١

Representation and Processing of EEG Signals Using ARMA Model

By

Raed Yousef Abu Hilal

Supervisor

Prof. Isam Zabalawi

Co-Supervisor

Dr. Souheil Odeh

Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Science in

Electrical Engineering / Communications

٩٩/١

١٤/١٦

University of Jordan

December 1998

This thesis was successfully defended and approved on 15/12/1998

Examination Committee	Signature
Prof. Isam Zabalawi	
Dr. Souheil Odeh	Souheil Odeh
Prof. Mohamed K. Abdelazeez	m. k. abdel
Dr. Mohamed Mismar	M. Misma
Dr. Mousa Abdallah	Abdall

DEDICATION

To my lovely father and mother

To my family

To my great love

Wafa'a

ACKNOWLEDGEMENT

With simple words, I would like to express my deep gratitude to those who helped me to complete this work successfully.

A great gratitude and respect to my supervisors Prof. Isam Zabalawi and Dr. Souheil Odeh, for their advice, support and efforts in helping me to finish my work as much perfect as possible. Also, I am extremely thankful to all the members of the Electrical Engineering Department.

Finally, I am so grateful to my lovely parents for their support, encouragement and continuous efforts in providing me with the comfortable atmosphere every time and during the preparation of this thesis.

TABLE OF CONTENTS

Appendices

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 INTRODUCTION

Because of the pioneering work by Berger 1929-1932, it became possible to record the brain activity by means of electrodes externally positioned upon the intact skull. The electrical activity in the cortical and subcortical layers of the brain gives rise to a potential at the surface of the scalp known as the electroencephalogram, or more simply, EEG. The EEG is a recording of the electrical activity of the brain, obtained by placing electrodes on various locations at the scalp. This activity may be due the collective activity of many nerve cells (called neurons) in a small area in the outer surface of the brain that is right under each electrode. When the potential in many neurons are changed synchronously, they create rhythms of various types.

An international system, the so called 10/20 system, has been defined for standardized symmetrical positioning of electrodes on the skull. To get a simultaneous survey of the activity in different parts of the brain, the activity of several electrodes is registered for 10 to 30 min by means of multichannel recorder with usually 8 to 16 channels.

The EEG, turned out to be a useful tool for studying the functional states of the brain, such as different states of wakefulness and sleeper metabolic disturbances, which, however, only emphasizes the importance of the EEG as a noninvasive diagnostic tool in wealth of neurological disorders, e.g., epilepsy, early diagnosis and localization of brain tumors. In addition, the EEG reflects abnormalities related to cerebrovascular disorders, infection diseases as well as metabolic endocrine disorders.

In the last few decades, progress has been made in extracting basic knowledge about brain functions from the brain electrical signals, and in developing clinical measures useful in diagnosing and treating several types of patient groups. The mostly used technique for analyzing EEG records still requires the visual examination and mental interpretation, with considerable importance placed on the detection of the abnormal patterns that emerge from the normal background. In most of these techniques, EEG recordings result in long traces that are difficult to analyze and interpret.

Over the years, EEG analysis have been conducted primarily in clinical settings, to detect the gross organic pathologies and the epilepsy. It have been shown that the cortical EEG patterns to be modified by a variety of variables which may or may not be pathological and as a result affect the EEG activity. Alternation of these variables produces a large variety of electrical patterns with different frequency and voltage characteristics. Therefore, this large variation in the electrical activity has limited the interpretation of the EEG to visual inspection by a trained electroencephalographer capable of distinguishing normal activity from localized or generalized abnormalities of particular types from relatively long EEG recorders. Thus, a complicating factor in using the EEG is the subjectivity with which it is interpreted.

Although much of the visual description has been standardized, not more than 60-80% concordance between pairs of assessors has been reported. This indicates that an objective method to quantify EEG characteristics could be of great value. So, several attempts have been made to reduce the EEG to more comprehensible form, many such methods have been developed for clinical use and are being used with varying success (Daskalova, 1988; Ferdjallah *et al.*, 1996; Goel *et al.*, 1996; Gvins, 1984; Isaksson *et al.*, 1981; Nakamura *et al.*, 1985).

Recently, a set of technical procedures involving the use of computer terminology in the recording, processing and analysis of brain electrical activity have been developed. In various forms these methods are referred as computerized EEG.

1.2 COMPUTERIZED EEG ANALYSIS

Computerized signal processing permits measurement and quantification of multiple aspects of brain electrical activity, thereby making available tremendous amounts of objective, precise information that is inaccessible by visual inspection of the EEG tracings. Computerized EEG analysis therefore supports and extends the evaluation of brain electrical activity by providing objective numerical data that can be used for graphical display, and mathematical and statistical analysis. Thus, the major thrust of quantitative EEG analysis was to produce an analytic equivalent of the expert neurologists visual assessment of strip chart tracing. The goals were to reduce the subjectivity of human judgments and to increase productivity.

Thus, because of the large amount of information which is present in the EEG signal, much of which cannot be adequately appreciated by visual inspection of analog tracings of the activity. Mathematical analysis of EEG tracings have been developed to provide correlations between neurophysiological events and its mathematical representation of electrical activity.

Improved method for quantitative analysis of signals of biological origin might be developed if the biophysical process underlying the generation of these signals were known in more detail. The use of parametric methods including linear models for EEG analysis seems to imply that such knowledge available.

Parametric modeling is a technique for time series analysis in which a mathematical model is fitted to a sampled signal. If the model forms a good approximation to the signals observed behavior it can then be used in wide range of applications, such as spectral estimation, speech analysis, and feature analysis for pattern classification problems.

The mathematical model that is most widely used is a rational transfer function, the exact form of which is determined by estimating suitable values for its parameters. If all these parameters lie in the transfer function's denominator then the model is termed an all-pole or Autoregressive (AR) model, while an all-zero or Moving-Average (MA) model has all of its parameters in the numerator. A model with parameters in both the numerator and denominator is termed as pole-zero or Autoregressive Moving-Average (ARMA) model.

In this thesis, we will be concerned in studying the ARMA model as a parametric method for representing and analyzing EEG signals. In the biomedical field, ARMA models are found to be a powerful tool in modeling and analyzing EEG (Cerutti *et al.*, 1988; Jansen, 1985; Korenberg *et al.*, 1989; Paarmann *et al.*, 1987).

The ARMA modeling technique can be formulated either in the frequency domain as a spectral matching problem or in the time domain as a linear prediction problem. To analyze EEG using ARMA modeling, a linear filter is given to describe the generating process of EEG. This filter is fed with white noise, and the output EEG signal has a spectral density depending upon the properties of the linear filter. The statistical of this output signal can be determined by using the regression of the signal upon itself. The linear prediction technique predicts the current output from linear combination of its past values and present and past values of the hypothetical input (Makhoul, 1975).

ARMA modeling have been successfully used for EEG signal analysis. They are used for finding out the similarity between mathematical models and biophysical models, for EEG segmentation and classification, and sharp transient detection. Furthermore, they can also be used to estimate the power spectrum of the EEG signals (Cadzow, 1982; Jansen, 1985; Kay *et al.*, 1981).

Thus, ARMA model represents a description of the EEG in a convenient form. Since the EEG signals can be described by frequency-related properties (rhythmic activities) which are called alpha, beta, and delta activities, ARMA model is used for this purpose to represent the rhythmic properties of the EEG signals. Its parameters have to be interpreted in terms which are familiar to the neurophysiologist.

The major aspect of ARMA modeling in analyzing EEG is the spectral analysis in which the EEG signals can be broken down into separate components, each component having a different frequency (Kay, 1988; Marple, 1987; Smith *et al.*, 1986). This decomposition of the waveform results in a frequency spectrum, which yields a distribution of amplitudes as a function of frequency for a given sample of EEG data. Therefore, the ARMA model is used to analyze the EEG signals leading to an estimation of the power spectral density of the signal.

1.3 THESIS CONTRIBUTIONS

In this thesis, the ARMA model is used to provide a quantitative analysis for the EEG signals. Utilizing the ARMA model, the following are achieved according to the EEG signals:

1- Data reduction and feature extraction. The EEG signals are represented with limit numbers of parameter using the ARMA model. We are interested in finding the suitable model order which can represent the EEG signals efficiently. This purpose is achieved by a new method for order estimation designated as Eigenvalue method (EV). This method is found to be superior to the previous methods. Utilizing this method it is found that the ARMA model can represent different types of simulated EEG signals with model order less than that which is required by the AR model to represent the same types of the EEG signals.

After estimating the suitable ARMA model order to represent the EEG signals, the parameters of the ARMA model are interpreted in such away that an important features , namely, frequency, bandwidth and power , that characterize the EEG signal are extracted from them using the Spectral Parameter Analysis (SPA) technique.

2- Estimating the Power Spectral Density (PSD) of the EEG signal. Using the ARMA model, it is possible to derive the PSD of the EEG signal efficiently from its parameters. In some cases, and due to the mixed-phase nature of the EEG signals, an inaccurate power spectral estimation is obtained, since some of the zeros of the ARMA model lie outside the unit circle. This problem has been solved by firstly converting the EEG signal to a minimum phase one by means of homomorhpic deconvolution then applying the ARMA model on the resulted signal to obtain the power spectral density. In this way an accurate power spectral estimation is achieved using the ARMA model.

3- Spike detection. It is found that the ARMA model can be used to detect EEG nonstationarities such as spikes when using it in the inverted form as a linear prediction error filter. The spikes are appeared in the error signal at the output of the prediction error filter and they are detected using special thresholding technique.

1.4 THESIS OUTLINE

This thesis aims at analyzing the EEG signals using the ARMA model as a parametric method. The outline of the thesis is as follows:

In Chapter 2, a general background of the EEG signals and its important characteristics is presented. In Chapter 3, the general concepts of the ARMA modeling are discussed. A general description about the ARMA process; its generation, the techniques used to estimate the model parameters, the estimation of the power spectral density, and the use of the ARMA technique in modeling both the stationary and the nonstationary signals are given in this chapter. This chapter also includes new approaches dealing with the ARMA models. It involves, a new method for ARMA order estimation based on eigenvalues of the covariance matrix. Besides, a new ARMA model dealing with systems with multiple inputs and delays is developed. At the end of this chapter, a technique for simulating EEG signals is presented.

In Chapter 4, the application of the ARMA modeling in analyzing the EEG signals is discussed. It shows how the important features of the EEG signals can be extracted using the ARMA model by means of spectral parameter analysis. A method for applying ARMA spectral modeling of the EEG in combination with homomorphic filtering is also considered in this chapter. This method models the minimum phase equivalent signal of the

EEG. Furthermore, this chapter includes a method that utilizes the ARMA model in detecting spikes. Spikes are described as sharp waves or transients present in a slowly varying background signal. This method uses the ARMA model in the inverted form to produce a linear prediction error filter to use it as spike detection technique.

Finally, the main conclusions that have been obtained from the simulation results of the various techniques presented in this thesis are summarized and discussed in Chapter 5.

CHAPTER 2

CONCEPTS OF EEG SIGNALS

In this chapter, the important characteristics and the basic concepts of the EEG signals are presented. The contents of this chapter were summarized from several texts (Binnie *et al.*, 1982; Cooper *et al.*, 1980; Daly *et al.*, 1990; Henry, 1980; Hughes, 1982; Tyner *et al.*, 1983). This chapter is presented to provide the reader with the minimum amount of information on the basic EEG principles that enables him/her to make a reasonable background about the EEG signals.

2.1 EEG TERMINOLOGY

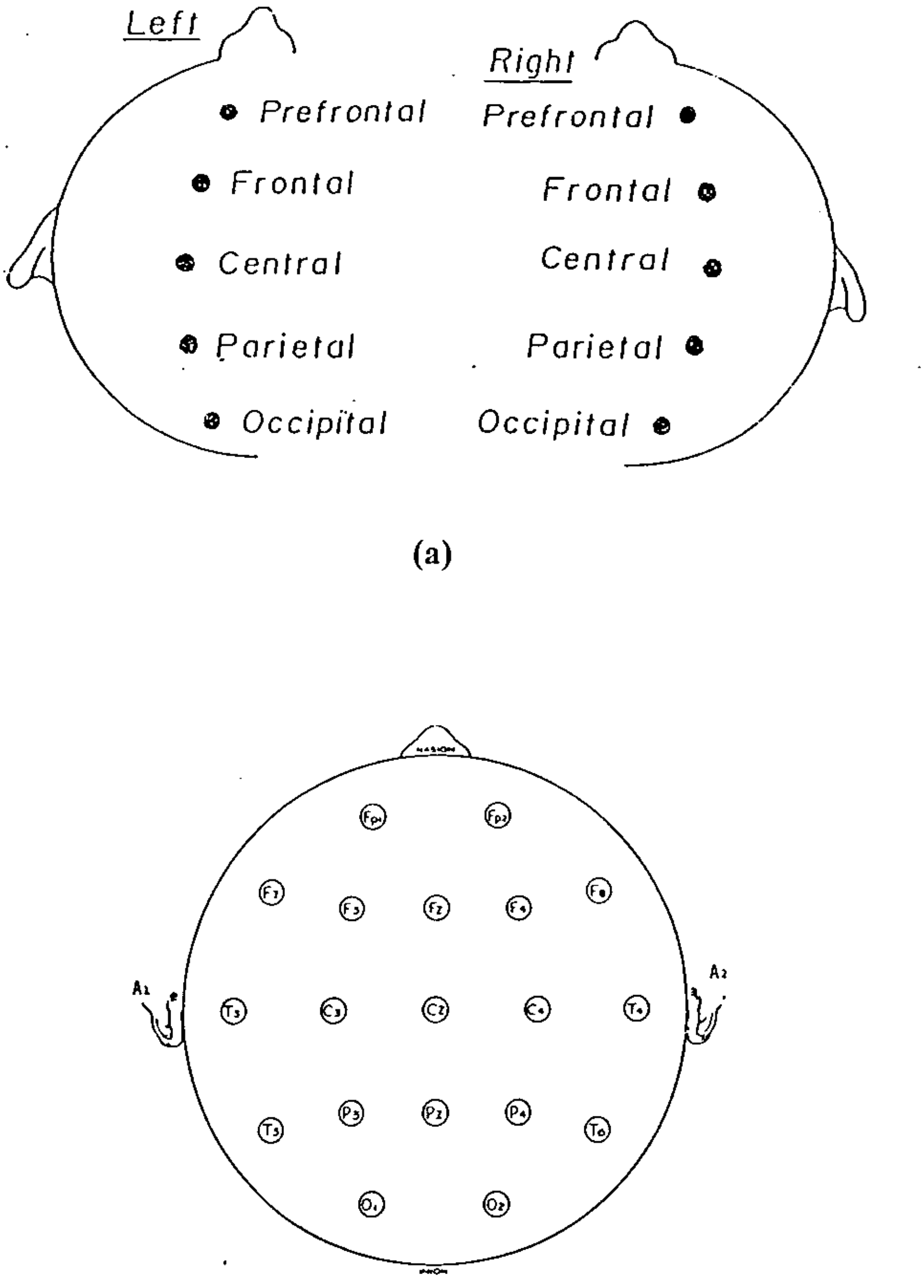
The discovery that it was possible to record the electrical activity of brain cells, the EEG, through the intact scalp of the human being appeared to offer fascinating possibilities for understanding how the brain works in healthy and/or sick human being. EEG is a recording of the electrical activity of the brain generated by the cell membranes of every living tissue.

The EEG consists of changing potential difference between one region and another of the scalp; these are generally of low amplitude. The activity recorded differs from one region of the scalp to another and it is therefore necessary, to record simultaneously from 20 or more electrodes distributed widely over the head.

The well-known 10-20 international system of electrode placement is used for this purpose. The resulting electrical activity from the electrodes describe the EEG signals of the brain in the various locations of the scalp. Each scalp location has a certain name as seen in Fig. 2.1(a). As a result, the electrodes have identifying names according to their position on the scalp: those on the left side have odd numbers, those on the right have even numbers, those near the midline have smaller numbers, and those more lateral have larger numbers.

The name includes the first letter of the general area where the electrode is placed. This is obvious in Fig. 2.1(b).

For descriptive purpose, the EEG signals may be divided into ongoing ‘background activity’ and episodic events or ‘transients’ which appear suddenly and are of relatively short duration. Thus, the analysis of EEG should be systematic process involving a series of orderly steps to characterize the electrical activity of the brain in terms of specific descriptors and measurements. Therefore, the EEG activity is described in terms of several characteristics, each of which must be considered in relation to the patient’s age and state. The expression state refers to the patient’s level of responsiveness (e.g., relaxed, awake, asleep and so on). These characteristics are:



(a)

F_p 1,2 = prefrontal
F3,4 = frontal
C3,4 = central
P3,4 = parietal
O1,2 = occipital
F7,8 = anterior temporal records rhythms from that general region but placed on frontal bone

T3,4 = mid-temporal
T5,6 = posterior temporal
A1,2 = ear (or mastoid)
Fz = front vertex
Cz = central vertex
Pz = parietal vertex
(note: z = zero)

(b)

Fig. 2.1 (a) Different locations on the scalp and their corresponding names.
(b) Position of the electrodes on the scalp.

1- Frequency:

Frequency refers to rhythmic repetitive EEG activity in Hz or the equivalent frequency of isolated single waves. Three terms are used relative to the frequency:

a) *Rhythmic:* EEG activity consisting of waves of approximately constant frequency.

b) *Arrhythmic:* EEG activity in which no stable rhythms are present.

c) *Dysrhythmic:* It refers to rhythms and/or patterns of EEG activity that characteristically appear in sick groups and rarely appear in healthy group.

2- Voltage:

When stating the amplitude (voltage) of a rhythm or other component, it is usual to refer to the peak-to-peak value in μV. In a given record one is more often concerned with the relative amplitudes either of one component with respect to another, or of one channel with respect to another. Three terms relating to the voltage are:

a) *Attenuation*: Reduction of amplitude of EEG activity resulting from decreased voltage. When activity is attenuated by stimulation, it is said to have been 'blocked'.

b) *Hypersynchrony:* Seen as an increase in voltage and regularity of rhythmic activity.

c) *Paroxysmal:* Activity occurring with sudden onset and termination and showing higher amplitude than the background.

3- Waveform:

It refers to the shape of the signal. The shape of a wave of an EEG activity is determined by the frequencies that combine to make up the waveforms and by their phase and voltage relationships. There are several terms associated with the waveform and they are:

a) *Monomorphic:* Distinct EEG activity appearing to be composed of one dominant frequency.

b) *Polymorphic:* Distinct EEG activity composed of multiple frequencies that combine to form a complex waveform.

c) *Sinusoidal:* Waves resembling sine waves.

d) *Transient:* An isolated wave or pattern that is distinctly different from background activity. Two terms relating to transient are:

i. *Spike:* a transient with a pointed peak and a duration from 20 to 70 msec.

ii. *Sharp wave:* a transient with a pointed peak and a duration from 70 to 200 msec.

e) *Complex:* A sequence of two or more waves, not necessarily of the same frequency, with distinct form or pattern from background activity, for example:

i. Spike-and-wave complex.

ii. Sharp-and-slow-wave complex.

4- Quantity:

Quantity refers to the amount of a particular type of EEG activity with respect to percent time present and/or to voltage. Four terms are related to the quantity are:

a) *Continuous:* Occurring without interruption. This term implies lack of marked frequency or voltage changes.

b) *Discontinuous* (or intermittent): Refers to patterns where there are changes of frequencies/voltages in the time domain appearing from time to time. The term carries only a descriptive message. In practice, several EEG patterns fall within this category.

c) *Regular:* Refers to the smoothness of the envelope of the waxing and waning of voltage that the EEG activity typically shows.

5- Location:

Location refers to scalp distribution of EEG rhythms or patterns, it implies cerebral location. Terms relating to the location are:

a) *Generalized:* Not limited to a specific area.

b) *Lateral:* Coming from one side, it is sometimes described as asymmetrical.

c) *Bilateral:* Coming from both sides, it is sometimes referred to as symmetrical.

d) *Focal:* Coming from a local area.

e) *Symmetry:* Equal distribution of EEG activity over homologous head areas.

6- Reactivity:

Reactivity refers to the reaction of the EEG to stimulation, stimulus-related change in morphology (waveform), and various physiological changes such as eye opening, mental calculation and movement.

7- Synchrony:

Synchrony is the simultaneous appearance of rhythmic or morphologically distinct patterns over different regions of the head, either on the same side or on both sides.

From the above discussion, we can say that the features of the EEG signals may be classified into waves, activities, rhythms and complexes, and that each feature should be described in terms of its frequency, amplitude, quantity, waveform and reactivity. A cursory glance through a small selection of EEG records will show that they contain components of three basic kinds: those that are fairly continuous and very often rhythmical, those that are transient and those that comprise the background activity, upon which the two preceding kinds are superimposed, either singly or together. Whilst background activity may be broken down by analysis into a spectrum of many frequencies which may be of clinical significance, a statement of frequency can be usefully made only of regular repetitive phenomena of which at least two complete cycles are present. Transient discharges are better described in terms of their duration and waveform.

As a matter of convenience, the EEG frequency spectrum is divided into bands that are designated as follows:

Delta (δ): less than 4 Hz.

Theta (θ): from 4 Hz to less than 8 Hz.

Alpha (α): from 8 Hz to 13 Hz.

Beta (β): more than 13 Hz.

Strictly speaking, these terms should be reserved for rhythmical discharges, but they are also used to describe irregular, non-rhythmical activity, the basic frequencies of which fall into a particular range. In general, they may be used irrespective of where the activity in question occurs. The only exception is the term *alpha rhythm* which refers to a particular phenomenon.

EEG signals are sometimes described as fast or slow, these terms imply that their dominant frequency is respectively above or below the alpha range. A polyrhythmic record is one in which two or more clear rhythmic components are present, whereas the term p*olymorphic* refers to irregular activity the individual waves of which are of variable period.

2.2 NORMAL VERSOUS ABNORMAL EEG ACTIVITY

The goal of clinical EEG is to make a useful diagnostic evaluation of particular patient. Because the technologist views the ongoing EEG activity as it is recorded, it is essential that he/she becomes familiar with and recognize the major features that appear in the ongoing record. On the basis of the features that do or do not appear, the recorded EEG signals may be classified as normal or abnormal EEG signals.

Normal EEG activity is any of the many kinds of distinctive activities that, qualitatively and quantitatively, appear in EEG signals of healthy groups of persons known to have no complaints, no neurological or other diseases. Abnormal EEG activity is that one which appears in groups of persons defined by specific complaints or specific neurological disease and does not appear in the healthy group.

To understand and recognize the pathological features, one must first get to know the normal patterns and their variability. The EEG changes continually, both in random manner, in association with changes in mental activity and with drowsiness or sleep. It also changes gradually over the lifetime of the individual and shows marked difference between one person and another. In adults, the limits of normal variation may be learned with no great difficulty, but in children the patterns are more varied. Because the EEG signals of the adults are the least complex in terms of variability from person to person, in both the waking and sleeping states, we will emphasize on EEG of the adults in these two states.

2.2.1 FEATURES OF THE NORMAL EEG

Before the various features of the normal EEG of adults are described, it is important to recognize that the identification of a particular activity or phenomenon may depend on its reactivity. An important element of the recording and its analysis is the testing of the reactions, or responses, of the features of the EEG to various physiological changes. These include eye opening and closing, repetitive movements of exterminates and sensory stimulation.

Normal EEG rhythms are found at different ages of the patients and also under different conditions. Usually, there is one dominant frequency, one which is the most prominent or obvious in the record, and this is called the 'background rhythm':

- Background rhythm in wake is equal to 8-10 Hz in adults (alpha rhythm).
- Background rhythm in sleep is equal to 5-6 Hz in light sleep (theta rhythm) and 2-3 Hz in deep sleep (delta rhythm).

Background rhythm can be considered as a general indication of the excitability of the central nervous system. It has the following features:

1) Alpha rhythm:

The most easily recognized activity in the waking EEG of the normal, relaxed adult whose eyes are closed is the alpha rhythm, prominent in the occipital region. It has a frequency ranging from 8 to 13 Hz, but averages 10 Hz. The average alpha voltage falls between 20 and 60 μV.

Alpha activity is usually rhythmic and is sometimes referred to as sinusoidal. This does not mean that an alpha wave is a sine wave, it only means that it resembles a sine wave.

Alpha rhythm may be entirely suppressed 'blocked' or at least attenuated when the subject opens his eyes. Alpha blocking is also produced by mental arithmetic and other task. Nevertheless, in most clinical contexts, when recording from conventionally spaced scalp electrodes, it suffices to regard the alpha rhythm as monorhythmic, although its frequency may be observed to vary about the mean by ± 0.5 Hz in normal persons.

A characteristic of the occipital alpha rhythm is that it may show what seems to be abrupt phase reversals, so that the result wave or waves have a frequency of half that of the ongoing alpha activity and in some instances have a greater amplitude. This concept is called *alpha variants* and refers to a mixture of an alpha rhythm and another rhythm that is usually half the frequency of the alpha rhythm, in this case it is called *slow alpha variant.* The slower activity is dominant, so that the typical alpha appears superimposed on it. In cases, less commonly alpha rhythm is replaced by activity within the beta frequency band and produce what is called *fast alpha variant.*

2) Beta rhythm:

Beta activity is defined as activity with a frequency higher than 13 Hz. Classically, it refers to more or less rhythmic activity in the 18 to 30 Hz frequency range. The voltage is not over 20 μV. It is present in nearly all normal persons.

On the basis of observed distributions and reactivity of beta activity, it has been inferred that there are three types: a) generalized, b) precentral, and c) posterior. It is commonly seen in the frontal and central regions.

Beta activity, particularly the generalized type, is bilaterally synchronous. Central beta activity may show symmetric features, and it may be reactive to movement-generated and stimuli, attenuating coincidentally with voluntary movement.

3) Theta rhythm:

Theta activity includes frequencies between 4 to 8 Hz. In the normal awake adult, small amount may appear in the central, temporal, and parietal areas, but it rarely is rhythmic. Its voltage is less than 15 μV. Theta activity in the temporal area is likely to be reduced by eye opening and may be quite asymmetrical, and that in the central areas are usually uninfluenced by eye opening but are enhanced during drowsiness. If the amplitude of the theta activity exceeds alpha amplitude by 50% or more, it may have pathological significance.

4) Delta rhythm:

Delta activity consists of rhythmic components below 4 Hz or isolated waves of more than 250 msec duration, it is not a usual feature of the awake adult EEG. Small amounts of delta activity may, however, be seen over the posterior temporal regions in young adults and represent the persistence of component which is conspicuous in childhood.

Sometimes delta activity is found superimposed with activity to form a pattern or a complex that resembles sharp-and-slow-wave activity. These waves are maximal in the posterior region with an amplitude ranging from 75 to 150 μV. These waves are often bilateral but asymmetric with greater amplitudes over the right hemisphere. Therefore they are called *posterior slow waves*. This phenomenon is not pathologically significant.

5) Mu rhythm:

The mu rhythm, a central rhythm of alpha activity frequency in which the individual waves have an arch-like shape, is present as a visually detectable rhythm in young adults. Its frequency is about 9 Hz but may range from 7 to 11 Hz. The voltage characteristics of the mu rhythm resemble those of the occipital alpha rhythm and are usually less than 100 μV.

Although the frequency may be within the alpha range, it is not an alpha rhythm. This is because its distribution is different and it has a unique reactivity. It does not block with eye opening but typically is blocked by movement or by tactile stimulation.

The mu rhythm may be either bilateral or unilateral. If bilateral, it often occurs in independent runs on either side of the scalp. It is commonly asymmetric in amplitude, but when present bilaterally it should have roughly the same frequency in both sides of the scalp.

6) Lambda waves:

Lambda waves are sharp waves with an equivalent frequency of 4 to 6 Hz and amplitude up to 20 µV. They occur in the occipital areas, but sometimes they may extend into the parietal regions. These waves are best seen when the person is actively engaged in looking at something that arouses his/her interest. They are not seen when the person's eye are closed.

Lambda waves have a waveform that is saw-toothed rather than spike-like, and the initial positive phase may be preceded and sometimes followed by a smaller one of opposite polarity.

7) Normal EEG activity in sleep

Dramatic events take place in the EEG when a person's state changes from relaxed waking activity to what is loosely called sleep. These events occur beginning with the wake state and continuing through the stages of sleep. During sleep the EEG follows a very constant pattern and the depth of sleep is classified on the basis of the EEG. There are five distinct sleep stages that can be identified as follows:

Stage 1:

Stage 1 usually called drowsiness. It is characterized by a decrease in voltage of the alpha rhythm in such a way it often disappears (attenuates) and then reappears several times during the initial period of drowsiness. Besides, a series of surface positive transients may emerge.

Stage 2:

As sleep continues to deepen in stage 2, the next distinct change involves appearance of a new 11 to 15 Hz rhythm (typically 14 Hz), which has a predominantly central vertex distribution. This activity appears in spindle-like bursts of rhythmic waves, which first increase, then decrease in voltage. Hence, they are called sleep spindles. The voltage of spindles may range up to 150 μV, although it is usually between 20 to 100 μV when measured in the central areas.

Stage 3:

Stage 3 differs quantitatively from stage 2. Delta activity below 1 Hz and greater in amplitude than 100 μV is present 20% to 50% of the time. Sleep spindles typically persist in this stage, although to lesser degree.

Stage 4:

In this stage, delta activity is present more than 50% of the time, with voltage greater than 75 μV. In addition, sleep spindles are present with slower frequency (e.g., 10 Hz rather than 14 Hz).

Stage 5:

This stage of sleep is defined not only by a change in EEG background activity, but also by the concomitant appearance of rapid eye movements

(REM). Hence, this stage is called REM sleep. The background activity has a variable appearance but typically is made up of theta and alpha components of low amplitude, sometimes with central theta bursts.

2.3 ABNORMAL EEG ACTIVITY

As indicated in the previous section, the normal EEG changes with age and with state of awakeness and in any case the limits of variation are broad. Abnormalities may appear as changes in the composition of background activity or as transient phenomena and may be generalized, unilateral or focal over a small region of the scalp. We previously stated that a definition of abnormal activity involves a comparison of EEG characteristics observed in groups of patients with neurological or other disease with those of healthy groups.

For an EEG to be considered abnormal, one or more of the following must be met:

1- The frequencies of basic rhythms are either faster or slower than that in the healthy groups. Sometimes normal rhythms are slowed or replaced by activity of lower frequency, this change may be diffused or localized. For instance, the alpha rhythm is replaced by rhythmic sinusoidal activity in the upper theta range.

2- The voltage of a particular EEG activity is either higher or lower than that in the healthy groups.

3- The locations of the various EEG activities are different than those in the healthy groups. As the normal EEG is fairly symmetrical, unilateral or more localized depression of normal activities is easily recognized and is one of the most reliable EEG signs of disease.

4- Unusual distinctive elements with a periodic character may appear such as slow waves and spike which are described in the following:

a) Slow waves: They are rhythms, appearing especially during wakefulness, that are slower than in the normal. Thus, an adult should not have theta or delta patterns in the waking record and if they appear, they are called slow waves abnormalities. Usually, the slower the frequency and the more often it appears, the greater the degree of abnormality. Abnormal slow waves appear when the brain cells are damaged.

b) Spikes (or sharp waves): The spike (or sharp wave) is a suddenly appearing (paroxysmal) electrical explosion. The sharp wave is the same as the spike except for a difference in the duration of the event. Spikes are shorter in duration, usually less than 70 msec, while sharp waves last from 70 to 200 msec. These two related patterns usually signify an epileptogenic region of the brain and are found in patients with seizures. Another pattern related to the spike is the spike-and-wave complex which consists of a spike followed by a wave. These complexes usually repeat themselves.

5- Loss of reactivity to stimulation occurs or an untypical response occurs.

Throughout the summarized description of the basic EEG concepts discussed above, it is noted that the EEG signals are described by an important characteristics which, in turn, reflect the normality and the abnormality present in a person. It was found that the most important of these characteristics are the amplitude, the frequency and the shape of the EEG signals.

Knowing these important features, one must determine how to estimate them efficiently. The amplitude (or indeed, the power) and the frequency are most easily estimated by the spectral estimation techniques, while the third characteristics can be determined using a special techniques. It is the aim of the next two chapters to produce techniques which can be used to achieve this purpose. Thus, the next chapter will be devoted for introducing an efficient method for describing the EEG signals. It is concerned with the analysis of the ARMA model through its various applications such as the power spectral estimation which is an important field in EEG analysis. Besides, the important morphylogical shape that is, the spikes, can be determined by utilizing the ARMA model as will be shown in the Chapter 4.

CHAPTER 3

ARMA MODELING

Extensive researches have been done in the field of EEG analysis. All of these researches shared the following interests which also serve to be the logical steps of any computerized EEG analysis system:

First, because of the large amount of data involved, there was an interest to replace the mass of original data by a small number of descriptive characteristics, that is, a big interest was lying in the reduction of data. Next, from the reduced data, one must easily find estimates of the true value which describe the EEG. Since the EEG signal shows certain periodicities (sharp resonance) accompanied by some randomness, it was of interest to detect the presence of position and strength of the different resonances, known as rhythms. Therefore, the basic interest was in describing the spectrum in a simple mathematical manner that would yield the characteristics of the different rhythms. Thus, the first step in EEG signal analysis is to construct a mathematical model which is sufficient to describe them.

