DATA COMPRESSION TECHNIQUES AS APPLIED TO THE ELECTROENCEPHALOGRAPH SIGNAL

BY

GALAL EZZ E. NADIM

A Thesis

Submitted to the Faculty of Graduate Studies in Partial

Fulfillment of the Requirements for the Degree of

Master of Science

Department of Electrical Engineering

University of Manitoba

December 1982

Winnipeg, Manitoba

DATA COMPRESSION TECHNIQUES AS APPLIED TO

THE ELECTROENCEPHALOGRAPH SIGNAL

BY

GALAL EZZ E. NADIM

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

This thesis presents a study of the Karhunen-Loéve expansion (KLE) data compression technique as applied to EEG signal. The KLE technique was found to give a data compression ratio of 2.56:1, which is found to be feasible for a mean-square error less than 15%. Typically, the Karhunen-Loéve expansion was found to cause high frequency information losses. Straight line interpolation techniques were also investigated. The straight line interpolation techniques were found to be inferior to KLE, because, in general, they introduce a higher amount of distortion to the signal. However, they were able to better approximate spike activity in the EEG. Terminal classifiers were also designed to test the KLE ability to preserve the EEG signal characteristics after approximation and to recognize the epileptic abnormalities associated with EEG.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation and gratitude to his supervisor, Dr. E. Shwedyk, for his valuable guidance, generous assistance and continuous encouragement during the progress of this work.

Thanks are also due to Mrs. Violet M. Lee for her outstanding typing. Special thanks are due to all who contributed to this work through discussion and suggestions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1-1 Overview

Electroencephalograms, or EEG's, are recordings of the electrical activity of the brain. These recordings are useful, because many physiological abnormalities produce characteristic waveforms in the brain's activity. Today, EEG's are used routinely to provide information relevant to the diagnosis, prognosis, and treatment of a great variety of abnormal conditions, including epilepsy, cerebral tumors and thrombosis, developmental abnormalities, and metabolic and endocrine disorders [1]. They are also used as a research tool in neurophysiology, and for monitoring sleep. The traditional method of EEG interpretation involves visual scoring by trained electroencephalographers. Keeping in mind what is known about a patient, the electroencephalographer observes the EEG's frequency and amplitude, as well as spatial and temporal relations between patterns which occur. The possible clinical significance of these observations is then evaluated.

EEG consists mainly of three basic types of activity - normal background rhythms, artefacts, and transients [24]. Normal background activity is generally classified, based on its dominant frequency, into one of 5 frequency bands: Delta (1-4 HZ), Theta (4-8 HZ), Alpha (8-12 HZ), Sigma (12-14 HZ) and Beta (14-35 HZ). The second category of waveforms observed in the EEG is artefact, which is defined as any potential difference due to an extracerebral source. Artefacts are generally undesirable disturbances which may be transient or last throughout an entire record. There are several sources of artefact, including head, arm or leg movement, eye blinks, faulty electrode connections, and extraneous physiological signals such as ECG, EOG, and EMG activity. The third type of activity present in the EEG is transient activity. Transients appear in a variety of forms, the most important of which are associated with epilepsy.

Unfortunately, there are many problems associated with traditional methods of EEG analysis. The clinical EEG is, to a large extent, more of an art than of a science [2]. Evaluation of records is very dependent on such factors as the amount and kind of training and experience. There is also considerable variation in evaluations by an individual interpretor, not only from day to day and subject to subject, but in the same recording [3]. This lack of consistency and standardization in the evaluation of EEG recordings limits their usefulness as a diagnostic tool.

Additional problems have been posed by the advent of prolonged EEG recordings. These recordings, which generally last 24 to 48 hours (compared with 20 minutes for a standard EEG), produce large amounts of data. For example, a 24 hour recording on a standard EEG machine would produce 2.5 km of writeout, and if it is sampled at a rate of 200 samples/second, it will need a memory size of nearly 69 Mbyte (using one byte to store each sampled value). It is clear that a huge storage area is required to store these recordings for later use, beside that the analysis by hand of such an amount of data on a routine basis presents a formidable task.

A possible solution to the above stated problems is to develop a computerized system for EEG analysis. Such a system could offer many advantages, including an objective, consistent evaluation of the EEG. Automated analysis would free highly skilled electroencephographers from the labour of evaluating a large number of easily classifiable records and allow them to concentrate on tasks which make better use of their expertise. Also, it is of interest to develop methods of achieving a substantial reduction in the data volume required to store these records. Computer-based techniques hold a particular promise for evaluating prolonged EEG recordings, and for reducing the storage needed by these recordings. Clearly, based on the preceeding discussion, there is considerable motivation for developing computerized systems for EEG analysis.

1-2 Format of the Thesis

This thesis describes research aimed at investigating the capability of various compression techniques in reducing the memory size required to store the EEG recording, and developing an automated system to recognize abnormalities in the original EEG signal as well as in the compressed signal. In particular, a computer-implemented system is presented which employs the Karhunen-Loéve expansion (KLE) technique to achieve an acceptable reduction in the required storage area. In addition, a study was made to implement a computer-based terminal classifier to detect abnormalities associated with epilepsy, and to check the validity of the data compression technique.

The primary objectives of this research were (1) to investigate the feasibility of using data compression techniques, in particular KLE, to compress the EEG signal, without altering its characteristics, (2) to investigate the feasibility of using pattern recognition techniques in recognizing epileptic abnormalities, and (3) to lay the ground work for future research.

This thesis consists of two major sections.

Initially, in chapter 2 and chapter 3, we discuss the basic theoretical background of different data compression techniques and their efficiency with the signal in question, i.e., the EEG signal. The remainder of the thesis is concerned with the designing of a terminal classifier to detect abnormalities associated with the EEG.

In Chapter 2, a theoretical introduction for the data compression problem by way of orthogonal transformations, with major emphasis on the Karhunen-Loève transform, and a discussion of a proposed system is provided.

In Chapter 3, an investigation of other data compression techniques - especially straight line interpolation - is illustrated, and a comparison is given between the different techniques with regard to their validity for the EEG signal.

In Chapter 4, a design of a terminal classifier is given, and a comparison is provided between various types of terminal classifiers regarding their efficiency in detecting EEG abnormalities.

Chapter 5 concludes this thesis by proposing possible improvements to the EEG analysis system, and by suggesting areas for future research.

CHAPTER 2

DATA COMPRESSION PROBLEM AND SOLUTION USING KARHUNEN-LOÈVE EXPANSION

2-1 Introduction

The EEG signal, in general, is a nonstationary random signal. But the assumption that the statistical properties do not change during short sections, leads to the general descriptive model of an EEG signal, which assumes the signal to consist of quasistationary segments on which transients may be superimposed [6]. The presence of such segments in the EEG suggests that its treatment as a locally stationary signal would be valid. Therefore, the first step in EEG analysis is often to divide it into manageable lengths. Different segmentation procedures are discussed in [4], [5]. In our case, a one second segement length is quite adequate, and ensures that the signal is stationary within this interval.

Prolonged records of EEG signal range from 12 hours to 48 hours, depending upon the case to be studied, with usually 4 recorded channels. If these records are to be stored for later purposes, a storage medium with a large memory capacity will be required. For example, a 12 hour recording, on a 4-channel EEG recorder, sampled at a rate of 128 samples/second, needs a memory size of almost 20 Mbytes, if each sampled value of an EEG is stored using one byte in storage. Therefore, it is necessary to develop methods to reduce the data volume needed for EEG recording storage. Data compression leads to savings in the amount of memory required to store the EEG in digital form. Such compression must be accomplished without sacrificing the information requirements of the user, e.g., the signal can be reconstructed within a specified error. Different techniques for accomplishing data compression, in many different fields of interest, are discussed in [8], [9], [10].

This chapter presents the results of an initial study to determine the feasibility of securing electroencephalograph (EEG) data compression via orthogonal transforms, in particular, the Karhunen-Loève Transform (KLT). A proposed system for implementing such a technique is given. Comparisons between the reconstructed EEG signal (from the compressed data) and the original EEG, are also given. Besides, the capability of KLE as a feature extraction technique is introduced.

