through the following assumption

$$E[P_N(\omega)] = \sum_{m=-N+1}^{N-1} R_{xx}(m) \exp(-j\omega m)$$

(3.5.9)
given data: \( x(1), x(2), \ldots, x(N) \)

segmented data: \( x(1), x(2), \ldots, x(M), 0, 0, \ldots, 0 \)

\( \text{L zeros} \)

In this case we would obtain sampled values of the periodogram at the
more finely spaced frequency \( \omega_k = 2k\pi/(M+L) \), for \( 0 \leq k \leq (M+L) \), instead of
the sampling by \( 2\pi k/M \), where \( 0 \leq k \leq M \) in this case.

The \( K \) periodograms will be computed such that

\[
I_{M}^{(i)}(\omega) = \frac{1}{M} |X^{(i)}(e^{j\omega})|^2, \quad 1 \leq i \leq K
\]

(3.6.3)

where

\[
X^{(i)}(e^{j\omega}) = \sum_{n=0}^{N-1} x(i)(n)e^{-j\omega n}
\]

(3.6.4)

If \( R_{xx}(m) \) is small for \( m > M \), then it is reasonable to assume that
the periodograms \( I_{M}^{(i)}(\omega) \) are independent of one another.

The spectrum estimate is defined by:

\[
B_{xx}(\omega) = \frac{1}{K} \sum_{i=1}^{K} I_{M}^{(i)}(\omega)
\]

(3.6.5)

\[
= \frac{1}{K[1/M]} \sum_{i=1}^{K} |X^{(i)}(e^{j\omega})|^2
\]

(3.6.6)

\[
= \frac{1}{N} \sum_{i=1}^{K} X^{(i)}(e^{j\omega})
\]

(3.6.7)

then \( E[B_{xx}(\omega)] = E[I_{M}^{(i)}(\omega)] \)
The Bartlett estimate is shown to be the convolution of the true spectrum with the Fourier transform of the triangular window function corresponding to a $M$-sample periodogram where $M=N/K$. Thus the Bartlett estimate is also a biased one. It is shown that the variance is inversely proportional to $K$, and therefore $B_{xx}(\omega)$ will have a variance approaching zero as $K$ becomes larger and hence the consistency of $B_{xx}(\omega)$. 
Fig. 3.4 Flow chart for computing the periodogram (ProG. #1)
START

DATA SS(N)

N = K*M

segment
SI(M)
i = 1, K

j = 1, N

j \geq M

NO

SI(J) = 0

YES

SI(J) = SS(j + iM - M)

S(j) = CMPLX(SI)

Find FFT

SFFT
Compute the averaged periodogram
\[ \text{PSD} = \frac{1}{K} \sum_{i=1}^{K} Y_i \]

Plot the PSD

STOP

END

Fig. 3.5 Flow chart for computing the averaged periodogram for smoothing the PSD (Prog. #2).
4.1 INTRODUCTION

Approximating second-order statistical relationships of time series is usually performed by seeking a rational fit whose parameters are typically selected so that these relationships best represent a set of autocorrelation lag estimates computed from time series observation. Two special types of rational functions referred to as the moving average (MA) and autoregressive (AR) models are being considered. The MA model was briefly analyzed in chapter two. When the MA(q) spectral estimate expression (2.23) is compared to the theoretical PSDF (2.22), a serious modeling mismatch can arise whenever the underlying autocorrelation lags are such that the elements of the autocorrelation matrix $\hat{R}_{xx}(m)$ are not approximately equal to zero for $m>q$. To avoid this shortcoming of MA models an alternate and more efficient model is being proposed which do not invoke the unnecessarily harsh requirement of a truncated autocorrelation lag behavior. A well-known utilized model is the autoregressive model representation. Namely, a spectral
model is said to be autoregressive model of order p, i.e. AR(p), if it may be put into the following form

\[ S_{AR}(e^{j\omega}) = \frac{b_0}{1 + a_1 e^{-j\omega} + \ldots + a_p e^{-jpu}} \]  

(4.1.1)

\[ \frac{|b_0|^2}{|A_p(e^{j\omega})|^2} = \frac{\prod_{k=1}^{p} (1-p_k e^{-j\omega})(1-p_k e^{j\omega})}{\prod_{k=1}^{p}} \]

\[ (p_k: \text{complex conjugate of } p_k, \text{poles of the polynomial } A_p(e^{j\omega})) \]

This AR(p) model has a functional behavior which is completely characterized by its (p+1) parameters \( b(0), a(1), a(2), \ldots, \) and \( a(p) \). The characteristic pth-order polynomial \( A_p(e^{j\omega}) \) is seen to influence the frequency behavior of the estimate while the parameter \( b_0 \) controls the level. The poles of such model occur in reciprocal pairs so that the AR(p) spectral model is also commonly referred to as an all-pole model.

4.2 AR PARAMETER IDENTIFICATION AND LINEAR PREDICTION

Error-prediction filter is designed to predict the sample value of a wide-sense weakly stationary time series one time unit ahead by using the present and past values of the time series. The signal in concern is modeled as a linear combination of its present and past values in the following manner

\[ x_n + \sum_{k=1}^{p} a_k x_{n-k} = e_n \]  

(4.2.1)
The excitation time series $e_n$ is taken to be a sequence of zero-mean and $\sigma^2$-variance uncorrelated variables. The estimated, noise-free AR($p$) time series is:

$$R_n = - \sum_{k=1}^{p} a_k x_{n-k}$$  \hspace{1cm} (4.2.2)

Determination of the criterion which should be followed for finding the filter (or AR process) parameters, $a_k$, stems out from the principle of minimization of mean-square error [18,54] which is defined by:

$$\mathcal{E}_s = \sum_{n} e_n^2$$  \hspace{1cm} (4.2.3)

where $e_n = |x_n - \bar{x}_n|$.

This is done by taking the derivative of $\mathcal{E}_s$ with respect to $a_k$ and equating it to zero such as:

$$\frac{\partial \mathcal{E}_s}{\partial a_i} = 0 \quad \text{for} \quad 1 \leq i \leq p$$  \hspace{1cm} (4.2.4)

which leads to the following set of linear equations

$$\sum_{k=1}^{p} a_k \sum_{n} x_{n-k} x_{n-i} = - \sum_{n} x_n x_{n-i}$$  \hspace{1cm} (4.2.5)

The total minimum squared error power is thus obtained and is given by:

$$P_{\text{min}} = \sum_{n} x_n^2 + \sum_{k=1}^{p} a_k \sum_{n} x_n x_{n-k}$$  \hspace{1cm} (4.2.6)

The range of $n$ can be assumed to be infinite.
By defining the autocorrelation function as:

\[ R_x(i) = \sum_{n} x_n x_{n-i} \quad 1 \leq i \leq p \quad (4.2.7) \]

we get the following set of what is known as the Yule-Walker equations:

\[ \sum_{i=1}^{p} a_k R_x(i-k) = -R_x(i) \quad 1 \leq i \leq p \quad (4.2.8) \]

Since, our signal is usually known over a finite duration of time it is modified by multiplying it by a window function [24,26].

Another way [22,26] to get the Yule-Walker equations is by performing the expectation operator (2.1) over \((x_n x_{n-k})\). For stationary process:

\[ E[x_{n+k} x_n] = R_x(k) \quad (4.2.9) \]

For stationary and ergodic process, the autocorrelation function can be computed as a time average as given by (2.21) for biased estimates or by (2.20) for unbiased estimates.

\[ E[x(n+k) x(n)] = E[x_{n+k}\left(\sum_{i=1}^{p} -a_i x_{n-i} + e_n\right)] \quad (4.2.10) \]

or

\[ = E[x_n\left(-\sum_{i=1}^{p} a_i x_{n-i+k} + e_{n+k}\right)] \]

which implies:

\[ R_x(k) = -\sum_{i=1}^{p} a_i R_x(k-i) + E[e_{n+k} x_n] \quad (4.2.11) \]
where
\[ E[e_{n+k}x_n] = \begin{cases} 0 & \text{for } k \geq 0 \\ \sigma^2 & \text{for } k = 0 \end{cases} \quad (4.2.12) \]

therefore,
\[ R_X(k) = \sum_{i=1}^{p} a_i R_X(k-i), \quad 1 \leq k \leq p \quad (4.2.13) \]
\[ R_X(0) = \sigma^2 - \sum_{i=1}^{p} a_i R_X(-i) \]

Linear prediction [26] can, also, be put under the following definition. Given some spectrum \( P(\omega) \), we wish to model it by another spectrum \( P_{\text{mod}}(\omega) \) such that the integrated ratio between the two spectra is minimized
\[
\text{const. } \pi \int_{-\pi}^{\pi} \frac{P(\omega)}{P_{\text{mod}}(\omega)} \, d\omega = \frac{\text{Es}}{2\pi} \quad (4.2.14)
\]
The parameters of the model spectrum are computed from the normal (Yule-Walker) equations where the needed autocorrelation coefficients \( R(i) \) are easily computed from \( P(\omega) \) by a simple Fourier transform.

To obtain the \( a_k \) coefficients of the AR(p) process by solving the set of \( p \) normal equations (autocorrelation method), it is required \((p^3)/3 + O(p^2)\) operations (multiplication or division) and \( p^2 \) storage locations. Further reductions can be achieved using the advantage of the special form of the normal equations, which in matrix form they can be written as
where the autocorrelation matrix can be shown to be a Toeplitz, i.e. it has the property of positive semidefiniteness and symmetry [24,pp.62-67].

Levinson derived an elegant recursive procedure which assumes the column vector of the right-hand-side of equation (4.2.15) to be a general column vector. By making use of the fact that this vector comprises the same elements found in the autocorrelation matrix, another method attributed to Durbin emerges which is twice as fast as Levinson's.

4.3 The Levinson-Durbin algorithm

Consider the Yule-Walker equations which describe the relationship between the AR process parameters and the autocorrelation function elements:

\[-a_k R(i-k) = R(i) \quad 1 \leq i \leq p \quad \text{(4.3.1)}\]

where \((1,-a_1, -a_2, \ldots, -a_p)\) represents the impulse response vector defining the prediction-error filter.

Let \(a_0 = 1\), and \(a_i = -a_i\), where \(1 \leq i \leq p\) \(\text{(4.3.3)}\).
Then, the Yule-Walker equations can be put under the following form:

\[
\sum_{k=0}^{p-1} a_k R_x(i-k) = R_x(i) \quad 1 \leq i \leq p \quad \text{(4.3.4)}
\]

The error-prediction filter minimum power is given by:

\[
P_p = P_{\text{min}} = R_x(0)
\]

\[
P_{\text{min}} = a_k R(-k) \quad \text{(4.3.5)}
\]

Then, the Yule-Walker equations can be put under the following form:

\[
\sum_{k=0}^{p-1} a_k R_x(i-k) = \begin{cases} P_p & \text{for } i = 0 \\ 0 & \text{for } i = 1, 2, \ldots, p \end{cases} \quad \text{(4.3.6)}
\]

Suppose we know the solution to the set of \(p\) equations pertaining to a prediction error filter of order \((p-1)\) as given by:

\[
\sum_{k=0}^{p-1} a_{p-1,k} R_x(i-k) = \begin{cases} P_{p-1} & \text{for } i = 0 \\ 0 & \text{for } i = 1, 2, \ldots, p-1 \end{cases} \quad \text{(4.3.7)}
\]

\((a_{i,j}: i \text{ designates the order and } j \text{ the index})\)

This form of equations is said to be obtained in forward direction which, in matrix form, can be written as:

\[
\begin{bmatrix}
R(0) & R(1) & \ldots & R(p-1) \\
R(1) & R(0) & \ldots & R(2-p) \\
\vdots & \vdots & \ddots & \vdots \\
R(p-1) & R(p-2) & \ldots & R(0)
\end{bmatrix}
\begin{bmatrix}
a_{p-1,0} \\
a_{p-1,1} \\
\vdots \\
a_{p-1,p-1}
\end{bmatrix}
= \begin{bmatrix}
1 \\
ap_{p-1,1} \\
\vdots \\
ap_{p-1,p-1}
\end{bmatrix}
\quad \text{(4.3.8)}
\]
making use of the fact that the [R] matrix is Toeplitz, then

$$R_x(i) = R_x(-i)$$

Substitute $p-1-k$ for $k$ we get what is known as the backward direction relations:

$$\sum_{k=0}^{p-1} a_{p-1,p-1} R_x(i-k) = \left\{ \begin{array}{c}
\begin{array}{c}
0 \\
\vdots \\
\vdots \\
1
\end{array}
\end{array} \right\} \quad i = 0, 1, \ldots, p-2$$

(4.3.9)

in matrix form:

$$\begin{pmatrix}
R(0) & R(1) & \cdots & R(p-1) \\
R(1) & R(0) & \cdots & R(p-2) \\
\vdots & \vdots & \ddots & \vdots \\
R(p-1) & R(p-2) & \cdots & R(0)
\end{pmatrix}
\begin{pmatrix}
a_{p-1,p-1} \\
a_{p-1,p-2} \\
\vdots \\
a_{p-1,1}
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
\vdots \\
1
\end{pmatrix}$$

(4.3.10)

Combining both forms given by (4.3.8) and (4.3.10) yields:

$$\begin{pmatrix}
R(0) & R(1) & \cdots & R(p) \\
R(1) & R(0) & \cdots & R(p-1) \\
\vdots & \vdots & \ddots & \vdots \\
R(p-1) & R(p-2) & \cdots & R(1) \\
R(p) & R(p-1) & \cdots & R(0)
\end{pmatrix}
\begin{pmatrix}
1 \\
a_{p-1,1} \\
\vdots \\
a_{p-1,p-1}
\end{pmatrix}
= \begin{pmatrix}
0 \\
a_{p-1,p-1} \\
\vdots \\
a_{p-1,1}
\end{pmatrix}$$

(4.3.11)

for filter order of $p$ we have
Comparing (4.3.11) and (4.3.12) we obtain the following recursive relations known as **Levinson's recursion formulas**:

\[ a_{p,i} = a_{p-1,i} + \rho_p a_{p-1,p-i} \quad i = 0, 1, \ldots, p \]  
\[ P_p = P_{p-1} + \rho_p \Delta_p \]  
\[ 0 = \Delta_p + \rho_p P_{p-1} \]  
\[ a_{p,i} = \begin{cases} \rho_p & \text{for } i = p \\ 0 & \text{for } i > p \end{cases} \]  
\[ P_p = P_{p-1}(1 - |\rho_p|^2) \]  

In particular, the recursive algorithm is initialized by:

\[ a_{1,1} = \frac{R_X(1)}{R_X(0)} \]  
\[ P_1 = \sigma_1 = (1 - |a_{1,1}|^2)R(0) \]  

with the recursion for \( k = 2, 3, \ldots, p \) we have

\[ a_{k,k} = \frac{[R_X(k) + \sum_{j=1}^{p-1} a_{k-1,j} R_X(k-1)]/P_{k-1}}{P_k} \]  
\[ a_{k,i} = a_{k-1,i} + a_{k,k} a_{k-1,k-i} \]  
\[ P_k = \sigma_p = (1 - |a_{k,k}|^2)P_{k-1} \]
If a process is actually an AR(p) one then \( a_{p+1,k} = a_p,k \) where \( k = 1,2,\ldots,p \) and hence \( a_{p+1,p+1} = 0 \). In general, for an AR(p) model \( a_{k,k} = 0 \) and \( P_k = \sigma_k = \sigma_p \) for \( k > p \). Hence the variance of the excitation noise is a constant for a model order equal to or greater than the correct order. Thus the point at which \( \sigma^2 \) does not change would appear to be a good indicator of the suitable order. The coefficients \( (a_{1,1}; a_{2,2}; \ldots; a_{p,p}) \) are often called the reflection coefficients. They have the property that for the set of autocorrelation coefficients \( (R_x(0), R_x(1), \ldots R_x(p)) \) to be a valid autocorrelation sequence it is necessary and sufficient that \( |a_{k,k}|^2 < 1 \) for \( k = 1,2,\ldots,p \). This condition also is used as a criterion for stability since a digital filter is said to be stable when all of its poles lie inside the unit circle.