Generally, methods for the analysis of EEG may be conveniently classified in two board categories, namely, the *parametric* and the *nonparametric* methods. The latter makes only general assumptions about the process that generates the EEG signal, essentially stationary during short interval. Parametric analysis is, in one sense, more specialized and, in another sense, more general.

3.1 PARAMETRIC REPRESENTATION OF EEG SIGNALS

Parametric analysis assumes that the EEG signals may be represented by means of a stochastic model involving specific parameters. Most often linear models have been used. With fixed parameters and appropriate conditions satisfied, the model will describe a stationary process or rather a class of such process. However the class may be generalized by allowing the parameters to vary with time. In this way nonstationary properties of the EEG signals may be taken into account.

To analyze an EEG signal by parametric methods, a linear filter is given to describe the generating process of EEG. This filter is fed with white noise, and the output EEG signal has a spectral density depending upon the properties of the linear filter. The statistics of this output signal can be determined by using the regression of the signal upon itself. The linear prediction technique predicts the current output, *y(n)*, from linear combination of past outputs and inputs, *x(n)*, as shown below

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) \qquad (3.1)$$

where a_k and b_k are the parameters of the hypothesized system with model order *(p,q)*. Equation (3.1) is the general Autoregressive Moving Average ARMA model. Several algorithm have been developed for system identification using ARMA modeling (Cadzow *et al.*, 1986; Pillai *et al.*, 1993; Politis, 1993).

ARMA model is considered to be a powerful mathematical method for time series analysis. It has the advantage of an immense data reduction, because it based on a few parameters in representing the signal. It has been successfully used for various applications such as prediction in the time domain, parametric spectral modeling, feature extraction and detection of the abrupt changes in the modeled signal. It is obvious by now that the ARMA model is a predictive model. It gives an explicit parametric representation of the signal, from which important properties of the data, such as the spectral density can be computed as a function of the model parameters.

In all the algorithms that will be developed in this chapter, it is assumed that the continuous-time signal $y(t)$ is sampled to obtain the discrete-time signal $y(nT)$, where n is an integer and T is the sampling interval with a corresponding sample frequency $f_s = 1/T$, which achieves the Nyquist criterion. We shall abbreviate $y(nT)$ by $y(n)$ with no loss of generality.

3.1.1 ARMA MODELS

In this section, we demonstrate that a wide-sense stationary random process may be represented as the output of a causal and causally invertible linear system excited by a white noise process. The condition that the system is causally invertible also allows us to represent the wide-sense stationary random process by the output of the inverse system which is a white noise process.

Many discrete-time random process encountered in practice are well approximated by a rational transfer function model. In this model, an input driving sequence *x(n)* and the output sequence *y(n)* that is used to model the data are related by the linear difference equation (3.1). This most general linear model is termed an ARMA model and shown in Fig. 3.1(a) with order *(p,q)*.

Equation (3.1) says that the output *y(n)* is a linear combination of past outputs and present and past inputs. That is, the signal *y(n)* is predictable from linear combinations of past outputs and inputs. So, the signal is modeled as a linear combination of its past values and present and past values of a hypothetical input to a system whose output is the given signal. Thus, if $\hat{y}(n)$ is an estimate of *y(n)*, then the prediction sequence is given by

$$\hat{y}(n) = -\sum_{k=1}^{p} \hat{a}_k y(n-k) + \sum_{k=0}^{q} \hat{b}_k x(n-k) \tag{3.2}$$

where $\hat{a}_k$ and $\hat{b}_k$ are the estimated model parameters.

Processes that can be approximated by rational transfer functions are quite common in nature. In ARMA model, the numerator of such function is referred to as the Moving Average (MA) part and expresses the contribution of the q most recent values of the input driving sequence (generally assumed to be white noise). The denominator or the Autoregressive (AR) branch reflects the functional relationship between p previous observations of the process and its most recent one. Thus, if all the a_k coefficients (except a_0=1) vanish for the ARMA parameters, then

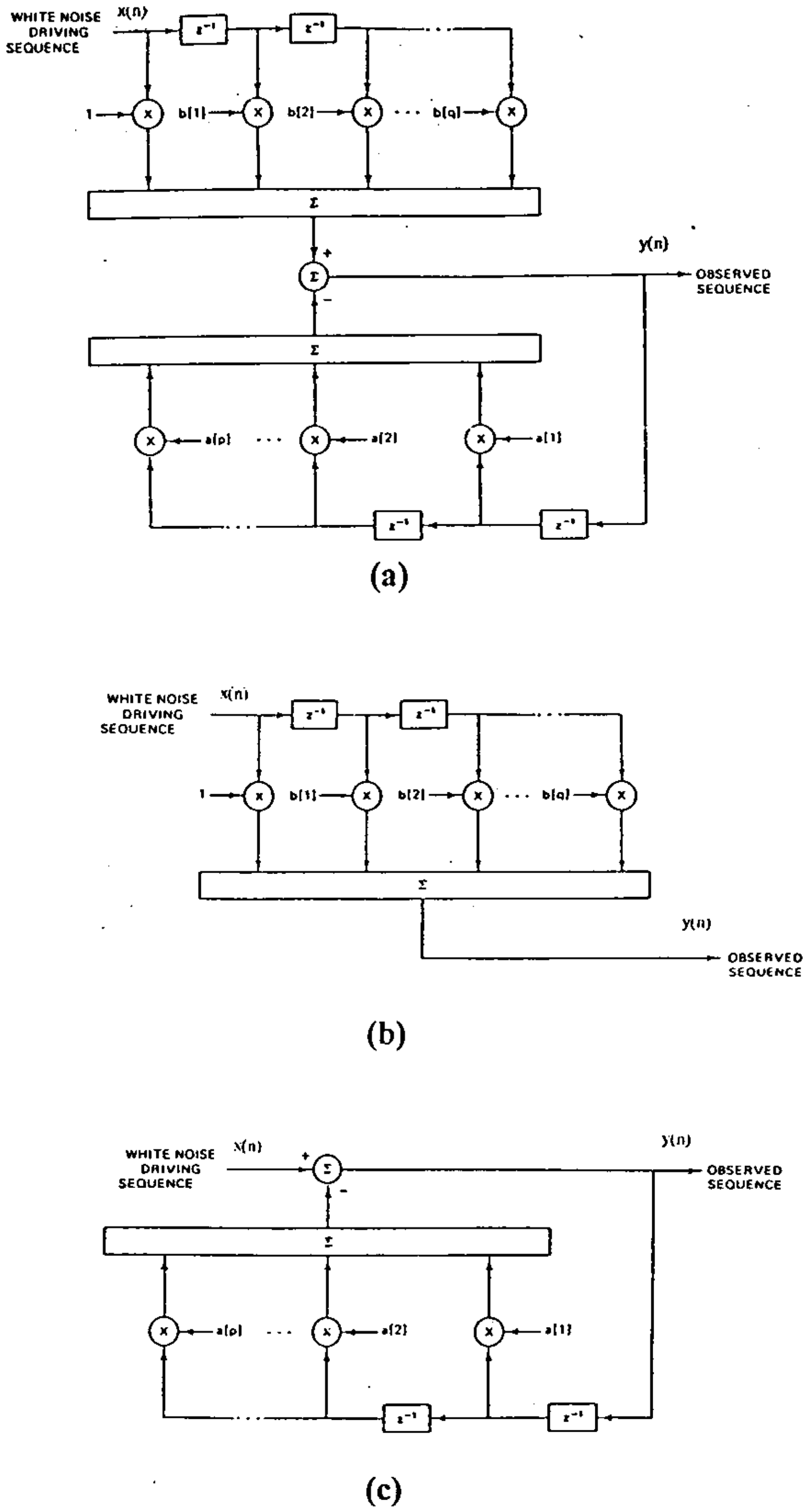


Fig. 3.1 (a) ARMA model of a random process. (b) MA model of a random process. (c) AR model of a random process.

$$y(n) = \sum_{k=0}^{q} b_k x(n-k) \tag{3.3}$$

and the process is strictly a Moving Average (MA) process of order q. This model is sometimes termed an *all-zero* model and shown in Fig. 3.1(b).

If all the b_k coefficients (except $b_0 = 1$) are zero in the ARMA parameters, then

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + x(n) \tag{3.4}$$

and the process is strictly an Autoregressive (AR) process of order p. This model is sometimes termed an *all-pole* model and shown in Fig. 3.1(c).

It is important to distinguish between the driving noise of the model *x(n)* and any observation noise. The ARMA model noise is not an additive or observation noise which is typically encountered in signal processing applications. *x(n)* is an innate part of the model and gives rise to the random nature of the observed process *y(n).* Any observation noise then needs to be modeled within the ARMA process by modification of its parameters.

Therefore, the time series *y(n)* is said to be ARMA process of the order *(p,q)* if it is generated (or can be modeled) according to the recursive relationship in (3.1) in which the excitation sequence *x(n)* is white noise process. In the z-domain, the system function *H(z)* between the input *x(n)* and the output *y(n)* for the ARMA process in (3.1) is the rational function

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3.5}$$

For the model to be stable, it is required that the polynomial

$$A(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-p}$$

have all its roots inside the unit circuit. If the polynomial

$$B(z) = b_0 + b_1 z^{-1} + \ldots + b_q z^{-q}$$

have all its roots inside the unit circuit, then the model is minimum-phase and therefore invertible. If these conditions are achieved then, the output of this filter to white noise input sequence *x(n)* is stationary random process *y(n)*. Conversely, the stationary random process *y(n)* may be transformed into a white noise process by passing *y(n)* through a linear filter with system function *1/H(z)*. We call this filter a noise *whitening filter*. Its output, denoted as *x(n)*, is called the *innovations process* associated with the stationary random process *y(n)*. These two relationships are illustrated in Fig. 3.2.

The prediction error *e(n)*, or the residual, will thus be defined by the following difference equation

$$e(n) = y(n) - \hat{y}(n) = y(n) + \sum_{k=1}^{p} \hat{a}_k y(n-k) - \sum_{k=0}^{q} \hat{b}_k x(n-k) \tag{3.6}$$

In this formulation the residual may be considered as either the output of the prediction error filter *A(z)/B(z)* which is also known as the inverse filter, or as the excitation signal of the ARMA model *H(z)*.

Given that the model is suitable one, therefore an estimate of the parameters of the model is required. The key to the performance of parametric modeling lies in the relative effectiveness of the various algorithms that can be used to estimate the model parameters.

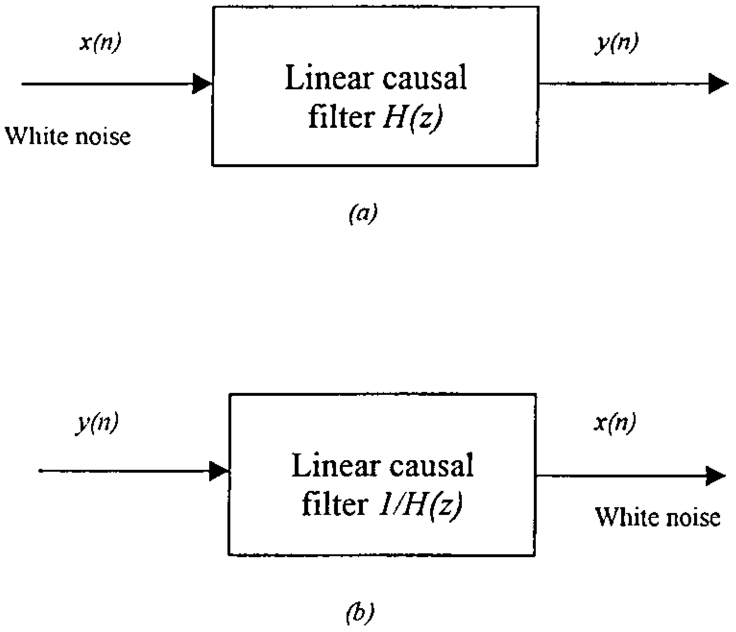


Fig. 3.2 Filter for generating (a) the random process *y(n)* from white noise *x(n)* and (b) the inverse filter.

3.2 ARMA PARAMETER ESTIMATION

The classical modeling problem is that of identifying the system's a_k and b_k parameters from a finite set of observations of the excitation and the response time series. The basic approach for calculating these parameters is to adopt a certain error criterion for the perfectness of fitting of the ARMA model to the set of data samples. The least squares (LS) criterion was chosen for this formulation due to its stability and physical meaning (Cadzow *et al.*, 1986).

In general, an ARMA filter is produced by cascading an AR filter with MA filter as shown in Fig. 3.3 (a). Although quite simple, the structure shown

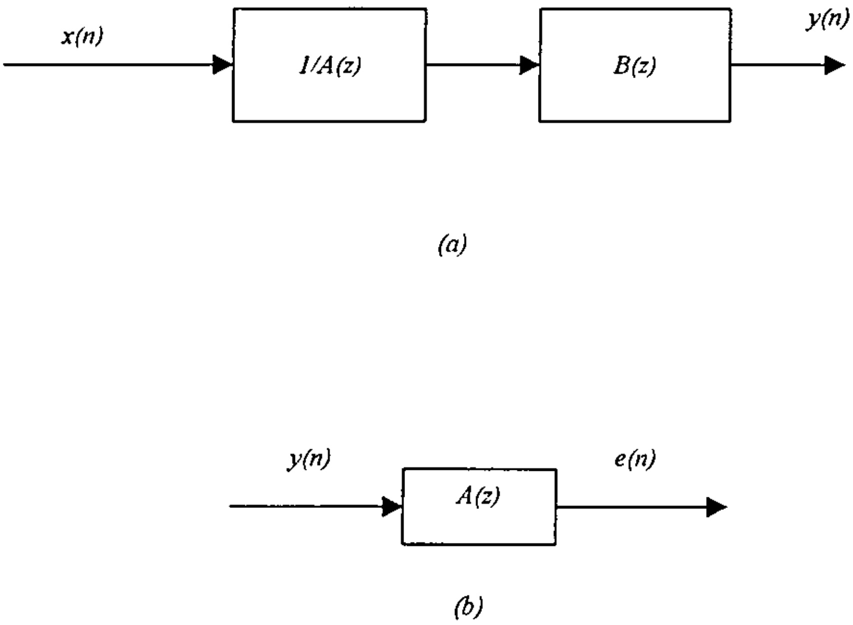


Fig. 3.3 (a) ARMA filter construction and (b) construction of residual sequence.

above is used extensively in various parameter estimation algorithms which generally estimate the AR and MA parameters separately. The normal method is to generate the AR parameters first, use these parameters to generate a residual time-series, and then calculate the MA parameters. Thus, if *y(n)* is an ARMA time-series and the AR filter is defined as *1/A(z)*, then the residual time-series *e(n)* can be produced by passing *y(n)* through *A(z)* as shown in Fig. 3.3(b). The sequence *e(n)* corresponds to a moving average sequence because the AR component in the original signal has been filtered out. Therefore, the MA parameters of the origin signal are determined using *e(n)* in the MA algorithm.

Before describing methods for estimating the parameters in an ARMA*(p,q)* process, it is useful to establish the basic relationship between the model parameters and the autocorrelation sequence.

For a parametric model with a rational transfer function, a basic relationship exists between the autocorrelation sequence $r_{yy}(n)$ and the parameters a_k and b_k of the linear filter $H(z)$ which generates the process by filtering the white noise sequence $x(n)$. This relationship may be obtained by multiplying the difference equation (3.1) by $y^*(n\text{-}m)$, where * is the conjugate operator, and then taking the expected value of both sides of the resulting equation. Thus, we have

$$E[y(n)y^*(n-m)] = -\sum_{k=1}^{p} a_k[y(n-k)y^*(n-m)] + \sum_{k=0}^{q} b_k[x(n-k)y^*(n-m)] \qquad (3.7)$$

Hence,

$$r_{yy}(m) = -\sum_{k=1}^{p} a_k r_{yy}(m-k) + \sum_{k=0}^{q} b_k r_{yx}(m-k) \qquad (3.8)$$

where $r_{yx}(m)$ is the cross correlation function (CCF) sequence between $x(n)$ and $y(n)$. The cross correlation is related to the filter impulse response. That is,

$$r_{yx}(m) = E[y^*(n)x(n+m)] = E\left[\sum_{k=0}^{\infty} h(k)x^*(n-k)x(n+m)\right] = \sigma^2 h(-m) \qquad (3.9)$$

where in the last step, we have used the fact that the sequence $x(n)$ is a white noise with variance σ^2. Hence,

$$r_{yx}(m) = \begin{cases} 0 & m > 0 \\ \sigma^2 h(-m) & m \leq 0 \end{cases} \qquad (3.10)$$

By combining (3.10) with (3.8) we obtain the desired relationship

$$r_{yy}(m)=\begin{cases}-\sum_{k=1}^{p}a_k r_{yy}(m-k) & m>q\\ -\sum_{k=1}^{p}a_k r_{yy}(m-k)+\sigma^2\sum_{k=0}^{q-m}h(k)b_{k+m} & 0\le m\le q\\ r_{yy}^{*}(-m) & m<0\end{cases} \tag{3.11}$$

This relationship in (3.11) applies in general to the ARMA process.

It should be noted that the relationship between the parameters of the ARMA process and the autocovariance function (ACF), $r_{yy}(m)$, is nonlinear. Given the ACF, we must solve a set of nonlinear equations to find the model parameters. This is due to the $\sum_{k=0}^{q-m}h(k)b_{k+m}$ term.

For an AR process (3.11) simplifies to

$$r_{yy}(m)=\begin{cases}-\sum_{k=1}^{p}a_k r_{yy}(m-k) & m>0\\ -\sum_{k=1}^{p}a_k r_{yy}(m-k)+\sigma^2 & m=0\\ r_{yy}^{*}(-m) & m<0\end{cases} \tag{3.12}$$

Thus, we have linear relationship between $r_{yy}(m)$ and the a_k parameters. These equations, called Yule-Walker equations, may be expressed in matrix form

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(-1) & r_{yy}(-2) & \cdots & r_{yy}(-p)\\ r_{yy}(1) & r_{yy}(0) & r_{yy}(-1) & \cdots & r_{yy}(-p+1)\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ r_{yy}(p) & r_{yy}(p-1) & r_{yy}(p-2) & \cdots & r_{yy}(0)\end{bmatrix}\begin{bmatrix}1\\ a_1\\ \vdots\\ a_p\end{bmatrix}=\begin{bmatrix}\sigma^2\\ 0\\ \vdots\\ 0\end{bmatrix} \tag{3.13}$$

Equation (3.11) is useful for the ARMA process if one knows $r_{yy}(m)$. In this case, it allows one to find the ARMA parameters from $r_{yy}(m)$ and $r_{yx}(m)$ by solving a set of linear equations.

After establishing the relationship of the model parameters and the correlation sequence, the model parameters can be estimated. The parameters of the ARMA model can be calculated in many ways. They can be estimated efficiently using the Modified Yule-Walker Equations (MYWE) method (Bruzzone *et al.*, 1984; Jacobs *et al.*, 1989; Kay, 1988). Although the ARMA process can be modeled by maximum likelihood techniques that minimize a nonlinear function, it has been shown that it is not the best method in practical applications (Kay, 1988). Therefore, in the next section, the MYWE method will be discussed in which the AR parameters are estimated directly by solving the set of linear equations that are given in (3.11) for $k = q+1, q+2, \ldots, p+q$. To find the AR parameters we need to solve

$$\begin{bmatrix} r_{yy}(q) & r_{yy}(q-1) & \cdots & r_{yy}(q-p+1) \\ r_{yy}(q+1) & r_{yy}(q) & \cdots & r_{yy}(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(q+p-1) & r_{yy}(q+p-2) & \cdots & r_{yy}(q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r_{yy}(q+1) \\ r_{yy}(q+2) \\ \vdots \\ r_{yy}(q+p) \end{bmatrix} \qquad (3.14)$$

These equations have been called the extended or modified Yule-Walker equations.

3.2.1 MODIFIED YULE-WALKER EQUATIONS (MYWE) METHOD

The ARMA estimation method described in this section uses the modified Yule-Walker equations (3.14) described above. Since these relationships hold when the ACF is known exactly, a reasonable approach is to replace the theoretical ACF samples by estimates and then solve the equations for AR parameters. The MA parameters are found in a separate step. This leads to the following estimation of the AR parameters:

$$\begin{bmatrix} \hat{r}_{yy}(q) & \hat{r}_{yy}(q-1) & \cdots & \hat{r}_{yy}(q-p+1) \\ \hat{r}_{yy}(q+1) & \hat{r}_{yy}(q) & \cdots & \hat{r}_{yy}(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{yy}(q+p-1) & \hat{r}_{yy}(q+p-2) & \cdots & \hat{r}_{yy}(q) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} = - \begin{bmatrix} \hat{r}_{yy}(q+1) \\ \hat{r}_{yy}(q+2) \\ \vdots \\ \hat{r}_{yy}(q+p) \end{bmatrix} \qquad (3.15)$$

From which we obtain the system

$$\hat{A}(z) = 1 + \sum_{k=1}^{p} \hat{a}_k z^{-k} \qquad (3.16)$$

Once the AR parameters have been estimated and *y(n)* filtered by $\hat{A}(z)$ to produce the approximate MA process, many methods may be used to estimate the MA parameters. Here, the Durbin's method for estimating the MA parameters is used which is given by

$$\hat{b} = -R_{aa}^{-1}\hat{r}_{aa} \qquad (3.17)$$

where

$$\left[\hat{R}_{aa}\right]_{ij} = \frac{1}{L-1} \sum_{k=1}^{L-|i-j|} \hat{a}_k \hat{a}_{k+|i-j|} \qquad i, j = 1, 2, \ldots, q$$

$$\left[\hat{r}_{aa}\right]_{i} = \frac{1}{L-1} \sum_{k=1}^{L-|i-j|} \hat{a}_k \hat{a}_{k+i} \qquad i = 1, 2, \ldots, q$$

and $q \le L \le N$. The $\hat{a}_k$ parameters are obtained by solving the MYWE.

The MYWE can be solved in an efficient manner using an extension of the Levinson recursion (Kay, 1988). The recursive algorithm is given by:

$$a_1(1) = -\frac{r_{yy}(q+1)}{r_{yy}(q)}$$

$$b_1(1) = -\frac{r_{yy}(q-1)}{r_{yy}(q)}$$

$$\rho_1 = (1 - a_1(1)b_1(1))r_{yy}(q)$$

Main Loop:

for $k = 2, 3, \ldots, p$

$$a_k(k) = -\frac{r_{yy}(q+k) + \sum_{m=1}^{k-1} a_{k-1}(m) r_{yy}(q-k-m)}{\rho_{k-11}}$$

$$a_k(i) = a_{k-1}(i) + a_k(k)b_{k-1}(k-i) \qquad i = 1, 2, \ldots., k-1$$

If $k = p$, exit; if not continue

$$b_k(k) = -\frac{r_{yy}(q-k) + \sum_{m=1}^{k-1} b_{k-1}(l) r_{yy}(q-k-m)}{\rho_{k-11}}$$

$$b_k(i) = b_{k-1}(i) + b_k(k)a_{k-1}(k-i) \qquad i = 1, 2, \ldots, k-1$$

$$\rho_{k1} = (1 - a_k(k)b_k(k))\rho_{k-1}$$

Thus, in this way, the ARMA parameter estimation is accomplished. With the background established above, we will now describe the power spectrum estimation method for the ARMA*(p,q)* process.

3.3 POWER SPECTRAL ESTIMATION

The major aspect of ARMA modeling in analyzing EEG signals is the spectral estimation, from which valuable features of the EEG signals can be extracted. Generally, the EEG contains information regarding changes in the electrical potential of the brain obtained from a given set of recording electrodes. These data include the characteristic waveform with its variation in amplitude, frequency, phase, etc. Any analysis procedure cannot simultaneously provide information regarding all of these variables. Consequently, the selection of any analytic procedure will emphasize change in one particular variable at the expense of the others. Power spectral analysis (the square of frequency spectra) provides a quantitative measure of the frequency distribution of the EEG at the expense of other details in the EEG such as the amplitude distribution.

In the context of modeling, spectral estimation is a three-step procedure.

- Select an appropriate model.
- Estimate the parameters of the assumed model using the available data samples.
- Obtain the spectral estimates by distributing the estimated model parameters into the theoretical PSD implied by the model.

As defined earlier, the transfer function of the ARMA model is given by equation 3.5. It is well-known that the z-transform of the ACF at the output of a linear filter, $p_{yy}(z)$, is related to the input $p_{xx}(z)$, as follows

$$P_{yy}(z) = H(z)H^*(1/z^*)P_{xx}(z) = \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}P_{xx}(z) \qquad (3.18)$$

when (3.16) is evaluated along the unit circle, $z = exp(j2\pi f)$, it becomes the PSD $p_{yy}(f)$. Often the driving process is assumed to be a white noise sequence of zero mean and variance σ^2. The PSD of the noise is then σ^2. The PSD of the ARMA output process becomes

$$P_{yy}(f) = P_x(f)\,|H(\exp(j2\pi f)\,|^2 = \sigma^2 \frac{\left|\sum_{k=0}^{q} b_k \exp(-j2\pi fk)\right|^2}{\left|\sum_{k=0}^{p} a_k \exp(-j2\pi fk)\right|^2} = \sigma^2 \frac{|B(f)|^2}{|A(f)|^2} \qquad (3.19)$$

Without loss of generality, we can assume that $a_0 = 1$ and $b_0 = 1$ since any filter gain can be incorporated into σ^2. By using this parametric representation, the spectral density can be calculated by a limited parameter set for every frequency argument.

Thus, as the ARMA parameters are estimated, one can estimate the power spectral density from (3.19). An efficient way for ARMA spectral estimation, is to find firstly the MA spectral by making use of the estimated AR parameters, then the PSD of the ARMA model can be estimated without requiring the b_k parameters.

As mentioned earlier, the AR and MA parameters of the ARMA model are estimated separately, once the parameters of the AR part of the model have been estimated, the sequence *y(n)* may be now filtered by the filter $\hat{A}(z)$ defined in (3.16) to yield the sequence

$$v(n) = y(n) + \sum_{k=1}^{p} \hat{a}_k y(n-k) \quad n = 0, 1, 2, \ldots, N-1 \qquad (3.20)$$

The cascade of the ARMA*(p,q)* model with $\hat{A}(z)$ is approximately the MA*(q)* process generated by the model *B(z)*. To be specific, the filtered sequence *v(n)* for $p \le n \le N-1$ is used to form the estimated correlation sequence $r_{vv}(n)$, from which we obtain the MA spectrum

$$P_{vv}^{MA}(f) = \sum_{m=-q}^{q} r_{vv}(m) e^{-j2\pi fm} \tag{3.21a}$$

then, the estimated ARMA power spectrum is given by

$$P_{yy}^{ARMA}(f) = \frac{P_{vv}^{MA}(f)}{\left|1+\sum_{k=1}^{p} a_k e^{-j2\pi fk}\right|^2} \tag{3.21b}$$

3.4 TIME-VARYING ARMA MODELS

Fitting ARMA models may be recommended to the analysis of stationary signals whose statistical characteristics, such as average amplitude and frequency content, do not vary with time. However, as a rule, biological or biomedical signals are not stationary; they have a time-varying power spectrum. Stationarity can be assumed only for single intervals. Consequently, modeling such signals is not suitable for linear models which cannot change in time. To account for these nonstationary signals, the construction of an ARMA model with parameters varying in time and the adaptation of the model to the changes of the structure of the signal at every point, will be discussed.

An EEG signal may vary much from one moment to another due to changes in the physiological state of the subject. When observed for longer intervals (more than 20 sec), the nonstationary character of the EEG signal will be apparent. EEG signals may be visualized as random signals occasionally containing dominant frequency called rhythm and/or pulse like activity called spike. Also it was found that although EEG signals may appear stationary over a short period of time, they exhibit a markedly nonstationary behavior when observed for longer periods of time (Jansen, 1985). For example, the amplitude and/or the frequency may change suddenly, and such changes may last for long durations or may occur as brief transients. Thus, since EEG is continually time-varying process, there must be a special way to deal with them. For such signals, an adaptive model can be used, in which its parameters are updated with the arrival of each new data sample making it a powerful tool for tracking the time dependency of the signal.

Systems possessing this characteristic were called adaptive systems because they can change their parameters in accordance with measured changes in their environment. Therefore, the corresponding ARMA models that can change their parameters in accordance with the changing nature of the incoming signal are called adaptive ARMA models. There are several algorithms that can be used as an adaptive ARMA model. The most common ones are Least Mean Square (LMS), Recursive Least Square (RLS) and the Kalman filters, besides, new developed algorithms (Grenier, 1983; Haseyama *et al.*, 1993; Isaksson *et al.*, 1981; Nehorai *et al.*, 1988).

Using adaptive ARMA modeling it is possible to determine the power spectrum at every time point. It provides a technique to calculate the time-varying power spectra in such a way that enables the estimation of frequency band powers with very high time resolution. In principle, the estimation is possible at each sample point with an arbitrarily high-frequency resolution. The dynamic spectral analysis can be performed with high-frequency resolution and the computing time allows an on-line calculation of spectral parameters.

In the following section an adaptive procedure of fitting a time-varying ARMA model to nonstationary signals is presented.

3.4.1 ADAPTIVE ARMA MODEL

In this section an adaptive recursive ARMA modeling as the basis for time-varying (dynamic) power spectral analysis will be introduced. This adaptive ARMA model is fit to nonstationary time series which allows self-exciting adaptation of the model parameters being estimated in every time point.

In general, time varying ARMA*(p,q)* models for the process *y(n)* can be written as

$$y(n) = -\sum_{k=1}^{p} a_k(n) y(n-k) + \sum_{k=0}^{q} b_k(n) x(n-k) \tag{3.22}$$

where the parameters $a_k(n)$ and $b_k(n)$ are estimated with an adaptive technique.

A recursive parameter estimation method for reducing the mean square linear prediction error has been developed as follows (Schack *et al.*, 1995)

$$\hat{a}_k(n) = 0 \qquad \text{for } \; n \le k, \; k=1, 2, \dots, p$$

$$\hat{b}_k(n) = 0 \qquad \text{for } \; n \le k, \; k=1, 2, \dots, q$$

$$\hat{a}_k(n) = \hat{a}_k(n-1) - c(n-1)e(n)y(n-k), \qquad \text{for } \; n > k, \; k=1, 2, \dots, p \tag{3.23}$$

$$\hat{b}_k(n) = \hat{b}_k(n-1) - c(n-1)e(n)e(n-k), \qquad \text{for } \; n > k, \; k=1, 2, \dots, q \tag{3.24}$$

The sequence of the prediction errors *e(n)* is calculated by the recursive algorithm

$$e(n) = y(n) + \sum_{k=1}^{p} \hat{a}_k(n-1)y(n-k) + \sum_{k=1}^{q} \hat{b}_k(n-1)e(n-k) \qquad \text{for } n > 0 \tag{3.25}$$

The estimation procedure for equations *(3.22)-(3.25)* uses the prediction error for the correction of the parameter estimation at every time point. This procedure is called the adaptive ARMA modeling of a stochastic signal. The control sequence *c(n),* which is called adaptation variable, needs to satisfy certain conditions for convergent estimating. To obtain a robust estimation procedure which can react to changes of the structure of the signal, *c(n)* is chosen as

$$c(n) = c_y(n)*f_y \tag{3.26}$$

where $c_y(n)$ is limited and depends on the properties of the signal observed, and the factor f_y is a positive number. The term $c_y(n)$ is chosen to be reciprocally proportional to the variance of the signal

$$c_y(n)=(\sigma^2{}_y(n))^{-1} \tag{3.27}$$

By considering the possibility of changes in the variance with time, a dynamic fit of the variance estimation is necessary at each point in time. Such adaptive estimation procedure for time-varying statistical parameters have been developed previously and have the following special form (Schack *et al.*, 1995)

$$\sigma^2{}_y(0) = 0$$

$$\sigma^2{}_y(n) = \sigma^2{}_y(n\text{-}1)\text{-}c_s(\sigma^2{}_y(n\text{-}1)\text{-}y^2(n)) \tag{3.28}$$

where the adaptation variable c_s fulfills the condition $0<c_s<1$. Similarly, the variance of the signal of the prediction error is estimated adaptively as

$$\sigma^2_x(0) = 0$$

$$\sigma^2_x(n) = \sigma^2_x(n-1)-c_x(\sigma^2_x(n-1)-e^2(n)) \tag{3.29}$$

with c_x fulfills the condition $0<c_x< 1$.

The adaptation speed can be adjusted by the choice of the factor f_y and the constant c_s. By increasing these factors, a quicker reaction of the estimation procedure is possible after rapid structure changes. The result of higher values of f_y and c_s is greater variation of the parameter estimation sequences around their true values during stationary intervals.

As a result, an instantaneous estimation of the spectral density (updated for each new sample point) can be given as

$$P_{yy}(n,f) = \frac{\left|\sum_{k=0}^{q} b_k(n)\exp(-j2\pi fk)\right|^2}{\left|1+\sum_{k=1}^{p} a_k(n)\exp(-j2\pi fk)\right|^2} \tag{3.30}$$

3.4.2 SIMULATION RESULTS

The performance of the adaptive ARMA procedure was tested on a simulated data, and different test systems were used. The systems are driven at the input with a white Gaussian noise *x(n)* with zero mean and variance σ_x^2.

a) Adaptation of the model parameters:

The first system is an ARMA process with order *(4,4)* described by

$$y(n) + 0.1y(n-1) + 1.61y(n-2) + .016y(n-3) + 0.74y(n-4) = x(n) + 0.34x(n-1) + 0.56x(n-2) + 0.21x(n-3) + 0.47x(n-4)$$

The adaptive estimations of the model parameters are shown in Fig. 3.4. As seen from these figures, the adaptive sequences quickly fit to the true parameters, varying around them after convergence.