2-2- Karhunen-Loève Expansion (KLE)

2-2-1 KLE as a Feature Extraction Technique

A possible solution to the problem of excessive feature space dimensionality is to perform a linear transformation on the data, mapping it into a lower dimensional space. A linear transformation could be applied to either continuous, or discrete data form. Appendix A illustrates the application of KLE to EEG in its continuous form. Here, we concentrate on its application to discrete data since this is the major corner-stone in this thesis. More precisely, let y(t) be the (random) EEG signal, and assume, for convenience, that it is a zero mean random process. If the epoch of interest spans the time interval t_1 to t_2, the N time samples taken at evenly spaced intervals of T seconds can be expressed by Nx1 dimensional random pattern vector:

$$Y = \begin{bmatrix} y(t_1) \\ y(t_1 + T) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y\left(t_1 + (N-1)T\right) \end{bmatrix} , \quad (N - 1)T = t_2 - t_1 \qquad (2\text{-}1)$$

Given, any set of N, N x 1 dimensional orthonormal basis vectors:

$$\{\phi_i\},\ i = 1, 2, \ldots, N \qquad \phi_i^T \phi_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2-2}$$

Y can be expressed as a weighted sum of those vectors:

$$Y = \sum_{i=1}^{N} \phi_i x_i \tag{2-3}$$

where the random coefficients, x_i, are linear functions of the random vector Y:

$$x_i = \phi_i^T Y \qquad i = 1, 2, \ldots, N \tag{2-4}$$

What is desired is an approximation of Y using fewer than N basis vectors and coefficients:

$$Y = \sum_{i=1}^{M} \phi_i\, x_i + e \qquad , M<N \tag{2-5}$$

where e is an Nx1 dimensional random vector representing the error incurred by the truncation of the series. Defining the Mx1 random vector X:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_M \end{bmatrix} \tag{2-6}$$

we may express the linear transformation of the high dimensional pattern vector Y into the lower dimensional feature vector X as:

$$X = [\phi_1 \ \phi_2 \ \ldots . \phi_M]^T Y \qquad (2\text{-}7)$$

The components of X can then be interpreted as a compact set of features which represent the original pattern in that the original EEG may be reconstructed from these features with some small error. If the linear transformation is successful, the dimensionality, M, of the feature vector, X, can be made much smaller than the dimensionality, N, of the pattern vector, Y, while still maintaining acceptably small values of the error vector. Several transformations have been developed; among them are the orthonormal exponential, Fourier, Haar, Discrete cosine, Walsh-Hadamard, and Karhunen-Loève transforms [11], [12], [13].

The Karhunen-Loève transform (KLT) is unique among them in that it has been shown to possess two optimal properties:

1) - it minimizes the mean-square of the representation when only a finite number of basis functions are used in the expansion (eqn. (2-21)).

2) - it minimizes the entropy function defined in terms of the average squared coefficients used in the expansion [14].

The maximization of population entropy is defined by:

$$h = - E \{Ln[p(X)]\} \tag{2-8}$$

where p(X) is the probability density function for x. Besides the above, another property is that the coefficients of the expansion (the resulting features) are uncorrelated (eqn. (2-18)).

The Karhunen-Loève transform, then, constitutes an optimal specification of the basis vectors. In mathematical terms, let R be the N X N correlation matrix:

$$R = E \{Y \ Y^T\} \tag{2-9}$$

It has been shown that the expected squared error given by:

$$E \{e^T e\} \tag{2-10}$$

is minimized over the choice of all orthornomal basis vectors if they are chosen to be solutions of the eigenequation:

$$R \ \phi_i = \lambda_i \phi_i \tag{2-11}$$

and if the subscripts are chosen such that:

$$\lambda_1 > \lambda_2 > \dots\dots > \lambda m > \dots\dots > \lambda n \tag{2-12}$$

The KLT features are computed by finding the projection of the vector of time samples along the unit magnitude eigenvectors corresponding to the dominant eigenvalues of the correlation matrix. The optimality of this transformation lies in the fact that it attempts to take advantage of any statistical regularity in the ensemble being represented. Consequently, relatively few coefficients may be used to represent a much higher dimensional data vector with minimal error.

From a signal space point of view, the KLT performs a rotation of the coordinate frame such that the axes are aligned with those directions in which maximum statistical deviation is observed in the signal ensemble being represented. Because this transformation is rotational (and to the extent that the representation is a good approximation of the original signal) the structure of the original signal space is preserved. Distances between various signals in the original space should be almost identical in the new feature space, and the Euclidean Distance algorithms employed by a number of researchers [15-16] should be just as applicable in the new feature space.

A number of researchers have already explored the application of KLT to ECG data [12-17], speech data [13] and image coding [18]. In this thesis, our interest is of course in applying the technique to EEG data. As mentioned we shall investigate it as a data compression technique. Also we shall discuss its validity with EEG as a pattern recognition technique.

2-2-2 KLE as a Data Compression Technique

Because much of the justification of the Karhunen-Loève transform (KLT) is based upon the fact that the original EEG signal may be reconstructed from its associated random coefficients (features), an evaluation of the KLT from the standpoint of the mean squared reconstruction error is presented in this section. The examination of the KLT from the point of view of EEG classification is left for chapter 4. As described in the previous section, the KLT is a linear feature extractor. That is, the MX1 feature vector:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \\ x_M \end{bmatrix} \tag{2-13}$$

is derived from the NX1 pattern vector:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{bmatrix} \tag{2-14}$$

through the transformation:

$$X = [\phi_1 \phi_2 \cdot\cdot\cdot\cdot \phi_M]^T Y \tag{2-15}$$

where the M basis vectors of the transformation matrix are chosen to be those eigenvectors corresponding to the M largest eigenvalues of the correlation matrix, R:

$$R\ \phi_i = \lambda_i \phi_i \tag{2-16}$$

$$\lambda_1 > \lambda_2 > \ldots > \lambda_m > \lambda_n$$

Because we assume that the correlation matrix is nonsingular, and because the correlation matrix is symmetric, its N eigenvectors form a complete orthogonal (henceforth presumed to be orthonormal) basis spanning the pattern space. Hence, the pattern vector may be represented exactly by N features (coefficients):

$$Y = \sum_{i=1}^{N} x_i \phi_i \tag{2-17}$$

where the features are given by eqn. (2-4). The features are also uncorrelated, i.e.:

$$E\ \{x_i\ x_j\} = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases} \tag{2-18}$$

An approximation of Y using M (<N) features (random coefficients) is:

$$Y = \sum_{i=1}^{M} x_i \phi_i + e \tag{2-19}$$

where e, the random error vector, is given by:

$$e = Y - \sum_{i=1}^{M} x_i \phi_i \tag{2-20}$$

The mean square error of the approximation is defined by:

$$E\ \{e^T e\} = E\{(Y - \sum_{i=1}^{M} x_i \phi_i)^T (Y - \sum_{i=1}^{M} x_i \phi_i)\} \qquad (2\text{-}21)$$

The eigenvalues are of particular interest because they indicate the degree to which the approximation of (2-19) is degraded when coefficients are deleted from the representation. Equation (2-21) can be re-expressed to show this. Carrying out the inner product in (2-21) and using (2-17) to substitute for Y in the middle term, (2-21) becomes:

$$E\{e^T e\} = E\{Y^T Y - 2 \sum_{i=1}^{M} x_i \phi_i^T\ Y + \sum_{i=1}^{M} \sum_{j=1}^{M} x_i \phi_i^T\ \phi_j x_j\}$$

$$= E\{Y^T Y - 2 \sum_{i=1}^{M} \sum_{j=1}^{M} x_i \phi_i^T \phi_j x_j + \sum_{i=1}^{M} \sum_{j=1}^{M} x_i \phi_i^T \phi_j x_j\} \qquad (2\text{-}22)$$