After the filter (AR process) parameters were determined, the power spectral density (PSD) estimate of the process is computed using the following expression

\[
P_{AR} = \frac{\sigma_p^2}{|1 + \sum_{k=1}^{p} a_k \exp(-j\omega k)|^2}
\]  

(4.3.15)
An alternative representation of the PSD is given by:

\[ P_{AR} = \sum_{n} r_{x}(n) \exp(-jwn) \quad (4.3.16) \]

where

\[ r_{x}(n) = \begin{cases} \frac{R_{x}(n)}{2} & \text{for } |n| < p \\ \frac{1}{2} & \text{for } |n| = 0 \\ \frac{1}{2} \left( a_{p} + a_{p}^{*} \right) & \text{for } |n| > p \end{cases} \quad (4.3.17) \]

From this it is easy to see that the AR PSD preserves the known auto-correlation lags and recursively extends the lags beyond the window of known lags. The AR PSD summation given by (4.3.16) is identical to the Blackman-Tukey PSD function (2.7.4) up to lag \( p \) but continues with an infinite extrapolation of the autocorrelation function rather than windowing it to zero. Thus AR spectra do not exhibit sidelobes due to windowing. It is the implied extrapolation (4.3.17) that is responsible for the high resolution property of the AR spectral estimate [24,26].

4.4 MODEL ORDER SELECTION

One of the main considerations in obtaining the AR model from raw time series observations is that of model order choice. It has been noticed that when \( p \) is selected too low, there will be generally too few model poles to adequately represent the underlying spectrum which means a highly smoothed spectrum is obtained. On the other hand, too high a choice of \( p \) will typically result in spurious effects (false
peaks) in the spectral estimate. The filter order can be determined experimentally by trying different values for \( p \), then choose the particular value which satisfies some predetermined requirements. However, for the proper choice of the optimal value \( p(\text{opt}) \) we may use several objective criteria [22,24]. One intuitive approach would be to construct AR models of increasing order until the computed prediction error power reaches a minimum (\( P_i = P_{i-1}(1-|a_i, i|^2) \)). As long as \( |a_i, i|^2 \) is nonzero (must be < 1), the prediction error power decreases, that is not sufficient to indicate the search be terminated.

Akaike has provided two criteria. The first criterion, called the final prediction error FPE, selects the order of the AR process so that the average error for a one-step prediction is minimized. For an AR(p) process the FPE is given by:

\[
(FPE)_p = \frac{P_p(N+p+1)}{(N-p-1)}
\]

(4.4.1)

where, \( N \) : number of data samples

\( P_p \) : prediction error power

The selected order, \( p \), is the one for which the FPE is minimum. Akaike suggested a second order selection criterion using a maximum likelihood approach to derive it. It was termed the Akaike Information Criterion (AIC). This criterion determines the model order by minimizing an
information theoretic function. Assuming the approach has a Gaussian statistical behavior, the AIC is defined as:

\[ \text{AIC}_p = \ln(P_p) + 2(p+1)/N \]  

(4.4.2)

4.5 MAXIMUM ENTROPY SPECTRAL ESTIMATION (MESE)

Burg's maximum entropy processing is a method for computing the power spectral density (PSD) from the first N lags of the autocorrelation function, but unlike the periodogram or the Blackman-Tukey approach, this method does not assume that the other lag values are zeros. Instead, the Burg method, mathematically, ensures that the fewest possible assumptions about unmeasured data are made by choosing the spectrum that maximizes the entropy of the process. The usefulness of this approach is marred to a considerable extent by the lack of a criterion for choosing the optimum length of the prediction error filter [24,36,39,40].

INFORMATION AND ENTROPY

The amount of information is defined by the relation:

\[ I_1 = K \ln(1/p_1) \]  

(4.5.1)

\[ I_1 = \log_2(1/p_1) \]  

(4.5.2)

where: \( p_1 \) is the probability of occurrence of M different things (\( m_1 \))

\( m_1 \) represents the number of times an \( x(t) \) occurs in M
realizations of the process.

If the observation time (T) is very large, we may expect $p_i T$ of $m_i$ things to have happened.

\[ I_{\text{total}} = K(p_1 T \ln(1/p_1) + p_2 T \ln(1/p_2) + \ldots + p_M T \ln(1/p_M)) \]

\[ M = K \sum_{i=1}^{M} p_i T \ln(1/p_i) \quad (4.5.3) \]

Entropy is defined as average of information per time interval such that:

\[ H = \frac{I_{\text{total}}}{T} = -K \sum_{i=1}^{M} p_i \ln p_i = -\sum_{i=1}^{M} p_i \log_2 p_i \quad (4.5.4) \]

Therefore, the entropy is the measure of the uncertainty described by a set of probabilities (a measure of disorder).

Given a data sequence $x(n)$ which possesses the PSD defined by

\[ S(\omega) = \sum_{n=-\infty}^{\infty} R_X(n) \exp(-j\omega n) \quad -\pi < \omega < \pi \quad (4.5.5) \]

where,

\[ R_X(n) = 1/2\pi \int_{-\pi}^{\pi} S_X(\omega) \exp(j\omega n) d\omega \quad \text{for all } n \quad (4.5.6) \]

In particular, for $|n| < p$ \quad (4.5.7)

In terms of information theory we require the entropy per sample time.
series be maximum, i.e.

\[ H = \int_{-\pi}^{\pi} \log_2 S(\omega) d\omega \quad (4.5.8) \]

is needed to be maximized under the constraints of (4.5.7).

The procedure of maximization [39] goes as follows:

\[ \frac{\partial \log_2 S(\omega)}{\partial R_x(n)} e^{-j\omega n} \frac{e^{-j\omega n}}{S(\omega)} = 0 \]

or

\[ \frac{\partial H}{\partial R_x(n)} = \frac{\pi}{-\pi} \int S(\omega) d\omega = 0 \quad \text{for } |n| > p \quad (4.5.9) \]

Let

\[ \frac{1}{S(\omega)} = [S(\omega)]^{-1} = \sum_{n=-\infty}^{\infty} \psi(n) e^{-j\omega n} \quad -\pi < \omega < \pi \quad (4.5.10) \]

where \( \psi(n) \) is the adopted autocorrelation function to \( [S(\omega)]^{-1} \) which means:

\[ \psi(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [S(\omega)]^{-1} e^{j\omega n} d\omega \quad \text{for all } n \quad (4.5.11) \]

for \( n \) tending to \(-n\)

\[ \int_{-\pi}^{\pi} [S(\omega)]^{-1} e^{-j\omega n} d\omega = 0, \quad \text{and} \]

\[ \frac{1}{2\pi} \int_{-\pi}^{\pi} [S(\omega)]^{-1} e^{-j\omega n} d\omega = 0 \quad \text{for } |n| > p \quad (4.5.12) \]
which implies

\[ \psi(n) = 0 \quad \text{for} \quad |n| > p, \quad \text{and} \]

\[ [S(\omega)]^{-1} = \sum_{n=-p}^{p} \psi(n) e^{-j\omega n} \quad (4.5.14) \]

this can be written under the following form:

\[ S(\omega) = \frac{\sigma^2}{\sum_{n=-p}^{p} \psi_n e^{-j\omega n}} \quad (4.5.15) \]

Let \( A(z) = \sum_{n=0}^{p} \psi_n z^{-n} = 1 + \psi_1 z^{-1} + \psi_2 z^{-2} + \ldots + \psi_p z^{-p} = A(z) \quad (4.5.16) \)

where \( z = e^{j \omega} \), and \( A(z) \) is a minimum delay polynomial, i.e. it has no zeros on or outside the unit circle \(|z|=1\). Thus

\[ S(z) = \frac{\sigma^2}{A(z) \cdot A(z^{-1})} \quad (4.5.17) \]

which shows that the maximum entropy process can be represented as an autoregressive model of order \( p \).

\[ S_{ME}(\omega) = \frac{\sigma^2}{|1 + \sum_{k=1}^{p} \psi_k e^{-j\omega k}|^2} \quad (4.5.18) \]

This estimate expresses maximum uncertainty with respect to the unknown information but is consistent with the known information.
Van Den Bos (1971) showed that maximizing the entropy of the corresponding PSD function is, in fact, equivalent to the least square fitting of an AR model. Smylie et al. (1973) gave an expression for the entropy of a discrete stationary Gaussian process \( x(n) \) as shown by

\[
H = \frac{1}{2} \log_2 \det[C]
\]  

(4.6.1)

where \( C \), the autocorrelation matrix, is given by

\[
C(M) = \begin{bmatrix}
\rho(0) & \rho(1) & \ldots & \rho(p) \\
\rho(1) & \rho(0) & \ldots & \rho(p-1) \\
\vdots & \vdots & \ddots & \vdots \\
\rho(p) & \rho(p-1) & \ldots & \rho(0)
\end{bmatrix}
\]  

(4.6.2)

Assume the first \((p+1)\) lags are known exactly; the idea behind the MESE theory is to choose the unknown autocorrelation coefficients \( \rho(p+1), \rho(p+2), \ldots \), in such a manner that the entropy of the process is maximized at each step, \( i \). Thus \( \rho(p+i) \) is determined by maximizing \( \det[C(p+i)] \) with respect to \( \rho(p+i) \). In fact \( \det[C(p+i)] \) has a single maximum as a function of \( \rho(p+i) \), by applying the rule of differentiation. Hence, \( \rho(p+i) \) is obtained for \( i=1,2,\ldots \).

Consider a discrete-time stochastic process \( x(n) \) given by

\[
x_n = \sum_{k=1}^{p} a_k x_{n-k} + e_n
\]  

(4.6.3)

As it was shown, the Yule-Walker (normal) equations are defined by
\[ p(k) = \sum_{i=1}^{p} a_i p(k-i) \quad k = 1, 2, \ldots, p \]  \hspace{1cm} (4.6.4)

Suppose now that the first (p+1) lags, (0), (1), \ldots, and (p), of the autocorrelation sequence are known. Substituting these values in the system of equations (4.6.4) we get the coefficients \( a_i \), where \( 1 \leq p \); the unknown coefficients (p+k) may be determined by solving the equation:

\[ \text{det}[C(p+k)] = 0 \]  \hspace{1cm} (4.6.5)

which is equivalent to the maximizing the entropy of the process. This equivalence extends to all the extrapolated autocorrelation coefficients. The conclusion follows, therefore, that the MESE theory is congruent to fitting an AR model to the random process [24, 40].

Z-transforming equation (4.6.3), we get:

\[ X(z) = X(z)[a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p}] + E(z) \]  \hspace{1cm} (4.6.6)

\[ X(z) = \frac{E(z)}{1 - \sum_{i=1}^{p} a_i z^{-i}} \]  \hspace{1cm} (4.6.7)

but \[ S(z) = |X(z)|^2 = \frac{|E(z)|^2}{|1 - \sum_{i=1}^{p} a_i z^{-i}|^2} \]  \hspace{1cm} (4.6.8)

Evaluating the Z-transform on the unit circle, \( z = e^{j\omega} \), we obtain half the power density and consequently the AR spectral estimate is given by
The AR process is identified by the coefficients \((1, a_1, \ldots, a_p)\).

\[
S_{AR} = \frac{2\sigma^2}{|1 - \sum_{i=1}^{p} a_i z^{-i}|^2} \quad (4.6.9)
\]
START

Input Data
S(N)

Assign AR order
p

Calculate p+1
autocorr. lags
(biased)

Initialize
Levinson's
Algorithm

begin recursion
k=2,3,...p

Find a_{k,k}
a_{k,i} & P_{min}
eq 4.3.14cde
Fig. 4.
Flow chart for computing the AR PSD using the Yule-Walker eq. and Levinson's algorithm. (Prog. #4)
The estimation process of a given data sequence is frequently done using a mathematical approximation approach which is helpful in certain situations. In many cases we seek to estimate the power spectral density, which is dependent on the infinite length autocorrelation sequence, from a finite set of time series observations. This incompatibility is resolved through the process of parameterization. The most widely employed procedure of parameterization is the rational spectral model. In this model an input driving sequence \( \{e_n\} \) and the output sequence \( \{x_n\} \) that is to model the data are related by the following linear difference equation:

\[
x_n = \sum_{i=0}^{q} b_i e_{n-i} - \sum_{k=1}^{p} a_k x_{n-k}\tag{5.1.1}
\]

The digital filter representing this equation is assumed to being driven by a zero-mean, \( \sigma^2 \) variance white noise \( \{e_n\} \) (see figure 5.1). This most general model is termed an ARMA model of order \( (p,q) \). By and large, spectral estimation investigators have been, primarily, concerned with such rational models which are seen to possess a finite number of \( q \)-zeros and \( p \)-poles.
The resulting spectral density function will be given by:

\[
S_{\text{ARMA}}(e^{j\omega}) = \frac{b_0 + b_1 e^{-j\omega} + \cdots + b_q e^{-jq\omega}}{1 + a_1 e^{-j\omega} + \cdots + a_p e^{-jp\omega}}^2
\]

(5.2)

\[
= \frac{B_q(e^{j\omega})}{A_p(e^{j\omega})}^2
\]

(5.3)

where, \( z_k \), and \( p_k \) represent the complex conjugate of \( z_k \) and \( p_k \) respectively.

An ARMA model is seen to have a frequency characterization which is the composite of a MA and AR models (mixed zero-pole rational...
function). The ARMA technique assumes that the noise corrupted AR(p) process is a general ARMA(p,p) process so that this approach is used to reduce the degradation of the AR spectral estimate in the presence of noise.

5.2 ARMA PARAMETER ESTIMATION

The main demanding task in the power spectral density estimation procedures is to determine the parameters, $a_i$ and $b_i$, of the adopted corresponding model, which are most compatible with the given autocorrelation lags, $R(0), R(1), \ldots, R(p)$. A considerable easing in computational requirements may be achieved if we evaluate the $a_i$ and $b_i$ separately. Some of the techniques exploited in solving this parameter identification problem were based on a least square error criterion which requires the solution of a set of linear equations with the most suboptimum case. These methods, generally, estimate the AR and MA parameters separately rather than jointly as required for optimal parameter estimation.

The maximum likelihood methods (MEM) can be used to find the ARMA parameters, $a_i$ and $b_i$, simultaneously but these methods entail the application of nonlinear programming solution procedures.

ALGORITHM 1

Once the parameters of a pure AR process have been consistently
identified, an ARMA model of the same process can be derived. The ARMA parameters and order can be determined directly from the parameters of the purely AR model. Making use of the fact that it is the same process, the transfer functions would, approximately, be equivalent to each other.