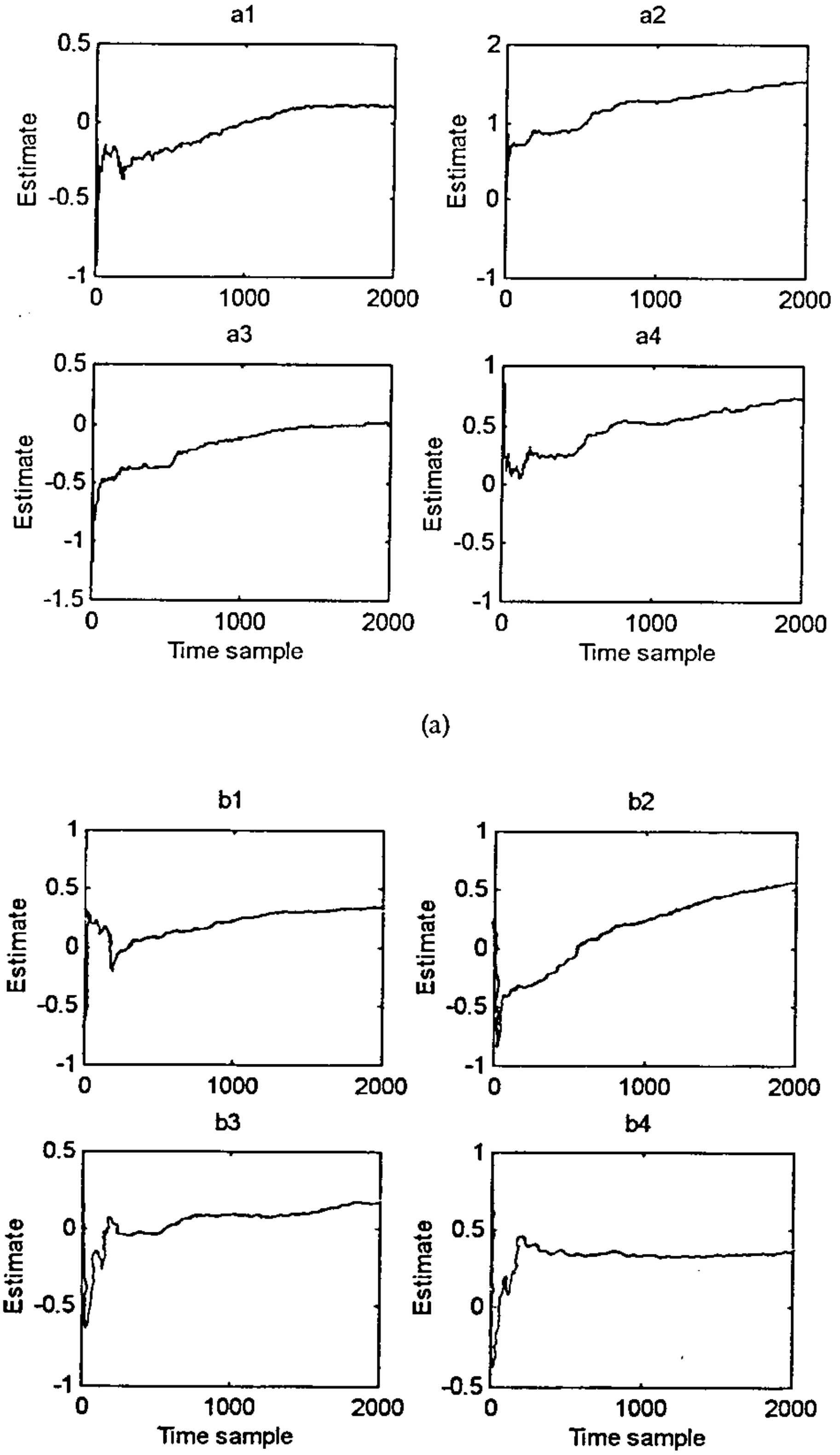
b) Tracking the structural changes of the modeled signals:

The ability of the adaptive model for fast fitting after structural changes of the modeled signal is demonstrated with the following example.

The test system used here is the ARMA*(2,2)* system given by

$$y(n) + a_1 y(n-1) + 0.98\,y(n-2) = x(n) + 2x(n-1) + 0.5x(n-2)$$

The value of the parameter a_1 *is* changed from 1.96 to -0.4 at the sample point n = 1000 as illustrated in Fig. 3.5 which shows the abrupt change of the estimating parameter after the change point and its convergence to the true value. Thus, the adaptive parameter estimation sequence try to fit to the new value. This property enables a continuous adaptation of the linear model to the nonstationary signals.



(a)

(b)

Fig. 3.4 Estimating of the parameters of the ARMA(4,4) process (a) a_k parameters, and (b) b_k parameters.

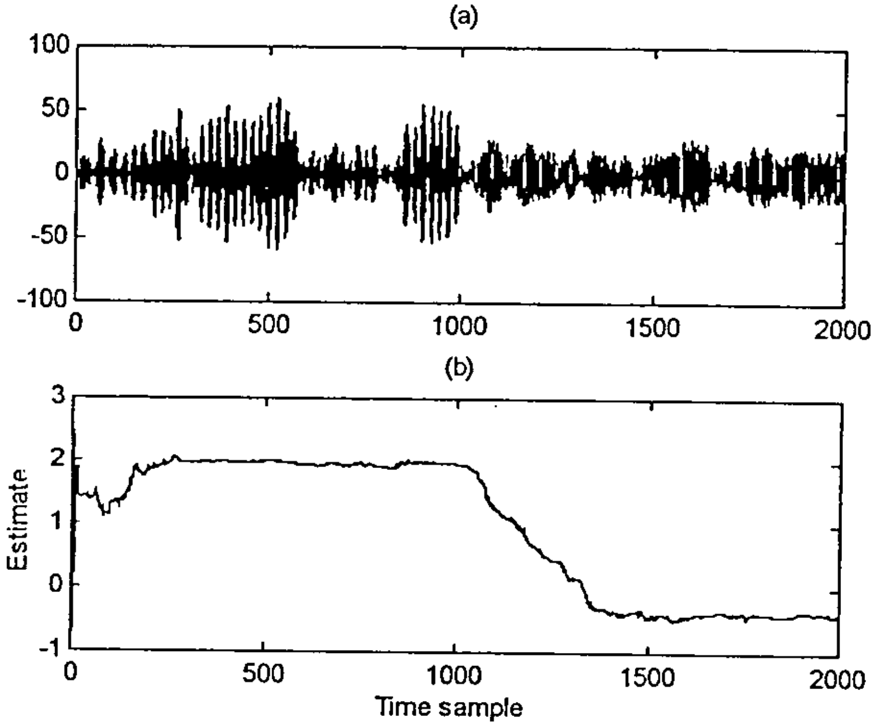


Fig. 3.5 (a) Simulation of time series consisting of two different ARMA(4,4) process (i.e., different in the a_l parameter). The parameter a_l is changed from 1.96 to -0.4 at sample point $n = 1000$. (b) adaptive estimation of the parameter a_l.

3.4.3 ADAPTIVE POLE-ZERO ESTIMATION

Systems are usually identified in terms of the coefficients of their characteristic polynomials or transfer function. However, one is often more interested in the roots of the polynomials than in their coefficients. For example, in biomedical engineering, the trajectory of the roots of an AR model of EEG signals has been used to predict the onset of seizure (Rogowski *et al.*, 1981). The z-plane trajectories of the zeros of the prediction error filter during the seizure period seem to provide a good way of predicting the onset of the seizure. It is precisely the transient behavior of the adaptive pole that is of interest.

In situation like this, representation of the signals or systems in terms of their roots is more appropriate than representation in terms of coefficients. Traditionally, the roots are located in a two-stage procedure. Initially, the polynomial coefficients are estimated using conventional identification techniques such as RLS algorithm, and then the roots of the resulting polynomial are found using standard factorization schemes. However, this approach is often computationally too intense to apply to on-line tracking of system roots where updated estimates are required at every sampling interval.

Thus, there exists a need for an algorithm which can satisfy the two simultaneous requirements of

a) directly providing the estimates of the system's poles (rather than the coefficients)

b) providing new estimates as each data sample is received.

For stationary processes, the estimates are required to become more accurate with each update. For nonstationary processes, the estimates are required to track the time-varying parameters.

Here an algorithm proposed by Nehorai *et al.* (1990) which satisfies the above requirements is presented. The method estimates the poles and zeros without explicitly estimating and factorizing the AR and MA polynomials. It is based on the direct parameterization of the ARMA process in terms of its poles and zeros rather than its coefficients a_k and b_k, and uses the Gauss-Newton Recursive Prediction Error (RPE) method (Soderstron *et al.*, 1989). The algorithm is derived as follows:

An ARMA process is given by equation (3.1). In the z-domain, it is represented as

$$Y(z) = H(z)X(z) = \frac{B(z^{-1})}{A(z^{-1})}X(z) \tag{3.31}$$

where *H(z)* is the transfer function between the input *x(n)*, which is a zero-mean white Gaussian noise with variance σ^2, and the output *y(n)*. The polynomials $A(z^{-1})$ and $B(z^{-1})$ can be represented as follows

$$A(z^{-1}) = \sum_{k=0}^{p} a_k z^{-k} = \prod_{k=1}^{p}(1-\lambda_{a_k} z^{-1}) \tag{3.32a}$$

$$B(z^{-1}) = \sum_{k=0}^{q} b_k z^{-k} = \prod_{k=1}^{q}(1-\lambda_{b_k} z^{-1}) \tag{3.32b}$$

The parameters λ_{a_k}, $k = 1, 2, ..., p$ and λ_{b_k}, $k = 1, 2, ..., q$ are the poles and the zeros of the system assumed to have magnitude less than unity. The coefficients a_k are assumed to be real, implying that the poles should be real or occur in complex conjugate pairs.

Define $\rho^{(a)}{}_{k}$, $\omega^{(a)}{}_{k}$ and $\rho^{(b)}{}_{k}$, $\omega^{(b)}{}_{k}$ to be the radius and positive angle of the k^{th} pair of the complex conjugate roots of $A(z^{-1})$ and $B(z^{-1})$ respectively, so that

$$\lambda_k = \rho_k \, exp(j\omega_k) \qquad (3.33)$$

The unknown parameter vector is defined as

$$\theta = [\rho^{(a)\,T} \; \rho^{(b)\,T} \; \omega^{(a)\,T} \; \omega^{(b)\,T} \,]^T \qquad (3.34)$$

where,

$$\rho^{(a)} = [\rho_1{}^{(a)}, \ldots, \rho_{ma}{}^{(a)} \,]^T$$

$$\omega^{(a)} = [\,\omega_1{}^{(a)}, \ldots, \omega_{ma}{}^{(a)} \,]^T$$

with $m_a = p/2$ which represent the radii and angles of the roots of $A(z^{-1})$. The superscript T means matrix transpose.

Similarly,

$$\rho^{(b)} = [\rho_1{}^{(b)}, \ldots, \rho_{mb}{}^{(b)} \,]^T$$

$$\omega^{(b)} = [\,\omega_1{}^{(b)}, \ldots, \omega_{mb}{}^{(b)} \,]^T.$$

with $m_b = q/2$ which represent the radii and angle of the roots of $B(z^{-1})$.

The polynomial coefficients derived from these roots are collected into the coefficient vector η given by

$$\eta = [a^T \, b^T \,]^T \qquad (3.35)$$

where,

$$a = [a_1, \ldots, a_{ma}]^T$$

$$b = [b_1, \ldots, b_{mb}]^T$$

in a recursive manner. The entries of the matrix $\xi_{11}(n)$ are calculated using the following recursion:

$$\frac{\partial a_i}{\partial \rho_k} = 2\rho_k \cos\omega_k \frac{\partial a_{i-1}}{\partial \rho_k} - \rho_k^2 \frac{\partial a_{i-2}}{\partial \rho_k} - 2\cos\omega_k a_{i-1} + 2\rho_k a_{i-2} \tag{3.44a}$$

for $2 \leq i \leq n$, $1 \leq k \leq m$ with boundary conditions

$$\frac{\partial a_0}{\partial \rho_k} = 0, \frac{\partial a_1}{\partial \rho_k} = -2\cos\omega_k \tag{3.44b}$$

where $n = p$ or q, m is the number of complex conjugate pairs of the roots.

The entries of the matrix $\xi_{13}(n)$ are calculated using the following recursion:

$$\frac{\partial a_i}{\partial \omega_k} = 2\rho_k \cos\omega_k \frac{\partial a_{i-1}}{\partial \omega_k} - \rho_k^2 \frac{\partial a_{i-2}}{\partial \omega_k} + 2\rho_k \sin\omega_k a_{i-1} \tag{3.45a}$$

for $2 \leq i \leq n$, $1 \leq k \leq m$ with boundary conditions

$$\frac{\partial a_0}{\partial \omega_k} = 0, \frac{\partial a_1}{\partial \omega_k} = -2\rho_k \sin\omega_k \tag{3.45b}$$

Similarly, the entries of $\xi_{22}(n)$ are calculated using (3.44) and those of $\xi_{24}(n)$ using (3.45), but with the coefficients b_k replacing a_k.

In the case of polar coordinates, computation of the coefficient vector $\hat{\eta}(n)$ from the parameter vector $\hat{\theta}(n)$ is accomplished by using the following recursion

$$a_k^{(j)} = a_k^{(j-1)} - 2\rho_j a_{k-1}^{(j-1)} \cos\omega_j + \rho_j^2 a_{k-2}^{(j-1)} \tag{3.46}$$

to find the AR coefficients from $\hat{\rho}^{(a)}(n)$ and $\hat{\omega}^{(a)}$. The same recursion is used to find the MA coefficients from $\hat{\rho}^{(b)}(n)$ and $\hat{\omega}^{(b)}(n)$.

Based on the above results, we now present the recursive Gauss-Newton RPE algorithm for ARMA pole-zero estimation as given below:

```
Initialize :$\hat{\theta}(0),\hat{\eta}(0),P(0),\Phi(1),\varphi(1),\varphi_F(1),w(1),w(\infty),w_0$
Main loop
Do for $n = 1,2,\cdots, N$
$x(n) = y(n) - \hat{\Phi}(n)\hat{\eta}(n-1)$
$L(n) = P(n-1)\Phi(n)/[w(n) + \Phi^T(n)P(n-1)\Phi(n)]$
$P(n) = [P(n-1) - L(n)\Phi^T(n)P(n-1)]/w(n)$
$\hat{\theta}(n) = \hat{\theta}(n-1) + L(n)x(n)$
*calculate $\hat{\eta}(n)$ from $\hat{\theta}(0)$
*calculate $\xi(n)$
$\bar{x}(n) = y(n) - \hat{\Phi}^T(n)\hat{\eta}(n)$
$y_F(n) = y(n) - \sum_{k=1}^{q} \hat{b}_k y_F(n-k)$
$x_F(n) = \bar{x}(n) - \sum_{k=1}^{q} \hat{b}_k x_F(n-k)$
*calculate $\varphi(n+1)$

*calculate $\varphi_F(n+1)$
$\varphi(n+1) = \xi^T(n)\varphi_F(n+1)$
$w(n+1) = w(\infty) - (w(\infty) - w(n))w_0$
```

where, the matrix $P(n)$ is defined as

$$P(n) = \left(\sum_{k=1}^{q} \Phi(n)\Phi^T(n) \right)^{-1} \tag{3.47}$$

and the vector $L(n)$ is the so-called gain vector of the recursive algorithm. The variables $\bar{x}(n)$ are the so-called a posterior prediction errors.

The recommended initial values for the algorithm are:

$P(0) = cI_d$ where c is some constant and I_d is the identity matrix.

$\theta(1) = \varphi(1) = 0$ the initial values for the pole estimates can be assigned using any prior knowledge concerning their likely position. In the absence of such knowledge, a possible choice is to distribute the poles uniformly around a circle inside the unit disk.

For stationary process, the recommended values for the forgetting factor are $w_0 = 0.99$, $w(1) = 0.95$ and $w(\infty) = 1$. Forgetting factor for nonstationary processes is usually set at a fixed constant value less than 1. The more rapid the parameter variation, the smaller (than 1) w becomes.

3.4.4 SIMULATION RESULTS

The performance of the adaptive pole-zero algorithm is investigated by using it to identify the roots of the following simulated process.

An ARMA process with order $(4,4)$ is simulated using the following difference equation

$$y(n) - 0.3y(n-1) + 0.69y(n-2) + 0.21y(n-3) + 0.45y(n-4) = x(n) + 0.5x(n-1) + 0.68x(n-2) + 0.62x(n-3) + 0.4x(n-4)$$

This process has poles at *0.9782 exp(± j1.0214)* and *0.6858 exp(± j 2.1248)*. The zeros are located at *0.9118 exp(± j1.2450)* and *0.6937 exp(± j2.4672)*. The transient convergence of the adaptive procedure are best seen by plotting the adaptive magnitude and phase of the adaptive poles and zeros as shown in Fig. 3.6. It can be seen that the algorithm converges rapidly to the true values of the poles and zeros. This is due to the fact of using the Gauss-Newton update with an exact expression for the gradient.

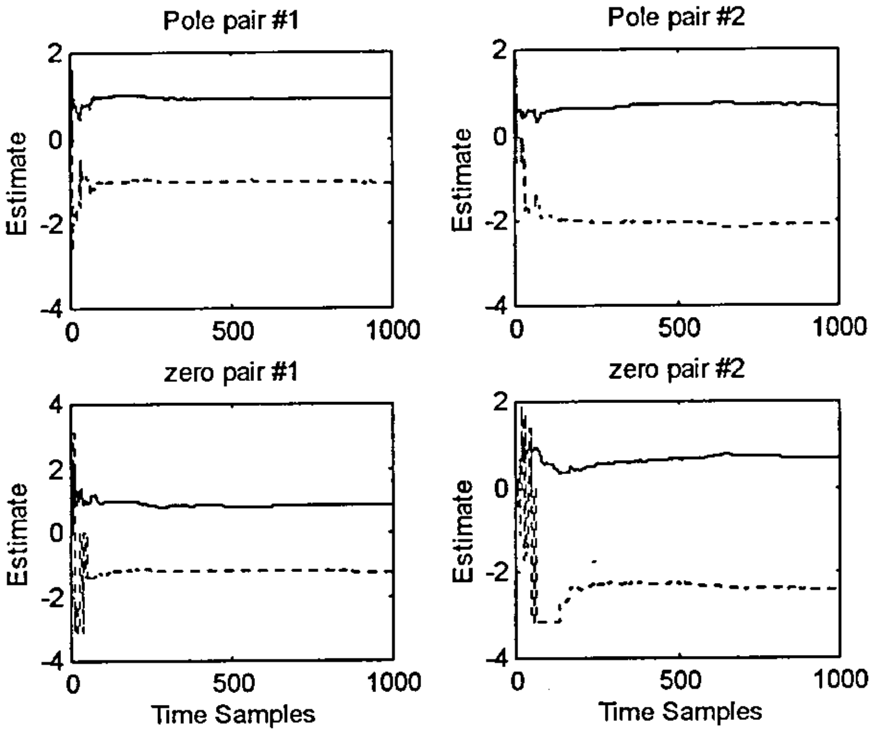


Fig. 3.6 Adaptive estimation for the magnitude (__) and phase (....) of the poles and zeros of the ARMA*(4,4)* process.

3.5 ORDER ESTIMATION OF THE ARMA MODEL

Estimating the parameters of the ARMA model is a major goal in system identification and signal modeling. Many techniques have been developed to solve this problem, but most of them assume prior knowledge of the model order (Ljung, 1987). In most realistic situation, the model order is not known and must be estimated prior to solving the parameter estimation problem.

Obviously, selecting the model order is a key first step toward the goal of estimating the model parameters. Several Criterions have been proposed for the model order selection task. The most well known of the proposed solution for this problem include the Final Prediction Error (FPE), Akaike Information Criterion (AIC), and Minimum Description Length (MDL) (Kay, 1988; Proakis *et al.*, 1992).

Model order determination techniques based on FPE, AIC, and MDL are computed from the prediction error variance, and are typically applied as in the following way:

First, the range of ARMA model orders to be considered is selected (e.g., $0 \leq p \leq p_{max}$ and $0 \leq q \leq q_{max}$). Next, for each *(p,q)* pair the numerator and denominator coefficients of the ARMA model are estimated under the assumption that p and q are the correct model orders. A prediction error variance of this model is then calculated. Finally, the *(p,q)* pair yielding the lowest value of the selected criterion is chosen as the best estimate of the true model order.

This approach suffers from the fact that, since the prediction error variances are used to compute the criteria, all of the parameter estimates must be first obtained in order to calculate the variances. Thus, in the next section a new method for determining the order of the ARMA model which does not require prior estimation of the model parameters is presented.

3.5.1 EIGENVALUE METHOD

In this section a new approach for model order determination based on the MDL criterion is proposed by Liang *et al.* (1993) and is shown to depend on the minimum eigenvalues of the covariance matrix derived from the observed data. The algorithm is derived as follows:

The ARMA model is given by the following difference equation:

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + \sum_{k=0}^{q} b_k x(n-k) \tag{3.48a}$$

In z-domain, it is represented as

$$Y(z) = H(z)X(z) = \frac{B(z^{-1})}{A(z^{-1})} X(z) \tag{3.48b}$$

The polynomials $A(z^{-1})$ and $B(z^{-1})$ are of degree p and q, respectively, given by

$$A(z^{-1}) = \sum_{k=0}^{p} a_k z^{-k} \tag{3.49}$$

$$B(z^{-1}) = \sum_{k=0}^{q} b_k z^{-k} \tag{3.50}$$

The relationship of (3.48a) can be rewritten in matrix format as (assuming the data length is N, that is, $n = 0, 1, 2, ..., N$-1):

$$\begin{bmatrix} y(0) & 0 & \cdots & 0 & x(0) & 0 & \cdots & 0 \\ y(1) & y(0) & \cdots & 0 & x(1) & x(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & \cdots & y(N-1-p) & x(N-1) & x(N-2) & \cdots & x(N-1-q) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \\ b_0 \\ b_1 \\ \vdots \\ b_q \end{bmatrix} = \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N-1) \end{bmatrix} \tag{3.51}$$

where *v(n)* is assumed to be zero-mean white Gaussian noise that represents any observation or modeling error. This may be written more compactly as

$$D_{pq}\,\theta_{pq} = v \tag{3.52}$$

in which D_{pq} is a composite data matrix, θ_{pq} is a parameter vector, and v is a noise vector. We define the data covariance matrix as

$$R_{pq} = D^{T}{}_{pq} D_{pq} \tag{3.53}$$

To estimate the input signal, *x(n)*, to be used in (3.51), the following procedure is performed:

It is well known that the observed output data can be modeled by a high order AR model (Kay *et al.*, 1981). Thus, (3.48) can be written as

$$\psi[y(n)] = x(n) + \psi_1[v(n)] \tag{3.54a}$$

In the z-domain,

$$\psi(z)Y(z) = X(z) + \psi_1(z)V(z) \tag{3.54b}$$

where ψ and ψ_1 are operators performed on *y(n)* and *v(n)* respectively.

The infinite order polynomial $\psi(z)$ is given by

$$\psi(z) = \frac{A(z^{-1})}{B(z^{-1})} = \sum_{n=0}^{\infty} \beta_n z^{-n} \approx \sum_{n=0}^{M} \beta_n z^{-n} \tag{3.55}$$

while $\psi_1(z) = 1/B(z^{-1})$.

If M is chosen large enough, a good approximation, $\psi(z)$, can be obtained using least squares approach. Then the input signal $\hat{x}(n)$ can be estimated using $\psi(z)$ as follows:

$$\hat{x}(n) = \sum_{i=0}^{M} \beta_i y(n-i) \tag{3.56}$$

where the β_i parameters are the elements of $\theta_2 = [\beta_1 \ \ \beta_2 \dots \beta_M]^T$ which is estimated by

$$\hat{\theta}_2 = \left[\frac{1}{N+1}\sum_{n=0}^{N}\phi_2(n)\phi_2^T(n)\right]^{-1}\frac{1}{N+1}\sum_{n=0}^{N}\phi_2(n)y(n) \tag{3.57}$$

where,

$$\phi_2(n) = [-y(n-1) \ \ -y(n-2) \dots -y(n-M)]^T$$

Thus, after the estimation of the input signal, $\hat{x}(n)$, it can be used now in (3.51) in the place of $x(n)$.

The MDL criterion is equal to the sum of the log-likelihood of the matrix likelihood estimator of the model parameters and a function that penalizes the use of the larger number of model parameters. The criterion is given by:

$$J_{MDL}(p,q) = -\log f(y|\theta) + \frac{1}{2} k \log N \tag{3.58}$$

Where,

$$y = [y(0) \ \ y(1) \dots y(N-1)]^T$$

and k is the number of free adjusted parameters *($k = p+q+1$)* in the model, $\hat{\theta}$ is the estimate of θ for a given p and q, and *f( )* is the probability density function of the observation noise/modeling error vector, $v =$ $[v(0)\ \ v(1) \ldots v(N-1)]^T$.

Since $v(n)$ is a zero-mean Gaussian noise with variance σ^2, we have

$$f(v) = f(y \mid \theta) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp[-\frac{1}{2\sigma^2} v^T v]$$
$$= \frac{1}{(2\pi\sigma^2)^{N/2}} \exp[-\frac{1}{2\sigma^2} \theta^T R_{pq} \theta] \tag{3.59}$$

then $J_{MDL}(p,q,\theta)$ reduces to

$$J_{MDL}(p,q,\theta) = \frac{N}{2}\log\sigma^2 + \frac{N}{2}\log 2\pi + \frac{1}{2\sigma^2}\theta^T R_{pq}\theta + \frac{1}{2}(p+q+1)\log N \tag{3.60}$$

For fixed p and q and constraining θ to have a unit norm, the choice of θ that minimize criterion (3.60) is found to be the eigenvector associated with the minimum eigenvalue $\lambda_{\min}$ of R_{pq}. With this choice of the $\theta = \theta_{min}$ we have

$$\frac{1}{N}\theta_{\min}^T R_{pq} \theta_{\min} = \frac{1}{N} v^T v = \frac{1}{N}\lambda_{\min} \approx \sigma^2 \tag{3.61}$$

substituting and dropping all terms that do not depend on p, q, or θ we obtain

$$J_{MDL}(p,q) = \frac{N}{2}\log\lambda_{\min} + \frac{1}{2}(p+q)\log N \tag{3.62}$$

The θ in the argument of J_{MDL} has been dropped, since the explicit θ dependence is suppressed (actually it has been incorporated into the $\lambda_{\min}$ term). Multiplying both sides by *2/N* and combining terms results in

$$\frac{2}{N} J_{MDL}(p,q) = \log\left[\lambda_{\min}(N^{1/N})^{(p+q)}\right] \tag{3.63}$$

Since log( ) is a monotonically increasing function, we can form a different criterion that contains exactly the same information as $J_{MDL}(p,q)$ (i.e., the new criterion has its minimum at the same place as $J_{MDL}(p,q)$. The new criterion is given by

$$J(p,q) = \lambda_{\min}(N^{1/N})^{(p+q)} \qquad (3.64)$$

From (3.64) it can be seen that for large N the $(N^{1/N})^{(p+q)}$ part is approximately one. This indicates that:

1- Model order selection is asymptotically simplified to examining the minimum eigenvalues of R_{pq} for different values of p and q.

2- The MDL criterion asymptotically (as N gets very large) provides no more information than the minimum eigenvalue of the covariance matrix R_{pq}. In other words, there is an asymptomatic equivalence between MDL and the minimum eigenvalue of this matrix.

If the noise/modeling error $v(n)$ is zero and the candidates orders p and q are chosen such that $p \geq n_a$ and $q \geq n_b$ (n_a and n_b are the actual model orders) then the $\lambda_{\min}$ will be zero, then we have $J(p,q)= 0$ if $p \geq n_a$ and $q \geq n_b$ which forms an infinite plane of zeros in the pq-plane. The smallest values of p and q such that $\lambda_{\min}$ is zero are $(p,q) = (n_a, n_b)$ and lie in the corner of this flat plane that is nearest the origin.

When the noise/modeling error $v(n)$ is not zero, a similar situation occurs. If p and q are chosen such that $p \geq n_a$ and $q \geq n_b$, $\lambda_{\min}$ will not be zero but it will tend to remain relatively small and flat for these values of p and q. If $p < n_a$ or $q < n_b$, the modeling error is significantly larger (because the model does not have enough parameters to fit the signal very well), making $\lambda_{\min}$ also significantly larger. This produces the same corner phenomenon at the correct values of $(p,q) = (n_a, n_b)$ described in the noiseless case above.

Therefore, we can calculate the table of $J(p,q)$ for all values of p and q, and then search for the corner where $\lambda_{\min}$ drops very quickly. We have observed that the corner that has the smallest $J(p,q)$ relative to $J(p-1,q)$ and $J(p,q-1)$ is the best estimate of the model order. That is, the corner for which the ratio

$$\frac{J(p,q)}{J(p-1,q)} \quad and \quad \frac{J(p,q)}{J(p,q-1)} \tag{3.65}$$

are the smallest values which provide the best choice of model order.

In order to facilitate the model order selection process, the following procedure was developed.

1- The $J(p,q)$ table is organized so that p increases from left to right, and q increases from top to bottom down the table.

2- Divide each row (element by element) of the $J(p,q)$ table by the previous row, so as to create a row ratio table.

3- Divide each column (element by element) of the $J(p,q)$ table by the previous column, so as to create a column ratio table.

4- From the row ratio table, an estimate of the order q is set equal to the row number that contains the minimum value of the row ratio table.

5- From the column ratio table, an estimate of the order p is set equal to the column number that contains the minimum value of the column ratio table.

3.5.2 SIMULATION RESULTS

To illustrate the idea of this new algorithm, a simulation was carried out on the following system which is an ARMA process of order *(6,4)* given by

$$y(n)+0.7907y(n-1)+0.042y(n-2)-0.5556y(n-3)-0.0247y(n-4)+0.3846y(n-5)$$
$$+0.3026y(n-6)=x(n)+0.3452x(n-1)+0.53x(n-2)+0.3985x(n-3)+$$
$$0.8138x(n-4)+e(n)$$

where, $x(n)$ is a white Gaussian noise with zero mean and unit variance, while $e(n)$ is an additive white Gaussian noise with zero mean and variance σ_e^2. The variance σ_e^2 was adjusted to set SNR = 20 dB.

The locations of the poles and zeros of the model is shown in Fig. 3.7 which shows that the zeros are closer to the unit circle than the poles and this require that the AR modeling order, M, to be large($M = 80$ is used here). Based on this information, the $J(p,q)$ table is constructed from which the row ratio and the column ratio tables are created as shown in Tables 3.1, 3.2 and 3.3.

The shaded cell in Table 3.1 is used to identify the best model order candidate, while the shaded cells in Tables 3.2 and 3.3 indicate the row or column that is the best candidate for the number of zeros (q) or poles (p) of the model. Thus, the true model order is obtained from the row and column ratios tables, i.e., $n_a = 6$ and $n_b = 4$.

For comparison purpose, the preceding process was simulated 25 times and the optimal order is obtained by applying the *Eigenvalue (EV)*, AIC and MDL methods.

For the AIC method the following criteria is used

$$AIC(p,q) = N \ln \sigma^2_{p,q} + 2(p+q)$$

in which the optimal order *(p,q)* is the one which gives the minimum value of AIC. While, for MDL method, the criteria used is

$$J_{MDL}(p,q) = (N/2) \log \sigma^2_{p,q} + 1/2\ (p+q)\log N$$

where the optimal order *(p,q)* is the one which gives the minimum value of MDL.

a) Comparing with respect to data length:

To demonstrate how the new method work for different data lengths compared with the other two techniques, different data lengths were taken in each run. The results are shown in Table 3.4, which gives the number of the correct order selection of the 25 trials for the preceding process. It can be seen that, the new method is better than the other two methods for different data lengths.

b) Comparing with respect to the effect of additive noise:

To study the performance of the new method at different SNRs, the preceding process was performed at SNRs ranging from 5 to 20 dB with 5 dB steps by adjusting the noise variance σ_e^2. For each SNR, 20 trials were run and the percent correct model order estimation was computed. This procedure is applied using the three algorithm and the results are shown in Fig. 3.8. The new method is seen to give more correctly model order estimate than the other two techniques in the case of different levels of noise.