The two double sums may be reduced by virtue of the orthonormality of the eigenvectors:

$$E\{e^T e\} = E\{Y^T Y - 2 \sum_{i=1}^{M} x_i x_i + \sum_{i=1}^{M} x_i x_i\}$$

$$= E\{Y^T Y - \sum_{i=1}^{M} x_i x_i\} \qquad (2\text{-}23)$$

using (2-4) to substitute for x_i, and bringing the expectation into the sum:

$$E\{e^T e\} = E\{Y^T Y - \sum_{i=1}^{M} \phi_i^T \, Y \, Y^T \phi_i\}$$

$$= E\{Y^T Y\} - \sum_{i=1}^{M} \phi_i^T \, R \, \phi_i$$

$$= \text{trace } (R) - \sum_{i=1}^{M} \phi_i^T \, R \, \phi_i \qquad (2\text{-}24)$$

where trace (R) is defined to be the sum of the diagonal elements of R. Lastly, using (2-16) and applying orthonormality we get:

$$E\{e^T e\} = \text{trace } (R) - \sum_{i=1}^{M} \phi_i^T \, \phi_i \lambda_i$$

$$= \text{trace } (R) - \sum_{i=1}^{M} \lambda_i \qquad (2\text{-}25)$$

From equation (2-25) it can be seen that:

1) - Trace (R) is the expected squared error incurred when no features (random coefficients) are used in the approximation of Y (that is when the best a priori guess E{Y}, is used as the approximation). This quantity has been referred to as "total energy" in the signal in question.

2) - The expected squared error decreases by an amount equal to the i^{th} eigenvalue when the i^{th} coefficient is included in the representation.

It can be shown that the sum of all N eigenvalues is equal to trace (R), and hence N features can completely account for the total energy.

It should be noted that the eigenvalues alone are not very informative in that their magnitudes depend upon the total energy of the signal itself. However, the ratio:

$$\frac{\lambda_i}{\text{trace (R)}} \tag{2-26}$$

is useful in deciding whether or not the i^{th} feature should be retained in the representation, for it measures the fraction of the total energy accounted for by the i^{th} coefficient. Similarly, the ratio:

$$\sum_{i=1}^{M} \lambda_i \;/\; \text{trace (R)} \tag{2-27}$$

is a measure of the fraction of total energy accounted for by the M dimensional representation. The error due to this representation can be re-expressed as:

$$\begin{aligned} E\{e^T e\} &= \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{M} \lambda_i \\ &= \sum_{i=N-M}^{N} \lambda_i \end{aligned} \tag{2-28}$$

The above discussion presumes that the pattern vector, Y, is a zero mean random vector. To ensure that this is true, the mean must be subtracted from the vector derived by sampling the EEG. If Z denotes the vector of time samples taken from the EEG, Y is derived by:

$$Y = Z - E\{Z\} \tag{2-29}$$

The reconstructed signal can be found by adding back the mean:

$$Z = E\{Z\} + \sum_{i=1}^{M} x_i \phi_i + e \qquad (2\text{-}30)$$

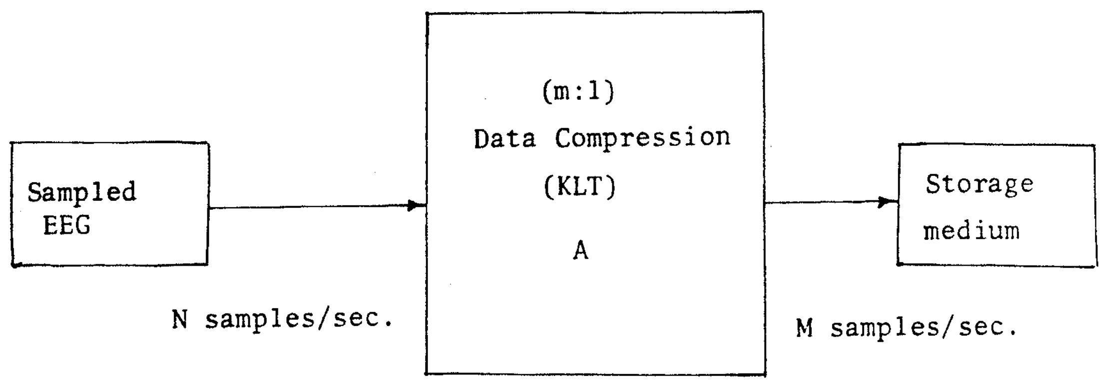
where e corresponds to the error vector defined by (2-20). From (2-29) it can be seen that the correlation matrix of Y is identical to the covariance matrix of Z.

Instead of storing the N samples of the EEG signal, we can store only the M random coefficients, to achieve a data compression ratio of N/M:1. Reconstruction of the EEG signal with allowable amount of error is then possible. The next section shows the results of the KLT application as a data compression technique to the EEG signal.

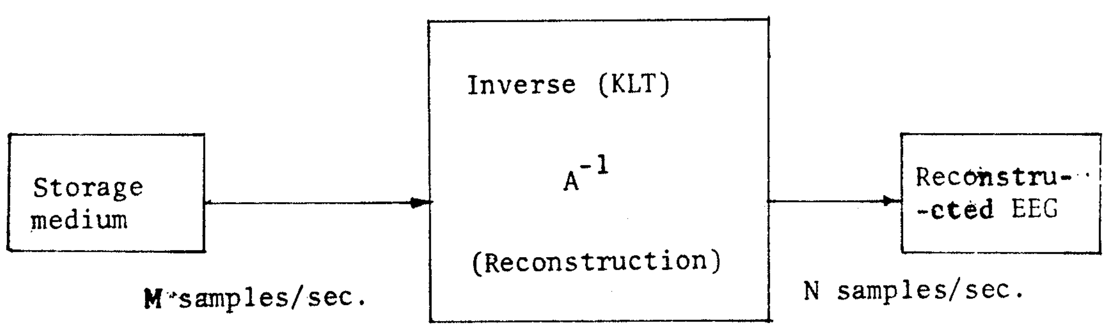
2-3 Analysis of an EEG Data Compression System

2-3-1 Proposed System

If each EEG signal (epoch of one second length) requires N words (bytes) to store its sampled values, then an m:1 data compression implies that on the average, N/m words per EEG are required to store its KLT random coefficients. Clearly the storage requirements are reduced by a factor m. Such compression must be realized essentially without loss of the features necessary for a physician to interpret the EEG. The effect of data compression with respect to EEG has significant implications. It could lead to the economical storage-retrieval of EEG's in data banks, and hence enable an institution to have readily available EEG records from large numbers of patients. Figure (2-1) shows a block diagram for a proposed system to achieve an m:1 data compression using KLT.



(a)



(b)

Fig.(2-1) KLT data compression of EEG signals by a factor of m:1 (m=N/M)
(a) storage (b) retrieval

2-3-2 Implementation and Experimental Results

The Karhunen-Loève transform, requires the solution of an eigenvalue, eigenvector problem for a large matrix (NxN). The solution, which must be obtained only once for all time, can be performed off line (with the hope that the same eigenvectors could be used for all patients).

The following steps are necessary to go through in order to find those eigenvectors:

1) - compute an (NxN) autocorrelation matrix R.

2) - obtain the eigenvalues and the corresponding normalized eigenvectors of R.

3) - re-arrange the eigenvalues in a descending order.

4) - for a pre-determined allowable mean-square error, determine the number M of the eigenvectors to be used in the expansion.

5) - form a set of these M eigenvectors to be the basis of the expansion and choose those eigenvectors which correspond to the largest eigenvalues of R. These eigenvectors are to be used in building the major block in figure (2-1).

A computer program was written, using Fortran Language, to perform these calculations, and is described in Appendix B. Jacobi's method for a symmetric matrix was used to solve the eigenvalue, eigenvector problem of an (NxN)* autocorrelation matrix R [19].

---

* N = 128, since the sampling rate being used was 128 sample/sec.