Consider the time series given by (5.1.1), where, for a purely AR process, \( b_i = 0 \) for \( i > 1 \).

\[
x_n = \sum_{k=1}^{p} a_k x_{n-k} + b_0 e_n \tag{5.2.1}
\]

Taking the Z-transform of both equations in (5.1.1) and (5.2.1) and equating their transfer functions we get:

\[
\frac{X(e^{j\omega})}{E(e^{j\omega})} = \frac{b_0}{1 + \alpha_1 e^{-j\omega} + \ldots + \alpha_L e^{-jL\omega}}
\]

\[
= \frac{b_0 + b_1 e^{-j\omega} + \ldots + b_q e^{-jq\omega}}{1 + a_1 e^{-j\omega} + \ldots + a_p e^{-jp\omega}} \tag{5.2.2}
\]

where \( \alpha_k, 1 \leq k \leq L \), are the parameters of the purely AR model of order \( L \) and, \( a_i \) and \( b_j \) with \( 1 \leq i \leq p \) and \( 1 \leq j \leq q \), are the respectively the AR and the MA components of the mixed ARMA\((p,q)\) model parameters of the same process. Cross-multiplying and equating the coefficients of like powers \( e^{-j\omega} \) we get the following set of relationships:
\[ a_2 = b_2 + b_1 a_1 + b_0 a_2 \]
\[ \vdots \]
\[ \vdots \]

\[ a_p = b_p + b_{p-1} a_1 + \ldots + b_0 a_p \]

\[ 0 = b_p a_i + b_{p-1} a_{i+1} + \ldots + b_0 a_{i+p} \quad i=1,2,\ldots,L \]

where \( b_j = 0 \) for \( j = q+1, q+2, \ldots, p \)

In matrix form it can be written as:

\[
\begin{bmatrix}
    a_p & a_{p-1} & \ldots & a_{p-q+1} \\
    a_{p+1} & a_p & \ldots & a_{p-q+2} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{p+q-1} & \ldots & a_p
\end{bmatrix}
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_{p+q}
\end{bmatrix}
= \begin{bmatrix}
    a_{p+1} \\
    a_{p+2} \\
    \vdots \\
    a_{p+q}
\end{bmatrix}
\]

(5.2.4)

and the set of equations (5.2.3):

\[
\begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_j \\
    \vdots \\
    a_p
\end{bmatrix}
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_{j-1} \\
    \vdots \\
    b_p
\end{bmatrix}
= \begin{bmatrix}
    1 & 0 & 0 & \ldots & 0 \\
    b_1 & 1 & 0 & \ldots & 0 \\
    \vdots & \vdots & \ddots & \ddots & \vdots \\
    b_{j-1} & \ldots & 0
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_j \\
    \vdots \\
    a_p
\end{bmatrix}
\]

(5.2.5)

where \( j \) should take on values \( 1, 2, \ldots, L \) but in this case our assumption

would be \( b_j = 0 \) for \( j > q+1 \) and \( a_j = 0 \) for \( j > p+1 \).
Clearly, with \( p \), \( q \), and the parameters \( \alpha_1, \alpha_2, \ldots, \alpha_{p+q} \) known, the wanted parameters \( a_i \) and \( b_k \) can be determined by solving the systems (5.2.4) and (5.2.5). The square matrix in (5.2.4) must be non-singular for the uniqueness of \( b_i \), such will be the case if the \( p \) and \( q \) are specified as minimal orders. The determination of the minimal orders of \( p \) and \( q \) from just knowledge of the AR parameters seems to be straightforward. Define a matrix involving the parameters \( a_i \) for some \( p=p_e \) and \( q=q_e \) such as \( [A_e(p_e,q_e)] \), whose rank is tested for a set of values of \( (p_e,q_e) \) such as \( (0,0); (1,1); (1,0); (2,2); \) etc. until for some \( p \) and \( q \) we get

\[
\text{Rank}[A_e(p_e,q_e)] = q \quad \text{for } p_e > p \text{ and } q_e > q \quad (5.2.6)
\]

or alternatively,

\[
\det[A_e(p_e,q_e)] = 0 \quad \text{for } p_e > p \text{ and } q_e > q \quad (5.2.7)
\]

In practice, only estimates of the true \( \alpha_i \) and of \( |A_e| \) are available. Hence, the condition (5.2.7) is replaced by a test:

\[
|A_e(p_e,q_e)|^2 < \varepsilon \quad \text{for some } p_e > p \text{ and } q_e > p \quad (5.2.8)
\]

and some \( \varepsilon > 0 \)

**Algorithm 2**

This technique is similar to the previous one but needs more manipulation with less restrictions. In fact, it proposes that a high-order AR model be obtained (L order) for the MA or the ARMA process, and the
AR sequence be further processed to estimate the MA parameters.

The method used here involves two stages: first, the estimation of the MA parameters, $b_1$, directly from the estimated autocorrelation sequence for the entire ARMA process, and the information given about the equivalent (pure) AR model parameters and order, followed by estimation of the AR parameters, $a_j$, from the residuals.

### MA parameter estimation

Designate by $S(z)$ the Fourier transform of the autocorrelation sequence, then we can write:

$$S(z) = |B_q(z)|^2 = \frac{\sigma^2}{|A_L(z)|^2}$$

$$= B_q(z)B_q(z^{-1}) \approx \tilde{S}(z) = \frac{\sigma^2}{A_L(z)A_L(z^{-1})}$$

which implies that:

$$B_q(z) = \frac{\sigma}{A_L(z)}$$

and hence that:

$$\gamma_L(n) * b_q(n) = \sigma \delta(n)$$

in fact

$$\gamma_L(n) * b_q(n) = \sigma \delta(n) + e(n)$$

where
\( \gamma_{L}(n) \) are found by solving the Yule-Walker equations (4.2.8) or (4.2.13)

\((*)\) represents convolution,

\( \delta(n) \) is the delta (Kronicker) function, and

\( e(n) \) is additive noise, assumed white

\( b_{q}(n) \) are obtained by minimizing the mean-square error of \( e(n) \) for a selected range of values of \( n \) such that \( n=1,2,\ldots,L-d \), where \( d \) is a small number introduced to avoid end effects near \( n=1 \).

On the other hand, equation (5.2.13) could be written in a more meaningful form such as:

\[
e(n) = \sum_{k=0}^{q} \gamma_{L}(n-k)b_{q}(k) - \sigma \delta(n)
\] (5.2.14)

\( b_{q}(0) \) can be found from:

\[
e(0) = \gamma_{L}(0)b_{q}(0) - \sigma = 0,
\]

\[
\sigma = \frac{b_{q}(0)}{\gamma_{L}(0)}
\] (5.2.15)

Equation (5.2.14) can be put in matrix form for \( n=1,2,\ldots,L-d \), as:

\[
e = \gamma_{L-d} \cdot \sigma + A \cdot b_{q}
\] (5.2.16)

Note: (5.2.16) represents a vectorial equation.

where

\[
e = [e(1), e(2), \ldots, e(L-d)]^T
\] (5.2.17a)
\[ \gamma_{L-d} = [\gamma_L(1), \gamma_L(2), \ldots, \gamma_L(L-d)]^t \]  
(5.2.17b)

\[ \text{(t stands for transpose)} \]

and

\[
A = \begin{bmatrix}
\gamma_L(0) & 0 & 0 & \cdots & 0 \\
\gamma_L(1) & \gamma_L(0) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\gamma_L(q-1) & \gamma_L(q-2) & \cdots & \gamma_L(0) \\
\vdots & \vdots & \cdots & \vdots \\
\gamma_L(L-d-1) & \gamma_L(L-d-2) & \cdots & \gamma_L(L-d-q)
\end{bmatrix}  
\]  
(5.2.17c)

and the wanted MA coefficients vector is:

\[
\underline{b}_q = [b_q(0) \ b_q(1) \ \cdots \ b_q(q)]  
\]  
(5.2.17d)

It is required to minimize the mean-square error with respect to the elements of the unknown vector \( \underline{b}_q \) proceeding as following:

\[
\text{Min}_{\underline{b}_q} \sum_{n=1}^{L-d} |e^2(n)| = e^t \cdot e = f(\underline{b}_q)  
\]  
(5.2.18)

Minimization process can be identified by taking the derivative of the function \( f(\underline{b}_q) \) with respect to \( \underline{b}_q \) and equating it to zero, such as:

\[
\frac{\partial f(\underline{b}_q)}{\partial \underline{b}_q} = \frac{\partial}{\partial \underline{b}_q} \left[ (\gamma_{L-d} \cdot \sigma + A \cdot \underline{b}_q)^t \cdot (\gamma_{L-d} \cdot \sigma + A \cdot \underline{b}_q) \right]  
\]  
(5.2.19)
after performing some algebraic calculations we obtain:

\[ 2\sigma A^T \gamma_L - d + 2A^T A b_q = 0 \]  \hspace{1cm} (5.2.20)

or

\[ (A^T A)b_q = -\sigma A^T \gamma_L - d \]

which gives

\[ b_q = (A^T A)^{-1} \sigma A^T \gamma_L - d \]  \hspace{1cm} (5.2.21)

Therefore solving the system of equations (5.2.21), the MA parameters \( b_i, 1 \leq i \leq q \), of the ARMA process, would be determined.

**AR parameter estimation**

Now consider the transfer function of the whole ARMA model given by:

\[ S_{\text{ARMA}}(z) = \frac{B_q(z)B_q(z^{-1})}{A_p(z)A_p(z^{-1})} \]  \hspace{1cm} (5.2.22)

the equivalent AR model transfer function of the same process is:

\[ S_{\text{AR}}(z) = \frac{\sigma^2}{A_L(z)A_L(z^{-1})} \]  \hspace{1cm} (5.2.23)

then,

\[ S_{\text{ARMA}}(z) = S_{\text{AR}}(z) \]  \hspace{1cm} (5.2.24)

which implies the following equality:
In terms of convolution theory this relationship can be written as:

\[ \gamma_L(n) * b_q(n) = \sigma a_p(n) \]  \hspace{1cm} (5.2.26)

In fact, with the additive noise \( e(n) \), (5.2.26) becomes:

\[ \gamma_L(n) * b_q(n) = \sigma a_p(n) + e(n) \neq a_{L+q}(n) \]  \hspace{1cm} (5.2.27)

where by \( a_{L+q}(n) \) we denoted the residual sequence.

After the MA parameter have been computed, a complete convolution could be operated on (5.2.27) to get the residual sequence \( a_{L+q}(n) \).

Then, to extract the \( p \)-th order AR coefficients estimates of the ARMA process from \( a_{L+q}(n) \) sequence, a simple truncation operation for \( n>p \) is applied to \( a_{L+q}(n) \). From the convolution theory, (5.2.27) can be written in the following form:

\[ \sum_{k=0}^{q} b_q(k) \cdot \gamma_L(n-k) = \sigma a_{L+q}(n) \]  \hspace{1cm} (5.2.28)

or in compact matrix form,

\[ \sigma a_{L+q} = [A_L] \cdot b_q \]  \hspace{1cm} (5.2.29)
which when broken down gives:

\[
\begin{array}{cccc}
    a_{L+q(0)} & y_L(0) & 0 & \ldots & 0 \\
    a_{L+q(1)} & y_L(1) & y_L(0) & \ldots & 0 \\
    a_{L+q(2)} & y_L(2) & \ldots & \ldots & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    a_{L+q(q)} & y_L(q) & y_L(q-1) & \ldots & y_L(0) \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    a_{L+q(L)} & y_L(L) & \ldots & \ldots & y_L(L-q) \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    a_{L+q(L+q)} & 0 & 0 & \ldots & 0 \\
\end{array}
\]

(5.2.30)

The value of \( L \) should be chosen as minimal as \((p+q)\) which means that \( e(n) \) in (5.2.27) can be made to be equal zero for \( p+1 \leq n \leq p+q \). \( L \), also, is required to be large enough to adequately represent the AR model in the ARMA process. It can be as large as accuracy and computational considerations will allow.
START

Data in S(N)

Find the AR(L) coeff. as in prog. 4

Assign p & q orders

Calc. the a_p and b_q parameters of the corres. ARMA

Compute the ARMA PSD

PLOT the PSD

END

Fig. 5.
Flow diagram for ARMA PSD computation by the equivalence with AR process
(Prog. #5)
CHAPTER 6

ELECTROMYOGRAPHY

AND

DIGITAL SIGNAL PROCESSING TECHNIQUES

6.1 INTRODUCTION

Because of its potential for providing an easily accessible quantitative estimate of the active state of muscle, myoelectric activity has seen wide application in diverse fields, such as clinical diagnosis, bioengineering, physical therapy, rehabilitation, prostheses, etc. Attempts to provide a quantitative myoelectric estimate of muscle activity have been plagued by the presence of an apparent noise component of very large amplitude and low frequency which is superimposed on the desired signal.

Electromyographic signals (EMG's) from the body's intact musculature have been suggested and utilized by many researchers as an effective noninvasive method to provide commands to control an electrically powered artificial limb. The introduction of microcomputers has made it possible to experiment with more sophisticated signal detection and motion control of human prostheses [45,55].
Muscles are defined as the organs of locomotion in man's body. The skeletal muscles will contract when stimulated electrically and, conversely, they produce a detectable current or voltage when they contract from any cause. The structural unit of contraction is called muscle fiber whereas the functional unit of a striated muscle is called motor unit (MU) which is shown in the figure below. An impulse descending the nerve axon causes all the muscle fibers in one MU to contract almost simultaneously; MU's contract for frequency pulses below 50/sec. Electromyographic signal is the electrical manifestation of the neuromuscular activation associated with a contracting muscle. The output pathway of the motor control system consisting of the ensemble of α motoneurons activates the skeletal muscle fibers. This output pathway is studied by recording action potentials (AP's) associated with the firings of individual MU's by means of indwelling or surface electrodes.
The detected waveform associated with one twitch of the MU is known as the motor unit action potential (MUAP). During a muscle contraction the MU's fire repetitively generating a motor unit action potential trains (MUAPT).

The muscle force produced is controlled primarily by varying the number of activated MU's and the rate of activation of an individual MU. Muscle force generation is also accompanied by electrical activity which, detected by surface electrodes, produces the electromyogram (EMG). The EMG is a complex temporal pattern and is best described as a random signal which may be modeled as a weighted sum of AP's of the MU's. Therefore, its statistical properties are dependent on the number of active MU's and their firing rate [45].

A model which describes myoelectric activity as a random process whose instantaneous amplitude distribution for a fixed level of muscle activity is Gaussian with zero mean

$$p(M(t)) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}(M(t)/\sigma(F))^2\right)$$

with \( E[M(t)] = 0 \), on the basis of experimental observations.

where,

- \( p(M) \): distribution function
- \( F \): muscle force
- \( M(t) \): myoelectric activity, and
\( \sigma(F) \): standard deviation

The time dependence between successive values of myoelectric activity is completely characterized by the autocorrelation function or, equivalently, the power spectral density function (PSDF). The PSD is assumed to be a rational function of frequency for mathematical convenience. The myoelectric activity, then, can be modeled as a zero-mean, Gaussian white noise process passed through a linear constant-coefficient filter. A physical situation corresponding to this model is that of a muscle contracting under isometric, nonfatiguing conditions. In this case, the parameters describing the myoelectric activity are a function only of the muscle force.

![Diagram of shaping filter](image)

**Fig. 6.2**

Functional mathematical model representing myoelectrical activity as a band-limited, zero-mean, Gauss-Markov process \( N(t) \) which is amplitude-modulated by a static function of muscle force.
The PSDF will have the following functional representation:

$$S_M(f) = K \ |H(f)|^2 \ \sigma(F)^2$$

where $H(f)$, a ratio of polynomials in $f^2$, is the shaping filter (fig. 6.2) function. This filter is a combination of the frequency content of subcutaneous myoelectric activity and the filtering effect of transmission through tissue and detection by electrodes.

Surface EMG signals are used to discriminate a set of arm and wrist function, for example, by developing some signal analysis techniques. The main work is aimed toward estimating the forces generated in a set of muscles and then using these estimates as the input to a model representing the subject under examination (limb, for example). Force estimation is done by measuring the total average power in an EMG signal and equating it to the force in the muscle directly below the electrode.

6.2 SCHEMATIC MODEL FOR THE MUAPT

When the muscle is at low contraction level, one single MU potential is distinguishable. This individual MU potential is the summation, at the electrode, of the electrical activity of each muscle fiber within the MU [45]. The shape of the AP has a significance in the diagnosis of some neuromuscular diseases. The MUAPT is characterized by
its interpulse intervals (IPI), the shape of the MUAP, and the average
firing rate, defined as

\[ \lambda(t,F) = [E[x]]^{-1} = \left\{ \int_{-\infty}^{\infty} xp(x,t,F)dx \right\}^{-1} \quad (6.2.1) \]

where \( p(x,t,F) \) is the IPI histogram (probability density function), \( F \)
is the force, and \( x \) represents the IPI's variable (fig. 6.3).