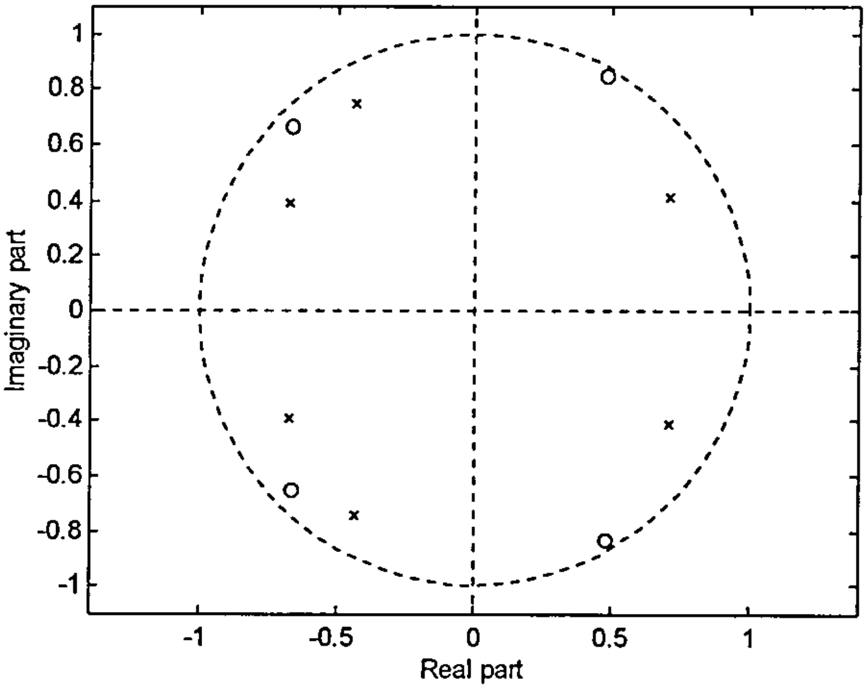


Fig. 3.7 Locations of the pole and zeros of the given ARMA*(6,4)* model.

Table 3.1 *J(p,q)* criterion

P q	0	1	2	3	4	5	6	7	8	9	10
0	135.4894	122.0512	120.4898	109.7341	109.2344	84.8486	54.8993	51.4400	45.5059	37.4408	36.7938
1	127.6560	123.3349	120.4963	111.3150	104.9612	86.4191	48.0179	48.8265	40.5373	30.9579	26.0795
2	99.1997	101.0499	99.9211	101.1514	94.4993	86.2791	48.5172	47.4815	39.6870	31.1439	24.0956
3	97.7253	83.2655	84.5393	84.6542	84.0770	82.3356	48.2288	47.3333	39.6222	29.0393	24.5523
4	94.2374	72.3109	60.3320	61.3658	61.6135	54.4267	$\boxed{30.2356}$	35.4650	32.1787	27.5780	24.9630
5	83.2057	71.2095	61.3143	62.3771	61.7034	54.6917	35.2637	34.8176	32.7981	27.8650	25.2469
6	82.7480	70.5455	61.6949	60.9741	61.9497	51.5577	35.0469	35.4243	26.8928	24.6659	23.9802
7	83.2520	71.8021	60.6955	60.5485	59.8738	46.0640	35.5772	35.7669	27.3881	22.9901	21.6938
8	74.3610	60.0466	56.4141	57.2952	57.0154	46.9218	29.9826	29.3432	24.1834	22.3070	22.1096
9	68.7246	61.1861	55.9919	57.0266	57.4138	46.5560	27.7306	27.7815	22.2859	19.7103	19.6642
10	68.1428	61.7898	57.0293	57.6545	58.2620	47.3769	27.4501	26.1553	21.5228	20.0677	19.4962

Table 3.2 Row ratio table

P q	0	1	2	3	4	5	6	7	8	9	10
1	0.9422	1.0105	1.0001	1.0144	0.9609	1.0185	0.8747	0.9492	0.8908	0.8269	0.7088
2	0.7771	0.8193	0.8292	0.9087	0.9003	0.9984	1.0104	0.9725	0.9790	1.0060	0.9239
3	0.9851	0.8240	0.8461	0.8369	0.8897	0.9543	0.9941	0.9969	0.9984	0.9324	1.0190
4	0.9643	0.8684	0.7137	0.7249	0.7328	0.6610	$\boxed{0.6269}$	0.7493	0.8121	0.9497	1.0167
5	0.8829	0.9848	1.0163	1.0165	1.0015	1.0049	1.1663	0.9817	1.0192	1.0104	1.0114
6	0.9945	0.9907	1.0062	0.9775	1.0040	0.9427	0.9939	1.0174	0.8199	0.8852	0.9498
7	1.0061	1.0178	0.9838	0.9930	0.9665	0.8934	1.0151	1.0097	1.0184	0.9321	0.9047
8	0.8932	0.8363	0.9295	0.9463	0.9523	1.0186	0.8427	0.8204	0.8830	0.9703	1.0192
9	0.9242	1.0190	0.9925	0.9953	1.0070	0.9922	0.9249	0.9468	0.9215	0.8836	0.8894
10	0.9915	1.0099	1.0185	1.0110	1.0148	1.0176	0.9899	0.9415	0.9658	1.0181	0.9915

Table 3.3 Column ratio table

P q	1	2	3	4	5	6	7	8	9	10
0	0.9008	0.9872	0.9107	0.9954	0.7768	0.6470	0.9370	0.8846	0.8228	0.9827
1	0.9662	0.9770	0.9238	0.9429	0.8233	0.5556	1.0168	0.8302	0.7637	0.8424
2	1.0187	0.9888	1.0123	0.9342	0.9130	0.5623	0.9787	0.8358	0.7847	0.7737
3	0.8520	1.0153	1.0014	0.9932	0.9793	0.5858	0.9814	0.8371	0.7329	0.8455
4	0.7673	0.8343	1.0171	1.0040	0.8834	$\boxed{0.5555}$	1.1730	0.9073	0.8570	0.9052
5	0.8558	0.8610	1.0173	0.9892	0.8864	0.6448	0.9873	0.9420	0.8496	0.9060
6	0.8525	0.8745	0.9883	1.0160	0.8323	0.6798	1.0108	0.7592	0.9172	0.9722
7	0.8625	0.8453	0.9976	0.9889	0.7694	0.7723	1.0053	0.7657	0.8394	0.9436
8	0.8075	0.9395	1.0156	0.9951	0.8230	0.6390	0.9787	0.8242	0.9224	0.9912
9	0.8903	0.9151	1.0185	1.0068	0.8109	0.5956	1.0018	0.8022	0.8844	0.9977
10	0.9068	0.9230	1.0110	1.0105	0.8132	0.5794	0.9528	0.8229	0.9324	0.9715

Table 3.4 Number of correct order estimates for the given example obtained by applying the Eigenvalue (EV), AIC and MDL methods for different data lengths.

Method	Data length (*N*)				
	150	300	500	1500	2000
AIC	9	10	10	12	15
MDL	13	13	16	18	19
EV	15	19	21	22	22

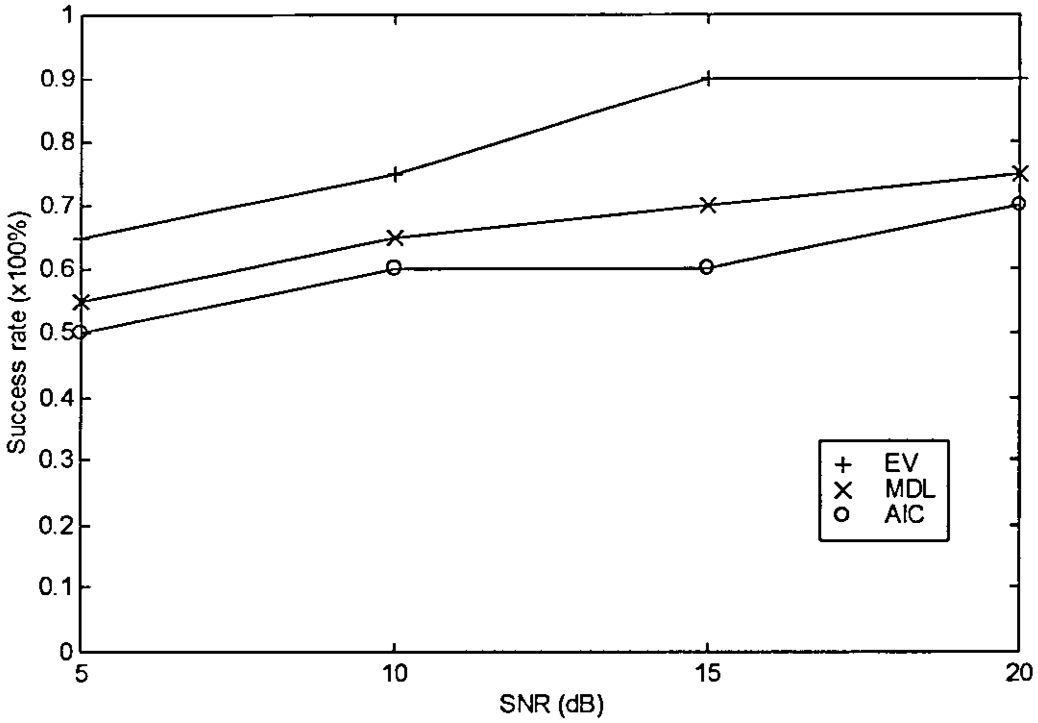


Fig. 3.8 Effect of the additive noise on the model order estimate obtained by applying the Eigenvalue (EV), AIC and MDL methods for different SNRs.

3.6 ARMA MODEL WITH MULTIPLE INPUTS AND DELAYS

This section presents a new approach to ARMA modeling which automatically seeks the best model order to represent investigated linear, time invariant systems using their input/output data. In this algorithm, which is proposed by Perrott *et al.* (1996), estimation results of an initial, over-parameterized model are incorporated to produce a set of lower order candidate models. Comparison of the models in this set leads to the selection of a model order to represent the system.

3.6.1 ARMA PARAMETER REDUCTION (APR) ALGORTHIM

For convenience, we will limit discussion to the two input ARMA model, the extension of the method to systems with more inputs should be clear.

Consider a two input LTI system whose input are $x_1(n)$ and $x_2(n)$, and whose output is $y(n)$. The corresponding ARMA model is denoted as:

$$y(n) = -\sum_{i=1}^{p} a_i y(n-1) + \sum_{k=s1}^{q_1} b_{1k} x_1(n-1) + \sum_{j=s2}^{q_2} b_{2j} x_2(n-1) + e(n) \qquad (3.66)$$

The sequence $e(n)$ is assumed to be white Gaussian noise with variance $E[e(n)^2] = \sigma^2$.

In the z-plane, the ARMA model can be written as:

$$A(z)y(z) = B_1(z)x_1(z) + B_2(z)x_2(z) \qquad (3.67)$$

in which we have defined the following:

$$\begin{aligned} B_1(z) &= \sum_{k=s1}^{q_1} b_{1k} z^{-k} \\ B_2(z) &= \sum_{k=s2}^{q_2} b_{2k} z^{-k} \\ A(z) &= \sum_{k=0}^{p} a_k z^{-k} \end{aligned} \tag{3.68}$$

From (3.67), we define the input output transfer function relationships

$$H_1(z) = \frac{y(z)}{x_1(z)} = \frac{B_1(z)}{A(z)}; \qquad H_2(z) = \frac{y(z)}{x_2(z)} = \frac{B_2(z)}{A(z)} \tag{3.69}$$

In attempting to represent the given physical system with such a model, two fundamental issues arise:

1) Estimation of the model order (i.e., the values of $\{p, s_1, q_1, s_2, q_2\}$ in (3.66))

2) Estimation of the parameter values once the model order is known.

As mentioned earlier, several different algorithms have been used that attempt to identify the best ARMA model order of an investigated physical system. Unfortunately, none of them consider systems with multiple inputs or delays (having s_1 and s_2 $\neq 0$ in (3.66)). Thus, this new algorithm that automatically estimates the ARMA model order associated with systems that are allowed to have multiple inputs delays.

In this method, the MDL criterion which makes use of 'residual error norms' is used to evaluate a given model's performance in comparison to other models. Its value is given by:

$$MDL(k) \equiv \left(1 + k\frac{\log N}{N}\right)\frac{\|\varepsilon_k\|^2}{N} \tag{3.70}$$

In the above formulation, k is defined to be the number of parameters contained within the model, ε_k, the prediction error sequence, and $\|\varepsilon_k\|$ is the residual error norm. When a set of candidate models are compared to each other, the model whose MDL value is less than the MDL values of the other models in the candidate set is said to perform better than these models. Thus, selection of a model using the MDL criterion amounts to finding the model among the candidate set which has the minimum MDL.

For a selection set of candidate models to be compared, we first mention the standard assumptions made for selecting ARMA models to represent systems without delay, namely,

AR parameters:

$a_i \neq 0$ for $i \in [1, p]$

$a_i = 0$, elsewhere.

MA parameters:

$b_{1k} \neq 0$ for $k \in [0, q_1]$

$b_{2j} \neq 0$ for $j \in [0, q_2]$

$b_{1j} = b_{2k} = 0$, elsewhere.

Unfortunately, the above assumption made on the MA parameters in ARMA modeling break down for systems with delays. In such cases, an appropriate set of constraints for these parameters is:

$b_{1k} = 0$ for $k \notin [s_1, q_1]$

$b_{2j} = 0$ for $j \notin [s_2, q_2]$

Here we not only lift the previous restriction that $s_1 = s_2 = 0$, but we also allow b_{1k} and b_{2j} to be zero within the range $k \in [s_1, q_1]$ and $j \in [s_2, q_2]$, respectively. Thus we allow for the nonzero MA parameters to be some combination of the total number of MA parameters contained in the sets $k \in [s_1, q_1]$ and $j \in [s_2, q_2]$.

Under the new MA constraint, the user now has to worry about varying five parameters, namely, $\{p, s_1, q_1, s_2, q_2\}$.

Referring back to the ARMA model in (3.66), define a 'data vector' associated with any specified $y(n)$ as follows:

$$\phi_t(n)^T = [x_1(n-s_1)\cdots x_1(n-q_1)x_2(n-s_2)\cdots x_2(n-q_2)y(n-1)\cdots y(n-p)] \qquad (3.71)$$

also, define a 'parameter vector' as:

$$\theta_t^T = [b_{1s1}\cdots b_{1q1}b_{2s2}\cdots b_{2q2}a_1\cdots a_p] \qquad (3.72)$$

where t corresponds to the dimension of θ_t (i.e., to the number of the nonzero parameters in the ARMA model).

Making use of these definitions, we rewrite (3.66) as:

$$y(n) = \phi_t(n)^T \theta_t + e(n) \qquad (3.73)$$

For N samples of the input and output sequences, we have:

$$Y = \begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix}; \quad \Phi_t = \begin{bmatrix} \phi_t(0)^T \\ \phi_t(1)^T \\ \vdots \\ \phi_t(N-1)^T \end{bmatrix}; \quad E = \begin{bmatrix} e(0) \\ e(1) \\ \vdots \\ e(N-1) \end{bmatrix} \qquad (3.74)$$

Using these definitions, the relationship between input, output and noise data is written for the ARMA model as:

$$Y = \Phi_t \theta_t + E \qquad (3.75)$$

which represents the actual 'true' parameterization of an investigated physical system.

Since the parameters contained within θ_t, along with their values, are unknown in the context of system identification, it is necessary to specify an equation which describes an 'estimation model' selected by the user, namely,

$$Y = \Phi_k \hat{\theta}_k + \varepsilon_k \qquad (3.76)$$

In the above equation, θ_k represents the ARMA parameter estimates, and ε_k the corresponding prediction error sequence.

In order to find the estimation model for which $k = t$, an approach that incorporates estimation results from an arbitrary, specified model is proposed. Thus, we initially choose k as some arbitrary, large number such that $k \geq t$ and define this as the 'maximal model'. A technique will be presented that attempts to remove the extra parameters from θ_k.

Given that the initial value of k has been chosen, the parameter estimates $\hat{\theta}_k$ are calculated with the least squares procedure (Kay, 1988). This operation is described compactly for the ARMA model as:

$$\hat{\theta}_t = (\Phi_k^T \Phi_k)^{-1} \Phi_k^T Y \qquad (3.77)$$

If we are to compare the chosen maximal model with the true model, it will be necessary to augment the θ_t vector with the extra parameters contained within $\hat{\theta}_k$. For this purpose, (3.75) is modeled as

$$Y = \Phi_k \hat{\theta}_k + E \tag{3.78}$$

combining (3.77) and (3.78) we obtain

$$\hat{\theta}_k = \theta_k + v_k \tag{3.79}$$

where

$$v_k = (\Phi_k^T \Phi_k)^{-1} \Phi_k^T E$$

is a corrupting noise vector whose autocovariance matrix is

$$E[v_k v_k^T] = \sigma^2 (\Phi_k^T \Phi_k)^{-1} \tag{3.80}$$

The diagonal elements within the above matrix correspond to the variance, or 'average energy' of the individual elements within v_k, which, in general, varies from parameter to parameter.

Our concern lies in determining which parameters within $\hat{\theta}_k$ are true (i.e., have an actual value that is nonzero). Intuitively, the larger in magnitude a parameter estimate is, the less likely it has an actual value of zero. However, each of the estimate was shown to be corrupted by noise whose average energy varies from parameter to parameter (3.79). Therefore, if we are to compare parameter estimates, it is more appropriate to first normalize their values by the level of the corruption occurring from noise. So, if we think of the true parameter component θ_k in each estimates as being the signal, and v_k as being the noise, then a signal to noise (SNR) ratio can be formed for each individual parameter as follows:

$$SNR(\hat{\theta}_k(i)) \equiv \sqrt{\frac{\theta_k(i)^2}{E[v_k(i)^2]}} \approx \sqrt{\frac{\hat{\theta}_k(i)^2}{E[v_k(i)^2]}} \tag{3.81}$$

where i designates the i^{th} element of the vector θ_k. One can see immediately that the SNR ratio effectively normalizes each estimate by the amount of corruption occurring from noise. Therefore, we propose to use the SNR of each parameter to compare the likelihood of each parameter being 'true'. Parameter estimates with high SNRs will be considered more important than those with low SNRs.

To develop a consistent model selection procedure, we need to consider AR and MA parameters separately in terms of evaluating their relative likelihood of being true. The algorithm which is referred to as the *ARMA Parameter Reduction (APR)* is summarized below:

1) Select a 'maximal model', a model that is believed to include all the true parameters of the system.
2) Remove the AR parameters from the maximal model one at a time starting from the highest index, while retaining all the MA coefficients in the maximal model (i.e., decrease p one value at a time), creating a set of lower order models.
3) Use an evaluation method (i.e., MDL) to choose the best performing 'reduced model' from among the set given by 2, that is, the MDL is used to compare the performance of each candidate model, and the best performing one is selected and defined as 'reduced model'. Note that this step implicitly estimates the AR order.

4) Using the 'reduced model' parameterization, the SNR ratios of the MA coefficients are calculated and then used to create another candidate set of models.

5) To form this new set, MA parameters are removed one by one from the reduced model according to their SNR ratios (i.e., MA parameters with low SNR ratios are removed before those with high SNR ratios). Thus, each model within this new candidate set contains the same AR coefficients but progressively fewer MA parameters.

6) Use the MDL criterion to compare the performance of each candidate model from the set given by (5), and the model with best performance is chosen and defined as the 'minimal model'. Note that this step estimates the MA order, thereby estimating the overall ARMA order.

7) The minimal model is the algorithm's best guess of the true model of the system.

3.6.2 SIMULATION RESULTS

To illustrate the above method, the following example is performed. Consider a system with delays given by the following ARMA process

$$y(n) = 1.2y(n-1) - 0.35y(n-2) - x_1(n-3) + x_2(n-1) - 1.3x_2(n-4) + e(n)$$

The model parameters are

AR parameters:

p=2, $a_1 = -1.2$, $a_2 = 0.35$

MA parameters:

s_1= 3, q_1 = 3, $\{b_{13}\}$ = -1

s_2= 1, q_2 = 4, $\{ b_{21}, b_{22}, b_{23}, b_{24}\}$ = {1, 0, 0, -1.3}

where $x_1(n)$ and $x_2(n)$ are white Gaussian noise with zero mean and unity variance and $e(n)$ is also Gaussian noise with zero mean and variance σ^2_e adjusted to give 0 dB.

To apply the APR algorithm, the following steps are performed

1- A 'maximal model' is first chosen which includes the true model. It is defined as $\{p, s_1, q_1, s_2, q_2\}$ = {5, 0, 5, 0, 5}. Thus, the maximal model has a total of 17 parameters.

2- A 'reduced model' is obtained by making use of the MDL values applied to the various models generated when the AR parameters of the maximal model are removed one by one. The results of the MDL criterion is shown in Fig. 3.9 (a) from which we can see that the minimum MDL occurred when the number of the parameters is equal to 14 (i.e., three of the AR parameters are removed). Thus the 'reduced model' is given by $\{p, s_1, q_1, s_2, q_2\}$ = {2, 0, 5, 0, 5}. From this step, the AR order is found to be $p = 2$.

3- The SNR for each MA parameter in the reduced model is calculated, and the results are shown in Fig. 3.9 (b) and (c) from which we can see that the parameters b_{13}, b_{21} and b_{24} have the highest SNR comparing to the other parameters. Thus, they are the most likely parameters to be true.

4- A 'minimal model' is obtained by making use of the MDL values applied to the various models generated when the MA parameters of the reduced model are removed one by one. The results of the MDL criterion is shown in Fig. 3.9 (d) which shows that the minimum MDL occurred when the number of the parameters is equal to 5 in which they construct the minimal model.

As found in step 2, the order of the AR part is $p = 2$, thus, the remaining three parameters in the minimal model belong to the MA part. Furthermore, since b_{13}, b_{21} and b_{24} have the highest SNR as shown before, the three MA parameters contained in the minimal model correspond to these parameters. This leads to estimate the delays to be $s_1 = 3$ and $s_2 = 4$.

Finally, the minimal model order for the system was selected as

AR parameters:

$p = 2$, with $a_1 = -1.33$ and $a_2 = 0.51$

MA parameters:

$s_1 = 3$, $q_1 = 3$, $\{b_{13}\} = -1.22$

$s_2 = 1$, $q_2 = 4$, $\{b_{21}, b_{22}, b_{23}, b_{24}\} = \{0.85, 0, 0, -0.98\}$

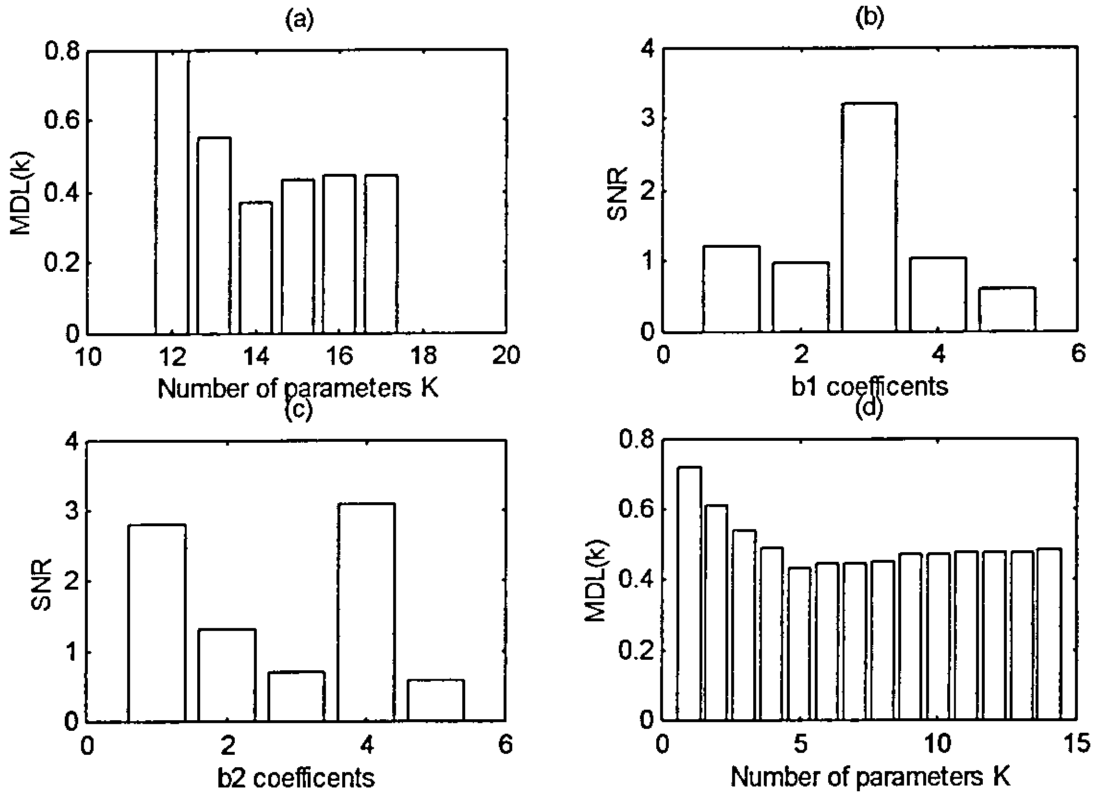


Fig. 3.9 (a) MDL criterion applied to the models produced by removal of AR parameters from the maximal model. (b) SNRs obtained for the MA parameters of the first input system (b_{1k} parameters). (c) SNRs obtained for the MA parameters of the second input system (b_{2k} parameters). (d) MDL criterion applied to the models produced by removal of MA parameters from the reduced model.

3.7 EFFICIENCY OF PARAMETRIC METHODS

In spite of its many successful applications, a parametric method is efficient only when its limitations and basic assumptions are acceptable. They are applicable if the result of inverse filtering of the EEG would be white Gaussian noise. A parametric model can be regarded as being efficient if the residual coming from inverting the fitted model constitutes a white noise process. The procedure for obtaining the residual, *e(n)*, of a parametric method (ARMA or AR model) is shown in Fig. 3.10.

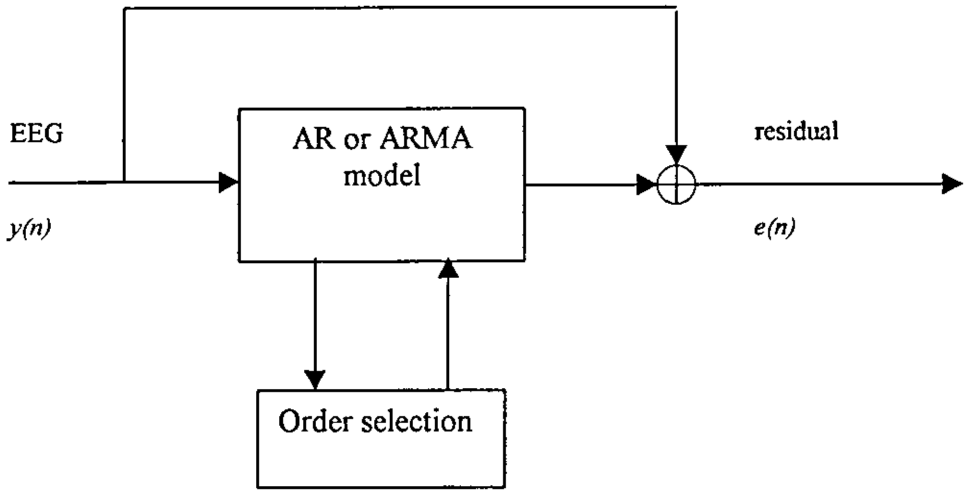


Fig. 3.10 Block diagram for obtaining the residual sequence from the fitted ARMA or AR model.

The testing procedures are described below. The following tests have been used on the residual for testing the efficiency of the selected model (Popinanov, 1992).

1- *Test 1:* A white noise process is uncorrelated with itself for all lags, except the lag equal to zero. In order to compare two time series with different scales of measurements, the normalization of the autocovariance, *R(k)*, is necessary. The alternative covariance function, *r(k)*, can be obtained by dividing the autocovariance estimates by the estimate of the variance of the residual signal. Thus,

$$r(k) = \frac{R(k)}{R(0)} \tag{3.85}$$

where,

$$R(k) = \frac{1}{N}\sum_{i=1}^{N-k}(e(i) - \bar{e})(e(i+k) - \bar{e}) \tag{3.86}$$

To test for the lack of 'local correlation' the *r(k)* must be within the range limited by $\pm\lambda/\sqrt{N}$ for k = 1, 2, ..., *N*/2, with λ = 1.96 fo. significance level $\rho<$ 0.05 (95% significance interval). Where the decision rule is that the alternative covariance function is acceptable if $-\lambda/\sqrt{N} \le r(k) \le \lambda/\sqrt{N}$.

2- *Test 2:* A white noise should have a constant power spectrum. The criterion of integrated spectrum has been used for detecting sine terms, i.e. the departures from whiteness caused by periodic effects. The cummulative periodogram is used here for the sample integrated spectrum computed at the harmonic frequencies $\omega_k = 2\pi k/N$. The normalized cummulative periodogram is used as a test for statistic *g(k)*, *k* = 1, 2, ..., *N*/2 computed according to

$$g(k) = \frac{\sum_{i=1}^{k} C_i^2}{\sum_{i=1}^{N/2} C_i^2} \qquad (3.87)$$

with

$$C_i^2 = \left(\frac{2}{N} \sum_{j=1}^{N} e(j)\cos(\omega_i . j) \right)^2 + \left(\frac{2}{N} \sum_{j=1}^{N} (e(j)\sin(\omega_i . j) \right)^2 \qquad (3.88)$$

To prove the efficiency of the selected model, i. e., the lack of the sine term in the residual, *g(k)* must be within the area limited by two straight lines

$$f = 2k/N \pm \lambda/\sqrt{N-1} \qquad \text{for } k = 1, 2, \ldots, N/2 \qquad (3.89)$$

with $\lambda = 1.35$ for significance level $p < 0.05$. The decision rule is that the normalized cumulative periodogram is acceptable if the following is achieved

$$2k/N - \lambda/\sqrt{\frac{N}{2} - 1} \leq g(k) \leq 2k/N + \lambda/\sqrt{\frac{N}{2} - 1} \qquad (3.90)$$

3.8 SIMULATED EEG

Because of the practical limitations of having a real EEG data to use them as a useful tool to investigate the performance of the various algorithms developed earlier, a simulation technique is used to simulate different types of EEG signals. Thus, algorithms used in this thesis will be tested on a simulated EEG data. The simulation technique used here is the one that is described by Weiss (1986) in which the simulated EEG signals were generated by linearly combining second order AR series. A summary of this technique is described below:

The EEG process is characterized by peaked spectra at one or more resonance frequencies. Each peak spectra has its own frequency location, bandwidth, and relative power (relative amplitude). Thus each peak in the spectrum is described by a second order AR model, then the total signal can be represented as the summation of these individual second order systems, i.e.,

$$EEG(n) = \sum_{k=1}^{M} \left(a_{1k} y(n-1) + a_{2k} y(n-2) + a_{3k} x(n) \right) \tag{3.91}$$

where the a_{jk}*'s* are the parameters of the second order AR models, M is the number of the resonances presented and *x(n)* is a white Gaussian noise. Furthermore, the power spectral density of the simulated signal is the linear combination of the spectra of each peak.

Thus, the actual simulation procedure involves the selection of the location, the bandwidth and the relative amplitude of each peak. This leads to different values of the parameters a_1, a_2 and a_3 and as a result, various types of EEG signals can be simulated.

Fig. 3.11 shows different types of simulated EEG signals with their corresponding power spectral density. These simulated EEG signals will be used in the next chapter to study the performances of the ARMA modeling techniques in analyzing the EEG signals.

After we have introduced the various aspects of the ARMA modeling, we will show in the next chapter how we can use the ARMA model to extract an detect important features of the EEG signals and how we can avoid the situation of having some zeros of the ARMA models outside the unit circle.

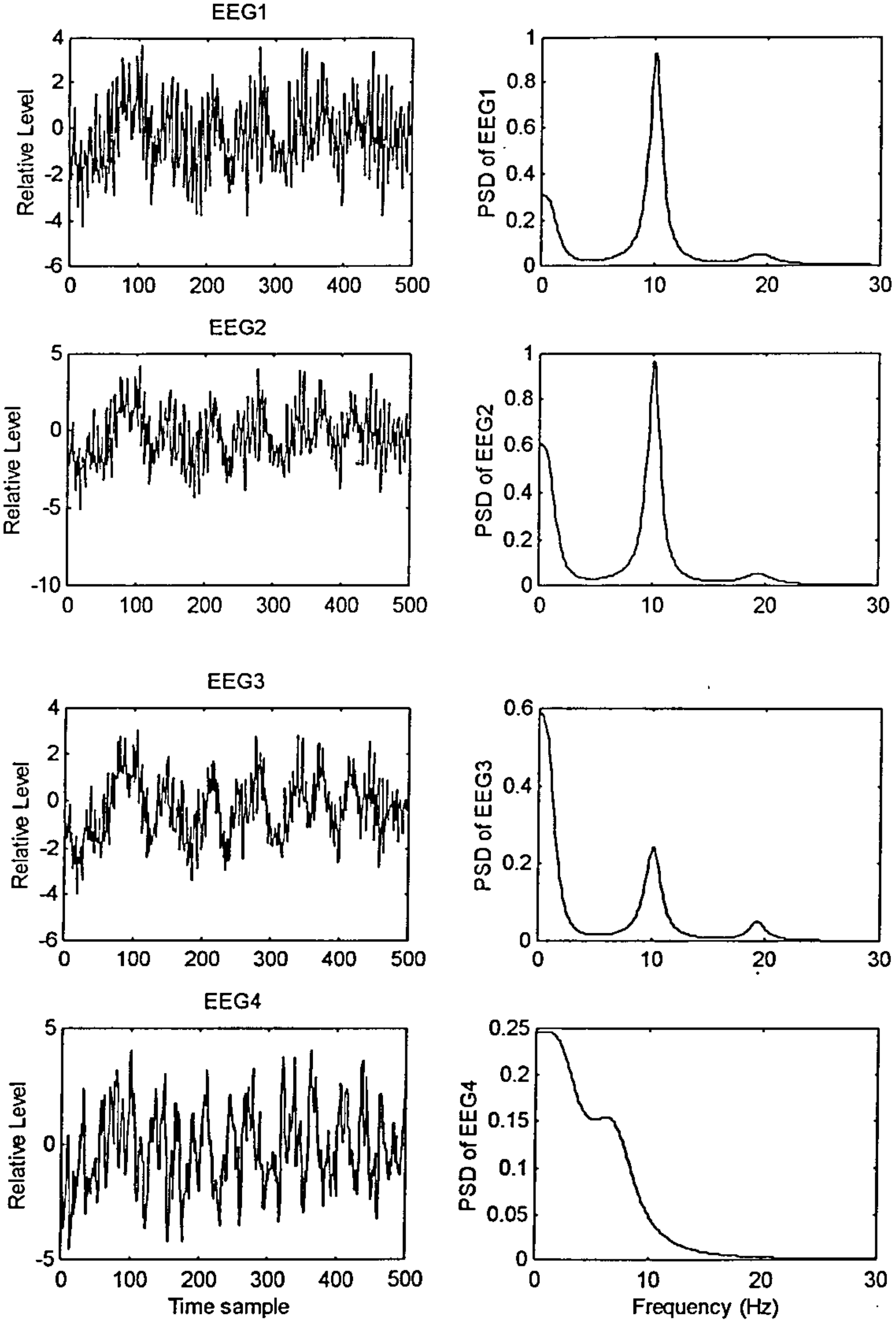


Fig. 3.11 Different types of simulated EEG signals with their corresponding power spectral density.