The eigenvalue, eigenvector problem was solved for seven different epochs chosen at random, each belonging to a different class. Table (1) illustrates the results of using five eigenvector sets as applied to different EEG epochs (each set consists of 50 eigenvector). It has been shown that one set (the 1st set) of these eigenvectors was more efficient than the others in representing the available EEG data with respect to the mean square error of the approximation. As an example, the first 16 eigenvectors of this set (obtained by using a normal epoch) are shown in Figures (2-2) through (2-9).

The mean-square error value of the approximation, is dependent on the number of the eigenvectors to be used in the expansion. Also, it is a measure of the efficiency of the reconstruction operation. Figure (2-10) illustrates the relation between the number of the eigenvectors used in the expansion, and the resultant mean-square error. A set of epochs was chosen at random to draw this graph. It is clear, from the figure, that as we increase the number of eigenvectors the mean-square error value decreases.

Experimental results showed that 2.56:1 data compression ratio is feasible. That is, if N(128) words are required to store an EEG epoch in its original form, then only M(50) words will be required to store it in terms of its transform components (the coefficients of the expansion).

Figures (2-11) through (2-22) show different examples of original and approximated epochs of EEG signal using only 50 eigenvectors. Each figure shows also the resultant point by point error of the approximation and the total mean-square error. In Appendix C, tables of the mean-square error of some epochs approximation, are provided. The mean-square error has been found to range from 0.06% - 50.4%, with 83.7% of the epochs having a mean-square error of less than 15%.

Table 1: Results of using five different eignevector sets with the EEG signal

epoch No.	Set #1	Set #2	Set #3	Set #4	Set #5
1	5.94	7.59	13.92	11.35	31
2	2.71	2.52	6.38	3.42	15.68
3	5.48	8.16	14.35	7.9	47.39
4	2.42	2.93	18.77	9.25	18.52
5	16.9	19.27	31.8	13.4	38.5
6	10.53	6.37	9.62	7.6	16.97
7	5.66	5.9	5.76	6.1	7.9
8	15.82	22.6	23.83	24.3	20.3
9	14.41	15.3	18.4	16.9	14.1
10	29.1	31.86	31.86	33.03	30.52
11	24.1	19.4	30.7	21.7	38.2
12	4.4	3.1	4.3	4.97	7.13
13	4.8	4.6	16.3	14.6	27.5
14	0.94	1.1	1.4	1.1	1.5
15	2.4	2.7	2.97	3.2	2.3
16	1.1	1.3	1.2	1.2	2.3
17	5.78	8.1	8.5	6.1	11.3
18	9.32	14.7	21.4	15.9	32.6
19	5.12	4.22	7.9	8.14	12.64
20	12.6	13.6	13.9	12.3	18.9
21	5.4	7	10.7	8.3	11.9
22	7.2	6.9	7.8	4.1	20.7

NOTE: The numbers listed in each row are the values of the mean-square error of the approximation, using the shown eigenvector sets.

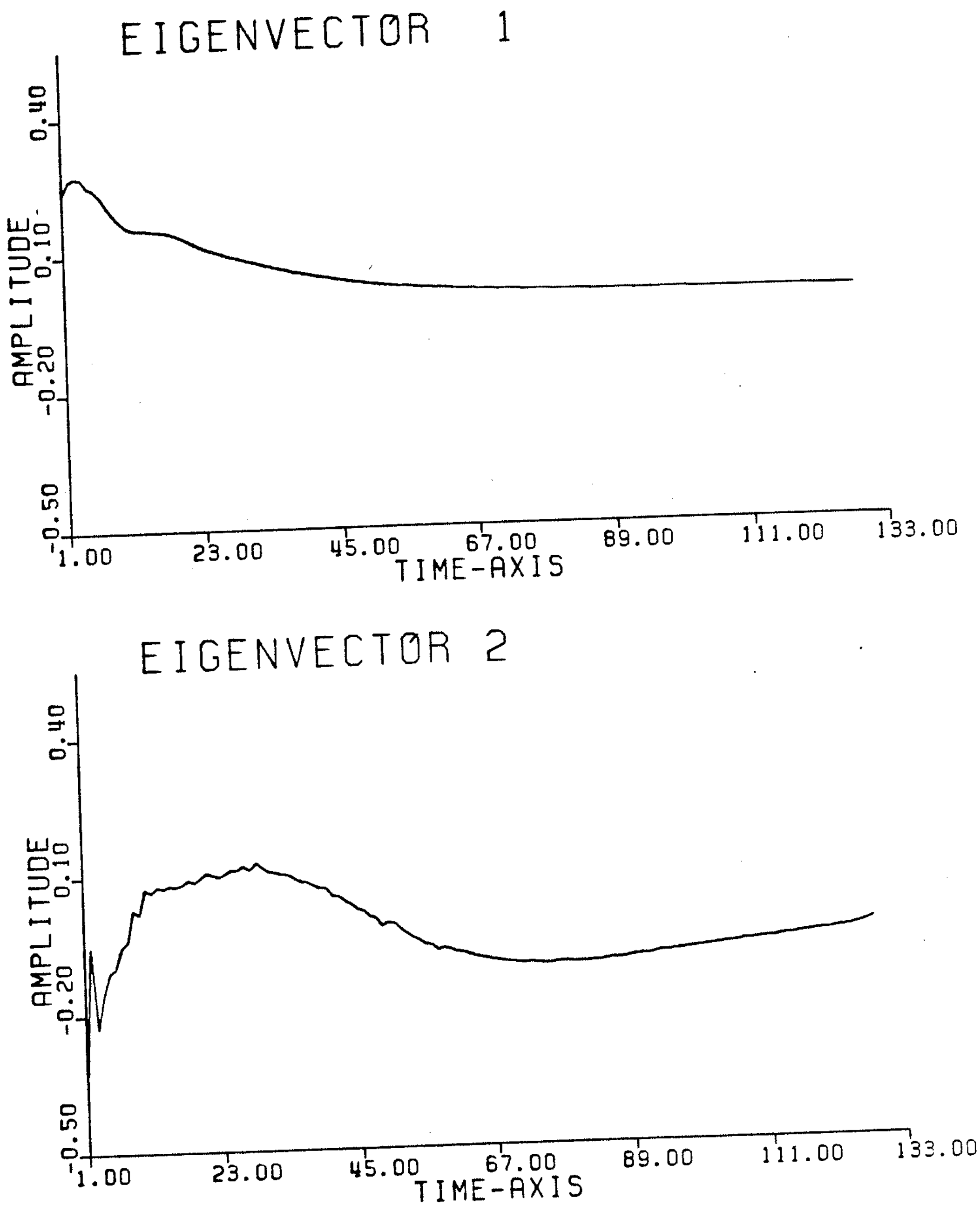


Fig.(2-2) The 1st and the 2nd eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.