The MUAPT's are decomposed into a sequence of Dirac impulses

\[ \delta_i(t) = \sum_{k=1}^{n} \delta(t-t_k) \quad (6.2.2) \]

in which \( t_k = \sum_{l=1}^{k} x_l; \quad k,l = 1,2,\ldots,n \quad (6.2.3) \)

where \( t_k \) denotes the time location of the MUAP's and \( n \) is the total
number of the IPI's in a MUAP.

then they are passed through a filter whose impulse response is \( h_i(t); \)

the output of the filter is \( u_i(t) = \sum_{k=1}^{n} h_i(t-t_k) \quad (6.2.4) \)

The parameters which define the EMG can be determined as the

\[
\text{Fig. 6.3}
\]

\[
\text{h(t)}
\]

\[
\delta(t-t_j)
\]

\[
t_j
\]

\[
\ldots
\]

\[
u(t)
\]

\[
x_1
\]
mean rectified value $E[u_i(t,F)]$, and

RMS value $MS[u_i(t,F)]$

In normal muscle, peak-to-peak MUAP, recorded by indwelling electrodes, range from few $\mu$V's to 5mV (typically 500$\mu$V). The number of phase of MUAP's recorded by bipolar electrodes may range from one to four; MUAP's with more than four phases appear usually in abnormal muscle tissue. The amplitude of the MUAP is directly proportional to the radius of the muscle fiber and inversely proportional to the distance filter-electrode.

By representing $h_i(t)$ by a Fourier series, it is shown that in the frequency range 0-40 Hz the power density spectrum is affected primarily by the IPI statistics. A noticeable peak appears in the PSD at the frequency corresponding to the firing rate and progressively lower peaks at harmonics of the firing rate. The amplitude of the peaks increases as the IPI's become more regular. Actually the pronounced peak in the 0-40 Hz frequency region of the PSDF appears during weak to moderate nonfatiguing constant isometric contractions. The height of the low-frequency peak decreases relatively to the remainder of the spectrum if the contraction strength increases from low to moderate values (see fig. 6.10). Beyond 40 Hz the PSD is essentially determined by the shape of $h_i(t)$. The myoelectric signal (ME) $m(t,F)$, then, will
be modeled as a linear spatial, and temporal summation of the MUAPT's detected by the electrode [46].

The superposition at the recording site forms the physiological ME signal \( m_p(t,F) \) which is not observable. When the signal is detected an electrical noise \( n(t) \) is introduced. The detected signal will also be affected by the filtering properties of the recording electrode \( r(t) \) and possibly other instrumentation. The resulting signal \( m(t,F) \) is the observable signal (fig. 6.4).

\[
\begin{array}{c}
\text{MUAPT's} \\
E
\end{array}
\quad
\begin{array}{c}
\text{n(t)} \\
\oplus
\end{array}
\quad
\begin{array}{c}
m(t,F) \\
r(t)
\end{array}
\]

**Fig. 6.4**

When an isometric tension is sustained, there is an increase in the average-half-wave or integrated amplitude of the EMG as the muscle fatigues. The frequency components of the surface EMG may provide an assessment of the degree of the muscle fatigue.

### 6.3 Modeling the EMG

A. The EMG could be described as amplitude-modulated noise[52]. The carrying wave is relatively wide-band noise modulated by a number of mechanical parameters such as force or velocity. A demodulator is intended for use with EMG's recorded from patients with pathological
Amplitude modulation is achieved by multiplying the filtered noise by a sine function (generated at $f_m = 5\text{Hz}$, a representative value) on which a dc bias is superimposed. Mathematically, it is expressed by:

$$x_t = n_t(1 + M\sin\omega t) \quad 0 \leq M \leq 1 \quad (6.3.1)$$
tremor. The demodulator will be composed of a high-pass filter, with a
cutoff frequency, \( f_c \), of 60 Hz, a full-wave rectifier, and a second-
order low-pass filter with \( f_c = 15 \) Hz. The whole operation is shown in
the block diagram below (fig. 6.5).

The mean variance of the amplitude modulated noise is:

\[
V_x = (1 + M^2/2) \cdot V_n
\]  
(6.3.2)

where, \( V_n \) designates the mean power (variance) of the noise \( n_t \).

The power spectrum of \( x_t \) can be expressed as

\[
S_{xx} = S_{nn} + M^2 \cdot S_{mm} \ast S_{nn}
\]  
(6.3.3)

\( S_{nn} \) represents the power spectrum of the carrying noise

\( S_{mm} \) represents the power spectrum of the modulating signal

\((*)\) is the convolution operator

\[
S_{mm} = 1/4[\delta(f + f_m) + \delta(f - f_m)]
\]  
(6.3.4)

\( S_{nn} \) is an ideal square power spectrum with a bandwidth of 100 Hz, which
is comparable to that of a real EMG tremor,

\[
S_{nn} = \begin{cases} 
A & \text{for } 50 \text{Hz} < |f| < 150 \text{Hz} \\
0 & \text{elsewhere}
\end{cases}
\]  
(6.3.5)
The method of demodulating amplitude modulated noise most commonly used is full wave rectification which converts the signal $x_t$ to $y_t$ with the following spectral density

$$S_{yy} = S_{n'n'} + M^2 S_{nn} * S_{n'n'} \quad 0 \leq M \leq 1 \quad (6.3.6)$$

where $S_{n'n'}$ is the power spectrum of $|n_t|$.

$S_{n'n'}$ shows a delta function at $f = 0$ Hz while $S_{nn}$ does not. $S_{n'n'}$ is calculated from a square wave noise spectrum $S_{nn}$ by a half wave rectification method. The assumptions made are a normal distributed noise $n_t$ with center frequency $f_c$ of the square noise spectrum higher than the width of the spectrum of $n_t$, $B_n$. 

**Fig. 6.6**

$S_{nn}$ spectrum
Fig. 6.7

$S_{n'n'}$ spectrum

Fig. 6.8a

$S_{yy}$ spectrum of the demodulated signal [52]

Fig. 6.8b

Power spectrum $S_{xx}$ after amplitude modulation of $nt$ with a sine function at $fm = 5$Hz [52]
B. Model of the ME signal considering the conduction velocity

During a sustained muscle contraction, the spectrum of the ME signal is known to undergo compression, as a function of time, related to the decreasing conduction velocity of the muscle fibers[51]. As a result of this restraint the frequencies of the ME signal will shift to the left of the spectrum, with fatigue, and a consistent increase in amplitude of the signal is also observed. This shift in spectrum and increase in amplitude may be accounted by a single physiological correlate, the conduction velocity. The decrease in conduction velocity can be monitored by tracking the changes in a characteristic frequency of the spectrum, such as the mode or the peak, the mean, and the median frequency. The median frequency $f_{\text{med}}$ is the frequency at which the spectrum is divided into two regions with equal powers while the mean frequency $f_{\text{mean}}$ is the average frequency. Mathematically these can be expressed by:

$$\int_{0}^{f_{\text{med}}} S(f) \, df = \int_{0}^{\infty} S(f) \, df = \frac{1}{2} \int_{0}^{\infty} S(f) \, df$$  \hspace{1cm} (6.3.7)

$$f_{\text{mean}} = \frac{\int_{0}^{\infty} fS(f) \, df}{\int_{0}^{\infty} S(f) \, df}$$  \hspace{1cm} (6.3.8)
The assumption of a Gaussian random process significantly defines the ME signal, however, it doesn't define the shape of the spectrum of the signal. Therefore, it is assumed that the PSD can be expressed as a rational function of two polynomials in frequency. A rational function which describes the shape of the spectrum (fig. 6.10) is given by

\[
G(f) = \frac{f^4 f^2}{h} \frac{f^2}{(f^2 + f^2) (f^2 + f^2)^2} \quad (6.3.9)
\]

where \( f_h \) and \( f_l \) represent the high and low cutoff frequencies, and \( K^2 \) is a scaling factor.

\[
f_l = \rho f_h \quad 0 < \rho < 1 \quad (6.3.10)
\]

To monitor the effect of the conduction velocity, the spectrum can be put under the following form:

\[
S(f) = \frac{1}{v^2} G(fd/v) \quad (6.3.11)
\]

Substituting the expression of \( G \) from (6.3.9) we get:

\[
= \frac{f^4 (f/v)^2}{h} \frac{1}{[(f/v)^2 + \rho^2 f^2][[(f/v)^2 + f^2]^2]} \quad (6.3.12)
\]

where \( v \) denotes the conduction velocity and \( d \) is the interelectrode distance. \( d \) can be chosen to be 1 for convenience.
We notice that the conduction velocity explicitly affects the signal amplitude by a factor of \(1/v\), it also inversely scales the frequency components of \(G\). It can be observed that a decrease in conduction velocity compresses the spectrum (fig. 6.10). Plotting the spectrum of a given EMG data as a function of frequency (fig. 6.11a) and using the logarithmic scale we can find the asymptotic behavior of the spectrum (fig. 6.11b). A flat asymptotic behavior about the mode (peak) of the spectrum is observed and as frequencies increased above this region a rolloff occurs in the spectrum. Below the flat region of the spectrum a more shallow rolloff can be noticed as frequency decreased. The shallower rolloff implies lower-order polynomials.

**BLOCK DIAGRAM**  (Signal plus noise model of myoelectric activity)

White noise \(w\)

\[
\begin{align*}
E[w] &= 0 \\
R_{ww}(\tau) &= \delta(\tau) \\
S_{ww}(f) &= 1
\end{align*}
\]

True ME

\[
\begin{align*}
E[u] &= 0 \\
R_{uu}(\tau) &= q\delta(\tau) \\
S_{uu}(f) &= q
\end{align*}
\]

estimate

**Fig. 6.9**

observ. signal = desired signal + noise

\[
\hat{m} = m + u
\]
PSD PEAK VARIATION
WITH CONDUCTION VELOCITY

Fig. 6.9
SHAPE OF EMG SPECTRUM
LINEAR SCALE

Spectrum as a function of frequency
Fig. 6.10

Asymptotic behavior of the spectrum
Fig. 6.11
In reality the noise is band-limited; however, it may be extended well beyond the frequency range of the desired signal. This situation may be modeled by passing the observed ME signal through a low-pass filter whose gain and cutoff frequency are optimally selected to minimize the average steady-state mean-square error, i.e. $E[(m-\hat{m})^2]$. In other words, a signal is low-pass filtered to reduce the contamination from ambient noise and its cutoff frequency is chosen to be sufficiently high to pass the majority of the desired signal and sufficiently low to remove most of the higher frequency components.

6.4 EMG AS A STATISTICAL TOOL

Electromyographic (EMG) signals are an indication of the electrical activity of muscles which arises whenever there is a voluntary or involuntary contraction of a muscle. Examination of the EMG is achieved by using the power spectral density function (PSDF). Myoelectric signals detected on the surface of the skin, with surface electrodes, are random signals while myoelectric activity observed is a function determined by the firings of muscle fibers whose action potentials sum to produce EMG. As it is known in engineering analysis, autocorrelation function and PSDF provide useful characteristics of random signals given that these quantities embody all first- and second-order statistical properties of the signals.
Using modern digital signal processing techniques, we can generate control signals for a multifunction lower arm prosthesis from a set of surface EMG signals [58]. This work is aimed toward estimating the forces generated in a set of muscles and then using these estimates as input to a kinematic model of the intact limb. The outputs of this model are limb function forces and/or velocities that can be used to control the servos of a particular prosthesis design. Force estimation is usually done by measuring the total average power in an EMG signal and equating that to the force in the muscle directly below the electrode taking into consideration some additional signal components, such as noise due to interaction between different leads and other muscle activities through the tissue. Graupe [55], in his work on EMG prosthesis control, has taken advantage of spectral properties of the EMG signal and shown that these properties change when conditioned on different limb functions. He defined the spectra of each limb function by an AR model.

\[ g_m(k) = \sum_{j=1}^{p} \alpha_{m,j} g_m(k-j) + e_m(k) \]  \hspace{1cm} (6.4.1)

where \( 1 < m < M \) represents each limb function

\( g_m(k) \) is the \( m \)-th limb function signal at time instant \( k \)

\( \alpha_{m,j} \) designates the \( j \)-th regression coefficient for the \( m \)-th limb function at time instant \( k \)
p represents the order of the autoregression (AR) model which could be determined by comparing the final results provided by different orders.

The model parameters are to be determined by, then, utilizing one of the methods discussed earlier (chapter 4), or for example, by a least-squares procedure that minimizes the set of cost functions

\[
J_m = \sum_{i=p+1}^{N} (e_m(i))^2
\]  
(6.4.2)

where N represents the number of samples.

Frequency components of the surface EMG may provide an assessment of the degree of muscle fatigue, the overall appearance of the EMG seems to change with fatigue, or, in another words, the EMG will be dominated by lower frequency components during fatigue. In an attempt to quantify fatigue the median frequency of the spectrum is examined and found to be decreasing linearly with time.

Myoelectric signals of long duration are probably nonstationary due to the physiology of the system. It is suggested that the myoelectric activity be examined over a short enough time period where these inputs are relatively constant and the EMG is perhaps weakly stationary. Only if this is the case does it make sense to discuss autocorrelation and PSD functions. The degree of stationarity of the EMG is
of interest since it is desired to calculate certain statistical averages of the random process by performing the corresponding time averages on an arbitrary individual sample function of the process. This is what is known as ergodicity. It is required that the sample record of the data be long compared to the random fluctuations of the data time history. The very small variations in the measured parameters over the short time intervals of interest indicate that the myoelectric signals may be considered wide-sense stationary. The mathematical formulas used to analyze the EMG are those derived for ergodic processes where the processing formulation is presented in discrete form to correspond to the digital data processing. The sample mean of an individual sample function, computed by time averaging, is given by

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

(6.4.3)

where, \(x_i\) are the data values, and \(N\) is the number of data points. The data can also be transformed to zero mean so that less error will be introduced into the computation and to remove the DC level

\[
y_n = x_n - \bar{x} \quad n=1,2,\ldots,N
\]

(6.4.4)

The estimated autocorrelation function is then given by

\[
R_x(kh) = R_k = \frac{1}{N-k} \sum_{i=1}^{N-k} y_i y_{i+k} \quad k=0,1,\ldots,M
\]

(6.4.5)

where \(h\) is the sampling interval, \(k\) represents the lag number and \(M=N\).
is the maximum lag number. The true mean square value of the data, the average power, is given by

\[ R_X(0) = \frac{1}{N} \sum_{i=1}^{N} (y_i)^2 = \bar{y}^2 \]  

(6.4.6)

For comparisons of the PSDF's of the EMG data, the frequency of the mean power, \( \bar{f} \), and the variance, var, about this mean value are calculated as given by

\[ \bar{f} = \frac{\sum_{f=f_0}^{f_c} f P(f)}{f_c - f_0} \]  

(6.4.7)

\[ \text{var} = \frac{\sum_{f=f_0}^{f_c} (f-\bar{f})^2 P(f)}{f_c - f_0} \]  

(6.4.8)

6.5 INSTRUMENTATION AND DATA ACQUISITION

Studying the EMG provides a useful clinical tool for diagnosis of skeletal muscles and the control of limb prostheses. Furthermore, the detection of the myoelectric signal, or the summed AP's, represents the overall activity of the muscle. Because of their ease of applica-
tion, surface electrodes are used for transducing the myopotentials whereas a large input impedance differential amplifier is utilized to amplify the raw EMG signal. The analog data are digitized and then stored on a memory tape or disc for further processing [50,54]. Surface EMG is sampled through a pair of bipolar adhesive surface electrodes, such as, silver-silver chloride type E5SH, with an active surface area of $2\text{cm}^2$ placed 7cm apart over the medial aspect of the forearm. The electrodes are interfaced with the skin surface through a layer of paste (REDUX, by Hewlet Packard) to increase conductivity. The transduced myoelectric potentials are of very small amplitude, $50\mu\text{V}_{pp}$ to $1\text{mV}_{pp}$, but then the raw EMG is amplified [fig. 6.12,13] using a differential amplifier (AC Microelectrode Preamplifier Model P15, GRASS Instruments) having an input impedance of $200\ \Omega$ differential, capacity coupled, and a CMMR factor of 10,000:1 ($\simeq 80 \text{ dB}$). After the electrodes are in place on the arm, position of the electrodes is determined by palpation, and prior to any muscular activity, the RMS amplitude of any $60 \text{ Hz}$ noise amplitude from the electrodes is assessed. For all experiments reported, the RMS noise amplitude was always around $8\mu\text{V}$, whereas during a maximal voluntary contraction, the RMS amplitude of the EMG was as high as $2\text{mV}$. The amplified EMG is then digitized using an A/D converter, incorporated in an Apple II microcomputer with a sampling rate of 1024 samples per second, which indicates a frequency response from
Fig. 6.12

Fig. 6.13
dc to 512 Hz. With the Apple computer parallel interface, the data are stored on a memory diskette in binary format. The data are then transformed into text (hexadecimal format) using an Apple Assembly program (BIN TO TEXT) and furthermore into decimal numbers with a Fortran program (DECIMAL) on the IBM 4341. Transfer of data from Apple computer to the IBM 4341 is done using the modem incorporated in the Apple through a package called COM WARE-II. The acquired data are then stored in the IBM computer where later were retrieved for further processing. Using procedures explained in earlier chapters, such as periodogram MA, AR, or ARMA parametric models, the PSDF is calculated. Computational results are then listed on a line printer and/or plotted using a CalComp plotter, such as HP 7225A plotter.