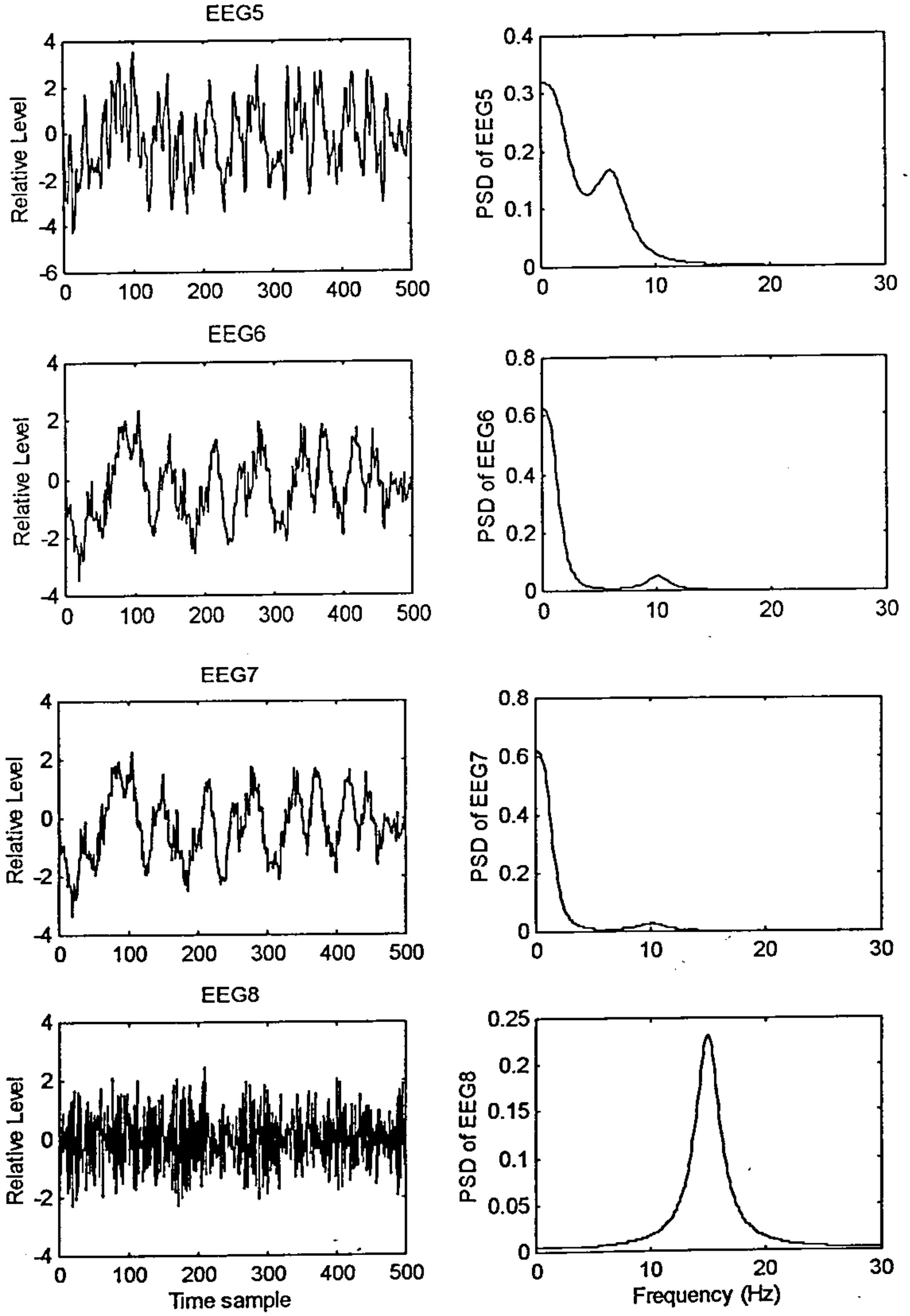


Fig. 3.11 (continued)

CHAPTER 4

EEG FEATURE EXTRACTION USING ARMA MODEL

The quantitative analysis of the EEG signals has been one of the most significant advances in the field of electrophysiology. Its primary purpose is to supply the neurophylogist with information that will complement the visual evaluation and hence improve the judgment. A somewhat idealized procedure may be described in three steps:

1- Appropriate sections are selected from a recording for further analysis.

2- Extraction of characteristic features of the process relevant to the purpose of analysis.

3- Classifying sections of EEG recordings and identifying the appropriate set of classes.

Thus, quantification of the EEG using computers is based on complex subjective processes of data reduction and feature extraction.

A common processing strategy used in computerized EEG is spectral analysis. EEG waveforms can be broken down into separate components, each component having a different frequency. This decomposition of the waveform results in a frequency spectrum, which yields a distribution of amplitudes as a function of frequency for a given sample of EEG data. The processed EEG data presented is based on power spectral analysis. The amplitude or power of each sine wave component of the EEG waveform is presented as a function of frequency, assuming no interaction between waves. This presents neuroelectric data as epochs of specific, definable information, which may be easier to interpret than raw EEG for less experienced personnel.

Spectral EEG data allows precise characterization of the frequency composition of brain electrical activity for a given period of time. Typical measures that are derived from spectral analysis of the EEG include: absolute power which is the amount of energy within a particular frequency expressed in microvolts squared, relative power which is the percentage of the total power spectrum that is present in each frequency band, and mean frequency which specifies the average frequency occurring in each frequency band.

As we have seen, the parametric models represent samples of the EEG signals through the linear relation given by (3.1). We have to show the validity of (3.1) as a model describing the rhythmic properties of the EEG. Thus, the next section will be devoted to the discussion of how a variety of important EEG signal parameters can be reliably estimated by using ARMA modeling.

4.1 FEATURE EXTRACTION

In parametric methods, one difficulty lies in interpreting the model parameters in terms that are familiar to the neurophysiologist. Presentation is often done by calculating the spectral density and plotting the results. Attempts have been made to describe the spectral density in terms that are closely related to the quantities used in the clinical neurophysiology. In this way, the spectral density may be partitioned into spectral components called δ, α, and β as shown in Fig. 4.1 (Isaksson *et al.*, 1981; Smith *et al.*, 1986). This is known as Spectral Component Parameter Analysis (SPA) of the background EEG. This technique resolves the power spectral density into a small set of smooth, peak-shaped "spectral components" each characterized by parameters giving its location in frequency, its width, and its area or power. This method represents the spectrum concisely and the spectral components parameters are directly related to the common terms used by the neurophysiologist to describe the EEG. The peak frequencies are denoted f_α and f_β and the bandwidth parameters σ_α, σ_β and σ_δ while the power parameters are written G_α , G_β and G_δ and expressed either in percentage or in absolute values.

The most direct approach to SPA is to fit an ARMA model to the EEG and then to compute the spectral component parameters from the ARMA coefficients. Processes characterized by ARMA models have spectra that exhibit clear peaks over a background of more widespread power contributions with possibly some "missing" frequencies. The spectral peaks are associated with the poles of the rational transfer function, and the valleys (or missing frequencies) with the zeros. In other words, the peaks in the power spectrum are characterized by the AR branch, and the valleys by the MA branch.

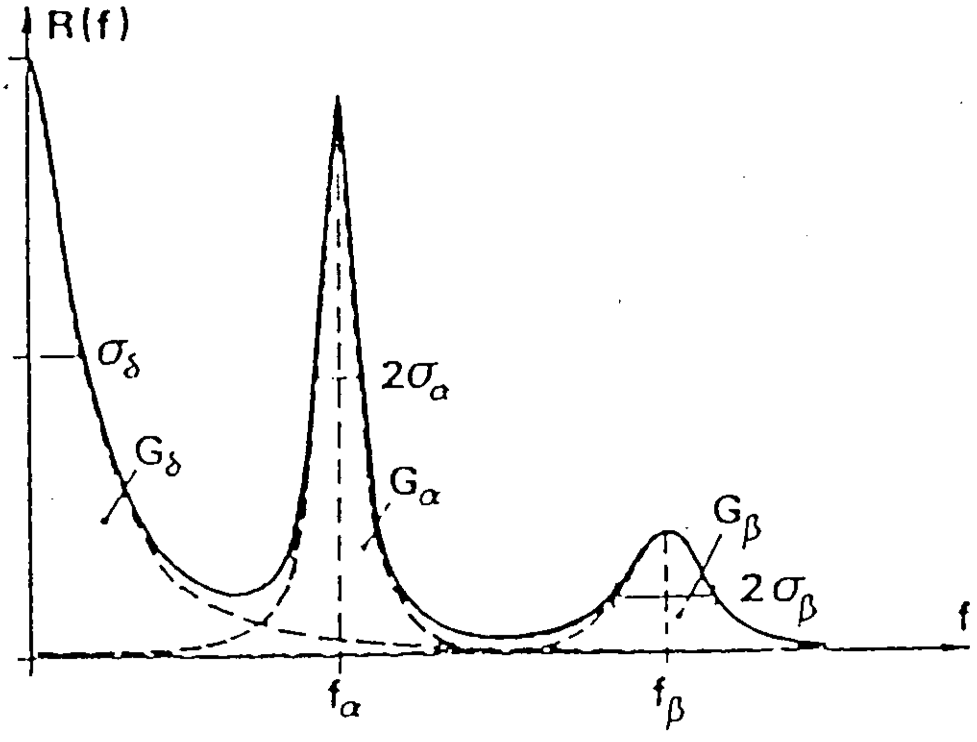


Fig. 4.1 Decomposition of the EEG spectrum into three spectral components corresponding to delta, alpha and beta activities. SPA parameter indicates. Dashed lines represent each component, while the solid line is the sum of all three components. (Isaksson *et al.*, 1981).

The number of peaks in the spectrum depends on the number of coefficients. The location of these peaks is determined by the position of the complex poles relative to the origin, and the peaks become more pronounced when the poles occur closer to the unit circuit.

4.1.1 SPECTRAL PARAMETER ANALYSIS

SPA assumes that the autocorrelaton of the EEG has the form of a sum of complex exponential:

$$R_{yy}(\tau) = \sum_{i=1}^{p} G_i e^{s_i|\tau|} \tag{4.1}$$

The real and imaginary parts of the poles $s_i = \sigma_i + j2\pi f_i$ give a total of $2p$ independent parameters. The one-sided power spectrum corresponding to each complex conjugate pair of terms and each real term in (4.1) has the form of smooth peak, called "spectral component". The EEG power spectrum density is the algebraic sum of these spectral components. The location in radian frequency of a spectral component is given by the imaginary part $(2\pi f_i)$ of the associated pole. A spectral component at zero frequency has a radian frequency half-power bandwidth equal to the negative of the pole's real part (σ_i) and a power equal to the associated real residue value G_i. A spectral component a way from zero frequency has half-power bandwidth of $-2\sigma_i$ and power $2\text{Re}(G_i)$.

An autocorrelation of the form in (4.1) is equivalent to an ARMA model of uniform samples of the EEG signal, $y(n) = y(nT)$, having the form

$$y(n)+ a_1y(n-1) +...+ a_p\, y(n-p) = b_0\, x(n) + b_1\, x(n-1) +...+ b_q\, x(n-q) \qquad (4.2)$$

As defined earlier, this model has transfer function

$$H(z)=\frac{B(z^{-1})}{A(z^{-1})} \qquad (4.3)$$

with

$$A(z^{-1}) = 1 + a_1 z^{-1} + ... + a_p z^{-p}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + ... + b_q z^{-q.}$$

In order to make the model stable it is required that the related polynomial

$$z^p A(z^{-1}) = z^p + a_1 z^{p-1} + ... + a_p \qquad (4.4)$$

has all its zeros (i.e, poles of the transfer function) inside the unit circle.

The spectral component parameters in (4.1) are directly related to the 2p ARMA coefficients in (4.2). The poles of (4.2) are given by the roots p_i of the polynomial

$$z^p + a_1 z^{p-1} + ... + a_p = 0 \qquad (4.5)$$

and related to the s_i in (4.1) by

$$p_i = e^{s_i T} \qquad (4.6)$$

Evaluating the expression in (4.5) at the unit circuit $z = e^{j\omega T}$, we get

$$exp(pj\omega T) + a_1\, exp[(p-1)j\omega T]+... + a_p = 0 \qquad (4.7)$$

Equation (4.7) can be factored and written in the form

$$(exp(j\omega T)-p_1)\ (exp(j\omega T) -p_2) ...\ (exp(j\omega T)-p_p) = 0 \qquad (4.8a)$$

That is,

$$(exp(j\omega_i T)-p_i) = 0 \qquad (4.8b)$$

Thus, the value of ω_i for which $\exp(j\omega_i T)$ match the phase of p_i represents the resonance frequency (the frequency at which there is a peak in the frequency spectrum of the data sequence) and it is known as the dominant frequency. Hence, the dominant frequencies, ω_i, are given by the phase of the roots of (4.8)

$$\omega_i = \frac{1}{T}\angle p_i = \frac{1}{T}\mathrm{Im}(\ln(p_i)) \qquad (4.9a)$$

and

$$f_i = \frac{1}{2\pi T}\mathrm{Im}(\ln(p_i)) \qquad (4.9b)$$

The bandwidth is given by

$$BW = \frac{1}{2\pi T}\ln(p_i) \qquad (4.10a)$$

for a spectral component at zero frequency, and by

$$BW = \frac{1}{\pi T}\ln(p_i) \qquad (4.10b)$$

for a spectral component a way from zero frequency.

The power at the dominant peak is given by the area under the peaks in the power spectrum between two cutoff frequencies. One way to obtain this power is to integrate the power spectral between the desired frequencies. Another method is to find the power at the peaks based on the residues (Res)

$$Pow(f_i) = 2Re\ (Res(p_i)) \qquad (4.11)$$

From the past discussion it is evident that the form of $A(z)$ is important in our development because it reflects the resonance structure of the spectral data. In (4.5) the poles p_i may be real or occur as complex conjugate pairs.

The real roots correspond to near DC (if p_i is +ve) or near $f_s/2$ (if p_i is -ve) while each complex pole-pair of p_i corresponds to a resonance (rhythm) whose resonance frequency and bandwidth are given in (4.9) and (4.10).

We have shown that the spectral analysis using ARMA models provides quantitative information on peak frequency, its associated power and bandwidth. The intended use of SPA is the on-line computation and display of a concise, clear trend plot of the EEG spectral information. SPA not only provides a summary of the EEG in convenient graphic form, but also facilitates statistical analysis of EEG effects which may not be evident by inspection. It provides a method of reducing selected EEG segments to a convenient quantitative summary.

4.1.2 SIMULATION RESULTS

To see how the ARMA model can be used to represent and extract important features of different classes of EEG signals, the following two examples are performed. Example # 1 shows a comparison between the ARMA and AR models in representing the EEG signals. While in example #2, the SPA technique is tested.

Exmple #1:

Four classes of simulated EEG waveforms known as, delta, theta, alpha, and beta are used to compare the performance of the ARMA model in modeling the different activities of the EEG signals with that of the AR model.

For each EEG class, different 30 records are created using the EEG simulation techniques described in section 3.8. Each record has a data length of 500 sample points. These records are fitted to both the ARMA and AR models. The EEG classes and their corresponding PSD are shown in Fig. 4.2 for a specific one record of each class.

a) Model order comparison:

The results of the optimal order of ARMA and AR modeling are represented in Tables 4.1 and 4.2, respectively. The order of the ARMA model is obtained by applying the eigenvalue method discussed in section 3.5, while, the order of the AR model is obtained by applying the AIC method in which the optimal order is selected according to the following criteria

$$AIC(p) = N \ln \sigma^2_x + 2p$$

where N is the data length and σ^2_x is the estimate of the white noise variance for the p^{th} order AR model. The appropriate order of the model is the value of p which gives the minimum value of AIC.

As shown in Table 4.1, the optimal order of the ARMA model has an average of 4.81 for the AR part and 2.58 for the MA part. Also, it can be seen that all classes of the EEG signals modeled by the ARMA model have nearly the same average number of parameters. Whereas, Table 4.2 shows that , the optimal order of the AR has an average of 8.89. Besides, the average order of some classes as alpha class is higher than those of the other classes.

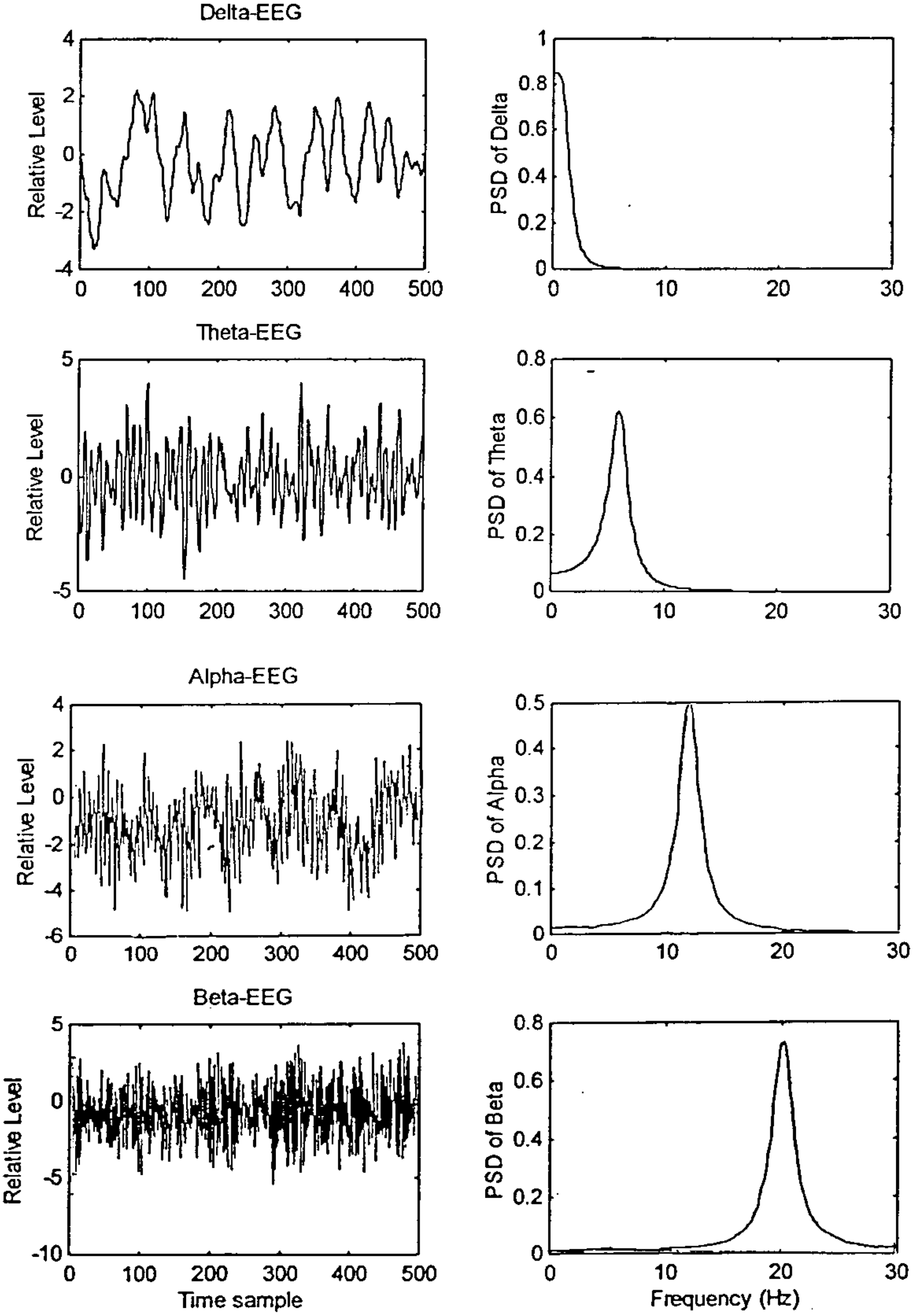


Fig. 4.2 Different classes of EEG signals with their corresponding PSD

Table 4.1 Optimal order of ARMA model using the Eigenvalue method.

Class of EEG	Order of ARMA model		
	Mean *(p,q)*	Maximum *(p,q)*	Minimum *(p,q)*
Delta	(5.03, 2.32)	(9, 4)	(4, 2)
Theta	(4.71, 3.23)	(8, 6)	(3, 1)
Alpha	(5.17, 2.51)	(9, 5)	(5, 2)
Beta	(4.33, 2.26)	(8,4)	(4, 2)
Total	(4.81, 2.58)		

Table 4.2 Optimal order of AR model using AIC method.

Class of EEG	Order of AR model		
	Mean	Maximum	Minimum
Delta	7.58	13	2
Theta	8.68	13	2
Alpha	10.95	16	2
Beta	8.34	15	3
Total	8.89		

Moreover, it can be seen that the average orders of the AR model is always higher than that of the ARMA model for each class. Thus, we can say that in general, the AR model needs larger order than the ARMA model to represent different EEG classes.

b) Residual tests:

To evaluate the efficiency of both the ARMA and AR models in representing the EEG signals, they are tested through the residual tests mentioned in section 3.7. The test of the lack of correlation (test 1) and the test of the lack of the sine terms (test 2) are performed. The results are shown in Tables 4.3 and 4.4. For illustration, tests 1 and 2 are shown in Figs. 4.3 and 4.4, respectively, for one record of alpha class.

As shown from Tables 4.3 and 4.4, the two tests are satisfactory for most the records of every EEG class. For the ARMA model, Table 4.3 shows that about 85% of the EEG classes passed test 1, 89.17% passed test 2 and 8.33% passed both the tests. Thus, we can say that, about 8.33% of the different classes of the EEG signals can be represented by the ARMA model.

For the AR model, it can be seen from Table 4.4 that, the percentage of the EEG signals that passed test 1 is 85.84%, and that passed test 2 is 88.34%, while about 82.5% passed both the tests. Therefore, about 82.5% of the EEG signals can be represented by the AR model.

From these results, we can say that, both the models are efficient in representing the EEG signals, keeping in mind that the ARMA model needs, in general, less parameters than the AR model to achieve this purpose.

Table 4.3 Percentage of EEG classes that can be modeled using ARMA model.

Class of EEG	Test 1 Accepted (%)	Test 2 Accepted (%)	Test 1 and Test 2 Accepted (%)
Delta	86.67	93.33	86.67
Theta	83.33	86.67	80.80
Alpha	86.67	90.0	83.33
Beta	83.33	90.0	83.33
Mean	85.0	89.17	83.33

Table 4.4 Percentage of EEG classes that can be modeled using AR model.

Class of EEG	Test 1 Accepted (%)	Test 2 Accepted (%)	Test 1 and Test 2 Accepted (%)
Delta	86.67	90.0	90.0
Theta	90.0	90.0	83.33
Alpha	80.0	86.67	76.67
Beta	86.67	86.67	80.0
Mean	85.84	88.34	82.50

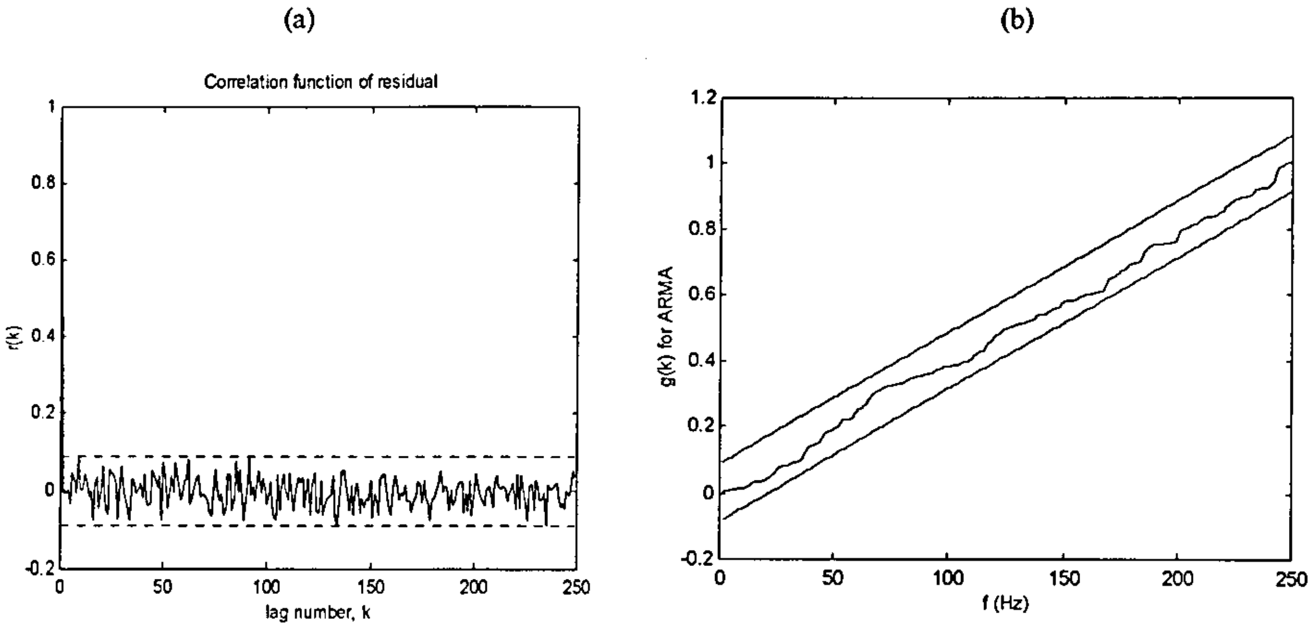


Fig 4.3 An illustration of test 1 and test 2 applied to the residual sequence obtained by the ARMA modeling of the alpha class. (a) test 1 for the lack of correlation and (b) test 2 for the lack of sine terms, where *g(k)* is the normalized cummulative periodogram.

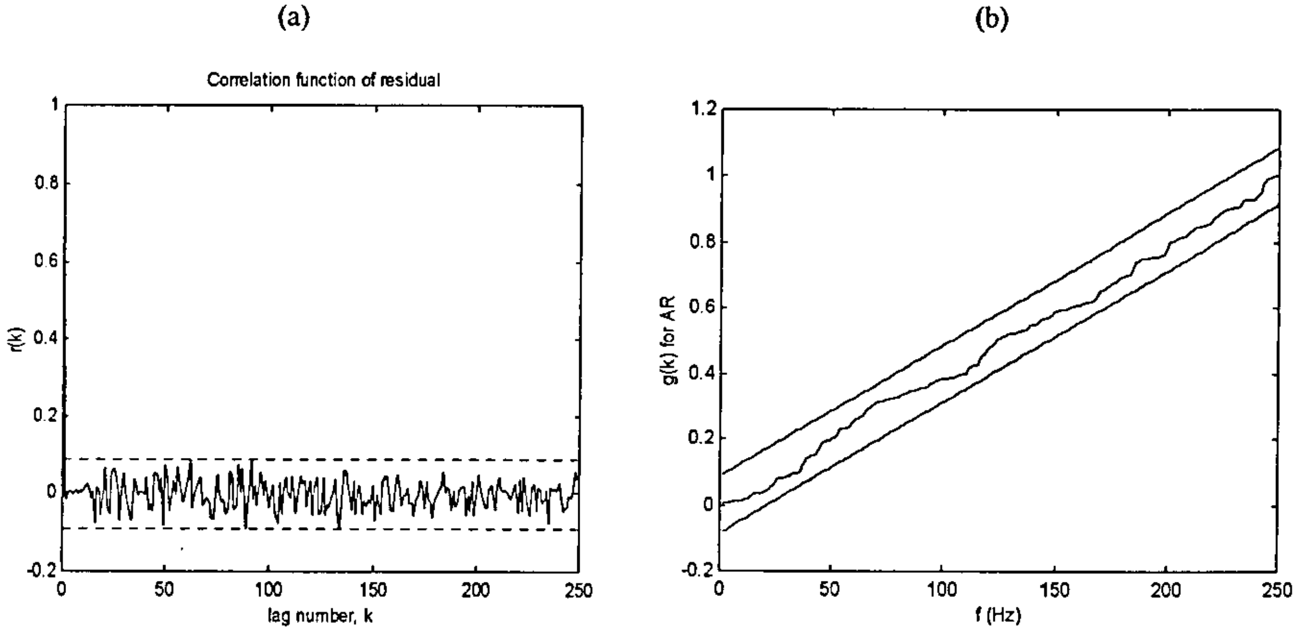


Fig 4.4 An illustration of test 1 and test 2 applied to the residual sequence obtained by the AR modeling of the alpha class. (a) test 1 for the lack of correlation and (b) test 2 for the lack of sine terms, where *g(k)* is the normalized cummulative periodogram.

Exmple #2:

The ARMA algorithm for SPA is applied on a simulated EEG signal designated as EEG1 shown in Fig. 4.5 with its corresponding power spectral density. EEG1 has a dominant alpha spectral component at 9.4 Hz and a weaker delta component at zero Hz and a beta component at 20.1 Hz. The true values of the spectral components of the EEG signal are shown in Table 4.5.

Different 10 records of the simulated EEG are used. These records are fitted to an ARMA model with order (6,4). The spectral components for each record are calculated using the SPA technique and a statistical analysis is performed, where the bias (average deviation of the parameters from their true value) is obtained as shown in Table 4.5. The estimated PSD of this signal using the ARMA model is shown in Fig. 4.6 for one record.

As shown in Fig. 4.6, the ARMA model is efficiently estimated the PSD of the given simulated EEG signal. It is clearly resolved the three spectral components of the signal. Furthermore, it can be seen from Table 4.5 that the ARMA model estimated the given spectral components with slight bias.

Thus, these results indicate that the ARMA model can be used as an efficient model to derive the PSD and extract the important features which describe the EEG signals.

Table 4.5 Statistics for SPA of the simulated EEG1.

Spectral component	True value	Bias
f_δ	0	0.15
σ_δ	5.3	-1.32
G_δ	0.13	0.062
f_α	9.4	0.035
σ_α	3.33	-0.34
G_α	0.63	0.04
f_β	20.1	-0.27
σ_β	6.1	-1.86
G_β	0.24	0.11

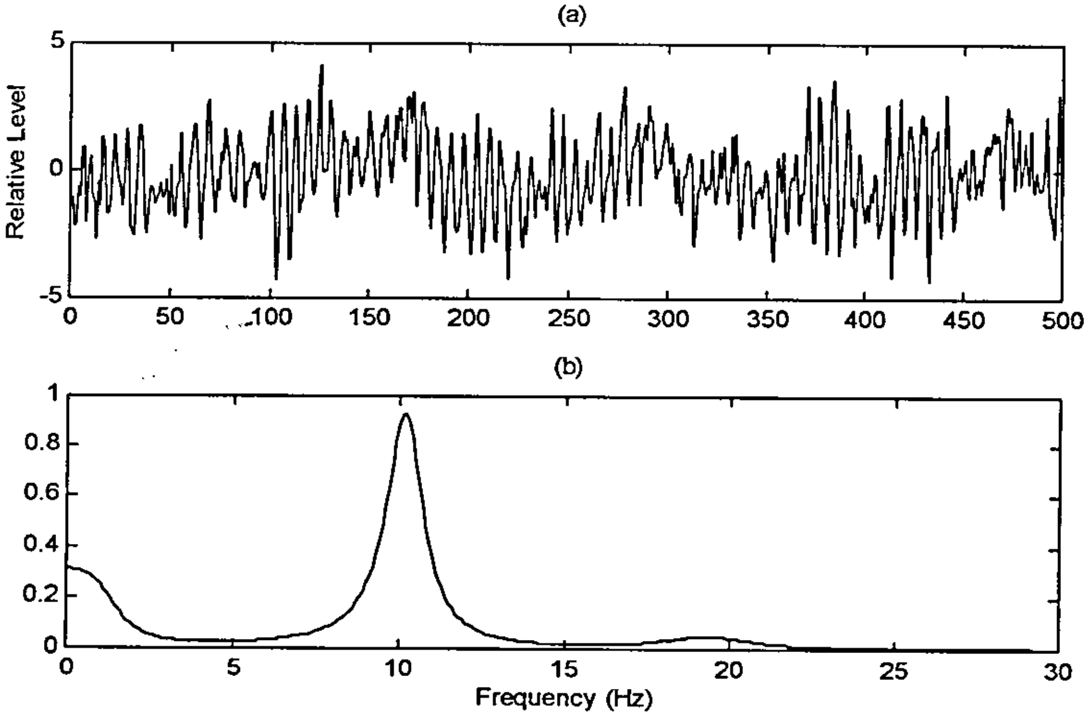


Fig 4.5 (a) Simulated EEG1 signal. (b) Its corresponding PSD.