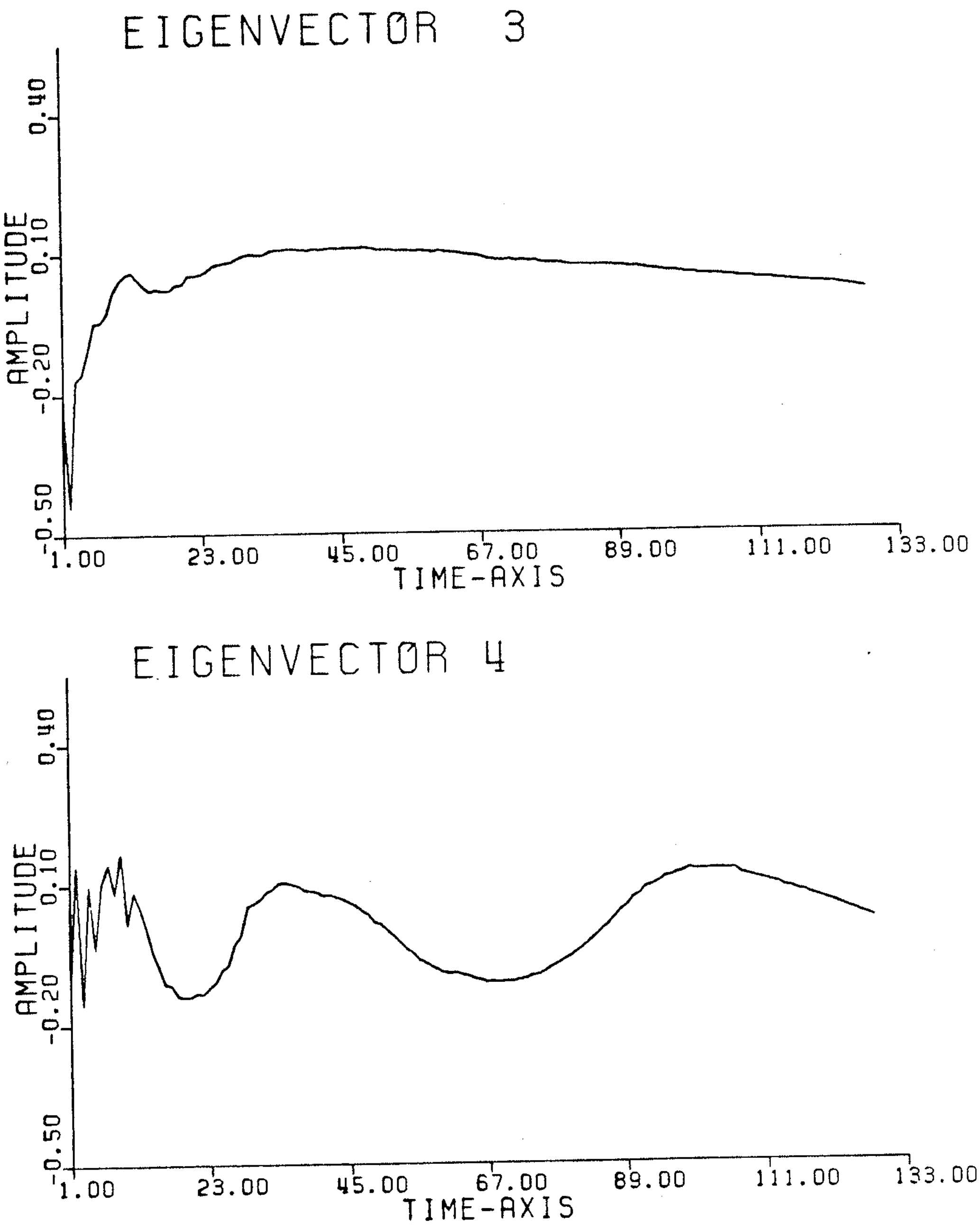
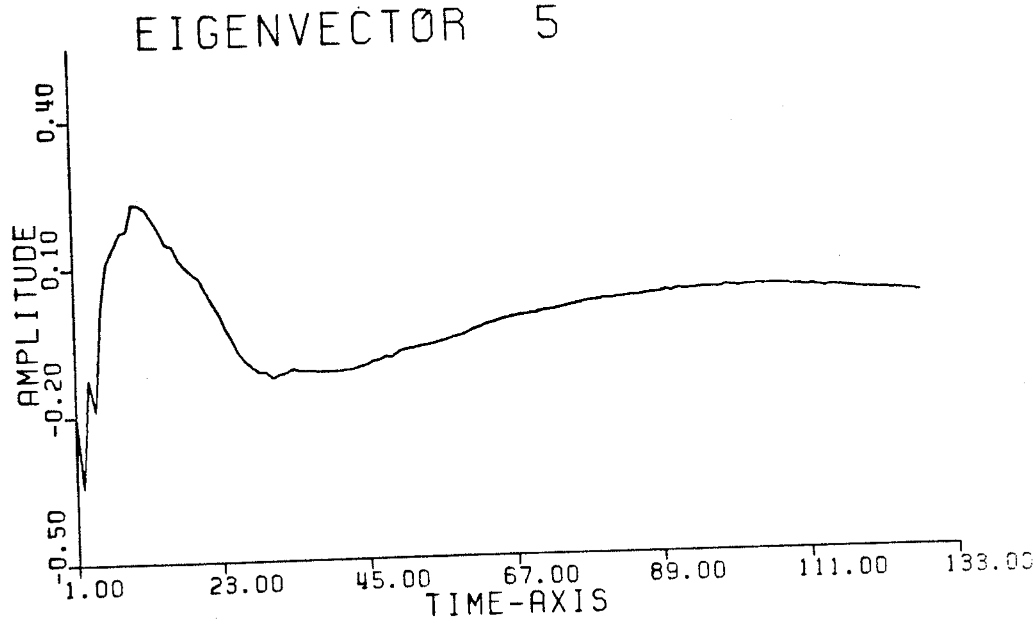


Fig.(2-3) **The third and the fourth eigenvectors used in the expansion.** Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.



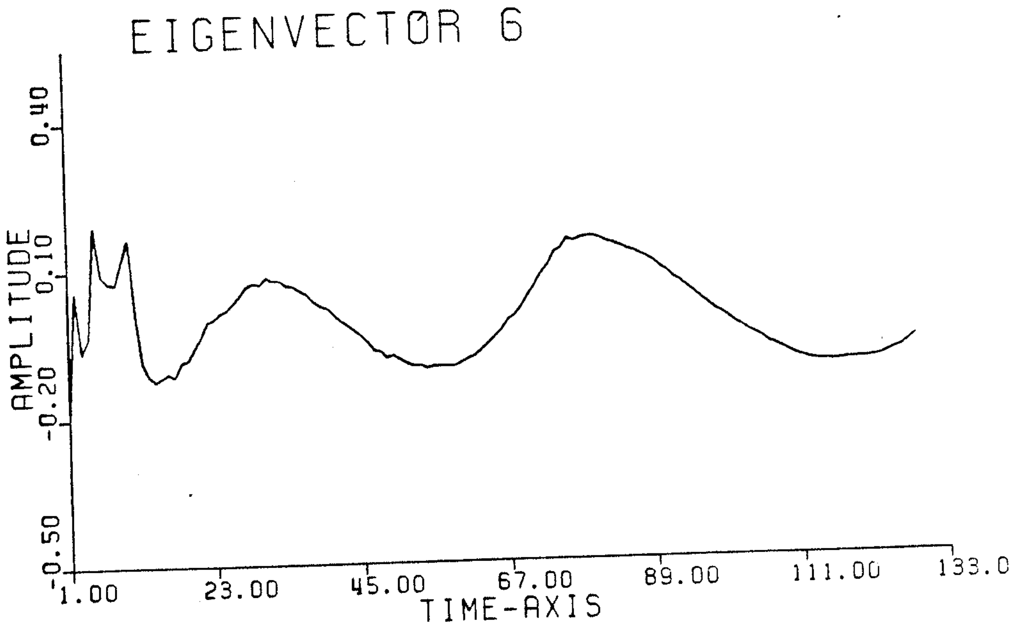
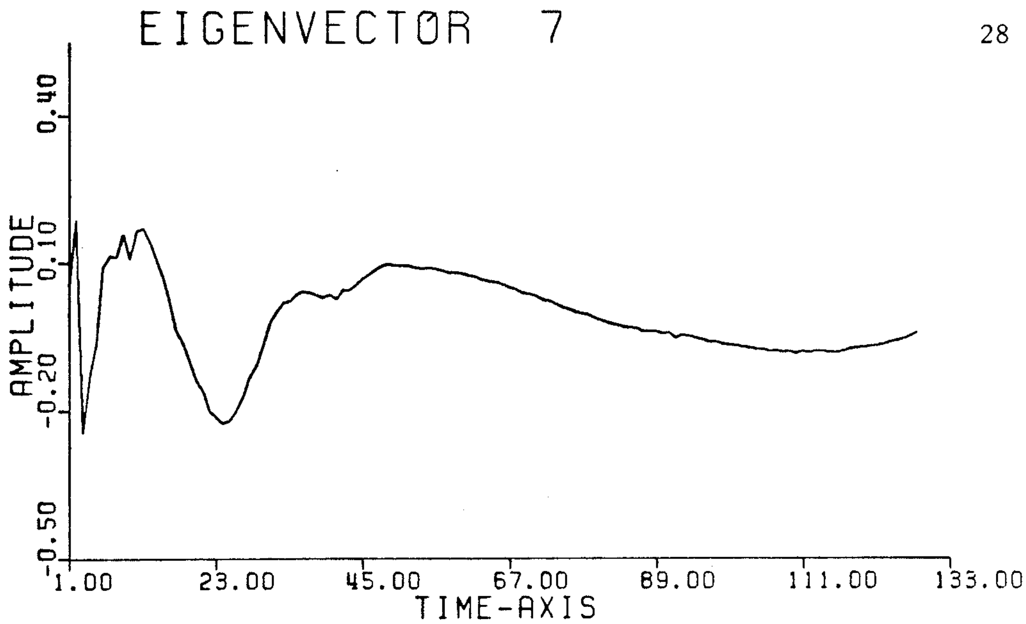


Fig.(2-4) The 5th and the 6th eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample, amplitude is normalized.