6.6 OBSERVATIONS

Shifts of the high-energy regions of the power spectrum can be inferred from the changes in the mean frequency, $\bar{f}$. Fatigue is characterized by more power in the lower half of the spectrum while the signals produced before fatigue have greater power in the upper one half. The apparent decrease in the mean frequency and the increase in the average power with fatigue indicates that more of the energy is found at lower frequencies. The increase in the average power over that before fatigue can also be attributed to the synchronization of the
AP's, i.e. if muscle tension is increased without fatigue occurring, the number of muscle fibers taking part in the muscle contraction increases and as a result there is an increase in the EMG amplitude due to the resultant summation. It is to be noted that both the shift in spectrum towards lower frequencies and the increase in the ME signal during a sustained contraction can be accounted by a single physiological correlate, the conduction velocity, v. The decrease in v can be monitored by tracking the changes in the central frequency.

6.7 NUMERICAL EXAMPLES

The EMG spectrum will be found by using some of the techniques discussed earlier. Take, for example, the raw EMG (DEMGOXO) shown in fig. 6.12 for N=512. The AR(p) PSD, for different orders, are plotted as can be seen in fig. 6.14. Low AR orders, p=4, produced high dc level, while AR(41) could still detect the highest peak at f=25 Hz accompanied with other low-level peaks in the low frequency region, at f=10 and 35 Hz. Increasing the AR order, over p=45, the total performance is distorted and the spectrum became unstable with dominant peaks in the high frequency region, for example f=150 Hz and higher. The optimal order turned out to be p=8 found by Akaike's FPE criterion (Section 4.4) but at p=18 we can still get an acceptable spectrum, see fig. 6.15. An interesting result could be drawn from fig. 6.16 where
1) ARMA(2,2) OF AR(8)
2) ARMA(4,4) OF AR(8)
3) ARMA(8,8) OF AR(30)

fig. 6.17

SHIFT IN THE ENH SPECTRUM AR(8)

fig. 6.18

1) ARMA(2,1) OF AR(12)
2) ARMA(8,4) OF AR(12)
3) ARMA(12,8) OF AR(24)

fig. 6.19
the same AR orders as in fig. 6.14 were used but the number of the AR coefficients used in the spectral density expression (4.1.1) was truncated to $p = p_{opt} - 8$, the spectrum of AR(41) looks very much smoother and closer to AR(8), fig. 6.15.

The same EMG data (DEMGX0) were used to plot the ARMA(p,q) PSD. The performance was accepted and good for $p > 4$ and $q > 4$ with $p > q$. Notice that the dc bias in the ARMA(2,2) is dominant while it is also considerable in ARMA(4,4), however, a good choice was ARMA(8,4) with smooth spectrum (fig. 6.17). This situation was observed at almost any L, of the equivalent AR process, in the vicinity of or slightly higher than the quantity $(p+q)$. Cases of $q > p$ produced a wider bandwidth with a shift in the peak towards lower frequencies whereas for $q < p > 12$ could maintain the bandwidth but with shift towards the right (fig. 6.18).

EMG spectra could also be obtained by the periodogram method. As an application we are going to use the raw EMG EMGLA, fig. 6.13. The spectrum for $N=512$ is shown in fig. 6.20 (notice the multitude of peaks in the low frequency region, 0-40 Hz, where it is supposed to be). This spectrum can be further smoothed to almost a typical EMG spectrum (see fig. 6.10) by applying the averaged periodogram as can be seen from fig. 6.21, and 6.22. The smoothest was obtained by averaging 16 periodograms of 32-point data each with the respective zero padding to
construct a 512-point FFT's (fig. 6.22). Another way to reduce the unwanted peaks in the PSD was done by zero padding the 512-point FFT thus forming a 1024-point FFT, and computing the corresponding periodogram (fig. 6.24)

6.8 COMMENTS ON THE EXAMPLES

As was indicated earlier in the discussion of the EMG performance, there must be a decrease in the central frequency and a corresponding increase in the power spectrum amplitude with fatigue, or increase in the conduction velocity. This case was examined by plotting the normalized spectra of two EMG data taken from the same subject (fig. 6.19) where the first represents the spectrum when the muscles of the biceps are at rest (DEMGX0) while the peak shifts to the left in the second spectrum representing the contracting muscle, weight was added in the palm (DEMGX2 not shown here).

The high performance of the parametric techniques was easily distinguished from the Fourier-based method, the periodogram. This could be examined from the smoothness of the AR and the ARMA spectra with the appropriate parameters selection of each method against the periodogram.
CHAPTER 7

NUMERICAL APPLICATIONS

7.1 INTRODUCTION

Two spectral estimation techniques based on Fourier transform operations have thus been developed

i) the PSD estimate based on the indirect approach via an auto-correlation estimate known also as the Blackman-Tukey, \( P_{BT} \), power spectral estimation (2.22) which is a special case of the more general MA spectral estimation.

ii) the other PSD based on the direct approach is, typically, referred to as the periodogram, \( P_{Per} \), (3.3.5) or (3.3.6), or for spectrum smoothing we make use of the so-called averaged periodogram discussed in section 3.6.

Recently parametric models for spectral estimation have gained more attention proving to be more spectrally efficient. The PSD of an AR process (fig. 2.1) will be evaluated by the Yule-Walker method with Levinson recursion as outlined in section 4.3. The comparative spectral estimation performance of the ARMA modeling (2.18) procedures will be investigated as developed in section 5.2, Algorithm 2.

7.2 NUMERICAL EXAMPLES (\( P_{BT} \) & \( P_{Per} \))

To demonstrate the utilization of these methods, the classical
problem of the detection and frequency identification of sinusoids in white noise case will be considered. The time series under examination is taken to be composed of \( m \) equiamplitude (unity) sinusoids in additive noise as specified by

\[
x(n) = \sum_{i=1}^{m} A_i \cos \omega_i n + e(n)
\]  
(7.1)

where in the case of two sinusoids \( m=2; A_i=1, \) and \( e(n) \) is a zero-mean, \( \sigma^2 \)-variance process. The example which is going to be our reference is given by

\[
x(n)=\cos(0.4\pi n)+\cos(0.2\pi n)+e(n)
\]  
(7.2)

for \( 0<n<N-1 \) where \( N \) being the record data length.

The PSDF associated with these cases is composed of \( 2m \) (i.e. 4) Dirac-delta impulses located at frequencies \( f_i (f_i=\omega_i/2\pi) \) riding on top of a constant value \( \sigma \), where \( f_1=0.1 \) and \( f_2=0.2 \) Hz. Depending on the underlying signal-to-noise ratios (SNR), the desired detection and frequency estimation will require certain precautions to be taken. For instance, if we consider computing \( \text{PBT} \) of a MA process of order \( q \), designated or as \( \text{MA}(q) \), or the AR PSD of order \( p \), \( \text{AR}(p) \), then \( q \) or \( p \) should be selected much larger than \( 2m \), (i.e., \( \gg4 \)). The SNR is defined as the ratio of the sinusoid power to the total power in the pass-band noise process.
\[ \text{SNR} = \frac{A_i}{\sigma} \]  

(7.3)

where \( A_i(rms) = A_i / \sqrt{2} \)

or, \[ \text{SNR(dB)} = 10 \log_{10}|\text{SNR}|^2 = 10 \log(1/2\sigma^2) \]  

(7.4)

where the case of \( A_i=1 \) is being taken care of (Table 7.1).

The PSD estimates for both \( P_{BT} \) and \( P_{Per} \) are plotted in fig. 7.1 and 7.2 respectively for the \( N=512 \) point data. Because of the symmetry in both spectra only half the data points will be used in plotting the PSD's so that instead of 4 Dirac pulses we will see only 2 at \( f_1 \) and \( f_2 \). The same situation will be followed with the AR and ARMA estimation. Also to be noted that the magnitude of the MA spectrum will be plotted, for easier comparisons with other spectra.

For the purpose of illustration, the MA(q) PSD as given by \( P_{BT} \) using q-autocorelation lags of the process will be discussed as a point of start with record length (\( N>10 \)) selected to be 128, 256, or 512. Examples of MA(4) at SNR of 9dB (fig. 7.3) show the disadvantage of using a low MA-order, it is unable to resolve the two sinusoids. The response around \( f=0.1 \) is as broad as the response at the high frequency end of the spectrum making it difficult to detect narrow-band components in a wide-band response. For \( q=12 \) at 3dB SNR the two dominant peaks are at \( f=0.085 \) and \( f=0.175 \) (fig. 7.4) but the response is still not well-defined and sharp. That is why we will consider the cases of
Table 7.1

<table>
<thead>
<tr>
<th>VARIANCE</th>
<th>SNR(DB)</th>
<th>STANDARD DEVIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.000</td>
<td>-20.000</td>
<td>7.071</td>
</tr>
<tr>
<td>39.716</td>
<td>-19.000</td>
<td>6.302</td>
</tr>
<tr>
<td>31.548</td>
<td>-18.000</td>
<td>5.617</td>
</tr>
<tr>
<td>25.059</td>
<td>-17.000</td>
<td>5.006</td>
</tr>
<tr>
<td>19.905</td>
<td>-16.000</td>
<td>4.462</td>
</tr>
<tr>
<td>15.811</td>
<td>-15.000</td>
<td>3.976</td>
</tr>
<tr>
<td>12.559</td>
<td>-14.000</td>
<td>3.544</td>
</tr>
<tr>
<td>9.976</td>
<td>-13.000</td>
<td>3.159</td>
</tr>
<tr>
<td>7.924</td>
<td>-12.000</td>
<td>2.815</td>
</tr>
<tr>
<td>6.295</td>
<td>-11.000</td>
<td>2.509</td>
</tr>
<tr>
<td>5.000</td>
<td>-10.000</td>
<td>2.236</td>
</tr>
<tr>
<td>3.972</td>
<td>-9.000</td>
<td>1.993</td>
</tr>
<tr>
<td>3.155</td>
<td>-8.000</td>
<td>1.776</td>
</tr>
<tr>
<td>2.506</td>
<td>-7.000</td>
<td>1.583</td>
</tr>
<tr>
<td>1.991</td>
<td>-6.000</td>
<td>1.411</td>
</tr>
<tr>
<td>1.581</td>
<td>-5.000</td>
<td>1.257</td>
</tr>
<tr>
<td>1.256</td>
<td>-4.000</td>
<td>1.121</td>
</tr>
<tr>
<td>0.998</td>
<td>-3.000</td>
<td>0.999</td>
</tr>
<tr>
<td>0.792</td>
<td>-2.000</td>
<td>0.890</td>
</tr>
<tr>
<td>0.629</td>
<td>-1.000</td>
<td>0.793</td>
</tr>
<tr>
<td>0.500</td>
<td>0.0</td>
<td>0.707</td>
</tr>
<tr>
<td>0.397</td>
<td>1.000</td>
<td>0.630</td>
</tr>
<tr>
<td>0.315</td>
<td>2.000</td>
<td>0.562</td>
</tr>
<tr>
<td>0.251</td>
<td>3.000</td>
<td>0.501</td>
</tr>
<tr>
<td>0.199</td>
<td>4.000</td>
<td>0.446</td>
</tr>
<tr>
<td>0.158</td>
<td>5.000</td>
<td>0.398</td>
</tr>
<tr>
<td>0.126</td>
<td>6.000</td>
<td>0.354</td>
</tr>
<tr>
<td>0.100</td>
<td>7.000</td>
<td>0.316</td>
</tr>
<tr>
<td>0.079</td>
<td>8.000</td>
<td>0.282</td>
</tr>
<tr>
<td>0.063</td>
<td>9.000</td>
<td>0.251</td>
</tr>
<tr>
<td>0.050</td>
<td>10.000</td>
<td>0.224</td>
</tr>
<tr>
<td>0.040</td>
<td>11.000</td>
<td>0.199</td>
</tr>
<tr>
<td>0.032</td>
<td>12.000</td>
<td>0.178</td>
</tr>
<tr>
<td>0.025</td>
<td>13.000</td>
<td>0.158</td>
</tr>
<tr>
<td>0.020</td>
<td>14.000</td>
<td>0.141</td>
</tr>
<tr>
<td>0.016</td>
<td>15.000</td>
<td>0.126</td>
</tr>
<tr>
<td>0.013</td>
<td>16.000</td>
<td>0.112</td>
</tr>
<tr>
<td>0.010</td>
<td>17.000</td>
<td>0.100</td>
</tr>
<tr>
<td>0.008</td>
<td>18.000</td>
<td>0.089</td>
</tr>
</tbody>
</table>
fig. 7.1

fig. 7.2
high orders with different SNR's. The MA(512) and MA(256) of 512-point data set yield identical PSD's at -10 dB (fig. 7.5,7.6). At 10 dB and 0 dB the MA(512) PSD's, of N=512, are remarkably similar with well-defined sharp peaks (fig. 7.7,7.8) while at -20 dB (fig. 7.9) the total resolution is lost and the spectral estimate exhibits a number of false peaks whose magnitudes are comparable to those of the two spectral peaks. For N=256 at -10 dB the MA(128) yields a poor spectral estimate with large number of slightly distinct peaks with relatively small amplitudes (fig. 7.10). The 512-point MA(128) PSD at 0 dB and the 256-point and 512-point at 10 dB PSD's are very much alike (fig. 7.11) where the peaks are well-defined and identifiable. In contrast, the MA(128) PSD's of 128-point at 10 and 0 dB SNR's are able to provide sharp response at one of the frequencies, $f=0.2$ (fig. 7.12). The MA(256) PSD of 256-point data set loses its spectral peaks to the neighbouring components of higher frequencies at -10 dB while at 0 dB it reveals them about $f=0.1$ and $f=0.2$ accompanied with few fluctuations of smaller magnitudes (fig. 7.13,7.14).

The periodogram, as implemented by the FFT, is used, next, in generating the spectral estimates for the same time series described earlier (7.2). It is known, and experimentally shown, that the periodogram does not exhibit sharp well-defined spectral peaks. Thus we will
choose values of N such as 128, 256, or 512. At low noise variances (0 to 10 dB) the periodograms produced good similar results; however, the higher the N the sharper the peaks (typical examples are shown in fig. 7.15 and 7.16). Increasing the noise variance (-10 dB) did not appreciably alter the results (fig. 7.17) but at -17 dB the spectrum became more complex where many spurious components appear with considerable amplitudes overriding those of the spectral peaks at some frequencies (fig. 7.18a, 7.18b).

In order to ease the potential ambiguity, created by the finite-frequency sampling (i.e., $\Delta \omega = 2\pi/N$), the concept of zero-padding and averaging is next incorporated. Using this approach, the original time series of length 256 will be appended with 256 zeroes, thus creating a 512-point data set (2*256). The resultant periodogram is then computed as the average of two 512-point padded FFT periodograms. Similarly, we could create 4 (4*128) or even 8 (8*64) 512-point padded FFT's of the same data set thus producing smoother spectra compared with original periodogram (fig. 7.19a,b,c). The smoothed periodograms could show identical performances, at 0 to 10 dB SNR, revealing well-defined sharp peaks at $f=0.1$ $f=0.2$ as can be seen from fig. 7.20a,b. The 2*256 averaged periodogram at 0 dB SNR looks smoother than the MA(256) PSD for record data length of 512. Similar cases are examined and found
identical as in MA(64) and MA(128) with 8*64- and 4*128-point periodogram, respectively. The Blackman-Tukey method at q=512 shows a better performance over the 512-point periodogram at 10dB and so is the case of MA(128) and MA(64) against the 4*128- and 8*64-point periodograms, respectively.