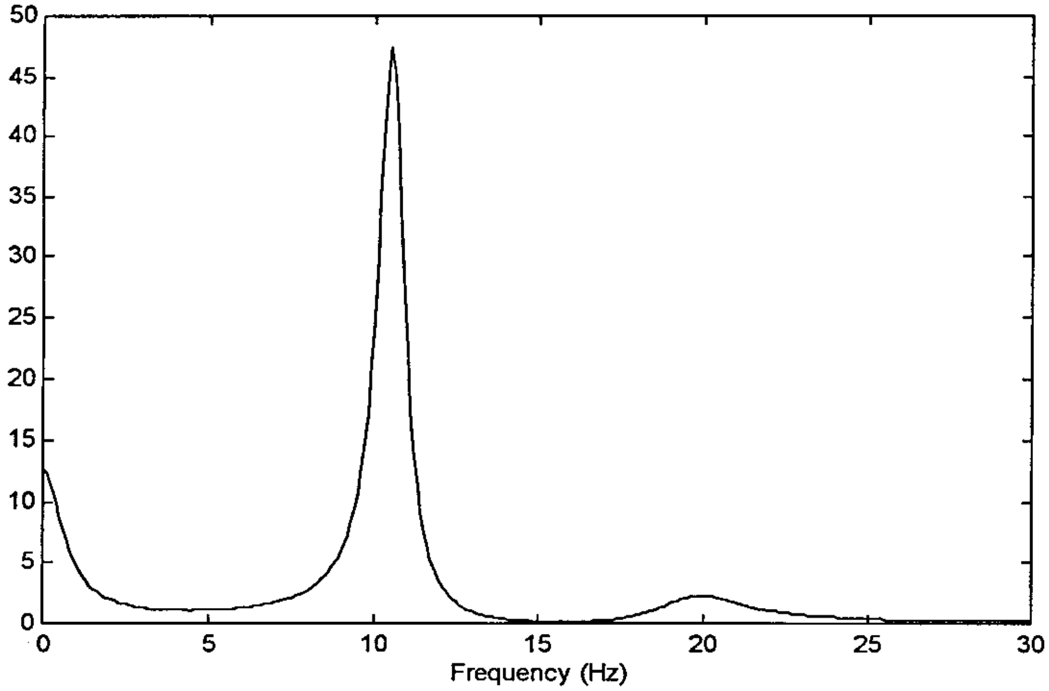


Fig. 4.6 Estimated PSD of the EEG1 signal in Fig. 4.5 (a) using ARMA model.

4.2 ARMA MODELS WITH HOMOMORPHIC FILTERING

The spectral information in the EEG registration plays a crucial role in the assessment, and hence, spectral estimation is very important in EEG processing. Among the many spectral estimation approaches, the parametric approach is found to be an efficient way in this aspect.

The popularity of ARMA spectral modeling based on the linear prediction (LP) technique is due to the efficiency of LP in extracting the structure of the signal. However, LP-based methods require that the signal to be either minimum or maximum phase (Makhoul, 1975). The real life signals are neither minimum or maximum phase but are of mixed phase nature. The spectral zeros which lie outside the unit circle in the z-plane, contribute to the excess phase and make the signal a mixed phase one. This limits the performance of the ARMA modeling methods based on LP. Fortunately, the mixed phase signals can be converted to an LP compatible form by homomorphic filtering (HF) (proakis *et al.*, 1992).

Thus, due to the mixed phase nature of the modeled signal in some cases, applying ARMA method directly resulted in inaccurate spectral estimate irrespective of the absence of the zeros outside the unit circle. This indicates that in order to get accurate spectral estimate of the modeled signal, the LP based on ARMA spectral modeling must be applied to the minimum phase equivalent of the signal rather than to the signal directly. This avoids the problems due to zeros outside the unit circle. To illustrate this idea, the following procedure is performed in which the mixed phase signal is converted to LP compatible form by homomorphic filtering then pole-zero estimation is made by LP. Let this method is designated as homomorphic prediction (HP) method.

4.2.1 HOMOMORPHIC DECONVOLUTION

Homomorphic deconvolution is a general method for separating two convolved signals by making use of the complex cepstrum to perform this separation. The complex cepstrum of the sequence $y(n)$ is defined as the sequence $c_y(n)$ which is the inverse z-transform of $C_y(z)$, where

$$C_y(z) = ln\, Y(z) \qquad (4.12)$$

In the frequency domain, if we express $Y(\omega)$ in terms of its magnitude and phase, say

$$Y(\omega) = |\, Y(\omega)|\, e^{j\theta(\omega)} \qquad (4.13)$$

Then

$$C_y(\omega) = ln\, |Y(\omega)| + j\theta(\omega) \qquad (4.14)$$

The complex cepstrum, as defined above, is a useful tool for performing deconvolution in some applications. To describe this, let us suppose that the signal $y(n)$ is obtained by convolving the two signals $x_1(n)$ and $x_2(n)$, that is

$$y(n) = x_1(n) * x_2(n) \qquad (4.15)$$

then

$$Y(z) = X_1(z)\, X_2(z) \qquad (4.16)$$

The logarithm of $Y(z)$ is

$$\begin{aligned} C_y(z) &= lnY(z) = ln\, X_1(z) + ln\, X_2(z) \\ &= C_{x1}\,(z) + C_{x2}\,(z) \qquad (4.17) \end{aligned}$$

Consequently, the complex cepstrum of the sequence $y(n)$ is expressed as the sum of the cepstrum of $x_1(n)$ and $x_2(n)$. That is,

$$c_y(n) = c_{x1}(n) + c_{x2}(n) \qquad (4.18)$$

Thus, we observed that the convolution of the two sequences in the time domain correspond to summation of the cepstrum sequences in the cepstral domain. The system for performing these transformations is called a homomorphic system and is illustrated in Fig. 4.7 (a).

In some applications, the characteristics of the cepstral sequences $c_{x1}(n)$ and $c_{x2}(n)$ are sufficiently different so that they can be separated in the cepstral domain. That is, if the signals $c_{x1}(n)$ and $c_{x2}(n)$ occupy disjoint time intervals then $x_1(n)$ and $x_2(n)$ can be recovered using an appropriate time domain window as shown in Fig. 4.7 (b). Once we have the cepstrum sequences $c_{x1}(n)$ and $c_{x2}(n)$ by windowing, the sequence $x_1(n)$ and $x_2(n)$ are obtained by passing $c_{x1}(n)$ and $c_{x2}(n)$ through the inverse homomorphic system, shown in Fig. 4.7 (c).

Thus, if $x(n)$ is the convolution of minimum phase component $x_{min}(n)$ and maximum component $x_{max}(n)$, then

$$c_x(n) = c_{min}(n) + c_{max}(n) \tag{4.19}$$

Since $c_{min}(n) = 0$ for $n<0$ and $c_{max}(n) = 0$ for $n>0$, the complex cepstrum provides a means of factorizing the signal $x(n)$ into its minimum and maximum phase component. Also, a mixed phase signal $x(n)$ is converted to a minimum phase signal $x_{mp}(n)$, called the minimum phase equivalent having the same spectral magnitude. This implies

$$|X_{mp}(e^{j\omega})| = |X(e^{j\omega})| \tag{4.20}$$

Since the minimum phase signal $x_{mp}(n)$ is a causal sequence then it can be reconstructed from its even part, this is due to the fact that the complex cepstrum of $x_{mp}(n)$ is also minimum phase one. Thus,

$$C_{mp}(n) = U(n)EV[c_{min}(n)] \tag{4.21}$$

where $U(n) = 0,1 , 2$ for $n<0$, $n = 0$, $n>0$, respectively.

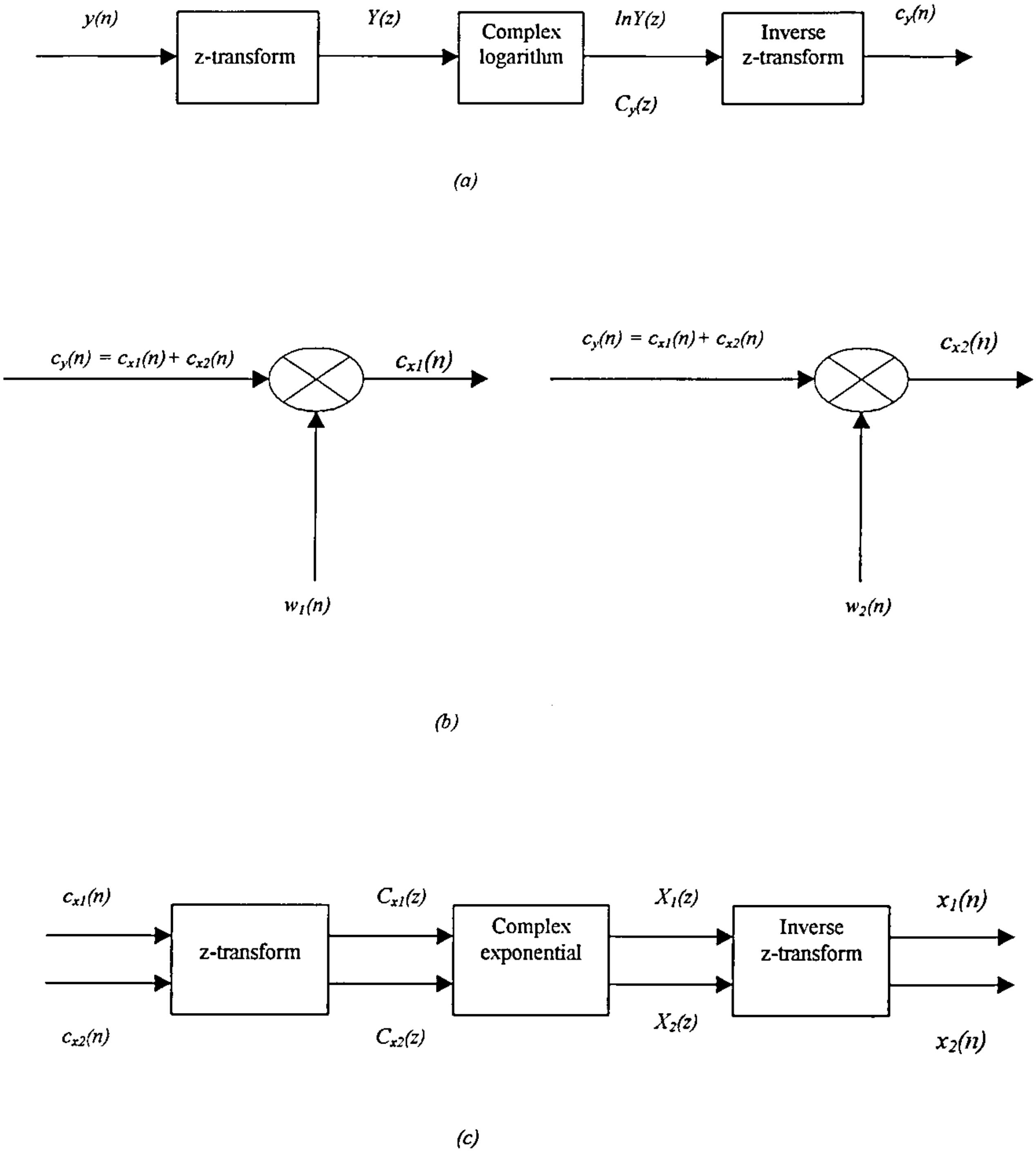


Fig. 4.7 (a) Homomorphic system for obtaining cepstrum $c_y(n)$ of the signal $y(n)$.
(b) Separating the two cepstral components by an appropriate windows $w_1(n)$ and $w_2(n)$.
(c) Inverse homomorphic system for recovering the signal $x_1(n)$ and $x_2(n)$.

Since the real part of $C_{mp}(z)$ is the z-transform of the even part of $c_{mp}(n)$, $EV[c_{min}(n)]$ is obtained from $log[|X(e^{j\omega})|]$. As a result, the minimum phase equivalent $x_{mp}(n)$ can be obtained from the inverse system as shown above. Thus, the method considered here involves the estimation of ARMA model of the minimum phase equivalent of the signal.

4.2.2 SIMULATION RESULTS

The performance of the HP method for simulated EEG signals is studied in the following example.

A simulated EEG signal with two spectral components at 10 Hz and 19.8 Hz is fitted directly (without homomorphic filtering) to an ARMA model with order *(6,4)*. The estimated PSD using the ARMA model is calculated. Besides, the locations of the poles and zeros of this model is found as shown in Fig. 4.8. Let us designate this procedure of fitting the EEG signal directly to the ARMA model as 'direct ARMA' method.

For the HP method, the simulated EEG is first fitted to a homomorphic filter to produce the minimum phase equivalent component of the EEG signal then, the resulted signal is fitted to an ARMA model of order *(6,4)*. Furthermore, the PSD and the pole-zero locations of the model representing the minimum phase signal are found as shown in Fig. 4.9.

As shown in Fig. 4.8 (b), some of the model zeros estimated by direct method are outside the unit circle. As a result, an inaccurate spectral estimation is obtained as shown in Fig 4.8 (d) from which we can see that the spectral component of the EEG signal at 19.8 Hz is not resolved by the direct ARMA method. The evidence of these zeros (which lie outside the unit circle), at least in this example, indicates that in general the EEG signals is not a minimum phase signal.

On the other hand, this is not the case when applying the HP method as shown in Figs. 4.9 (b) and (d), from which we can see that the two spectral components of the simulated EEG signal are resolved clearly.

Fig. 4.10 shows the PSD estimated by both methods. The difference in these estimates can be seen more clearly when observing Fig. 4.11 which shows the pole spectrum and the zero spectrum for both methods. As shown in Fig. 4.11 (a), the pole spectrum for both methods are almost the same, while the zero spectrum differs significantly. It is evident that only the zero spectral estimate is affected by direct estimation and not the pole spectrum. This is obvious since for a stable signal all the poles lie within the unit circle and hence the poles satisfy the assumption of LP technique. On other hand, for zeros no such restriction exists and can lie anywhere in the z-plane. However, the zeros which lie out side the unit circle will not satisfy the assumption of the LP and this leads to an inaccurate spectral estimates.

Thus, by applying the ARMA model to the minimum phase equivalent component of the EEG signal rather than to the original EEG signal, an accurate spectral estimation is achieved.

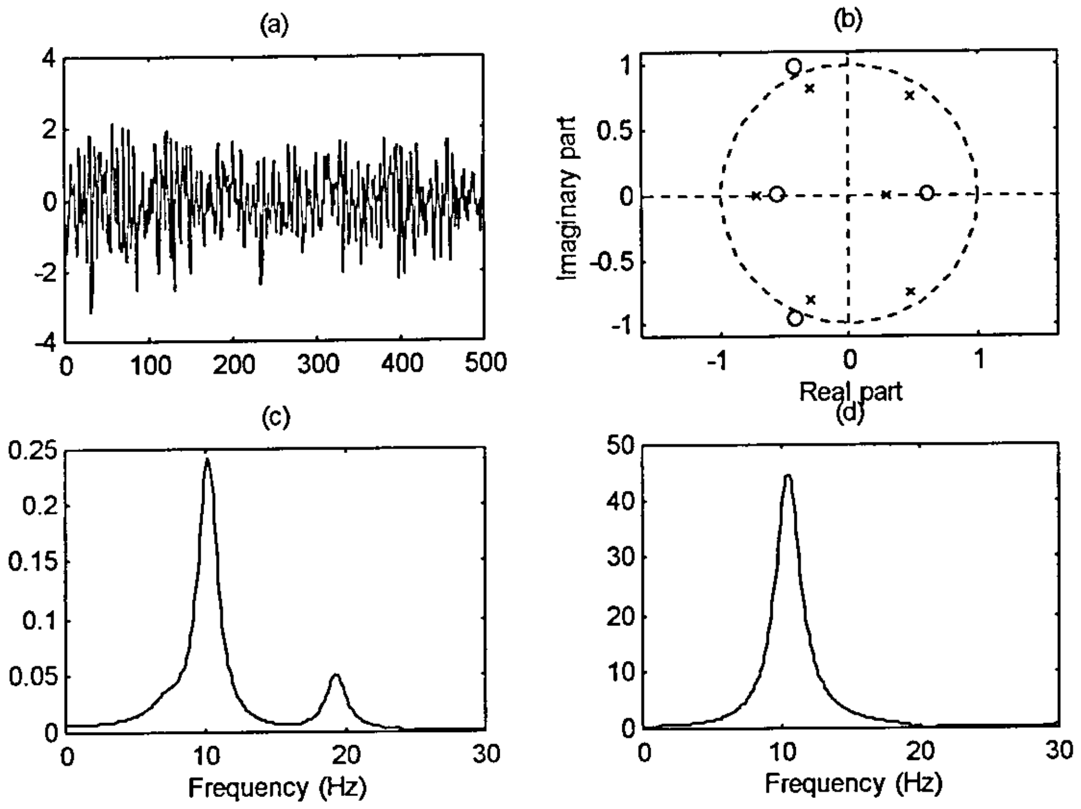


Fig.4.8 (a) Original EEG signal. (b) Pole-zero locations of the direct ARMA model.
(c) Original PSD of EEG. (d) Estimated PSD using the direct ARMA.

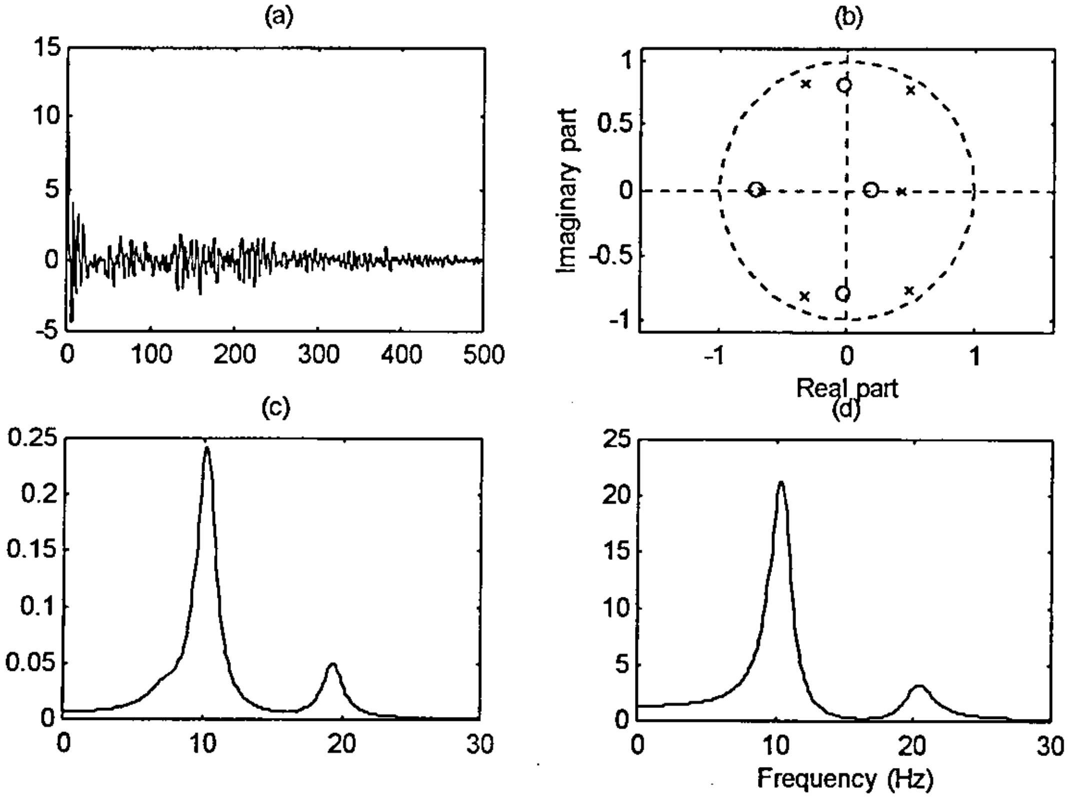


Fig.4.9 (a) Minimum phase equivalent component of the original EEG.
(b) Pole-zero locations of the HP model.
(c) Original PSD of EEG.
(d) Estimated PSD using the HP.

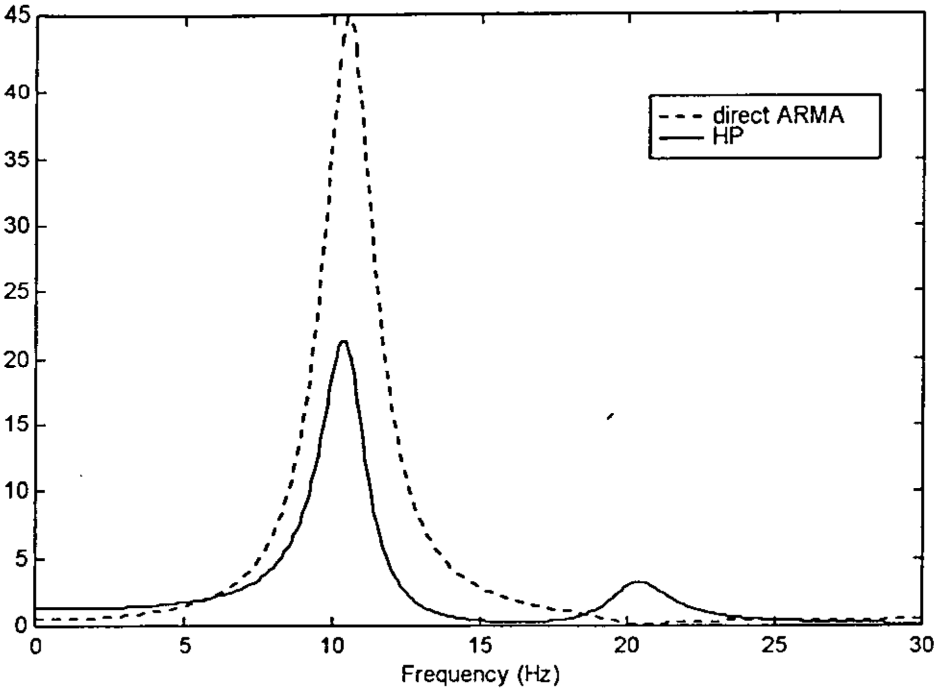


Fig. 4.10 Estimated PSD for direct ARMA and HP methods.

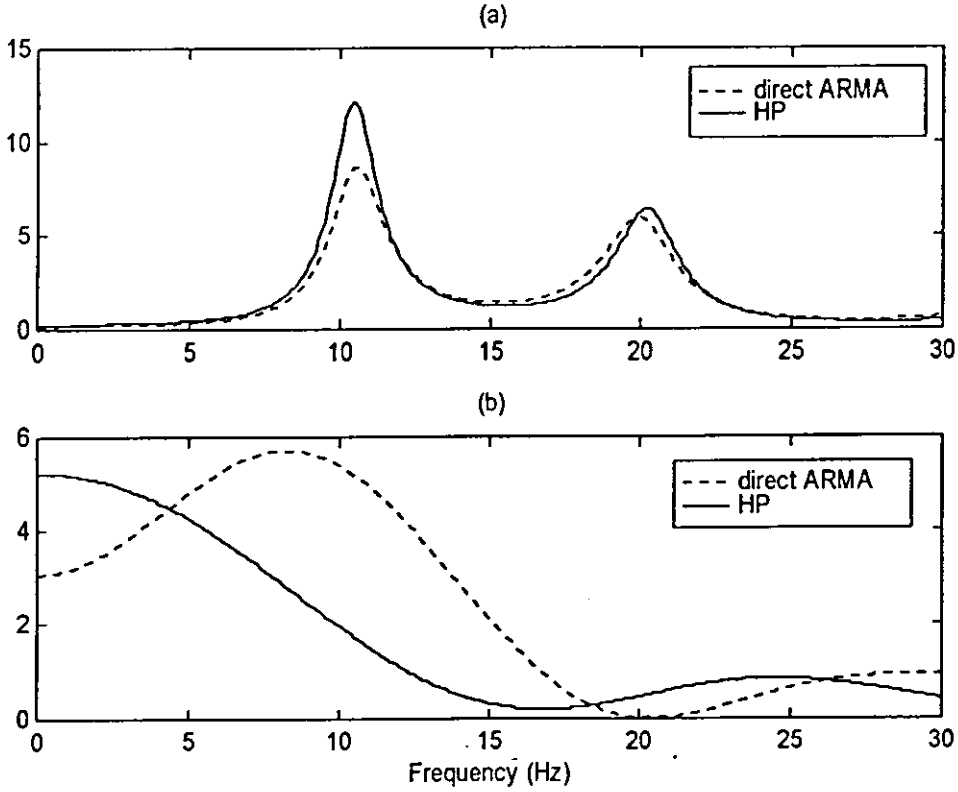


Fig 4.11 (a) Pole spectrum for both methods. (b) Zero spectrum for both methods.

4.3 SPIKE DETECTION

In the analysis of biomedical signals, spikes are important for diagnosis. Sometimes impulses of low occurrence rate are superimposed on predictable process, as spikes in EEG due to epilepsy. Therefore, the EEG signal is considered to be composed of background (stationary) waves and paroxysmal (nonstationary) waves. Spikes can be regarded as nonstationary waves, because they occur paroxysmally. In such cases, nonstationary components are symptomatic and are of more interest than the stationary counterpart.

In signal processing techniques, the word "spike" means 1) localized high frequency and 2) increase in instantaneous energy. The quantitative description of the amplitude and spectrum of spikes vary from signal to signal, subject to subject, and it even varies from time to time for the same subject. This is precisely why the detection of spikes becomes difficult. As the spike base width increases, energy is concentrated more in low-frequency band where the energy of the background signal is also located and the detection becomes more difficult in the frequency domain. Hence, the difficulty of spike detection increases with the increase in spike width.

Computerized EEG analysis for detecting the nonstationary waves, especially spike waves of epileptic EEG, has been actively studied. Any algorithm for spike detection can be divided into two stages; the first stage pre-emphasizes the spike and the second stage determines the spike positions using some thresholding technique. Various types of algorithms have been tried for spike detection (Fender *et al.*, 1986; Qian *et al.*, 1988; Ray, 1994).

In analyzing the EEG effectively, the separation of nonstationary waves from stationary waves is desirable. Such separation is useful not only for the detection of the nonstationarities but also for the separate analysis of the two components. For example, one may wish to analyze the stationary waves in the frequency domain while the nonstationary waves are analyzed in the time domain. In the next section, the nonstationary component is separated from the stationary one using a prediction error filter. It may be separated as an error signal which is corrupted with carried-over error.

4.3.1 SPIKE DETECTION USING PREDECTION ERROR FILTER

As defined earlier, the prediction and the prediction error sequences for an ARMA process are given by equations (3.2) and (3.6) respectively. In the z-domain, the error filter *E(z)* is defined as

$$E(z) = \frac{1 + \sum_{k=1}^{p} a_k z^{-k}}{\sum_{k=0}^{q} b_k z^{-k}} \qquad (4.22)$$

Since EEG can be regarded as a summation of stationary background waves and nonstationary paroxysmal waves, the input signal, *y(n),* is a summation of two types of signals: the stationary signal, *s(n)*, which can be represented by the ARMA model, and the nonstationary signal, *g(n)*, composed of random impulsive waves of low occurrence rate. That is,

$$y(n) = s(n) + g(n) \qquad (4.23)$$

The impulses are also represented by the triangular waves of linearly rising and falling edges and of width nT ($n = 2m$, $m = 1, 2, 3, \ldots.$ and T is the sampling interval) having the peaks at $nT/2$.

Since the nonstationary part consists of random impulsive waves of low occurrence rate, it may be separated as an error signal which is corrupted with carried-over error. The latter is removed using signal-inversion technique.

Therefore, the nonstationary components appear as an error signal, $e(n)$, which is indeed the prediction error sequence. The carried-over error is defined as:

If a spike exists at n and the front end of the prediction sequence just touches $(n-1)$, the error which is almost equal to the magnitude of the spike, is detected at n because the prediction coefficients can not predict sudden rise or fall in the magnitude of $y(n)$. The error signal is defined here as carried-over error, till the spike remains within the prediction sequence. This may be cancelled partially by thresholding and almost completely by the signal inversion.

The nonstationary and stationary components of the input signal $y(n)$ are separated as follows:

Representing the error filter $E(z)$ as in (4.22), the output of the filter will be $Y(z)E(z)$. Since $y(n)$ is the linear summation of a predictable signal, $s(n)$, and a train of random pulses, $g(n)$, the output of the error filter will be $[S(z)E(z)+G(z)E(z)]$. The first part will have very small magnitude (since $s(n)$ is predictable). The second part corresponds to the output of the error filter when the input is *g(n)* and since the latter is not predictable, the entire *g(n)* along with the carried-over errors will appear as output.

Random pulses of low occurrence rate are linearly summed with a slowly varying background signal as shown in Fig. 4.12 (a). The error signal, *e(n),* is shown in Fig. 4.12 (b). The carried-over errors after each spike are seen to be overshoots and undershoots followed by gradual decay. These spikes can not be fully recovered simply by thresholding because the height of some of low-magnitude spikes may be less than the magnitude of the overshoot/undershoot. A simple way to separate the spike is to invert the error signal. This is shown in Fig. 4.12 (c) and is designated as $e_i(n)$. It may be shown that $e_i(n)$ is also an error signal of the input $y(n)$ with a reversed sign.

Assume that a threshold (shown as a dotted line in Fig. 4.12 (b), is gradually lowered till it touches the highest spike at *n*. With a prior information of the width of the spikes the corrected error, $e_c(n)$, signal (which represents the separated nonstationary signal containing the spikes) may be written as the summation of *e(n)* and $e_i(n)$ for both sides of the spike, i.e., if the width of the spike is $2mT$, then:

$$e_c(n+m+j) = e(n+m+j) + e_i(n+m+j) \tag{4.24}$$

and

$$e_c(n\text{-}m\text{-}j) = e(n\text{-}m\text{-}j) + e_i(n\text{-}m\text{-}j) \tag{4.25}$$

for the right-hand side and left-hand side of the spike, respectively, where $j =$ 0, 1, 2, ..., *r*, and *r* depends on the density of the spikes.

The spike itself is recovered as

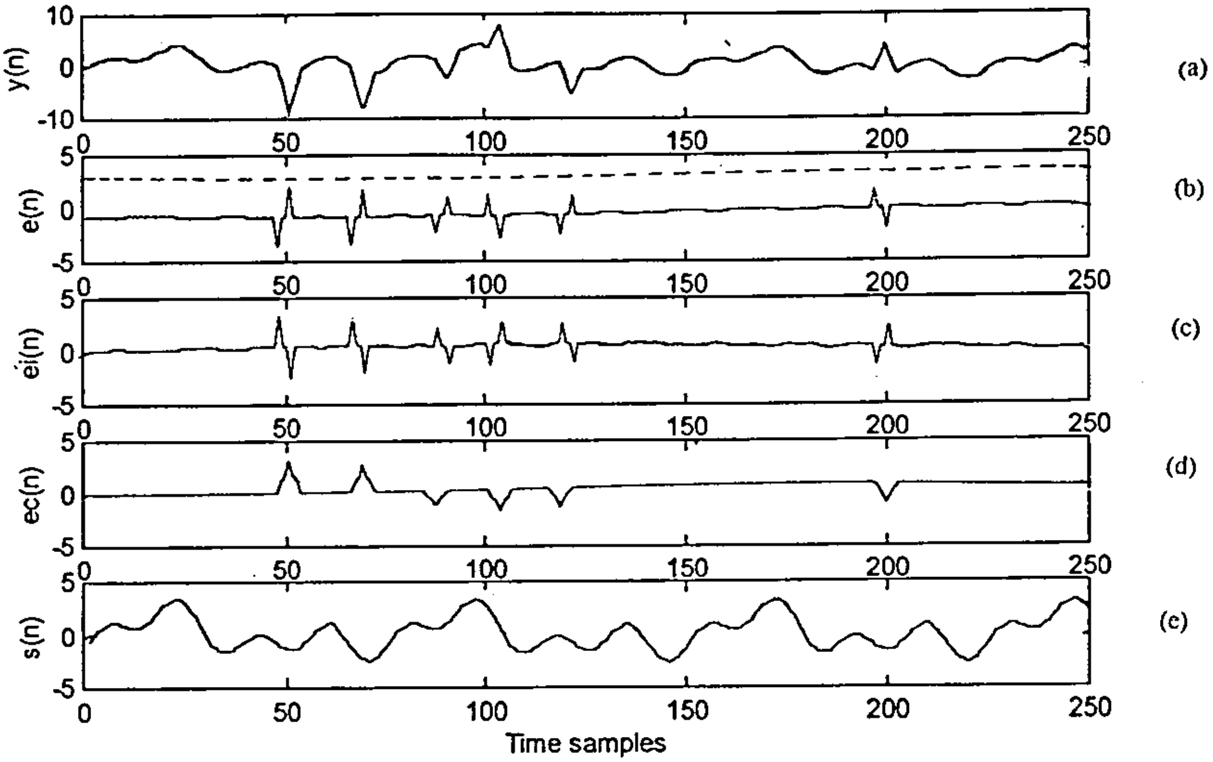


Fig. 4.12 Separation of nonstationary and stationary parts using linear prediction error filter of an ARMA model. (a) y(n): signal with spikes, (b) e(n): prediction error sequence, (c) $e_i(n)$: inverted prediction error sequence, (d) $e_c(n)$: nonstationary part, and (e) separated stationary part.

$$e_c(n+s) = V_r / (1+|s|) \tag{4.26}$$

where, V_r is the height of the threshold, and

$$s = -(m-1), \ldots, 0, \ldots, (m-1) \tag{4.27}$$

The threshold line is then taken to the negative side and the same operation is repeated. With this type of thresholding, the spikes of higher magnitudes are detected first and the largest overshoots/undershoots are killed. The spikes of low magnitudes are then gradually detected by lowering the threshold. If the rest of the baseline is zero padded, the nonstationary and stationary parts are almost completely separated as shown in Figs. 4.12 (d) and (e), respectively.