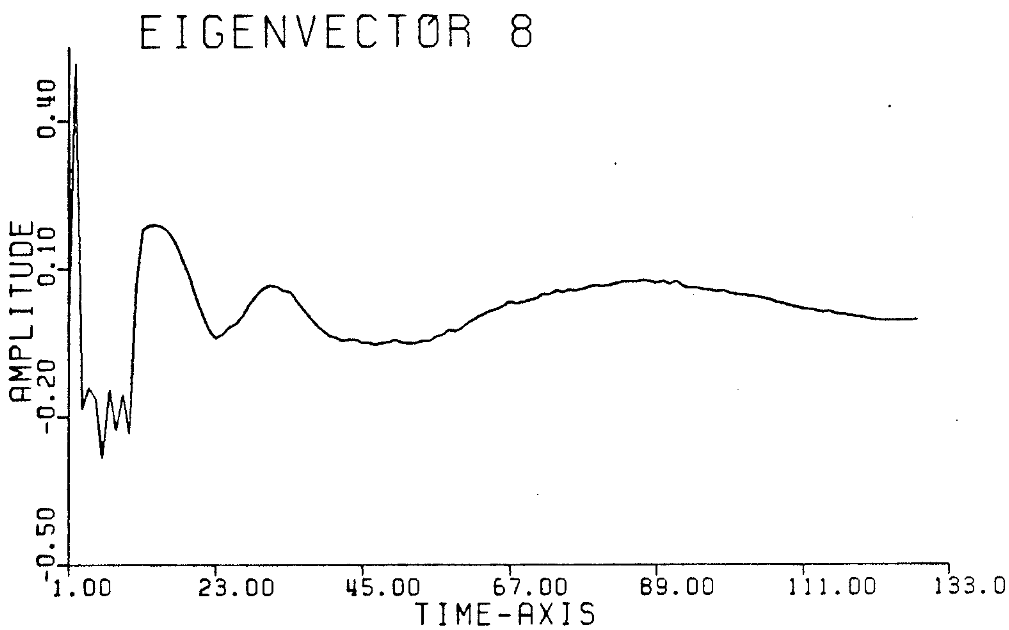
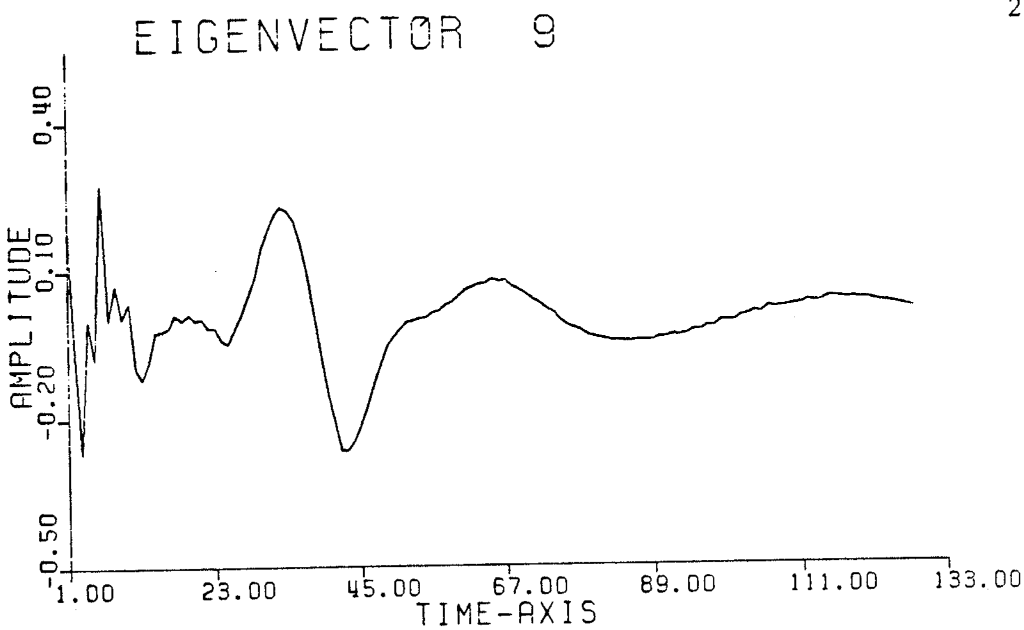


Fig.(2-5) The 7th and the 8th eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.



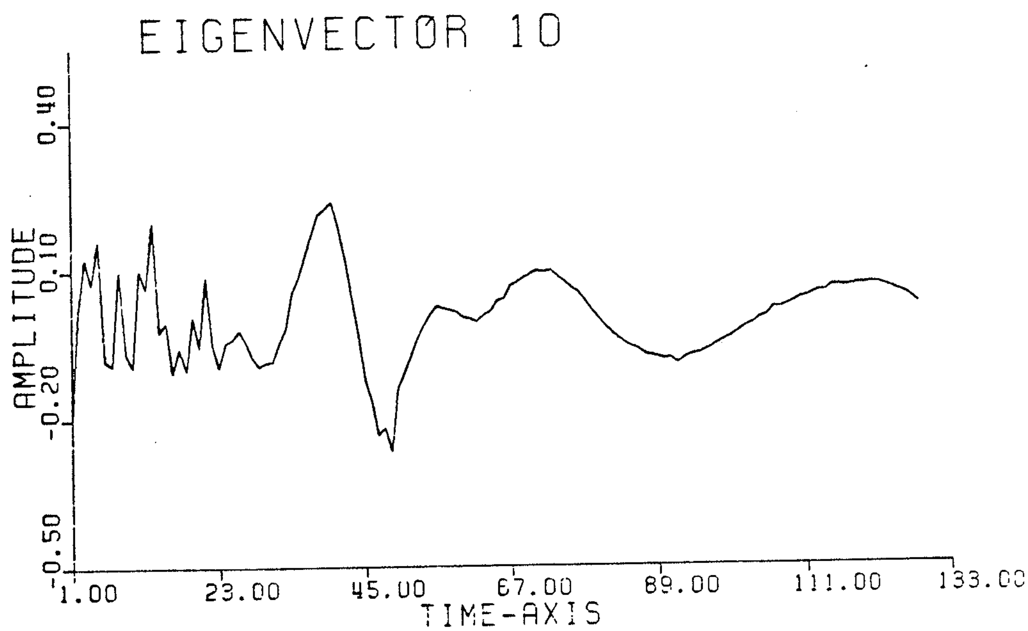


Fig. (2-6) The 9th and the 10th eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.