7.3 AR(p) AND ARMA(p,q) ESTIMATES

Windowing the data (P_{Per}) or autocorrelation lags (P_{BT}) makes an implicit assumption that the unobserved data or lag values are zero outside the window which is, normally, an unrealistic assumption. Use of a priori information may permit the selection of an exact model for the process that generated the data samples or, at least, a model that is a good approximation to the actual underlying process. In this context we need to select, first, a time series model (AR or ARMA), then, estimate the parameters of the assumed model, and finally obtain the spectral estimates by substituting the estimated model parameters into the theoretical PSD expression implied by the corresponding model. For clarification, the time series described previously will be spectrally investigated.

The AR spectral estimation, in essence, requires the determination of the error-prediction filter order, p. One approach is to try different values of p and then choose the optimal order which satisfies
some predetermined requirements. Small p's are shown to yield insufficient resolution in the spectral estimates, fig. 7.21a, whereas large values of p make the estimates statistically unstable, fig. 7.21b. Thus p is expected to be in the vicinity of a percentage of N, the data record length. Akaike's final prediction error (FPE) criterion was able to produce an optimal order of p=33, for N=512 (fig. 7.22a), and p=41, for N=1024 (fig. 7.22b). Good AR spectral estimates of lower orders, such as p=12 or 19, and higher orders, such as p=40, were also obtained, for low noise variances (0 to 10 dB), with very sharp peaks at the fractional sampling frequencies f=0.1 and f=0.2 (fig. 7.23a,b). At SNR=-10 dB the AR(12) spectrum was able to resolve the two spectral frequencies but with wider bandwidths exhibiting a "bump" at f=0.375 and a dc component (fig. 7.24a) while the AR(6) had a large bandwidth around f=0.16 and a dc component (fig. 7.24b).

To investigate the ARMA(p,q) spectral estimation, the p and q orders are to be chosen in accordance with L, the order of the equivalent AR model. It was experimentally shown that the best method of selecting the parameters is satisfying the inequality L>p+q. It was shown the ARMA(4,4) spectra exhibits a bandwidth centered at f=0.17 (fig. 7.25a) for L=12 or 18 while for L=40 a wider bandwidth could be seen at the same central frequency (fig. 7.25b). As a result choosing L
Fig. 7.27a
AR of M=33, Q=8
ARM ARMA of P=8
0 DB
0 1.0 2.0 3.0 4.0 5.0 6.0 7.0
NORM. FREQ.

Fig. 7.27b
AR of M=33, Q=8
+10 DB
0 1.0 2.0 3.0 4.0 5.0 6.0 7.0
NORM. FREQ.

Fig. 7.27c
AR of M=33, Q=8
+10 DB
0 1.0 2.0 3.0 4.0 5.0 6.0 7.0
NORM. FREQ.

Fig. 7.27d
AR of M=33, Q=8
0 DB
0 1.0 2.0 3.0 4.0 5.0 6.0 7.0
NORM. FREQ.
Linear time-invariant systems known as digital filters represent to a good extent the statistical time series of a random process. Modeling statistical time series and finding the parameters that identify them were the main theme of this thesis. Random processes under study were assumed weakly stationary in wide-sense and ergodic. Statistical or time averages of first- and second-order were then obtained to mathematically characterize the process. The second-order moment known also as the autocorrelation function is a time-domain description of the time series under analysis. Frequency-domain or spectral analysis was then applied where the power spectral density function was to be found using different digital signal processing techniques.

Classical and modern spectral analysis methods were presented. The periodogram (\(P_{\text{Per}}\)) and the Blackman-Tukey (\(P_{\text{BT}}\)) power spectra make part of the first category while the AR and ARMA power spectra which define the parametric models of the time series make part of the second category. As a demonstration, a time series consisting of sinusoids with white noise was used. A practical example was also obtained from an EMG data to show the performance of each technique and the advantage or disadvantage of one over the other.
The spectral estimates $P_{BT}$ and $P_{Per}$ are shown to be not identical. However, if the biased autocorrelation estimate (2.21) is used and as many as $q=N-1$ data samples are computed, then the $P_{BT}$ estimate and the periodogram spectral estimate yield identical results as can be seen from fig. 7.7 and 7.16 or 7.5 and 7.18a.

In spite of its efficient computation and reasonable results, the PSD based on the FFT has a limited performance when analyzing short data records manifested by the inability to distinguish the spectral responses of the two signals. The data is implicitly windowed when processing with FFT. The resolution is improved by smoothing the periodogram where the average of individual periodograms, performed on a segment of data, is being taken; the data is known to be zero-padded, or choosing an appropriate window would fulfill the job.

The PSD based on the Blackman–Tukey procedure is characterized by sidelobes, some of which produce negative values of the spectrum. Low MA orders makes it uneasy to identify the two sinusoidal responses. Upon increasing the MA-order, two reasonably well-defined peaks are revealed in the PSD estimate. This, in turn, has a drawback and that is because spectral estimation by the Blackman–Tukey method is computationally efficient if only a few lags are needed, which is not, usually, the case for better performance.

To avoid windowing the data ($P_{Per}$) or autocorrelation lags ($P_{BT}$)
we make use of a priori information which permit the selection of an exact model for the process that generated the data samples. In this context we needed to select, first, a time series model (AR or ARMA), then, estimate the parameters of the assumed model, and finally obtain the spectral estimates by substituting the estimated model parameters into the theoretical PSD expression implied by the corresponding model. The AR spectral estimation, in essence, requires the determination of the error-prediction filter order, \( p \). To investigate the ARMA\((p,q)\) spectral estimation, the \( p \) and \( q \) orders are to be chosen in accordance with \( L \), the order of the equivalent AR model. It was experimentally shown that the best method of selecting the parameters is satisfying the inequality \( LJp+q \).

Parametric techniques offer an improvements in resolution determined by the ability to fit an assumed model with few parameters to the measured data. When the model is an accurate representation of the data, spectral estimates can be obtained whose performance exceed that of the traditional BT or periodogram. The improvement in performance is manifested by higher resolution and a lack of sidelobes.

When the data characteristics are unknown the ARMA spectral estimates are more desirable than the AR spectral estimates. We were able to see from the different examples shown previously that the ARMA\((p,q)\) power spectrum is able to resolve the two sinusoids with fewer parame-
ters than the AR(p) case. The ARMA method used for determining the spectral estimates could show strong peaks with wide valleys unlike the AR(p) spectrum where only sharp narrow-banded peaks arise. It is to be noticed that the broad-band response of the MA spectrum, computed by the BT method, stands in contrast to the AR narrow-banded response. The resolution of the Yule-Walker estimated spectrum is in fact only a little better than that of an unsmoothed periodogram. The main difference between them is that the sidelobes evident in the periodogram spectrum are not present in the YW estimate. More accurate and efficient AR estimates based on Burg technique or maximum likelihood and maximum entropy methods could be found in [22] reference.

EMG signal is the electrical manifestation of the neuromuscular activity associated with a contracting muscle. It could described as a random signal modeled as a weighted AP's of MU's. Frequency will shift to the left of the spectrum during a sustained muscle contraction. It is observed that the shift in spectrum towards lower frequencies and the increase in the ME signal during a sustained contraction can be accounted by a single physiological correlate, the conduction velocity. The high performance of the parametric techniques was easily distinguished from the Fourier-based method, the periodogram. This could be examined from the smoothness of the AR and the ARMA spectra with the appropriate parameters selection of each method against the periodogram.
A table which shows the performance of each technique used in computing the EMG spectrum is listed below taking into consideration some characteristic values.

<table>
<thead>
<tr>
<th>Technique</th>
<th>3dB BW (Hz)</th>
<th>shape</th>
<th>f_peak (Hz)</th>
<th>operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periodogram</td>
<td>15.23</td>
<td>7 peaks in the 0-40 Hz region</td>
<td>21.17</td>
<td>N\log_2 N</td>
</tr>
<tr>
<td>Av. periodog.</td>
<td>28.56</td>
<td>Smoother in the 0-40 Hz region a peak at 75.88 another at 101.18</td>
<td>22.11</td>
<td>N\log_2 M</td>
</tr>
<tr>
<td>N = k*M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AR(8) p = 8</td>
<td>15.23</td>
<td>Smooth /Sharp</td>
<td>22.35</td>
<td>p^2</td>
</tr>
<tr>
<td>ARMA(8,8) p=q=6</td>
<td>15.23</td>
<td>Smooth /wide valleys</td>
<td>21.07</td>
<td>q \cdot p^2</td>
</tr>
</tbody>
</table>
APPENDIX

PROG. #1 (FFTH2.FOR)

THIS PROGRAM COMPUTES THE DFT OF A GIVEN SIGNAL USING THE FAST
FOURIER TRANSFORM ALGORITHM (FFT) AND FINDS THE PERIODограм
AS A METHOD OF DETERMINING THE PSD.

S(J): THE GIVEN SIGNAL AT INSTANT J
SS(J): THE COMPLEX FORM OF S(J)
X : ARRAY OF ABSCISSAS REPRESENTING THE SAMPLED FREQUENCY
Y : ARRAY OF ORDINATES REPRESENTING THE SPECTRAL AMPLITUDE
N : NUMBER OF DATA POINTS AS A POWER OF 2 (2**M)
M : EXPONENT OF 2
XR : SIMULATED WHITE NOISE OF ZERO MEAN AND VARIANCE GIVEN BY
VAR, WHERE STD IS THE STANDARD DEVIATION.

START

COMPLEX S(1024),XF(1024)
DIMENSION SS(1024),XL(12),YL(12),X(1024),Y(1024),IBUF(5000)
DATA XL/'F','R','E','Q','U','N','C','Y',3*' '/
DATA YL/'A','M','P','L','I','T','U','D',3*' '/

PI=3.1415926
M=9
N=2**M
VAR=0.5
STD=VAR**0.5
DO 2 J=1,N
CALL ANORM(XR)
SS(J)=1.00*COS(.4*PI*(J-1))+1.00*COS(.2*PI*(J-1))+STD*XR
1 FORMAT(5X,F12.5)
S(J)=CMPLX(SS(J),0.0)
2 CONTINUE

CALL THE SUBROUTINE FFT TO FIND THE SPECTRUM SSFFT

CALL FFT(S,M)
YT=0.0
DO 4 J=1,N
SFFT=CABS(S(J))
X(J)=(J-1)/FLOAT(N-1)
Y(J)=SFFT
C
C COMPUTE THE PERIODOGRAM, AND THEREBY FINDING THE PSD
C $P_{\text{per}} = \frac{1}{N} |SFFT|^2$
C
Y(J)=SFFT*SFFT/FLOAT(N)
WRITE(6,3) SS(J),XFFT
3 FORMAT(' ',F12.5,5X,F14.7)
IF (Y(J).GT.YT) YT=Y(J)
4 CONTINUE
DO 5 J=1,N
5 Y(J)=Y(J)/YT
NO2=N/2
C CALL MMPLOT(X,Y,NO2,1,XL,YL)
CALL ADMPLT(X,Y,NO2,1,XL,YL)
C
C INVOKE THE HP PLOTTER AND PLOT THE SPECTRUM
C
CALL PLOTS(IBMUF(5000),20000,11,-3)
CALL PLOT(1.0,1.0,-3)
CALL SCALE(X,8.0,NO2,1)
CALL SCALE(Y,6.0,NO2,1)
CALL AXIS(0.0,0.0,'FREQUENCY',-9,8.0,0.0,X(NO2+1),X(NO2+2))
CALL AXIS(0.0,0.0,'NORM. AMPLITUDE',15,6.0,90.0,Y(NO2+1),Y(NO2+2))
CALL PLOT(0.0,0.0,3)
CALL FLINE(X,Y,NO2,1,0,0)
CALL SYMBOL(4.0,5.0,0.18,'SNR = 0 DB',0.0,11)
CALL SYMBOL(4.5,7.25,0.176,'PERIODOGRAM N=512',0.0,18)
C CALL SYMBOL(4.5,7.25,0.176,'PERIODOGRAM N=512',0.0,18)
C CALL SYMBOL(3.5,7.25,0.176,'SNUSOID W/O NOISE',0.0,18)
C CALL SYMBOL(3.5,7.25,0.176,'EMG',0.0,5)
CALL PLOT(0.0,0.0,999)
STOP
C
C C LIST SUBROUTINE FFT
C
C PROG. #2 (FFTH5.FOR)
C
C THIS PROGRAM COMPUTES AND PLOTS THE SPECTRUM OF A DIGITAL SIGNAL
C USING THE AVERAGED PERIODOGRAM FOR SMOOTHING THE SPECTRUM
C
C DEFINING THE VARIABLES
C
C N : TOTAL NUMBER OF DATA GENERATED BY THE SIGNAL
C MP : INDEX
C K : NUMBER OF SEGMENTS USED TO SECTION THE DATA
C M : NUMBER OF DATA IN EACH SEGMENT
C N = K * M = 2 ** MP
C VAR : VARIANCE OF THE EMBEDDED WHITE NOISE
C STD : STANDARD DEVIATION, STD = VAR ** 0.5
C ANORM : GENERATES THE SIMULATED WHITE NOISE OF ZERO MEAN AND
C VARIANCE = VAR
C SS(N) : DATA GENERATED, GIVEN BY SIGNAL
C S(N) : COMPLEX FORM OF THE DATA SS(N)
C SI(M) : SEGMENTED DATA, KSEGMENTS OF M POINTS EACH
C
C COMPLEX XF(1024), S(1024)
DIMENSION SS(1024), IBUF(5000), XL(12), YL(12)
DIMENSION SI(1024), X(1024), Y(1024), YY(8, 512)
DATA XL / ' S' , ' A' , ' M' , ' P' , ' L' , ' E' , ' D' , ' F' , ' R' , ' Q' /
DATA YL / ' S' , ' P' , ' E' , ' C' , ' ' , ' D' , ' E' , ' N' , ' S' /
C
C GENERATE A SIGNAL COMPOSED OF TWO SINUSOIDS AUGMENTED WITH WHITE
C NOISE, OR READ FROM ANOTHER FILE THE EMG DATA
C
PI = 3.1415926
K1 = 2
I = 7
K = 2 ** K1
M = 2 ** I
MP = K1 + I
N = K * M
N = 2 ** MP
VAR = 0.05
STD = VAR ** 0.5
DO 22 J = 1, N
CALL ANORM(XR)
SS(J) = COS(.4 * (J - 1) * PI) + COS(.2 * (J - 1) * PI) + STD * XR
C READ (5, 1) SS(J)
1 FORMAT (5X, F12.5)
C WRITE (6, 11) J, SS(J)
11 FORMAT (1X, I4, 3X, F6.2)
CONTINUE

C FORM THE K SEGMENTS OF M POINTS THEN ZERO PAD THE REST
C OF N-M POINTS SO THAT WE STILL CAN COMPUTE THE FOURIER
C TRANSFORM OF N POINTS
C
DO 15 II=1,K
  DO 3 JJ=1,N
    IF(JJ.LE.M) GO TO 88
    SI(JJ)=0.0
  GO TO 80
88  SI(JJ)=SS(JJ+II*M-M)
80  S(JJ)=CMPLX(SI(JJ),0.0)
3  CONTINUE
C
C TAKE THE FOURIER TRANSFORM AND FIND THE SPECTRUM SFFT
C
CALL FFT(S,MP)
  DO 15 J=1,N
    SFFT=CABS(S(J))
15  CONTINUE
C
C COMPUTE THE INDIVIDUAL PERIODOGRAM, AND PRINT OUT THE RESULTS
C
    Y=(1/M)|SFFT|^2