For any detection algorithm, there is a need to evaluate its performance objectively. The following performance indeces are considered:

1) False-Negative Ratio: The ratio of number of the missed spikes to the actual number of spikes is the false-negative (FN) ratio and is defined by

$$FN = \frac{Number\ of\ spikes\ missed}{Actual\ number\ of\ spikes}$$

2) False-Positive Ratio: The ratio of the number of false-detected spikes to the actual number of spikes is the false-positive (FP) ratio and is defined by

$$FP = \frac{Number\ of\ false\ spikes\ \mathrm{det}ected}{Actual\ number\ of\ spikes}$$

3) Bias: The bias of spike detection in terms of the number of sampling points is defined by

$$Bias = \frac{1}{M}\sum_{i=1}^{M}\left|n_i - \hat{n}_i\right|$$

where n_i and $\hat{n}_i$ are the actual and estimated locations, respectively, of the i^{th} spike, and M is the number of correctly detected spikes. The location of the maxima of the spike is taken as the spike position.

4.3.2 SIMULATION RESULTS

To evaluate the performance of the above method, the following simulation is performed.

The slowly varying background signal $s(n)$ shown in Fig. 4.13 (a) is chosen as

$$s(n) = sin(\omega n) - sin(2\omega n+\theta) + sin(4\omega n)$$

where ω =2π/75, θ =π/2. The spike train $g(n)$ is

$$g(n) = \sum_{i=1}^{r} d(n-k_i)$$

where r is the number of spikes, $d(n-k_i)$ is the spike with base width $2m$ at k^{ith} instant, and it represented by

$$d(n-k_i) = A_i(n-k_i) + \sum_{j=1}^{m} A_i \frac{(m-j)}{m}[\delta(n-k_i+j)+\delta(n-k_i-j)]$$

where, $\delta(n)$ is the unit impulse function. The spike position, amplitude, and sign of the spike are generated by random number generator.

As an example, spike train with six spikes and base width $6T$ is shown in Fig. 4.13 (b). The signal with spikes, $y(n) = s(n) + g(n)$, is shown in Fig. 4.13 (c). To detect the spikes, an ARMA model with order $(6,4)$ is used, the resulting prediction error sequence is shown in Fig. 4.13 (d) and the detected spike sequence is shown in Fig. 4.13 (e).

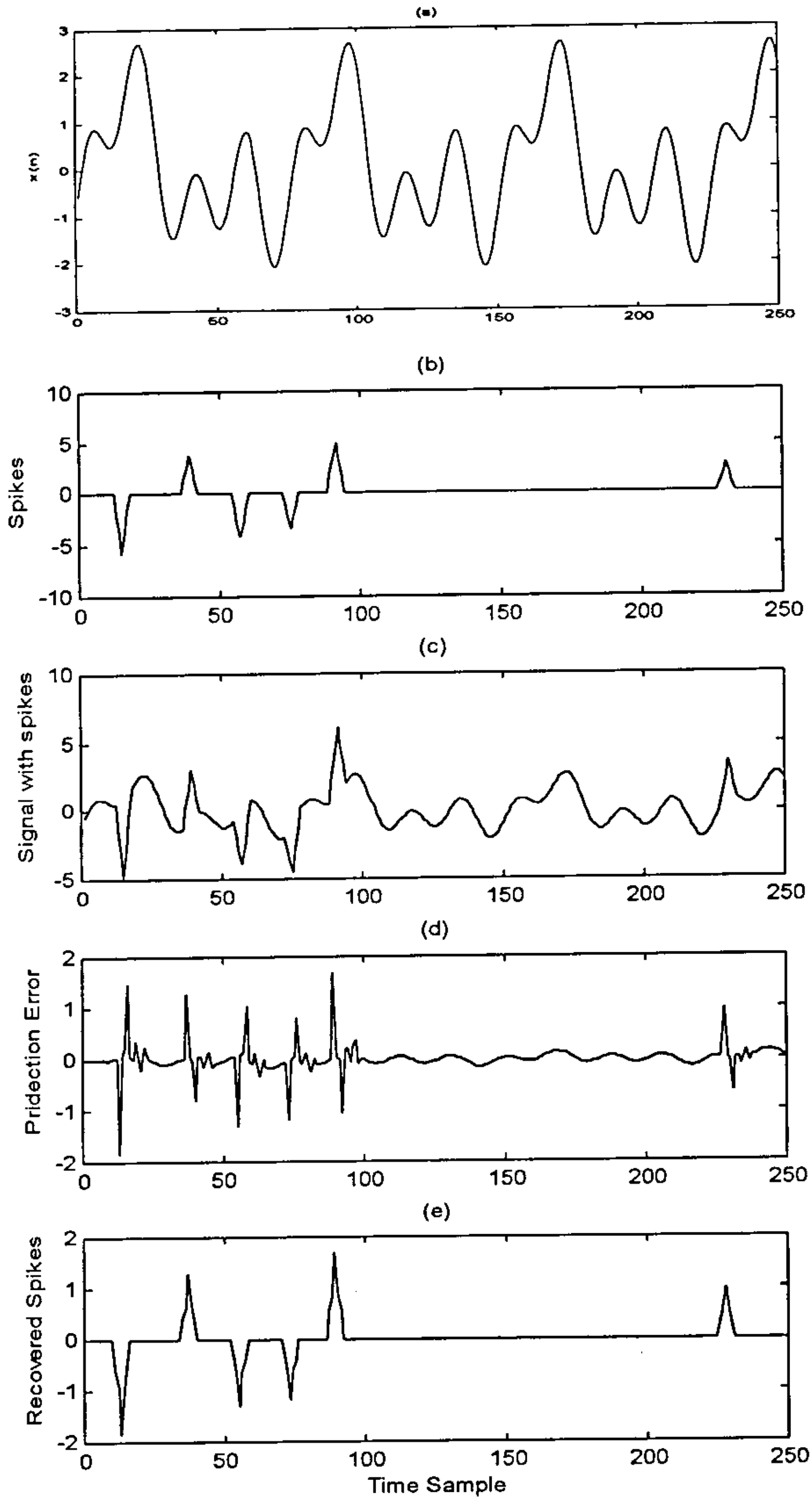


Fig. 4.13 (a) The slowly varying signal, (b) Spike train, (c) signal with spike, (d) prediction error sequence of the ARMA model, and (e) the detected spikes.

a) Effect of additive noise:

To demonstrate the performance of this method when the spikes are corrupted with noise, an additive white Gaussian noise *v(n)* with zero mean and variance σ_v^2 is added to the signal *y(n)*. The varianceσ_v^2 is adjusted to give the required SNR.

Table 4.6 shows the performance indeces of this method with different cases of additive noise (no-noise, and noise at 5 dB and 20 dB) in such a way that the method is run 20 times for each case.

b) Effect of the spike width:

To demonstrate how the spike width affect the performance of the this method, the spike width is varied from $2T$ to $12T$. A 25 trials are performed and the average value of these indeces is calculated. The results are shown in Fig. 4.14.

Based on the performance indeces, it can be shown from Table 4.6 that, the spike detection technique became robust when the signal corrupted by noise. At low SNR, the FP ratio is high which indicates that the number of the false spikes detected is increased. While, on the other hand, the FN ratio has a small value. Thus, a linear combination of both these ratios is used to evaluate the performance of this method, since either the FP or the FN ratio can be improved by adjusting the threshold level but at the cost of the other.

Furthermore, it is seen from Fig. 4.14 that the performance indeces increase with the increase of spike width, which indicate that, the difficulty of the spike detection increases. Since in this case, as the spike width increases, the energy is concentrated in the low frequency bands where the energy of the stationary background signal is located.

Table 4.6 The performance indeces of the spike detection method for the simulated signal for both the noiseless case and the noisy case at (5 dB and 20 dB).

Performance indeces	No-noise	SNR = 20 dB	SNR = 5 dB
	mean ± sd	mean ± sd	mean ± sd
FP	0.32 ± 0.17	0.36 ± 0.16	4.24 ± 0.31
FN	0.25 ± 0.21	0.26 ± 0.17	0.12 ± 0.11
Bias	2.61 ± 0.63	2.58 ± 0.55	2.33 ± 0.37

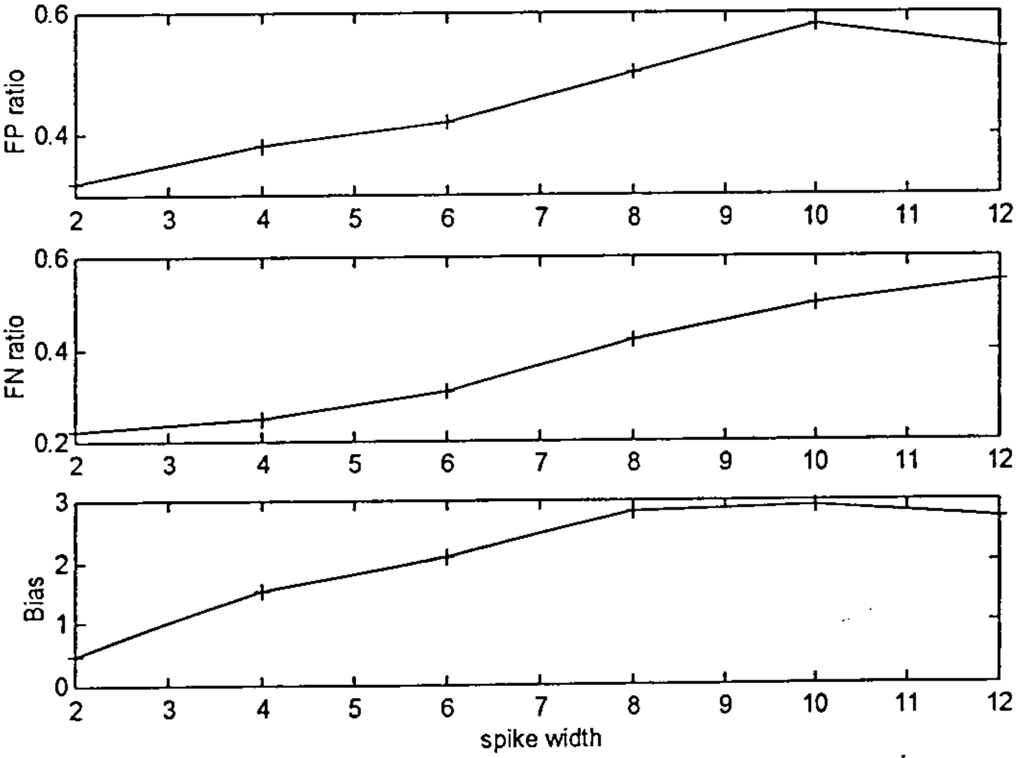


Fig. 4.14 Effect of the spike width on the performance indeces of the spike detection method for the noiseless case. These results are mean performances of 25 trials.

CHAPTER 5

DISCUSSION AND CONCLUSION

In this thesis we are concerned in studying the various concepts and methods being used for EEG signal analysis. In this respect, we are interested in how to perform data reduction on an EEG signal while preserving information over the relevant EEG patterns, this involved the problem of finding the most appropriate method for feature extraction. For this purpose, it was found that the ARMA model is considered to be a suitable mathematical method for describing the EEG signals. Different algorithms dealing with ARMA modeling were presented in this thesis. For the purpose of studying the performance of these algorithms, they have been applied to selected examples of simulated systems. The results indicate that these algorithms are useful in various applications.

The use of the ARMA modeling in EEG signal analysis provides practically useful method to quantify EEG signals for several reasons:

1- The EEG signal can be realized in terms of small number of parameters. Utilizing the ARMA parameters, a considerable degree of data reduction is achieved. The computation problem is therefore to estimate the model parameters. An important element in this estimation is to define the minimum number of parameters to be computed, that is, the model order.

Selecting the model order is a critical first step toward the goal of modeling an ARMA process and estimating the model parameters. A new method for the determination of the model order was discussed in section 3.5.1 In this method, a relationship between the minimum eigenvalue of a covariance matrix and the MDL criterion is constructed. A table for this relation is established for the different values of the model order *(p,q)* for which a search for the location, where the minimum eigenvalue drops very quickly, is an indication for the best model order. The performance of this method has been tested and compared to previous methods, namely, AIC and MDL in section 3.5.2. The comparison was with respect to the effect of the data lengths of the modeled signal and the effect of the additive noise. The simulation results presented in section 3.5.2 indicate the superiority of the new ARMA model order technique over the previous methods. The main advantage of the new method over the previous ones is that the model order is determined without requiring prior estimation of the model parameters. Furthermore, it was found that the new method is better in the sense that it can provide more efficiently the correct model order estimate for the cases of different data lengths and different levels of noise.

2- It is possible to derive power spectra and extract the important features of the EEG signal from the ARMA parameters.

The ARMA modeling is a powerful method in spectral estimation problem. It is an efficient way in describing the spectrum of the EEG signal which contains both peaks and valleys. This is an advantage of the ARMA model over the AR model which is, in turn, more suitable for signals whose spectrum has only peaks. Thus, when an AR model is used to represent an EEG signal, large number of parameters (large model order) is required since many poles are required to approximate the effect of a zero which produces the valley in the EEG spectrum. Therefore, the ARMA model in general requires less parameters than what the AR model require to describe the EEG signal. For the purpose of comparison, different classes of simulated EEG signals, namely, delta, theta, alpha and beta, have been fitted to both the ARMA and AR models. It is found from the simulation results of example #1 presented in section 4.1.2 that both the models are suitable for describing the EEG signals since the residual tests, which are used to test the efficiency of the parametric methods, are accepted when applying both models. It is found that about 83.33% of these classes can be efficiently represented by the ARMA model, and about 82.5% of them can be efficiently represented by the AR model.

Moreover, it is found that, the ARMA model requires an average order of (4.81, 2.58) to represent these classes, while the AR model requires an average order of 8.89. This indicates that the AR model requires a higher average model order than that of the ARMA model when modeling these signals.

The performance of any EEG signal analysis method depends to a large extent on the extraction of relevant features characteristics of the signals. By means of ARMA modeling, it is possible to interpret the model parameters into terms that are closely related to the quantities used in the clinical neurophysiology. Thus, by fitting an ARMA model to the EEG signal and using the SPA technique, important features, namely: power, frequency and bandwidth, that characterize the EEG signal were extracted from the ARMA parameters. These features are of importance in order to characterize particular patho-physiological states. The simulation results of examples #2 in section 4.1.2 show how efficiently the ARMA model could derive the PSD of the EEG signals. Besides, it is found that the ARMA model could be expected to estimate the spectral component parameters of the EEG signals with a slight bias. This indicates that, the ARMA model is a useful tool in extracting the important features of the EEG signals

Another point of discussion is the validity of the ARMA in representing the EEG signal when it is considered as a mixed phase signal. It has been shown from the simulation results in section 4.2.2 that the spectral estimates obtained after filtering the EEG signal by a homomorphic filter is better than the direct estimations of the ARMA spectrum. The ARMA model has been fitted directly (without homomorphic filtering) to a simulated EEG signal with two spectral components at 10 Hz and 19.8 Hz. The direct ARMA modeling of the EEG signal has been found to affect the zero spectrum because some of the zeros of the model lie outside the unit circle, resulting in an inaccurate spectral estimate, that is, it failed to resolve the spectral component at 19.8 Hz. This problem has been solved by applying the ARMA model to the minimum phase equivalent component of the EEG signal obtained by filtering it by a homomorphic filter. This ensures that the zeros of the ARMA model lie inside the unit circle resulting in accurate zero spectral estimation, and therefore, an accurate pole-zero spectral estimation is achieved, that is, both the spectral components have been resolved clearly.

3- It is possible to detect EEG transient nonstationarities such as epileptiform spikes using the inverse ARMA filtering. The ARMA model can also be used in an inverted form, which leads to the so called inverse filtering operation. In section 4.3.1, a method for spike detection based on linear prediction error filter was presented. Spikes are of diagnostic importance in the biomedical signals. They are described as sharp waves or transients present in a slowly varying background signal. In the frequency domain, spikes can be described as high frequency components. As a result, high pass filtering techniques have been used for the separation of spikes. Because of the high frequency contents, the spikes appear in the error signal when the linear prediction filtering scheme is used.

In section 4.3.2, the performance of this method with respect to the effect of additive noise and spike width has been tested on a simulated signal. The simulation results presented in section 4.3.2 indicate that the presented spike detection technique is a useful one specially for the case of high SNR and when the spikes have short durations. It is found that, the difficulty of spike detection increases in the presence of large amount of noise and when the width of the spike becomes large, since in this case, the spikes with large width will have more energy at the low frequency bands where the energy of the background signal is also located.

For the purpose of introducing the various aspects of ARMA modeling, the ARMA model is used as an adaptive model to track the structural changes of the modeled signal. Both adaptive ARMA parameters and adaptive ARMA roots techniques are discussed in sections 3.4.1 and 3.4.3 respectively.

Furthermore, a new ARMA model dealing with systems with multiple inputs and delays is presented in section 3.6. This algorithm determines the best ARMA model to represent a system using its input and output data. It uses the parameter estimates of an overly specified model to construct a set of lower order models that are compared via a criterion that makes use of the residual error norms such as the MDL criteria. Thus, this algorithm automatically estimates the ARMA model order associated with systems that are allowed to have multiple inputs and delays. An over-parameterized (i.e., has higher order than necessary) model are used to infer the order of the best ARMA parameterization that describe the system.

It can be stated that using ARMA modeling a reasonable computerized EEG system can be achieved. It is worth to say that, the methods have been mentioned in this thesis as methods for EEG analysis are just a part of a wide research area which is still evolving. Although, these methods are believed to be good techniques in representing and describing the EEG signals, there might exist others which are better to some extent. However, ARMA modeling is not restricted to the processing of EEG signals alone, it can be used in a wide range of different related fields, e.g., speech analysis, radar, sonar, etc.

Finally, all the presented algorithms have been programmed in MATLAB and run on a PC. These algorithms were tested on simulated data. Nevertheless, acceptable results have been achieved which agree with the particular strategy used in this thesis. It is believed that valuable results will be achieved if these algorithms are tested on real data.

APPENDICES

Appendix-A: Computer programs for ARMA Parameter Estimation

```
function [A,B,SIG]=mywen(p,q,N,y)

%=====================================================================
%       Computer program for Modified Yule-Walker Equation (MYWE) method
%
%       This program implements the modified Yule-Walker equations to
%       estimate the AR parameters of an ARMA process. The MA parameters
%       and the white noise variance are estimated by filtering the data
%       by an estimate of A(z) and applying Durbin's method.
%
%       Input Parameters:
%       p               The AR order  of the ARMA model
%       q               The MA order  of the ARMA model
%       N               Number of data samples desired
%       y(i), i=1:N     The observed signal
%
%       Output Parameters:
%
%       A(i), i=1:p     The estimated AR parameters
%       B(i), i=1:q     The estimated MA parameters
%       SIG             Excitation white noise variance estimate
%=====================================================================


% To compute the autocorrelation estimates
x=y;
R=corrn(N,p+q+1,0,x,x);
% To solve the Yule-Walker equations
 [A,IFLAG]=solveywe(p,q,R);
% IFLAG is an indicator for the solution of the modified Yule-Walker equations
if (IFLAG==-1)
return
end
% Filter data by estimated A(z) to yield approximate MA time series
y2=czerfltn(p,N,A,x);
% Apply Durbin's method with large order AR model to estimate the MA
% parameters
L=round(N/5);
[B,SIG]=durbinn(q,N,L,y2);

function R=corrn(N,LAG,MODE,x,y)
for k=0:LAG-1
Nk=N-k; sum=0;
for J=1:Nk
```

```
sum=sum+conj(x(J))*y(J+k);
end
if(MODE==0)
R(k+1)=sum/(N-k);
elseif (MODE~=0)
R(k+1)=sum/N;
end
end


function [A,IFLAG]=solveywe(p,q,R)

IFLAG=0; Eps=10^(-15); AA(1,1)=-R(q+2)/R(q+1);
if(p==1)
A(1)=AA(1,1);
return
end
BB(1,1)=-R(q)/R(q+1);
RHO=(1-AA(1,1)*BB(1,1))*R(q+1);
if(abs(RHO)<Eps)
IFLAG=-1
return
end
for k=2:p
c=-R(q+1+k);
for l=1:k-1
LAG=q+k-l;
if(LAG>=0)
cor=R(LAG+1);
else
cor=conj(R(1-LAG));
end
c=c-AA(l,k-1)*cor;
end
AA(k,k)=c/RHO;
for I=1:k-1
AA(I,k)=AA(I,k-1)+AA(k,k)*BB(k-I,k-1);
end
if (k==p)
for I=1:p
A(I)=AA(I,p);
end
return
end
LAG=q-k;
if (LAG>=0)
d=-R(LAG+1);
else
d=-conj(R(1-LAG));
end
for l=1:k-1
LAG=q-k+l;
if (LAG>=0)
cor=R(LAG+1);
else
```

```
cor=conj(R(1-LAG));
end
d=d-BB(l,k-1)*cor;
end
BB(k,k)=d/RHO;
for I=1:k-1
BB(I,k)=BB(I,k-1)+BB(k,k)*AA(k-I,k-1);
end
RHO=(1-AA(k,k)*BB(k,k))*RHO;
if(abs(RHO)<0)
IFLAG=-1
return
end
end


function y=czerfltn(p,N,A,x)

y(1)=x(1);
if(p==1)
for I=p+1:N
y(I)=x(I)+A(1)*x(I);
end
else
for I=2:p
y(I)=x(I);
for J=1:I-1
y(I)=A(J)*x(I-J);
end
end
for I=p+1:N
y(I)=x(I);
for J=1:p
y(I)=y(I)+A(J)*x(I-J);
end
end
end

function [B,SIG2]=durbinn(q,N,L,x)

% Fit a large order AR model to the data using the  autocorrelation method

[A,SIG2]=autcorr(L,N,x);
L1=L+1;
Ap(1)=1;
for I=2:L1
Ap(I)=A(I-1);
end
% Use the autocorrelation method to generate MA  parameter estimates
[B,Pb]=autcorr(q,L1,Ap);

function [A,SIG2]=autcorr(p,N,x)

% To compute the autocorrelation sequence
R=corrn(N,p+1,1,x,x);
```

```
% To solve Yule-Walker equations using Levinson recursion
[A,AA,RHO]=levisonn(p,R);
SIG2=RHO(p);


function [A,AA,RHO]=levisonn(p,R)

RHO0=R(1);
AA(1,1)=-R(2)/R(1);
RHO(1)=(1-abs(AA(1,1))^2)*R(1);
A(1)=AA(1,1);
if(p==1)
return
end
for I=2:p
B=-R(I+1);
for k=1:I-1
B=B-AA(k,I-1)*R(I+1-k);
end
AA(I,I)=B/RHO(I-1);
for k=1:I-1
AA(k,I)=AA(k,I-1)+AA(I,I)*conj(AA(I-k,I-1));
end
RHO(I)=(1-abs(AA(I,I))^2)*RHO(I-1);
end
for I=1:p
A(I)=AA(I,p);
end
```

Appendix-B:
Computer programs for Power Spectral Density Estimation

```
function pwsd=nfpsd(p,q,A,B,sig,NEXP)

%=====================================================================
%       Computer program to compute the Power Spectral Density (PSD)
%       of a parametric model.
%       This program computes the PSD values across the normalized
%       frequency band [-1/2,1/2], given the parameters of the ARMA
%       model.
%
%       Input Parameters:
%
%       p                    AR model order (for MA process p=0)
%       q                    MA model order (for AR process q=0)
%       A(i), i=1:p          The AR parameters
%       B(i), i=1:q           The MA parameters
%       sig                   Variance of the excitation white noise
%       NEXP                  Power of two which determines number of
%                             frequency samples desired
%
%       Output Parameters:
%
%       pwsd(i), i=1:NEXP    Array of the PSD values
%=====================================================================

EX=2^NEXP; SIG2=sig; a=A; b=B;
% To compute the denominator frequency function
if(p~=0)
den(1)=1;
for I=1:p
den(I+1)=a(I);
and
for I=p+2:EX
den(I)=0;
end
den=fftn(den,1,NEXP);
end
% To compute the numerator frequency function
if(q~=0)
xnum(1)=1;
for I=1:q
xnum(I+1)=b(I);
end
for I=q+2:EX
xnum(I)=0;
end
xnum=fftn(xnum,1,NEXP);
end
```

```
% To compute the PSD values
% For AR Model
if (p==0)
for I=1:EX
P(I)=SIG2*abs(xnum(I))^2;
end
% For MA Model
elseif (q==0)
for I=1:EX
P(I)=SIG2/abs(den(I))^2;
end
else
% For ARMA Model
for I=1:EX
P(I)=SIG2*(abs(xnum(I))^2)/(abs(den(I))^2);
end
end
% Transpose halves of FFT outputs so that first PSD value is at frequency of -1/2.
for I=1:EX/2
pwsd(I+EX/2)=P(I);
pwsd(I)=P(I+EX/2);
end

function x=fftn(x,INVRS,M)

N=2^M;
y(1)=x(1); for I=2:N num=I-1; J=0; l=N;
for k=1:M
l=l/2;
if(num>=l)
J=J+2^(k-1); num=num-l;
end
end
y(I)=x(J+1);
end
% Begin computation of FFT
ldft=1; ndft=N;
for k=1:M
ldft=2*ldft; ndft=ndft/2;
for I=1:ndft
for J=1:ldft/2
ARG=-(2*pi*INVRS)*(J-1.)/ldft;
W=cos(ARG)+sin(ARG)*j; np=J+ldft*(I-1); nq=np+ldft/2;
save=y(np)+W*y(nq); y(nq)=y(np)-W*y(nq);
y(np)=save;
end
end
end
for I=1:N
if(INVRS==1)
x(I)=y(I);
end
if(INVRS==-1) x(I)=y(I)/N;
end
end
```

```
function pspd=psdarma(A,B,sig,fmax,Fs)

%==========================================================================
%       Computer program to compute the Power Spectral Density (PSD) of a parametric
%       model. This program computes the PSD in the frequency range from 0 to Fmax Hz
%       with step equal to 0.1.
%
%       Input Parameters:
%
%       A(i), i=1:p+1        The AR parameters
%       B(i), i=1:q+1        The MA parameters
%       fmax                 The largest frequency to compute the PSD up to it
%       sig                  Variance of the excitation white noise
%       Fs                   Sampling frequency
%
%       Output Parameters:
%
%       pwsd(i), i=1:NEXP  Array of the PSD values
%==========================================================================

f=0:0.1:fmax;
H=FREQZ(B,A,f,Fs);
pspd=sig*(abs(H).^2);
```

Appendix-c:
Computer programs for ARMA Order Estimation

```
function  [p,q]=evarma(pmax,qmax,N,M,y)

%==========================================================================
%       Computer program to compute the order of the ARMA model using
%       the eigenvalue method.
%
%       Input Parameters:
%
%       pmax             The maximum AR order of the ARMA  model to be considered.
%       qmax             The maximum MA order of the ARMA  model to be considered.
%       N                Number of samples desired
%       M                Large order AR process to estimate the unobservable innovations
%       y(i), i=1:N     The observed signal (column vector)
%
%       Output Parameters:
%
%       p               The estimated AR order of the ARMA model
%       q               The estimated MA order of the ARMA model
%==========================================================================

% To estimate the unobservable innovations
N=N-1
for n=0:N
```

```
l=n+1;
for I=1:M
if((n-I)<0)
Y(I)=0;
else
Y(I)=y(n-I+1);
end
end
yy(l,:)=-Y;
end
TH=yy; sum2=zeros(M,M); sum1=zeros(M,1);
for l=1:N+1;
sum1=sum1+TH(l,:)'*y(l); sum2=sum2+TH(l,:)'*TH(l,:);
end
term1=sum1/(N+1); term2=inv(sum2/(N+1)); B=term2*term1;

for n=0:N
l=n+1; e(l)=B'*(-yy(l,:))'; e(l)=e(l)+y(l);
end
e=e';
% To find candidate models with different orders
for q=0:qmax
q1=q+1;
for p=0:pmax
p1=p+1; yp=zeros(N,p+1); I=1;
for J=1:p+1
yp(I:N,J)=y(1:N-I+1,1); I=I+1;
end
E=zeros(N,q+1); I=1;
for J=1:q+1
E(I:N,J)=e(1:N-I+1,1); I=I+1;
end
D=[yp,E]; Rpq=D'*D;
EIG=eig(Rpq); mineig=min(EIG);
JJ(q1,p1)=mineig*(N^(1/N))^(p+q);
end
end
% Row ratio Table
MINq=min(JJ(2,:)./JJ(1,:)); MAorder=1;
for I=1:qmax
Jq(I,:)=JJ(I+1,:)./JJ(I,:); minq=min(Jq(I,:));
if(minq<MINq) MINq=minq; MAorder=I;
end
end
% column ratio Table
MINp=min(JJ(:,2)./JJ(:,1)); ARorder=1;
for J=1:pmax
Jp(:,J)=JJ(:,J+1)./JJ(:,J); minp=min(Jp(:,J));
if(minp<MINp)  MINp=minp; ARorder=J;
end
end
p= ARorder;     q= MAorder
```

```
function [p,q]=aicmdl(pmax,qmax,N,y,FLAG)
%=========================================================================
%       Computer program to compute the order of the ARMA model using
%       the Akaike Information Criterion (AIC) or Minimum Description
%       Length (MDL) methods.
%
%       Input Parameters:
%
%       pmax            The maximum order of the AR part of the ARMA
%                       model to be considered.
%       qmax            The maximum order of the MA part of the ARMA
%                       model to be considered.
%       N               Number of samples desired
%       y(i), i=1:N     The observed signal
%       FLAG            Set equal to: 1 for AIC method
%                                     2 for MDL method
%
%       Output Parameters:
%
%       p               The estimated AR order of the ARMA model
%       q               The estimated MA order of the ARMA model
%=========================================================================

for p=1:pmax
for q=1:qmax
 [A,B,SIG]=mywen(p,q,N,y);
var=SIG;
if(FLAG==1)
cr=N*log(var)+2*(p+q);
elseif(FLAG==2)
cr=(N/2)*log(var)+0.5*(p+q)*log(N);
end
if(p==1&q==1)
MINCR=cr;
end
if cr<=MINCR
MINCR=cr; arord=p; maord=q;
end
end
end
p=arord; q=maord;
```

```
function p=aicar(pmax,N,y)
%=========================================================
%       Computer program to compute the order of the AR model using
%       the Akaike Information Criterion (AIC).
%
%       Input Parameters:
%
%       pmax            The maximum order to be considered.
%       N               Number of samples desired
%       y(i), i=1:N     The observed signal
%
%       Output Parameters:
%
%       p               The estimated AR order
%=========================================================

for p=1:pmax
nn=p; th=ar(y',nn); E=pe(y',th);
var=(std(E))^2; cr=N*log(var)+2*p;
if(p==1) MINCR=cr;
end
if (cr<=MINCR) MINCR=cr; arord=p;
end
end
end
p=arord;
```