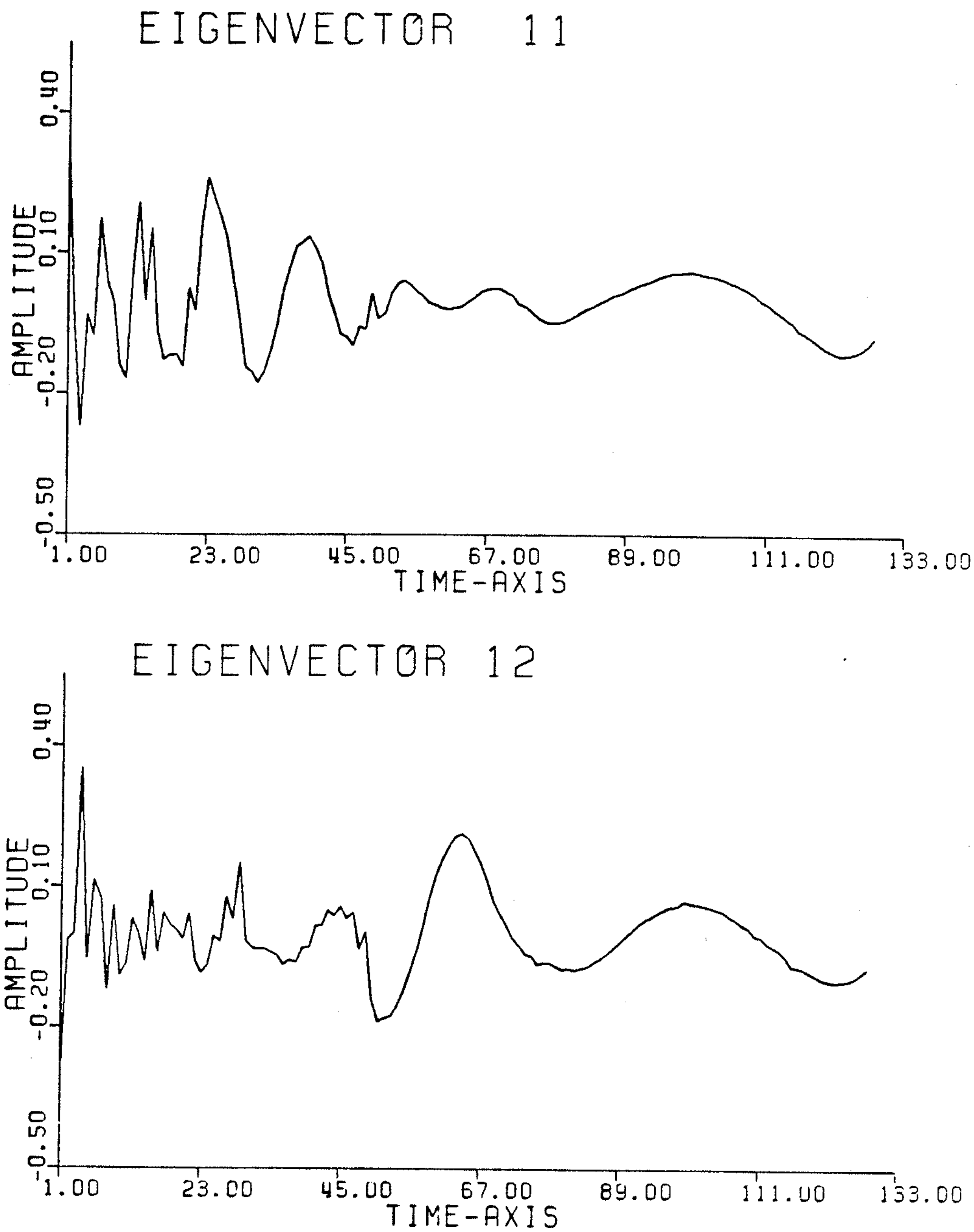
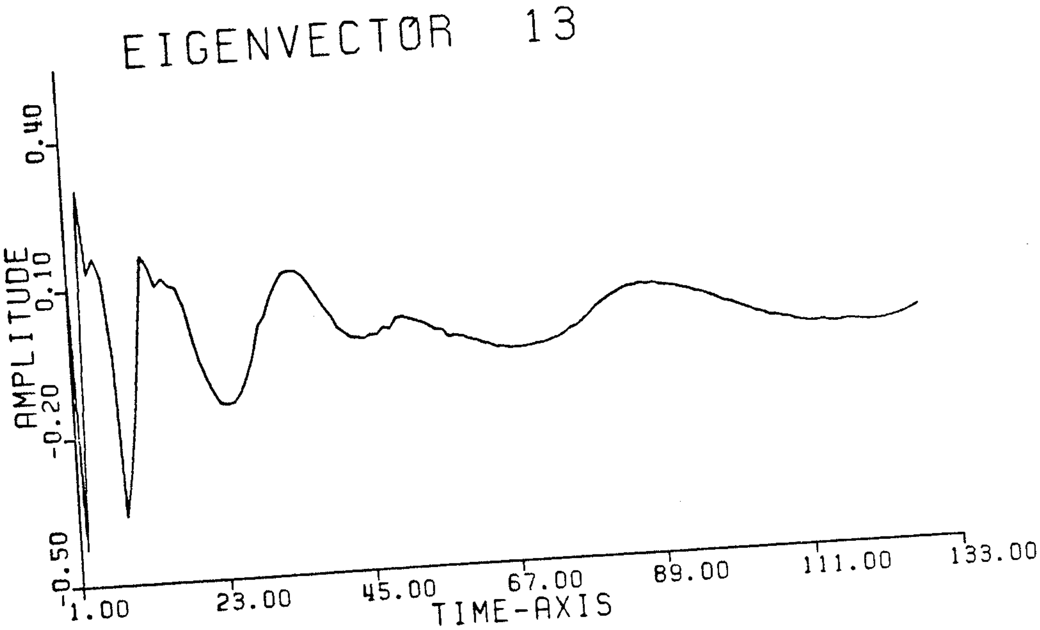


Fig. (2-7) The 11th and the 12th eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.