Y(J)=SFFT*SFFT/M
YY(II,J)=Y(J)
JM1=J-1
C WRITE(6,8)JM1,SI(J),Y(J),YY(II,J)
8  FORMAT(//8X,I5,11.7,8X,F11.7,5X,11.7)
15  CONTINUE
C
C COMPUTE THE PSD AS THE AVERAGED PERIODOGRAM
C
    PSD=1/K @ Y
C
DO 23 J=1,N
  SUMP=0.0
  DO 21 I=1,K
    SUMP=SUMP+YY(I,J)
21  X(J)=(J-1)/FLOAT(N-1)
  Y(J)=SUMP/K
C WRITE(6,2) JM1,Y(J)
2  FORMAT(1X,I5,5X,F11.7)
23  CONTINUE
YT=0.0
DO 16 J=1,N
IF (Y(J).GT.YT) YT=Y(J)
16 CONTINUE
DO 10 J=1,N
10 Y(J)=Y(J)/YT
C
NO2=N/2
C WRITE(6,12) (X(J),Y(J),J=1,N)
12 FORMAT(5X,F5.0,5X,F11.7)
C CALL MMPLT(X,Y,NO2,1,XL,YL)
CALL ADMPLT(X,Y,NO2,1,XL,YL)
C
C PLOT THE SPECTRUM USING THE HP (CAL-COMP) PLOTTER
C
CALL PLOTS(IBUF(5000),20000,11,-1)
CALL PLOT(1.0,1.0,-3)
CALL SCALE(X,8.0,NO2,1)
CALL SCALE(Y,8.0,NO2,1)
C CALL AXIS(0.0,0.0,'NORM. FREQ.',-10,8.0,0.0,X(NO2+1),X(NO2+2))
C CALL AXIS(0.0,0.0,'NORM. AMPL.',10,6.0,90.0,Y(NO2+1),Y(NO2+2))
CALL PLOT(0.0,0.0,3)
CALL FLINE(X,Y,NO2,1,0,0)
CALL SYMBOL(2.5,6.5,0.18,'AV. PERIODGRAM',0.0,15)
CALL SYMBOL(2.5,6.0,0.15,'N=4*128',0.0,8)
CALL SYMBOL(2.5,5.5,0.15,'-10 DB SNR',0.0,10)
CALL PLOT(0.0,0.0,999)
STOP
END
C PROG. #3 (MA2.FOR)
C
C PROGRAM TO COMPUTE THE SPECTRAL DENSITY OF A MOVING-AVERAGE
C TIME SERIES BY THE BLACKMAN-TUKEY METHOD
C
N : DATA RECORD LENGTH
IQ : NUMBER OF AUTOCORRELATION LAGS, (MA PROCESS ORDER)
R(K) : AUTOCORRELATION ELEMENT AT LAG I
D : ARRAY OF DATA ELEMENTS
OMEGA : FREQUENCY IN RAD/SEC.
SPEC : SPECTRAL DENSITY
XR : RANDOM NUMBER GENERATOR OF VARIANCE VAR

C START
C
DIMENSION D(1026),R(1026),SPEC(1026),OMEGA(1026),XL(12),YL(12)
DIMENSION BUF(5000)
DATA XL,YL/12*' ',12*' '/
M=9
N=2**M
PI=3.1415926
C
DO 100 IQ=16,64,16
IQ=16
C
C GENERATE THE DATA D(N)
C
WRITE(6,1)
1 FORMAT(1X,' DATA POINTS',I1)
C
C TWO SINUSOIDS WITH WHITE NOISE
C
DO 3 I=1,N
CALL ANORM(XR)
D(I)=SIN(0.4*PI*(I-1))+SIN(0.43*PI*(I-1))+1.0*XR
C
C OR READ EMG DATA
C
READ (5,2) D(I)
WRITE (6,2) D(I)
2 FORMAT(5X,E12.5)
3 CONTINUE
C
C CALCULATE THE AUTOCORRELATION SEQUENCE
C
\[ R_x(k) = \frac{1}{(N-k)} \sum_{i=0}^{N-k} d(i)d(i+k) \]
WRITE(6,4)
FORMAT(1H1,1X,'AUTOCORRELATION COEFFICIENTS',/)
SS=0.0
DO 5 K=1,N
  SS=SS+D(K)*D(K)
  RO=SS/FLOAT(N)
C WRITE(6,6) RO
6 FORMAT(3X,'R(0)  = ',E12.5)
DO 9 I=1,N
  SUM=0.0
  NMI=N-I+1
  DO 7 J=1,NMI
    SUM=SUM+D(J)*D(J+I-1)
  R(I)=SUM/FLOAT(NMI)
C WRITE(6,8)I,R(I)
8 FORMAT(3X,'R(',I4,')= ',E12.5)
9 CONTINUE

C COMPUTE THE SPECTRAL DENSITY
C w(n)=1, rectangular window
WRITE(6,10) IQ
10 FORMAT(1H1//1X,'MA(Q) PROCESS FOR Q= ',I3//)
C WRITE(6,11)
11 FORMAT(1X,'FREQUENCY(RAD.),5X,'SPEC. DENS.(NORMALIZED)',//)
SI=0.0
DO 14 I=1,N
  OMEGA(I)=PI*(I-1)/FLOAT(N-1)
  S=0.0
  DO 12 J=I,IQ
    S=S+R(J)*COS(J*OMEGA(I))
  SPEC(I)=2*S+RO
  WRITE(6,13) OMEGA(I),SPEC(I)
13 FORMAT(2X,E12.5,6X,E12.5)
C
C COMPUTE MAXIMUM AMPLITUDE AND NORMALIZE
C
IF (SI.LT.SPEC(I)) SI=SPEC(I)
CONTINUE
WRITE(6,15) SI
15 FORMAT(//1X,'MAXIMUM AMPLITUDE ',E12.5)
DO 16 I=1,N
  SPEC(I)=SPEC(I)/SI
C
C PLOT THE SPECTRUM USING MMPLOT SUBROUTINE
CALL ADMLT(OMEGA,SPEC,N,1,XL,YL)

C PLOT THE SPECTRUM ON THE HP- PLOTTER USING CAL-COMP PACKAGE
C
XLE=8.0
YLE=7.0
CALL PLOTS(BUF,20000,11,-1)
CALL PLOT(1.0,1.0,-3)
CALL SCALE(OMEGA,XLE,N,1)
CALL SCALE(SPEC,YLE,N,1)
CALL AXIS(0.0,0.0,'FREQ./RAD',-9,XLE,0.0,OMEGA(N+1),OMEGA(N+2))
CALL AXIS(0.0,0.0,'AMPLITUDE',9,YLE,90.0,SPEC(N+1),SPEC(N+2))
CALL PLOT(0.0,0.0,3)
CALL LINE(OMEGA,SPEC,N,1,0,0)
CALL SYMBOL(3.0,8.0,0.17,'SPECTRAL DENSITY',0.0,16)
CALL SYMBOL(3.0,7.5,0.15,'USING MA PROCESS, Q=16',0.0,22)
CALL PLOT(0.0,0.0,999)
STOP
END
THIS PROGRAM FINDS THE POWER SPECTRAL DENSITY OF AN AUTOREGRESSIVE (AR) PROCESS USING THE YULE-WALKER METHOD. SOLUTION IS GIVEN BY LEVINSON'S RECURSION ALGORITHM.

S(N) : DATA OF RECORD LENGTH N
M : ERROR-PREDICTION FILTER ORDER, OR AR ORDER
R(I) : AUTOCORRELATION COEFFICIENT OF LAG I, THERE EXISTS M+1 LAGS
P : ERROR-PREDICTION FILTER MINIMUM POWER
A(I) : I-TH AR COEFFICIENT
FPE : FINAL PREDICTION ERROR ELEMENTS ARRAY
OMEGA : FREQUENCY IN RAD/SEC
Y : SPECTRAL DENSITY AS A FUNCTION OF OMEGA
RHO : REFLECTION COEFFICIENT

START

COMPLEX CB,CBI,CW,COMEGA(1024)
DIMENSION P(45),DELTA(45),A(45),S(1024),Y(1024),BUF(5000)
DIMENSION X(1024),R(45),RHO(45),AA(45),FPE(45)
DIMENSION XL(12),YL(12),OMEGA(1024)

PUT DATA S(N)

PI=3.14159265
MP=9
N=2**MP
VAR=0.05
STD=VAR**0.5
DO 200 J=1,N

IF DATA CONSISTS OF TWO SINUSOIDS WITH NOISE

CALL ANORM(XR)
S(J)=COS(.4*PI*(J-1))+COS(.2*PI*(J-1))+XR*STD

OR, READ EMG DATA

READ (5,23) S(J)
23 FORMAT(5X,E12.5)
WRITE(6,23) S(J)
200 CONTINUE

ASSIGN FILTER ORDER, M, AND COMPUTE THE M+1 AUTOCORRELATION LAGS
C
M=4
MN=M+1
C WRITE(6,9)
9 FORMAT(1H1//3X,'AUTOCORRELATION MATRIX COEFFICIENTS'/)
DO 300 L=1,MN
LM1=L-1
Q=0.0
K=N-L+1
DO 400 J=1,K
400 Q=Q+S(J)*S(J+L-1)
R(L)=Q/FLOAT(N)
C WRITE(6,50) LM1,R(L)
50 FORMAT(1X,'R(',I2,')=' ,E13.6)
300 CONTINUE
C WRITE(6,82) M
82 FORMAT(1H1,'COEFFICIENTS OF AR PROCESS OF ORDER' ,I3,18X,'FPE' #,22X,#'OUTPUT POWER'/)
C
C INITIALIZE LEVINSON'S ALGORITHM
C
A(1)=-R(2)/R(1)
AA(1)=A(1)
DELTA(1)=0.0
P(1)=R(1)-(R(2)*R(2)/R(1))
FPE(1)=P(1)*(N+2)/FLOAT(N-2)
C WRITE(6,72)
72 FORMAT(/2X,'ORDER = 1'
C WRITE(6,40)A(1),FPE(1),P(1)
40 FORMAT(20X,'A(1)= ',E13.6,15X,E13.6,15X,'P(1)= ',E13.7)
C
C START RECURSION ALGORITHM TO FIND THE AR COEFFICIENTS
C CORRESPONDING TO EACH ORDER L UP TO M
C
DO 12 L=2,M
12 WRITE(6,19) L
19 FORMAT(/2X,'ORDER =',I2)
SUM1=0.0
LL=L-1
DO 14 J=1,LL
14 SUM1=SUM1+R(L-J+1)*A(J)
DELTA(L)=SUM1+R(L+1)
RHO(L)=-DELTA(L)/P(L-1)
IF (RHO(L).GT.1.) GO TO 1000
DO 20 J=1,LL
AA(J)=A(J)+RHO(L)*A(L-J)
1000 CONTINUE
C
C
A(L)=RHO(L)
P(L)=P(L-1)+RHO(L)*DELTA(L)
C WRITE(6,121)J,AA(J)
121 FORMAT(/20X,'A(',12,')=' ,E13.6)
20 CONTINUE
DO 44 I=1,LL
A(I)=AA(I)
C IF (L.EQ.M) GO TO 75
C GO TO 44
C75 WRITE(6,122) I,A(I)
122 FORMAT(/20X,'A(',12,')=' ,E13.6)
44 CONTINUE
C IF (L.EQ.M) GO TO 76
GO TO 12
C76 WRITE(6,77)L,A(L),P(L),FPE(L)
77 FORMAT(/20X,'A(',12,')=' ,E13.6,2(15X,E13.6))
12 CONTINUE
C
C FIND THE OPTIMUM FILTER ORDER M(OPT) BY THE FPE CRITERION
C
LOC=1
DO 999 I=2,M
FPE(I)=P(I)*(N+1+1)/FLOAT(N-I-1)
IF (FPE(LOC).LT.FPE(I)) GO TO 999
LOC=I
999 CONTINUE
WRITE (6,534) FPE(LOC),LOC
534 FORMAT(///5X,'FPE(MIN)=' ,E13.6,5X,'CORRESPONDING TO AR(',I3,')')
WRITE(6,256)
256 FORMAT(lH1//6X,'INDEX' ,11X,'OMEGA' ,lOX, 'SPECTRAL DENSITY'/)
YN=0.0
C
C COMPUTE THE AR SPECTRAL DENSITY
C
DO 773 I=1,N
OMEGA(I)=2.0*PI*(I-1)/FLOAT(N-1)
COMEGA(I)=CMPLX(0.0,OMEGA(I))
CW=COMEGA(I)
CB1=(1.0,0.0)
DO 771 K=1,M
771 CB1=CB1+A(K)*CEXP(-CW*K))
ABSA=CABS(CB1)
SFA=ABSA*ABSA
Y(I)=P(M)/SFA
C
C NORMALIZE
C
IF (YN.LT.Y(I)) YN=Y(I)
X(I)=(I-1)/FLOAT(N-1)
C
WRITE(6,772) I,X(I),Y(I)
772 FORMAT(5X,I4,8X,E13.6,8X,E13.6)
773 CONTINUE
DO 775 K=1,N
775 Y(K)=Y(K)/YN
C
C PLOT THE AR SPECTRUM BY MMPLT OR ON A HP PLOTTER (CAL-CMP)
C
NO2=N/2
C
CALL MMPLT(X,Y,512,1,XL,YL)
C
CALL ADMPLT(X,Y,NO2,1,XL,YL)
C
XLE=7.0
YLE=7.0
CALL PLOTS(BUF,20000,11,-1)
CALL PLOT(1.0,1.0,-3)
CALL SCALE(X,XLE,NO2,1)
CALL SCALE(Y,YLE,NO2,1)
CALL AXIS(0.0,0.0,'FREQUENCY',-9,XLE,0.0,X(NO2+1),X(NO2+2))
C
CALL AXIS(0.0,0.0,'NORM. AMPL.' ,11,YLE,90.0,Y(NO2+1),Y(NO2+2))
CALL PLOT(0.0,0.0,3)
CALL LINE(X,Y,NO2,1,0,0)
CALL SYMBOL(3.0,9.0,0.15,'AR M=4 P=4 N=512 ',0.0,23)
CALL SYMBOL(3.0,6.5,0.15,'-10 DB',0.0,6)
CALL PLOT(0.0,0.0,999)
GO TO 1002
1000 WRITE(6,1001)
1001 FORMAT(//,2X,'FILTER ORDER IS INCONVINIENT',/)
1002 STOP
END
C PROG. #5 (ARMA5.FOR)

C THIS PROGRAM WILL FIND THE PSD OF AN AUTOREGRESSIVE-MOVING AVERAGE
C (ARMA) PROCESS BY THE EQUIVALENCE METHOD WITH AN AR PROCESS
C
C ELEMENTS OF THE FIRST PART UNTILL THE COMPUTING OF THE EQUIVALENT
C AR COEFFICIENTS ARE THE SAME AS WAS PROCEEDED IN PROG. #4 (AR1).
C
COMPLEX CB,CB1,CW,COMEGA(1024)
DIMENSION P(50),DELTA(50),A(50),S(1024),Y(1024)
DIMENSION X(1024),R(50),RHO(50),AA(50),T(40),AP(50),AM(50,15),
# AT(25,50),AD(40),APM(45,40),BUF(5000)
DIMENSION XL(12),YL(12),RC(25),E(25),OMEGA(1024),BQ(25)
COMMON/BLOCK1/AL(20,20),B(20),C(20),IQ
PI=3.14159265
MP=9
N=2**MP
VAR=0.05
STD=VAR**0.5
DO 200 J=1,N
C C READ (5,23) S(J)
CALL ANORM(XR)
S(J)=COS(0.4*PI*(J-1))+COS(0.2*PI*(J-1))+STD*XR
23 FORMAT(5X,E12.5)
C C WRITE(6,23) S(J)
200 CONTINUE
M=24
MN=M+1
IQ=4
IP=20
C C WRITE(6,9)
9 FORMAT(1H1//3X,'AUTOCORRELATION MATRIX COEFFICIENTS'//)
DO 300 L=1,MN
LM1=L-1
Q=0.0
K=N-L+1
DO 400 J=1,K
400 Q=Q+S(J)*S(J+L-1)
R(L)=Q/FLOAT(N)
C C WRITE(6,50) LM1,R(L)
50 FORMAT(1X,'R(',I2,')=' ,E13.6)
300 CONTINUE
C C WRITE(6,81)
81 FORMAT(1H1//25X,'A',29X,'DELTA',22X,'P',/)
WRITE(6,82) M
A(1) = -R(2)/R(1)
AA(1) = A(1)
DELTA(1) = 0.0
P(1) = R(1) - (R(2)*R(2)/R(1))