Appendix-D:
Computer programs for Adaptive ARMA Models

```
function [A,B]=adarma(p,q,y,N,cs,cp,ff)

%=========================================================
%       Computer program for adaptive ARMA parameters.
%
%       Input Parameters:
%
%       p               AR order  of the ARMA model
%       q               MA order  of the ARMA model
%       y(i), i=1:N     Array for data samples
%       N               Number of data samples
%    0<cs<1             Adaptation variable of the adaptive variance of the observed signal.
%    0<cp<1             Adaptation variable of the adaptive variance of the prediction error.
%       ff              adaptation factor
%
%       Output Parameters:
%
%       A(i,j), i=1:N   The estimated Time-varying AR parameters
%               j=1:p
%       B(i,j), i=1:N   The estimated Time-varying MA parameters
%               j=1:q
%=========================================================
vary(1)=0; varpe(1)=0; e(1)=0; c(1)=0;e=wgn(N,1);
for n=1:N
sum1=0;sum2=0;
for J=1:p
if(J>=n)  A(n,J)=0;
else
A(n,J)=A(n-1,J)-c(n-1)*e(n)*y(n-J);
end
end
for k=1:q
if(k>=n) B(n,k)=0;
else
B(n,k)=B(n-1,k)-c(n-1)*e(n)*e(n-k);
end
end
% adaptive prediction error
for J=1:p
if (J>=n) sum1=sum1;
else
sum1=sum1+A(n-1,J)*y(n-J);
end
end
for k=1:q
if (k>=n) sum2=sum2;
else
sum2=sum2+B(n-1,k)*y(n-k);
end
end
e(n)=y(n)+sum1+sum2;
% adaptive variances
if(n==1) vary(n)=cs*(y(n)^2);
varpe(n)=cp*(e(n)^2);
else
vary(n)=vary(n-1)-cs*(vary(n-1)-y(n)^2);
varpe(n)=varpe(n-1)-cp*(varpe(n-1)-e(n)^2);
end
% adaption fcator
c(n)=ff/vary(n);
end
```

```
function [pmag,pph,zmag,zph]=adppz(p,q,y,N,inab,inpz)
%=====================================================================
%      Computer program for adaptive pole-zero estimation for the ARMA model.
%
%      Input Parameters:
%
%      p                        AR order  of the ARMA model
%      q                        MA order  of the ARMA model
%      y(i), i=1:N              Array for data samples
%      N                        Number of data samples
%      inab(i), i=1:p+q         The initial values of the ARMA   parameters such that the AR
%                                parameters are located first
%      inpz(i), i=1:(p+q)/2     The initial values of the roots of the ARMA model
%                                represented in  polar coordinate such that the radii are
%                                located first.
%
%      Output Parameters:
%
%      pmag(i,j), i=1:N         The estimated magnitudes of the poles
%                 j=1:p/2
%      pph(i,j),   i=1:N        The estimated phases of the poles
%                 j=1:p/2
%      zmag(i,j), i=1:N         The estimated magnitudes of the zeros
%                 j=1:q/2
%      zph(i,j),  i=1:N         The estimated phases of the zeros
%                 j=1:q/2
%=====================================================================

yf=zeros(1,N);ef=zeros(1,N); ee=zeros(1,N); winf=1; wz=0.99;w(2)=0.97; P=.1*eye(p+q);
ma=p/2;mb=q/2; ab(1,:)=inab; pz(1,:)=inpz; zerojac1=zeros(p,mb); zerojac2=zeros(q,ma);
EG(2,:)=zeros(1,p+q);RV(2,:)=zeros(1,p+q);

for t=2:N
n=t-1;
e(t)=y(t)-RV(t,:)*ab(t-1,:)';
L(t,:)=((P*EG(t,:)')/(w(t)+EG(t,:)*P*EG(t,:)'))';
P=(P-L(t,:)'*EG(t,:)*P)/w(t);
pz(t,:)=(pz(t-1,:)'+L(t,:)'*e(t))';   % column
pmag(t,:)=abs(pz(t,1:ma));
zmag(t,:)=abs(pz(t,ma+1:ma+mb));
pph(t,:)=pz(t,ma+mb+1:2*ma+mb);
zph(t,:)=pz(t,2*ma+mb+1:2*ma+2*mb);
% calculate the parameters from the roots
a(t,:)=c2par(p,pmag(t,:),pph(t,:));
b(t,:)=c2par(q,zmag(t,:),zph(t,:));
ab(t,:)=[a(t,:),b(t,:)];
% to calculate the jacobian
n1=p;
[JACRP1,JACPHP1]=jacobi(a(t,:),p,ma,pmag(t,:),pph(t,:));
JACRP=JACRP1(2:n1+1,:);
JACPHP=JACPHP1(2:n1+1,:);
n1=q;
[JACRZ1,JACPHZ1]=jacobi(b(t,:),q,mb,zmag(t,:),zph(t,:));
JACRZ=JACRZ1(2:n1+1,:);
JACPHZ=JACPHZ1(2:n1+1,:);
```

```
% construct the jacobian matrix
JACZP=[JACRP,zerojac1,JACPHP,zerojac1;zerojac2,JACRZ,zerojac2,JACPHZ];
ee(t)=y(t)-RV(t,:)*ab(t,:)'; yyf(t,:)=grv(q,n,yf);
yf(t)=y(t)-b(t,:)*yyf(t,:)'; EEf(t,:)=grv(q,n,ef);
ef(t)=ee(t)-b(t,:)*EEf(t,:)';
for I=1:p
if(n-I+1)<0
Y1(I)=0; Y2(I)=0;
else
Y1(I)=y(n-I+2); Y2(I)=yf(n-I+2);
end
end


for I=1:q
if((n-I+1)<0) Y3(I)=0; Y4(I)=0;
else
Y3(I)=ee(n-I+2); Y4(I)=ef(n-I+2);
end
end
RV(t+1,:)=[-Y1,Y3]; RVF(t+1,:)=[-Y2,Y4];
EG1=JACZP'*RVF(t+1,:)';          EG(t+1,:)=EG1';
w(t+1)=winf-(winf-w(t))*wz;
end


 function  [JACR,JACPH]=jacobi(A,n1,m1,mag,ph)

A=[1,A]; JACR(1,:)=zeros(1,m1);
JACPH(1,:)=zeros(1,m1);
k=1:m1;
JACR(2,:)=-2*cos(ph(k)); JACPH(2,:)=2*mag(k).*sin(ph(k));
for I=3:n1+1
for k=1:m1
JACR(I,k)=2*mag(k)*cos(ph(k))*JACR(I-1,k)-(ph(k)^2)*JACR(I-2,k)-2*cos(ph(k))*A(I-
1)+2*mag(k)*A(I-2);
JACPH(I,k)=2*mag(k)*cos(ph(k))*JACR(I-1,k)-(ph(k)^2)*JACR(I-
2,k)+2*mag(k)*sin(ph(k))*A(I-1);
end
end

function a1=c2par(m,mag1,ph1)

aa=zeros(m,m/2);
for J=1:m/2
for I=1:2*J
if(J==1&I==1) aa(I,J)=-2*mag1(J)*cos(ph1(J));
elseif(J==1&I==2) aa(I,J)=mag1(J)^2;
elseif(I==1) aa(I,J)=aa(I,J-1);
elseif(I==2) aa(I,J)=aa(I,J-1)-2*mag1(J)*aa(I-1,J-1)*cos(ph1(J));
else
aa(I,J)=aa(I,J-1)-2*mag1(J)*aa(I-1,J-1)*cos(ph1(J))+mag1(J)^2*aa(I-2,J-1);
end
end
end
```

```
for I=1:m
a1(I)=aa(I,m/2);
end

function Y=grv(M,n,y1)

for I=1:M
if((n-I)<0) Y(I)=0;
else
Y(I)=y1(n-I+1);
end
end
```

Appendix-E: Computer program for ARMA modeling of systems with multiple inputs and delays

```
function  [p,q1,s1,q2,s2,par]=apr(pmax,q1max,q2max,s1,s2,N,x1,x2,e,y)

%=================================================================
%      Computer program for ARMA modeling of systems with multiple
%      inputs and delays.
%
%      Input Parameters:
%
%      pmax                The maximum AR order of the ARMA
%                          model to be considered.
%      q1max               The maximum MA order of the first input
%                          to be considered.
%      q2max               The maximum MA order of the second input
%                          to be considered.
%      N                   Number of samples desired.
%      s1                  The initial delay of the first input signal
%      s2                  The initial delay of the second input signal
%      x1(i), i=1:N        The first input signal
%      x2(i), i=1:N        The second input signal
%      e(i), i=1:N         Additive white noise
%      y(i), i=1:N         The observed signal
%
%      Output Parameters:
%
%      p                   The estimated AR order of the ARMA model
%      q1                  The estimated MA order of the first input
%      q2                  The estimated MA order of the second input
%      s1                  The estimated delay of the first input
%      s2                  The estimated delay of the second input
%   par(i), i=1:q1+q2+p    The estimated parameter values arranged as
%                          follows: MA parameters for the fist input,
%                          MA parameters for the second input and the
%                          AR parameters.
%=================================================================

q=q1max; q2=q2max; p=pmax; x=x1'; x2=x2'; y=y'; xp=zeros(N,q+1);sig2=(std(e))^2; I=1;

for J=1:q+1
xp(I:N,J)=x(1:N-I+1,1); I=I+1;
end
xp2=zeros(N,q2+1); I=1;
for J=1:q2+1
xp2(I:N,J)=x2(1:N-I+1,1); I=I+1;
end
yp=zeros(N,p); I=1;
for J=1:p
yp(I:N,J)=y(1:N-I+1,1); I=I+1;
end
y1=yp(1:N-1,:); y2=[zeros(1,p);y1];
```

```
D=[xp,xp2,y2]; x=x'; y=y';
pv=inv(D'*D)*D'*y';     v=inv(D'*D)*D'*e';

% AR parameters removal
kmax=pmax+q1max-s1+q2max-s2+2;
L=1;    k=kmax;             mtnp1=k;   ARord=pmax;
acv=sig2*inv(D'*D);     acv1=acv;
pe(L,:)=(y'-D*pv)'; nors=norm(pe(L,:))^2;
[q,r]=qr(D);
pv1=pv;           mind1=(1+k*(log(N)/N))*(nors/N);

for I=1:pmax
mdl1(k)=(1+k*(log(N)/N))*(nors/N);
if(mdl1(k)<=mind1)
mind1=mdl1(k);                      mtnp1=k;
pv1=pv(1:k,1);  D1=D(:,1:k); acv1=acv(:,1:k); L2=L;
end

% remove one parameter
q1=q;   k=k-1;  L=L+1;
[q,r]=qr(D(:,1:k)); clear r
pe(L,:)=(pe(L-1,:)'+q1(:,k+1)*q1(:,k+1)'*y')';
nors=norm(pe(L,:))^2;
pv(1:k,1)=inv(D(:,1:k)'*D(:,1:k))*D(:,1:k)'*y';
v(1:k,:)=inv(D(:,1:k)'*D(:,1:k))*D(:,1:k)'*e';
acv=sig2*inv(D(:,1:k)'*D(:,1:k));
end
num1=kmax-mtnp1; ARord=pmax-num1;

% calculating the SNR of MA parameters
for I=1:mtnp1
nm(I)=acv1(I,I);
end
pv1s=pv1'.^2;   ratio=pv1s./nm;
SNR=ratio.^0.5;
SNRMA1=SNR(1,1:q1max+1);
SNRMA2=SNR(1,q1max+2:mtnp1-ARord);

% MA parameters removal
k=mtnp1;        L3=1;
pe2(L3,:)=(y'-D1*pv1)';
nors=norm(pe2(L3,:))^2;
mind2=(1+k*(log(N)/N))*(nors/N);
pv2=pv1(1:k,1);[q2,r]=qr(D1);
clear r

for I=1:mtnp1-1
mdl2(k)=(1+k*(log(N)/N))*(nors/N);
if(mdl2(k)<=mind2)
mind2=mdl2(k);                      mtnp2=k;
pv2=pv1(1:k,1);           D2=D1(:,1:k);
acv2=acv1(:,1:k);
end
```

```
% remove one parameter
q3=q2;  k=k-1;  L3=L3+1;
[q2,r]=qr(D1(:,1:k)); clear r
pe2(L3,:)=(pe2(L3-1,:)'+q3(:,k+1)*q3(:,k+1)'*y')';
nors=norm(pe2(L3,:))^2;
pv1(1:k,1)=inv(D1(:,1:k)'*D1(:,1:k))*D1(:,1:k)'*y';
v1(1:k,:)=inv(D1(:,1:k)'*D1(:,1:k))*D1(:,1:k)'*e';
acv1=sig2*inv(D1(:,1:k)'*D1(:,1:k));
end

MAord=mtnp2-ARord+1;
pqr=pv2;

SNRT=[SNRMA1,SNRMA2];
jj=sort(SNRT);
k=length(jj);
for I=1:k
maxsnr(I)=jj(k);
k=k-1;
end
z1=1;   z2=1;
for I=1:MAord
for k=1:q1max
if(bcoef(I)==SNRMA1(k))
maxb1(z1)=SNRMA1(k); ind1(z1)=k; z1=z1+1;
end
end
end

for I=1:MAord
for k=1:q2max
if(bcoef(I)==SNRMA2(k))
maxb2(z2)=SNRMA2(k);ind2(z2)=k; z2=z2+1;
end
end
end
[d1,d2]=max(maxb1);   [d3,d4]=max(maxb2);
[r1,r2]=min(maxb1);   [r3,r4]=min(maxb2);
q1=ind1(d2); s1=ind1(r2); q2=ind2(d4); s2=ind2(r4);
p=ARord; par=pv2;
```

Appendix-F:
Computer program for Parametric Methods efficiency

```
function  []=residtes(p,q,N,y)

%==================================================================
%      Computer program to evaluate the efficiency of the parametric models for both
%      ARMA and AR models.
%
%      Input Parameters:
%
%      p                 The AR order of the ARMA model
%      q                 The MA order of the ARMA model  (q=0 for an AR process)
%      N                 Number of data samples
%      y(i), i=1:N       The observed signal
%==================================================================

sum1=0; sum2=0;s1=0; s2=0;
% For AR Model
if(q==0) nn=p;  th = ar(y',nn);
else
%) For ARMA Model
nn=[p,q]; th=armax(y',nn);
end
% Test for the lack of correlation terms
M=N/2; E1 = pe(y',th);
r=covf(E1,M); var=(std(E1))^2;
rn=r/var; limit=1.96/sqrt(N);
% 2) Test for the lack of sine terms
k2=1.35;
for k=1:N/2
w(k)=2*pi*k/N;
tole1(k)=(2*k/N)+k2/sqrt((N/2)-1);
tole2(k)=(2*k/N)-k2/sqrt((N/2)-1);
end
for I=1:N/2
for J=1:N/2
sum1=sum1+E1(J)*cos(w(I)*J);
sum2=sum2+E1(J)*sin(w(I)*J);
end
tcrm1=((2/N)*sum1)^2; term2=((2/N)*sum2)^2;
sqc(I)=term1+term2; sum1=0;sum2=0;
end
for I=1:N/2
s1=s1+sqc(I);
end
for K=1:N/2
for I=1:K
s2=s2+sqc(I);
end
g(K)=s2/s1; s2=0;
end
```

```
% To plot the two tests
figure(1)
for I=1:M
limit2(I)=limit;
end
s=1:M;
plot(s,limit2,s,-limit2,s,rn);
figure(2)
k=1:N/2; tol1=tole1;     tol2=tole2;
plot(k,tol1,k,tol2,k,g)
xlabel('f (Hz)'); ylabel(' g(k)');
```

Appendix-G:
Computer program for Simulating EEG signals

```
function [EEG,PF]=EEGSIMN(M,F,B,r,N,Fs,fmax)

%==========================================================================
%       Computer program to simulate different types of EEG signals with certain
%       characteristics. The simulated EEG signal is represented by a summation of
%       second  order AR series.
%       This simulation technique is described by Weiss (1986).
%
%       Input Parameters:
%
%       M                  Number of resonances
%       F(i), i=1:M        Frequency associated with each resonance
%       B(i), i=1:M        Bandwidth associated with each resonance
%       r(i), i=1:M        Relative amplitude associated with each resonance
%       N                  Number of samples desired
%       Fs                 Sampling frequency
%       fmax               The largest frequency to compute the PSD up to it
%
%       Output Parameters:
%
%       EEG(i), i=1:N   The simulated EEG signal
%       PF(i), i=1:Fr   The power spectral density of the EEG signal
%==========================================================================

sum2=0; sum3=0; sum4=zeros(1,N); T=1/Fs; Fr=fmax;
for I=1:M
a2(I)=0.06*B(I)-0.9;
a1(I)=(4*a2(I)/(a2(I)-1))*cos(2*pi*F(I)*T);
D(I)=ps(a1(I),a2(I),F(I),T);
y1(I)=(1-a2(I))^2-a1(I)^2; y2(I)=(1+a2(I))/(1-a2(I));
sum1(I)=r(I)*D(I)*y1(I)*y2(I);
end
for I=1:M
sum2=sum2+r(I); sum3=sum3+sum1(I);
end
```

```
kn=sum2/sum3;
for I=1:M
a3(I)=sqrt(kn*r(I)*D(I));
end
% To generate a second order AR series
sum1=zeros(1,N); x=wgn(N,1);
for I=1:M
A=[1,-a1(I),-a2(I)];                B=a3(I);
yy(I,:)=filter(B,A,x); sum1=sum1+yy(I,:);
end
EEG=sum1;
% To compute the PSD of the simulated EEG signal in the frequency
% range from 0 to Fr Hz with step 0.1 as illustrated in the function eegpsd.

PF=eegpsd(a1,a2,a3,M,T,Fr);


function w=wgn(N,VAR)

%==================================================================
%       Computer program to generate White Gaussian noise.
%
%       Input Parameters:
%
%       N               Number of white noise samples desired
%       VAR             Variance of white noise samples desired
%
%       Output Parameters:
%
%       w(i), i=1:N     The generated white noise signal
%==================================================================

M=N;
for I=1:M
w(I)=rand;
end
L=M/2;
for I=1:L
u1=w(2*I-1); u2=w(2*I);
temp=sqrt(-2*log(u1));
w(2*I-1)=temp*cos(2*pi*u2)*sqrt(VAR);
w(2*I)=temp*sin(2*pi*u2)*sqrt(VAR);
end

function PF=eegpsd(a1,a2,a3,M,T,Fr)
fr=0:0.1:Fr; sum5=zeros(1,length(fr));
for I=1:M
DD(I,:)=ps(a1(I),a2(I),fr,T);
pp(I,:)=(2*T*a3(I)^2)./DD(I,:);
sum5=sum5+pp(I,:);
end
PF=sum5;

function D1=ps(A1,A2,fr,T)
```

```
for J=1:length(fr)
D1(J)=1+A1^2+A2^2-2*A1*(1-A2)*cos(2*pi*fr(J)*T)-2*A2*cos(4*pi*fr(J)*T);
end
```

Appendix-H:
Computer program for Spectral Parameter Analysis

```
function [fr,BW,pown]=spcomp(p,q,A,B,sig,Fs)
%==========================================================================
%     Computer program to compute the spectral component parameters of an EEG signal
%     using the Spectral Parameter Analysis (SPA)  technique. This program computes the
%     frequency, bandwidth and power of the various resonances in the signal.
%
%     Input Parameters:
%
%     p                The AR order of the ARMA model
%     q                The MA order of the ARMA model
%     A(i), i=1:p+1    The AR parameters
%     B(i), i=1:q+1    The MA parameters
%     sig              Variance of the excitation noise
%     Fs               The sampling frequency
%
%     Output Parameters:
%
%     fr(i), i=1:p     The estimated spectral component frequencies
%     BW(i), i=1:p     The estimated spectral component bandwidths
%     pown(i), i=1:p   The estimated spectral component relative powers
%==========================================================================


sum=0; T=1/Fs; aa=A; bb=B; pr=roots(A); zr=roots(B); a=A(1,2:p+1);
% for an AR model
if(size(B)==1)b=1;
else
b=B(1,2:q+1);
end
% To compute the frequencies and bandwidths of the various resonances
for k=1:p
fr(k)=(1/(2*pi*T))*imag(log(pr(k)));
if(fr(k))==0
BW(k)=-log(abs(pr(k)))/(2*pi*T);
else
BW(k)=-log(abs(pr(k)))/(pi*T);
end
end
% to compute the relative powers of the various resonances
for k=1:p
mul=1;
for J=1:p
if(J~=k)
mul=mul*(pr(k)-pr(J))*(1/pr(k)-conj(pr(J)));
end
```

```
end
res(k)=1/(pr(k)*(1/pr(k)-conj(pr(k)))*mul);
end
for I=1:p
if(imag(pr(I))==0)
pow(I)=sig*abs(real(res(I))); sum=sum+pow(I);
elseif(imag(pr(I))>0)
pow(I)=2*sig*abs(real(res(I))); sum=sum+pow(I);
end
end
pown=pow/sum;
```

Appendix-I:
Computer program to find the Minimum Phase Component of a Signal

```
function mphy=minph(N,y)

%==================================================================
%       Computer program to find the minimum phase of a given signal
%
%       Input Parameters:
%
%       N                  Number of data samples
%       y(i), i=1:N        The given signal to find its minimum phase component
%
%       Output Parameters:
%
%       mphy(i), i=1:N     The minimum phase signal
%==================================================================


mphy= zeros(1,N);
x1 = real(ifft(log(abs(fft(y)))));
if(rem(N,2)==0)
w = [1; 2*ones((N/2)-1,1) ; 1; zeros((N/2)-1,1)];
else
w = [1; 2*ones((N/2)-1,1) ;  zeros((N/2)-1,1)];
end
x2=w.*x1';          x3=fft(x2);
x4=exp(x3);         x5=ifft(x4); mphy=real(x5)';
```

Appendix-J:
Computer program For spike detection

```
function ST=spikes(N,r,sw)

%=====================================================================
%       Computer program to generate spike train signal. The spike has
%       a width of nT (n=2sw, sw=1,2,3,..., and T is the sampling
%       interval). The spike position, amplitude and sign of the spikes
%       are generated by a random number generator.
%
%       Input Parameters:
%
%       N                  Number of samples desired for the spike signal
%       r                  Number of spikes
%       sw                 spike width
%
%       Output Parameters:
%
%       ST(i), i=1:N    The generated spike train signal
%=====================================================================

qq=sw; delta1=zeros(1,N); delta2=zeros(1,N); d1=zeros(1,N);d2=zeros(1,N);

for I=1:r
s=round(rand); amp(I)=(-1)^s*(2.5+5*rand); pos(I)=2*qq+(N-4*qq)*rand;
end
pos1=sort(pos); pos(1)=pos1(1); pos(r)=pos1(r); dif=round((pos(r)-pos(1))/r);
for I=2:r-1
pos(I)=pos(I-1)+((dif/2)+dif*rand);
end
pos=round(pos);
for n=1:N
for k=1:r
if n==pos(k) delta1(n)=amp(k);
end
end
end
for n=1:N
for k=1:r
for m=1:qq
if n==pos(k)-m
d1(n)=(amp(k)*(qq-m)/qq);
end
end
for m=1:qq
if n==pos(k)+m
d2(n)=(amp(k)*(qq-m)/qq);
end
end
delta2(n)=d1(n)+d2(n);
end
end
ST=delta1+delta2;
```

```
function [spik,FP,FN,BIAS]=spikdet(p,q,N,y,sw)

%=====================================================================
%      Computer program for spike detection using linear prediction error filter. The spike
%      detection technique is evaluated using  the performance indeces, namely, false
%      positive ratio, false negative ratio and the bias.
%
%      Input Parameters:
%
%      p                  The AR order of the ARMA model
%      q                  The MA order of the ARMA model
%      N                  Number of data samples
%      y(i), i=1:N        Signal contains both the stationary and
%                         nonstationary (spike) signals
%      sw                 spike width
%
%      Output Parameters:
%
%      FP                 False positive ratio
%      FN                 False negative ratio
%      BIAS               The bias of spike detection
%      spik(i), i=1:N     The recovered spike signal
%=====================================================================


qq=sw; spkN=0; nn=[p,q]; TH1 = armax(y',nn);
E1 = pe(y',TH1); E2=-E1; rr=2; kk=1;
for I=1:2*qq-1
s(kk)=-(qq-I); kk=kk+1;
end
spik=zeros(1,N); EE1=E1;
z=0;    flag=1;
while(flag==1)
[ps,indp]=pnsep(E1',N,1); [m1,ind]=max(ps);
if(z==1)
Emax=m1;
end
z=z+1; n=indp(ind);
if(E1(n+qq/2+1)<0)
if(rem(qq,2)==0)
pos1(z)=n+(qq/2);
else
pos1(z)=n+(qq-1)/2;
end
else
if(rem(qq,2)==0) pos1(z)=n-(qq/2);
else
pos1(z)=n-(qq-1)/2;
end
pos1(z)=n-qq;
end
if(m1<Emax/2)
flag=0;
end
```

```
E1(n-qq:n+qq,1)=zeros(2*qq+1,1);
end
r=z; E1=EE1; pamp=r;  namp=r; pos=pos1;
E1=EE1; L=1; pspk=0; nspk=0; d1=0;
for k=1:pamp
[ps,indp]=pnsep(E1',N,1);
[vr,ind]=max(ps); n=indp(ind);
xx=EE1(n-qq:n+qq,1); xxmin=min(xx);
while(vr<abs(xxmin))
E1(n-qq:n+qq,1)=zeros(2*qq+1,1);
[ps3,indp3]=pnsep(E1',N,1);
[vr,ind3]=max(ps3); n=indp3(ind3);
xx=EE1(n-qq:n+qq,1); xxmin=min(xx);
end
spos(L)=n;
for I=1:length(s)
spik(n+s(I))=vr/(1+abs(s(I)));
E1(n+s(I))=0;
if s(I)==0
spkN=spkN+1;
d1=d1+1;
spkpos(d1)=n;
end
end
for J=0:rr
spik(n+qq+J)= EE1(n+qq+J)+E2(n+qq+J);
spik(n-qq-J)=EE1(n-qq-J)+E2(n-qq-J);
E1(n+qq+J)=0; E1(n-qq-J)=0;
end
L=L+1; pspk=pspk+1;
end
E1=EE1; L=1; d2=0;
for k=1:namp
[ns,indn]=pnsep(E1',N,2); [vr,ind]=min(ns);
spos(L+pspk)=indn(ind); n=spos(L+pspk);
for I=1:length(s)
spik(n+s(I))=vr/(1+abs(s(I)));
E1(n+s(I))=0;
if s(I)==0
spkN=spkN+1;
d2=d2+1;
spkpos(d1+d2)=n;
end
end
for J=0:rr
spik(n+qq+J)= EE1(n+qq+J)+E2(n+qq+J);
spik(n-qq-J)=EE1(n-qq-J)+E2(n-qq-J);
E1(n+qq+J)=0; E1(n-qq-J)=0;
end
L=L+1; nspk=nspk+1;
end
spkpos=spkpos+2; l=1; NTS=0;
for I=1:spkN
for k=1:r
if spkpos(I)==pos(k)
```

```
NTS=NTS+1;
cspos(l)=pos(k);
l=l+1;
end
end
end
FN=(r-NTS)/r; FP=(spkN-NTS)/r;
sumb=0;
for l=1:NTS
sumb=sumb+abs(pos(l)-spkpos(l));
end
BIAS=sumb/NTS;


function [pns,ind1]=pnsep(em,N,flag)

J=1;
if flag==1
for n=1:N
if em(n)>=0
pss(J)=em(n); indp(J)=n; J=J+1;
end
end
pns=pss; ind1=indp;
end
if flag==2
for n=1:N
if em(n)<0
ns(J)=em(n); indn(J)=n; J=J+1;
end
end
pns=ns; ind1=indn;
end
```

References

- Binnie, C. D., Rowan, A.J. and Gutter, TH. 1982. A Manual of Electroencephalographic Technology. Cambridge University Press, London.
- Bruzzone, S. P. and Kaveh, M. 1984. Information Tradeoffs in Using the Sample Autocorrelation Function in ARMA Parameter Estimation. IEEE Transactions on Acoustics, Speech, and Signal Processing, 32: 701-715.
- Cadzow, J. A. 1982. Spectral Estimation: An Overdetermined Rational model Equation Approach. Proceeding of the IEEE, 70 (9): 907-939.
- Cadzow, J. A., Solomon, O. M. 1986. Algebraic Approach to System Identification. IEEE Transactions on Acoustics, Speech, and Signal Processing, 34 (3): 462-469.
- Cerutti, S., Chiarenza, G., Liberati, D. Mascellani, P. and Pavesi, G. 1988. A Parametric Method of Identification of Single-Trial Event-Related Potentials in the brain. IEEE Transactions on Biomedical Engineering, 35 (9): 701-711.
- Cooper, R., Osselton, J. W., and Shaw, J. C. 1980. EEG Technology, *third edition*. Butterworths, England.
- Daly, D. D. and Pedley, T.A. 1990. Current Practice of Clinical Electroencephalography, *second edition*. Raven Press, New York.

- Daskalova, M. I. 1988. Wave Analysis of the Electroencephalogram. Medical and Biological Engineering and Computing, 26: 425-428.
- Fender, D. H., Arakawa, K., Harashima, H., Miyakawa, H. and Saitoh, Y. 1986. Separation of a Nonstationary Component from the EEG by a Nonlinear Digital Filter. IEEE Transactions on Biomedical Engineering, 33 (7): 724-726.
- Ferdjallah, M. and Barr, R. E. 1996. Enhanced Period-Peak Analysis of Electroencephalograms Using a Fast Sinc Function. Medical and Biological Engineering and Computing, 34: 431-435.
- Goel, V., Brambrink, A. M., Baykal, A., Hanely, D. F., and Thakor, N. V. 1996. Dominant Frequency Analysis of EEG Reveals Brain's Response during Injury and Recovery. IEEE Transactions on Biomedical Engineering, 43 (11): 1083-1092.
- Grenier, Y. 1983. Time-Dependent ARMA Modeling of Nonstationary signals. IEEE Transactions on Acoustics, Speech, and Signal Processing, 31 (4): 899-911.
- Gvins, A. S. 1984. Analysis of the Electromagnetic Signals of the Human Brain: Milestones, Obstacles, and Goals. IEEE Transactions on Biomedical Engineering, 31 (12): 833-850.
- Haseyama, M., Nagai, N. and Miki, N. 1993. An Adaptive ARMA Four-Line Lattice Filter for spectral Estimation with Frequency Weighting. IEEE Transactions on Signal Processing, 41 (6): 2193-2207.

- Henry, C. E. 1980. Current Clinical Neurophysiology Update on EEG and Evoked Potentials. Elsever, North-Holand.
- Hughes, R. 1982. EEG in Clinical Practice. Butterworths, England.
- Isaksson, A., Wennberg, A. and Zetterberg, L. H. 1981. Computer Analysis of EEG Signals with Parametric Models. Proceeding of the IEEE, 69 (4): 451-461.
- Jacobs, M. H., Rao, S. S. and Jose, G. V. 1989. Parametric Modeling of Somatosensory Evoked Potentials. IEEE Transactions on Biomedical Engineering, 36 (3): 392-403.
- Jansen, Ben H. 1985. Analysis of Biomedical Signals by Means of Linear Modeling. CRC Critical Reviews in Biomedical Engineering, 12 (4): 343-392.
- Kay, S. M. and Marple, S. L. 1981. Spectrum Analysis- A Modern Perspective. Proceeding of the IEEE, 69 (11): 1380-1418
- Kay, S. M. 1988. Modern spectral Estimation: Theory and Application. Prentice Hall, New Jersey.
- Korenberg, M. J. and Paarmann, L. D. 1989. An Orthogonal ARMA Identifier with Automatic Order Estimation for Biological Modeling. Annals of Biomedical Engineering, 17: 571-592.
- Liang, G., Wilkes, M. and Cadzow, J. A. 1993. ARMA Model Order Estimation Based on the Eigenvalues of the covariance Matrix. IEEE Transactions on Signal Processing, 41 (10): 3003-3009.

- Ljung, L. 1987. System Identification-Theory for the User, Englewood Cliffs, Nj: Prentice Hall.
- -Makhoul, J. 1975. Linear Prediction: A Tutorial Review. Proceeding of the IEEE, 63 (4): 561-581.
- Marple, S. 1987. Digital Spectral Analysis with Applications, Englewood Cliffs, Nj: Prentice Hall.
- Nakamura, M., Nishida, S. Neshige, R. and shibasaki, H. 1985. Quantitative Analysis of 'Organization' by Feature Extraction of the EEG Power Spectrum. Electroencephalography and Clinical Neurophysiology, 60: 84-89.
- Nehorai, A. and Staper, D. 1990. Adaptive Pole Estimation. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38 (5): 825-838.
- Nehorai, A. and Stoica, P. 1988. Adaptive Algorithms for Constrained ARMA Signals in the Presence of Noise. IEEE Transactions on Acoustics, Speech, and Signal Processing, 36 (8): 1282-1291.
- Paarmann, L. D., Yarman, F. and Onaral, B. 1987. Parametric Estimation of the Spontaneous EEG Under Acceleration Stress. IEEE Engineering in Medicine and biology.
- Proakis, J. G. and Manolakis, D. G. 1992. Digital Signal Processing Principles, Algorithms and Applications, *second edition*. Macmillan, New York.

- Perrott, M. H. and Cohen, R. J. 1996. An Efficient Approach to ARMA Modeling of Biological Systems with Multiple Inputs and Delays. IEEE Transactions on Biomedical Engineering, 43 (1): 1-13.
- Pillai, S. U., Shim, T. I. and Youla, D. C. 1993. A New Technique for ARMA-System Identification and Rational Approximation. IEEE Transactions on Signal Processing, 41 (3): 1281-1304.
- Politis, D. N. 1993. ARMA Models, Prewhitening, and Minimum Cross Entropy. IEEE Transactions on Signal Processing, 41 (2): 781-787.
- Popinanov, D. 1992. Time Series Analysis of Brain Potentials Preceding Voluntary Movements. Medical and Biological Engineering and Computing, 30: 9-14.
- Qian, J., Barlow, J. S. and Beddoes, M. P. 1988. A Simplified Arithmetic Detector for EEG Sharp Transient-Preliminary Results. IEEE Transactions on Biomedical Engineering, 35 (1): 11-18.
- Ray, G. C. 1994. An Algorithm to Separate Nonstationarity Part of a Signal Using Mid-Prediction Filter. IEEE Transactions on Signal Processing, 42 (9): 2276-2279.
- Rogowski, Z., Gath, I. And Bental, E. 1981. On the Prediction of Epileptic seizures. Biol. Cybernet., 42: 9-15.
- Schack, B., Bareshova, E., Grieszbach, G. and Witt, H. 1995. Methods of Dynamic Spectral analysis by Self-exciting Autoregressive Moving Average Models and Their Application to Analyzing Biosignals. Medical and Biological Engineering and Computing, 33: 492-498.