WRITE(6,72)

72 FORMAT(/2X,'ORDER = 1')
WRITE(6,40) A(1), DELTA(1), P(1)

40 FORMAT(20X,'A(1) = ',E13.6,15X,E13.6,15X,'P(1) = ',E13.7)

DO 12 L=2,M
WRITE(6,19) L

19 FORMAT(/2X,'ORDER = ',I2)
SUM1 = 0.0
LL = L-1
DO 14 J=1,LL
SUM1 = SUM1 + R(L-J+1)*A(J)
DELTA(L) = SUM1 + R(L+1)
RHO(L) = -DELTA(L)/P(L-1)
IF (RHO(L) .GT. 1.) GO TO 1000
DO 20 J=1,LL
AA(J) = A(J) + RHO(L)*A(L-J)
A(L) = RHO(L)
P(L) = P(L-1) + RHO(L)*DELTA(L)
WRITE(6,121) J, AA(J)

121 FORMAT(/20X,'A(',I2,') = ',E13.6)

20 CONTINUE
DO 44 I=1,LL
A(I) = AA(I)
IF (L .EQ. M) GO TO 75
GO TO 44

75 WRITE(6,122) I, A(I)

122 FORMAT(/20X,'A(',I2,') = ',E13.6)

44 CONTINUE
IF (L .EQ. M) GO TO 76
GO TO 12

76 WRITE(6,77)L, A(L), DELTA(L), L, P(L)

77 FORMAT(/20X,'A(',I2,') = ',E13.6,15X,E13.6,15X,'P(',I2,') = ',E13.6)

12 CONTINUE

C START COMPUTING THE ELEMENTS OF THE ARMA PROCESS (SECTION 5.2)
C IQ : ORDER OF THE MA PARAMETERS OF THE ARMA PROCESS
C IP : ORDER OF THE AR PARAMETERS OF THE ARMA PROCESS
C
IQ = 2
IP = 2

C SOLVE FOR B(Q) COEFFICIENTS
C
IP1=IP+1
ID=1
LD=M-ID
WRITE(6,320)
320 FORMAT(1H1)
DO 100 I=IP1,LD
  AD(I)=A(I)
  DO 100 J=1,IQ
  AM(I,J)=A(I-J)
100 CONTINUE
DO 55 I=1,IQ
  DO 55 J=IP1,LD
  AT(I,J)=AM(J,I)
55 CONTINUE
WRITE(6,54) ((AM(I,J),J=1,IQ),AD(I),I=IP1,LD)
54 FORMAT(32X,'COEFFICIENTS MATRIX A :',12X,'VECTOR AD :
#3(2X,F7.4),16X,F7.4)
WRITE(6,541) ((AT(I,J),J=IP1,LD),I=1,IQ)
541 FORMAT(IISX,'TRANSPOSE
MATRIX AT :
#3(2X,F7.4))
DO 501 I=1,IQ
  T(I)=0.0
  DO 500 K=1,LD
    AL(I,J)=AL(I,J)+AT(I,K)*AM(K,J)
 500 T(I)=T(I)+AT(I,K)*AD(K)
  C(I)=-1.*P(M)**0.5*T(I)
501 CONTINUE
WRITE(6,222)
222 FORMAT(/2X,'C(I)=-SIGMA*AT(I,J)*AD(J)''
WRITE(6,555) (I,C(I),I=1,IQ)
555 FORMAT(2X,'C(',12,')=',E13.6)
C********** SOLVE SYSTEM : BQ(I)=INVAL(I,J)*C(J)
C**********ELEMEINS OF SYSTEM MATRIX...
C**********AL(I,J)...
C**********ELEMEINS OF CONSTANTS VECTOR
C**********C(I)
C**********ORDER OF THE SYSTEM
C********** IQ
C
WRITE (6,1) ((AL(I,J),J=1,IQ),I=1,IQ)
1 FORMAT(/5X,'SYSTEM MATRIX AL(I,J)=AT(I,K)*A[K,J] : '///
#3(2X,E13.6))
WRITE(6,3)
3 FORMAT(/3X,'SOLUTION VECTOR ',15X,'RESULTANT VECTOR',7X,'ERROR
#VECTOR'////)
   BO=P(M)**0.5
WRITE(6,4)IQ,BO
4 FORMAT(/5X,I7,'_ORDER MA COEFFICIENTS OF ARMA PROCESS' ,//8X,
   '#BQ(I)=INV[AL(I,J)*C(I) ',//5X,'BQ(0)=' ,E13.6)
   CALL GAUSS
WRITE(6,32) (L,B(L),L=1,IQ)
32 FORMAT(5X,'BQ(' ,IZ,')=' ,E13.6)
   DO 31 I=1,IQ
   C SUM3=0.0
   DO 10 L=1,IQ
   C SUM3=SUM3+AL(I,L)*B(L)
   C RC(I)=SUM3
   C E(I)=ABS(C(I)-RC(I))
   C10 CONTINUE
   WRITE(6,21) RC(I),E(I)
21 FORMAT(34X,E13.6,10X,E13.6)
C
C FIND THE AR(P) COEFFICIENTS
C
   LQ1=LQ+1
   IQ1=IQ+1
   DO 150 I=1,LQ1
   DO 150 J=1,IQ1
   IF (I.GT.M) GO TO 255
   IF (I-J) 88,66,99
88 APM(I,J)=0.0
   GO TO 150
66 APM(I,J)=1.0
   GO TO 150
99 APM(I,J)=A(I-J)
   GO TO 150
255 CONTINUE
   IF (I-M-J) 51,61,71
71 APM(I,J)=0.0
   GO TO 150
61 APM(I,J)=A(M)
   GO TO 150
51 APM(I,J)=A(M-J+(I-M))
C51 APM(I,J)=A(I-J)
CONTINUE
WRITE(6,26)
FORMAT(1H1)
WRITE(6,551) ((APM(I,J),J=1,IQ1),I=1,LQ1)
551 FORMAT(/'APM MATRIX: ',///,4(2X,F7.4))
BQ(I)=BO
DO 35 I=2,IQ1
35 BQ(I)=B(I-1)
WRITE(6,45)IP
45 FORMAT(/'ORDER AR COEFFICIENTS OF ARMA PROCESS'/)
WRITE(6,46)
46 FORMAT(/'AP(I)=APM([I,J]*BQ(J)/SIGMA'/)
DO 135 I=1,LQ1
AP(I)=0.0
DO 134 J=1,IQ1
134 AP(I)=AP(I)+APM(I,J)*BQ(J)
AP(I)=AP(I)/(P(M)**0.5)
IF(I.GT.(IP+1)) AP(I)=0.0
IM1=I-1
WRITE(6,257) IM1,AP(I)
257 FORMAT(5X,'AP(',12,')=',E13.6)
135 CONTINUE
WRITE(6,256)
256 FORMAT(1H1//6X,'INDEX',11X,'OMEGA',1OX,'SPECTRAL DENSITY/)
C
C COMPUTE THE SPECTRAL DENSITY
C
YN=0.0
DO 773 I=1,N
OMEGA(I)=2.0*PI*(I-1)/FLOAT(N-1)
COMEGA(I)=CMPLX(0.0,OMEGA(I))
CW=COMEGA(I)
CB=CMPLX(B0,0.0)
DO 770 K=1,IQ
770 CB=CB+B(K)*CEXP(-CW*K)
ABSB=CABS(CB)
SFB=ABSB*ABSB
CBI=(1.0,0.0)
DO 771 K=2,IP1
771 CBI=CBI+AP(K)*CEXP(-CW*(K-1))
ABSA=CABS(CB1)
SFA=ABSA*ABSA
Y(I)=SFB/SFA
C
C NORMALIZE
C
IF (YN.LT.Y(I)) YN=Y(I)
X(I)=(I-1)/FLOAT(N-1)
CWRITE(6,772) I,X(I),Y(I)
772 FORMAT(5X,I4,8X,E13.6,8X,E13.6)
773 CONTINUE
DO 775 K=1,N
775 Y(K)=Y(K)/YN

C
C PLOT THE ARMA SPECTRUM (MPLOT OR CALCOMP PLOTTER)
C
NO2=N/2
C CALL MMPLOT(X,Y,NO2,1,XL,YL)
CALL ADMPLT(X,Y,NO2,1,XL,YL)
WRITE(6,722) M,IP,IQ
722 FORMAT(/5X,'M=',I2/5X,'P=',I2/5X,'Q=',I2//)
XLE=7.0
YLE=5.0
CALL PLOTS(BUF,20000,11,-1)
CALL PLOT(1.5,1.5,-3)
CALL SCALE(X,XLE,NO2,1)
CALL SCALE(Y,YLE,NO2,1)
CALL AXIS(0.0,0.0,'NORM. FREQ.',-11,XLE,0.0,X(NO2+1),X(NO2+2))
CALL AXIS(0.0,0.0,'NORM. AMPL.',11,YLE,90.0,Y(NO2+1),Y(NO2+2))
CALL PLOT(0.0,0.0,0.3)
CALL LINE(X,Y,NO2,1,0,0)
CALL SYMBOL(2.5,6.0,0.15,'AR(24)',0.0,5)
CALL SYMBOL(2.5,5.5,0.15,'ARMA(20,4)',0.0,11)
CALL SYMBOL(2.5,5.10,0.15,'-10 DB',0.0,6)
CALL PLOT(0.0,0.0,999)
GO TO 1002
1000 WRITE(6,1001)
1001 FORMAT(/,2X,'FILTER ORDER IS INCONVINIENT',/)
1002 STOP
END

C
C LIST SUBROUTINE GAUSS
C
C
C PROG. #6 (DECIMAL.FOR)
C
C THIS PROGRAM PERFORMS THE OPERATION OF TRANSFORMING TEXT DATA IN
C HEXADECIMAL FORMAT INTO A DECIMAL FORMAT.
C
C THE PROGRAM WILL READ FIRST THE INDIVIDUAL HEXADECIMAL NUMBERS FROM
C 0 TO F IN THE ARRAY A(16), THEN IT WILL READ THE ACTUAL DATA FROM
C THE TWO ARRAYS B(16) AND C(16) WHICH REPRESENTS THE 32 CHARACTERS
C FOUND IN EVERY DATA LINE BY CALLING THE SUBROUTINE BIN . LATER THE
C SUBROUTINE BTD WILL TRANSFORM THE DATA INTO BINARY AND FURTHERMORE
C INTO DECIMAL. THE OUTPUT DECIMAL DATA WILL APPEAR IN THE ARRAY Y(16)
C WHICH IS TAKEN FROM OUT(16) FOUND IN SUBROUTINE BIN AND COMPUTED
C VIA SUBROUTINE BTD.
C
C THE PROGRAM IS DESIGNED TO TRANSFORM OUTPUT VALUES TO +5 OR -5 VOLTS
C THAT VALUES FROM 0 TO 128 HAVE A HIGH OF 5 VOLTS, FROM 129 TO 256 WILL
C BE HAVING VALUES BETWEEN ) AND -5 VOLTS.
C
C START (MAIN PROGRAM)
C
C DIMENSION A(16),B(16),C(16),OUTO(16),OUT(16),Y(16)
READ(5,10) (A(I),I=1,16)
10 FORMAT(16Z1)
C WRITE(6,11) (A(I),I=1,16)
11 FORMAT(5X,16Z1)
DO 100 J=1,48
CALL BIN(A,OUT)
DO 13 I=1,16
Y(I)=OUT(I)
13 CONTINUE
WRITE(10,15) Y
15 FORMAT(5X,E12.5)
100 CONTINUE
STOP
END
C
C SUBROUTINE BIN
C
C SUBROUTINE BIN(A,OUT)
DIMENSION A(16),B(16),C(16),OUT(16)
READ(5,6) (B(N),C(N),N=1,16)
6 FORMAT(5X,32Z1)
CALL BTD(A,B,C,OUT).
RETURN
END
SUBROUTINE BTD

SUBROUTINE BTD(A,B,C,OUT)
DIMENSION OUT1(16),OUT0(16),OUT(16),A(16),B(16),C(16)
DO 11 N=1,16
   DO 1 I=1,16
      IF (A(I).NE.C(N)) GO TO 1
      OUT0(N)=FLOAT(I-1)
   1 CONTINUE
11 CONTINUE
   DO 22 N=1,16
      DO 2 I=1,16
         IF (A(I).NE.B(N)) GO TO 2
         OUT1(N)=FLOAT(I-1)
      2 CONTINUE
22 CONTINUE
   DO 3 I=1,16
      OUT(I)=16.0*OUT1(I)+OUT0(I)
      OUT(I)=(OUT(I)-128.0)*5.0/128.0
   3 CONTINUE
RETURN
END
LISTING OF THE SUBROUTINE FFT

SUBROUTINE FFT(XF,M)
COMPLEX XF(1024),U,W,T
N=2**M
N2=N/2
N1=N-1
J=1
DO 3 I=1,N1
  IF(I.GE.J) GO TO 1
  T=XF(J)
  XF(J)=XF(I)
  XF(I)=T
1  K=N2
2  IF(K.GE.J) GO TO 3
  J=J-K
  K=K/2
  GO TO 2
3  J=J+K
PI=3.141592653589793
DO 5 L=1,M
  LE=2**L
  LE1=LE/2
  U=(1.0,0.0)
  W=CMPLX(COS(PI/LE1),SIN(PI/LE1))
  DO 4 J=1,LE1
    DO 4 I=J,N,LE
      ID=I+LE1
      T=XF(ID)*U
      XF(ID)=XF(I)-T
      XF(I)=XF(I)+T
4    U=U*W
5  RETURN
END
C THIS IS SUBROUTINE GAUSS
C
SUBROUTINE GAUSS
COMMON/BLOCK1/AL(20,20),B(20),C(20),IQ
C******** ELIMINATE I-TH UNKNOWN
 NM=IQ-1
 DO 15 I=1,NM
C******** FIND LARGEST PIVOT
 AMAX=0.0
 DO 11 II=I,IQ
 IF (ABS(AL(II,I)).LT.AMAX) GO TO 11
 ISTAR=II
 AMAX=ABS(AL(II,I))
11 CONTINUE
C******** RETURN IF PIVOT IS TOO SMALL
 IF (AMAX.LT.(10**(-10))) GO TO 13
C******** SWITCH ROWS
 DO 17 J=I,IQ
 DUM=AL(ISTAR,J)
 AL(ISTAR,J)=AL(I,J)
17 AL(I,J)=DUM
 DUM=C(ISTAR)
 C(ISTAR)=C(I)
 C(I)=DUM
C******** PIVOT
 IP=I+1
 DO 15 II=IP,IQ
 PIVOT=AL(II,I)/AL(I,I)
 C(II)=C(II)-PIVOT*C(I)
 DO 15 J=1,IQ
15 AL(II,J)=AL(II,J)-PIVOT*AL(I,J)
C******** RETURN IF LAST PIVOT IS TOO SMALL
 IF (ABS(AL(IQ,IQ)).LT.(10**(-10))) GO TO 13
 B(IQ)=C(IQ)/AL(IQ,IQ)
C******** BACK SUBSTITUTE
 DO 41 K=1,NM
 L=IQ-K
 B(L)=C(L)
 LP=L+1
 DO 16 J=LP,IQ
16 B(L)=B(L)-AL(L,J)*B(J)
41 B(L)=B(L)/AL(L,L)
 GO TO 22
13 WRITE(6,18)
18 FORMAT(' ILL-CONDITIONED')
22 RETURN
END
REFERENCES