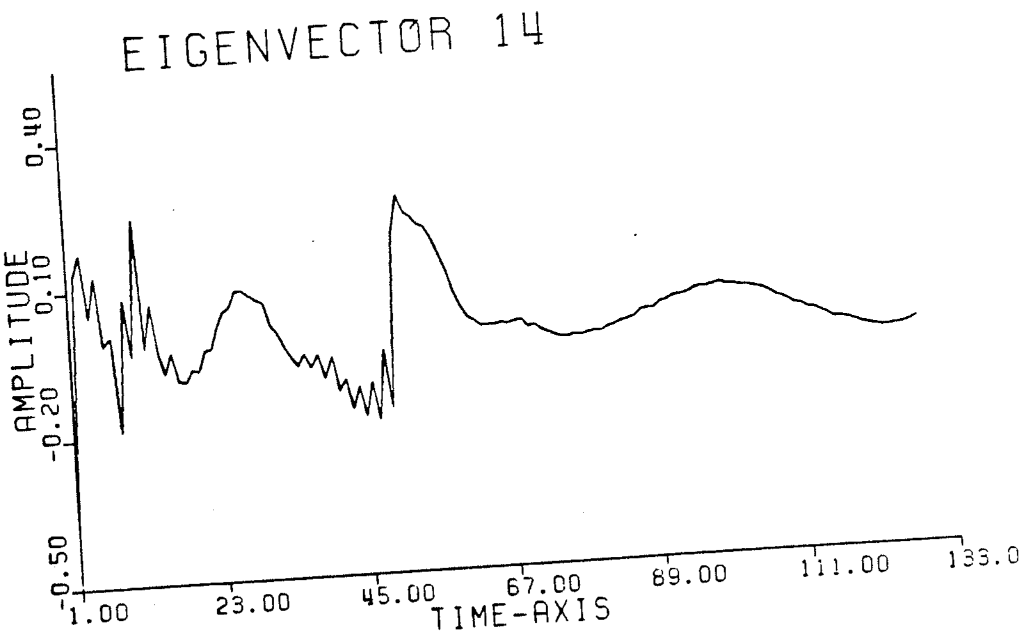
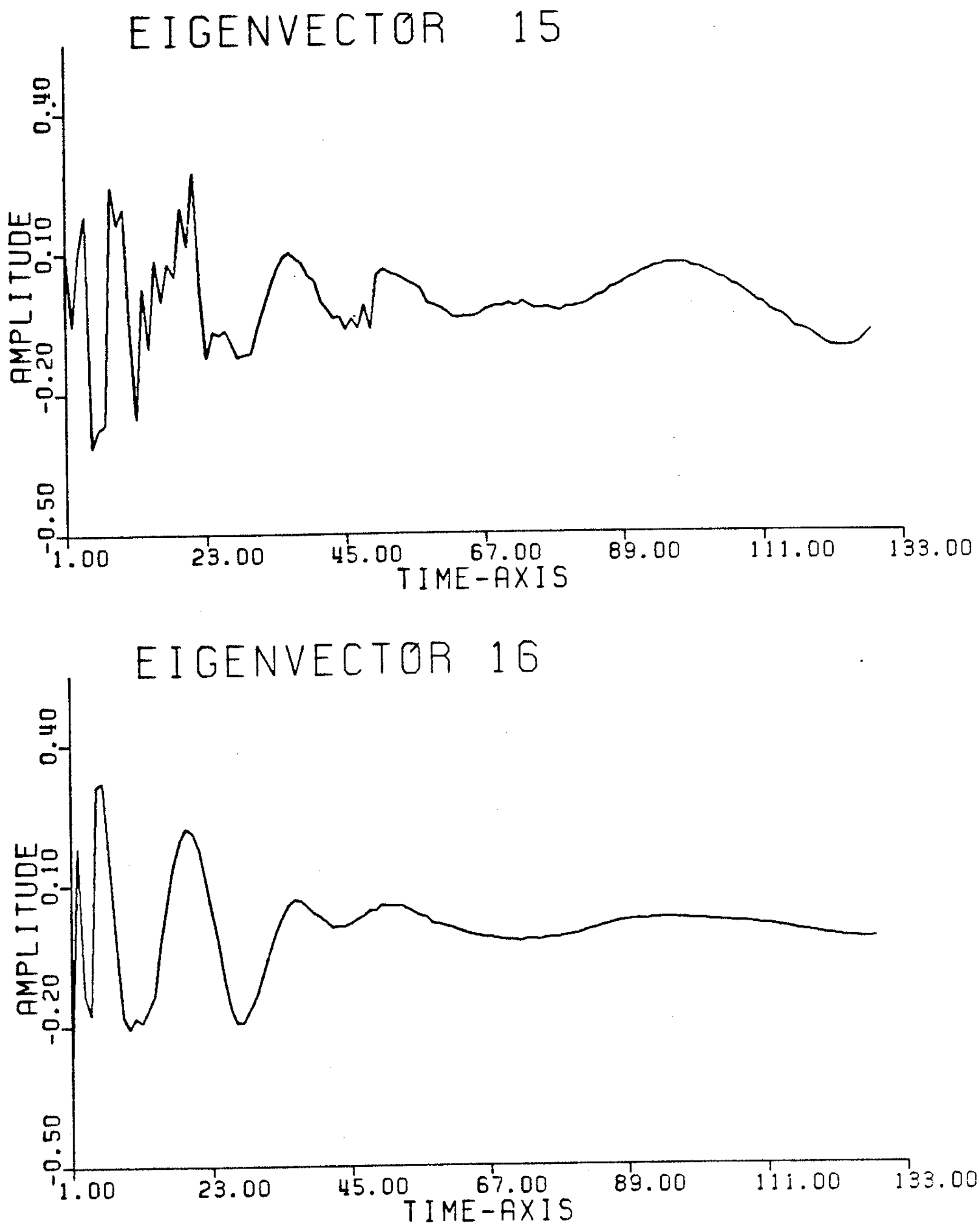


Fig.(2-8) The 13th and the 14th eigenvectors used in the expansion. Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.



Fig(2-9) **The 15th and the 16th eigenvectors used in the expansion.** Time axis is in terms of number of samples. 1 sec= 128 samples. t=0 corresponds to the 1st sample. Amplitude is normalized.

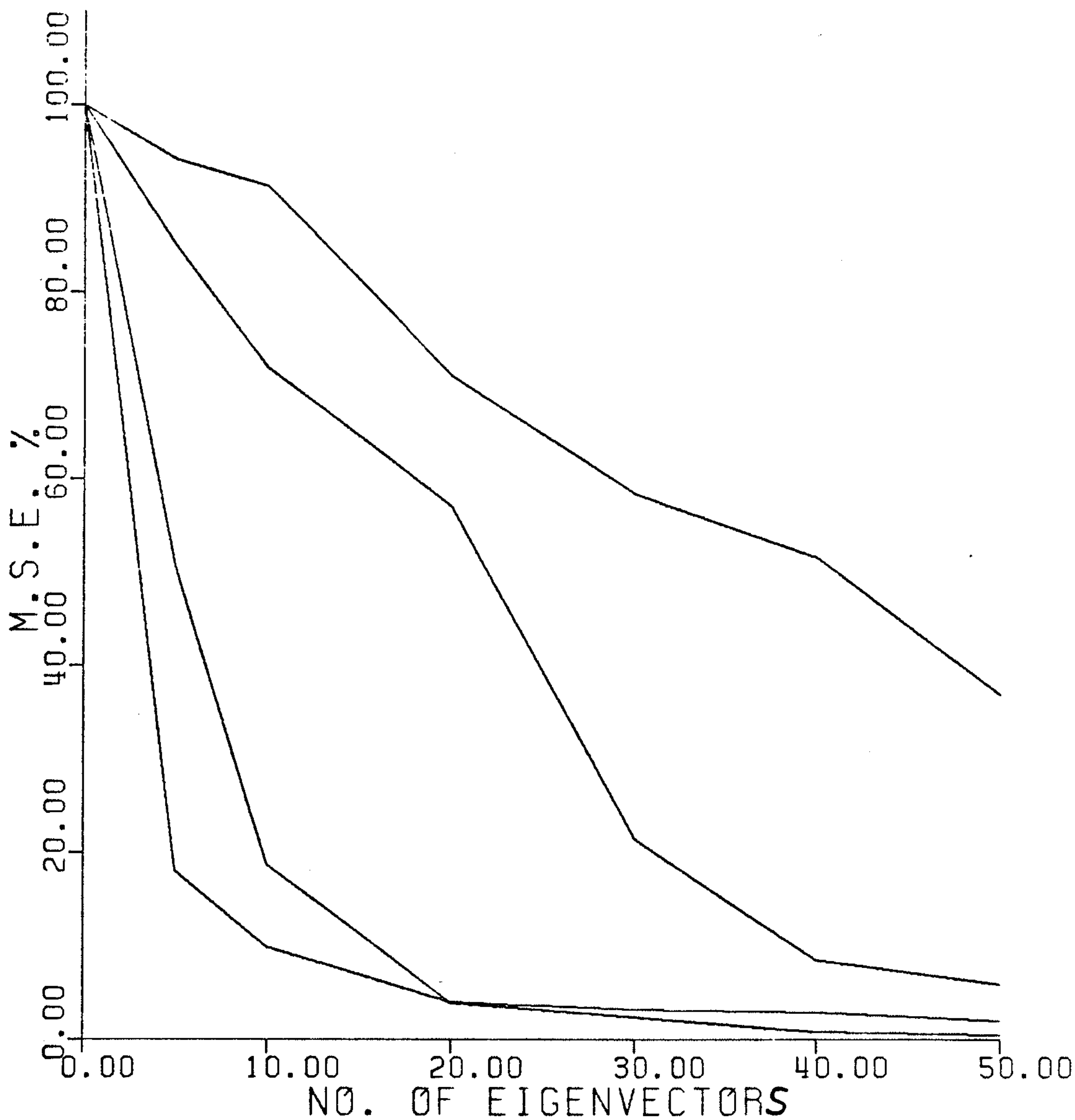


Fig.(2-10) **Relationship** between the mean-square error value and the number of eigenvectors used in the KLE .

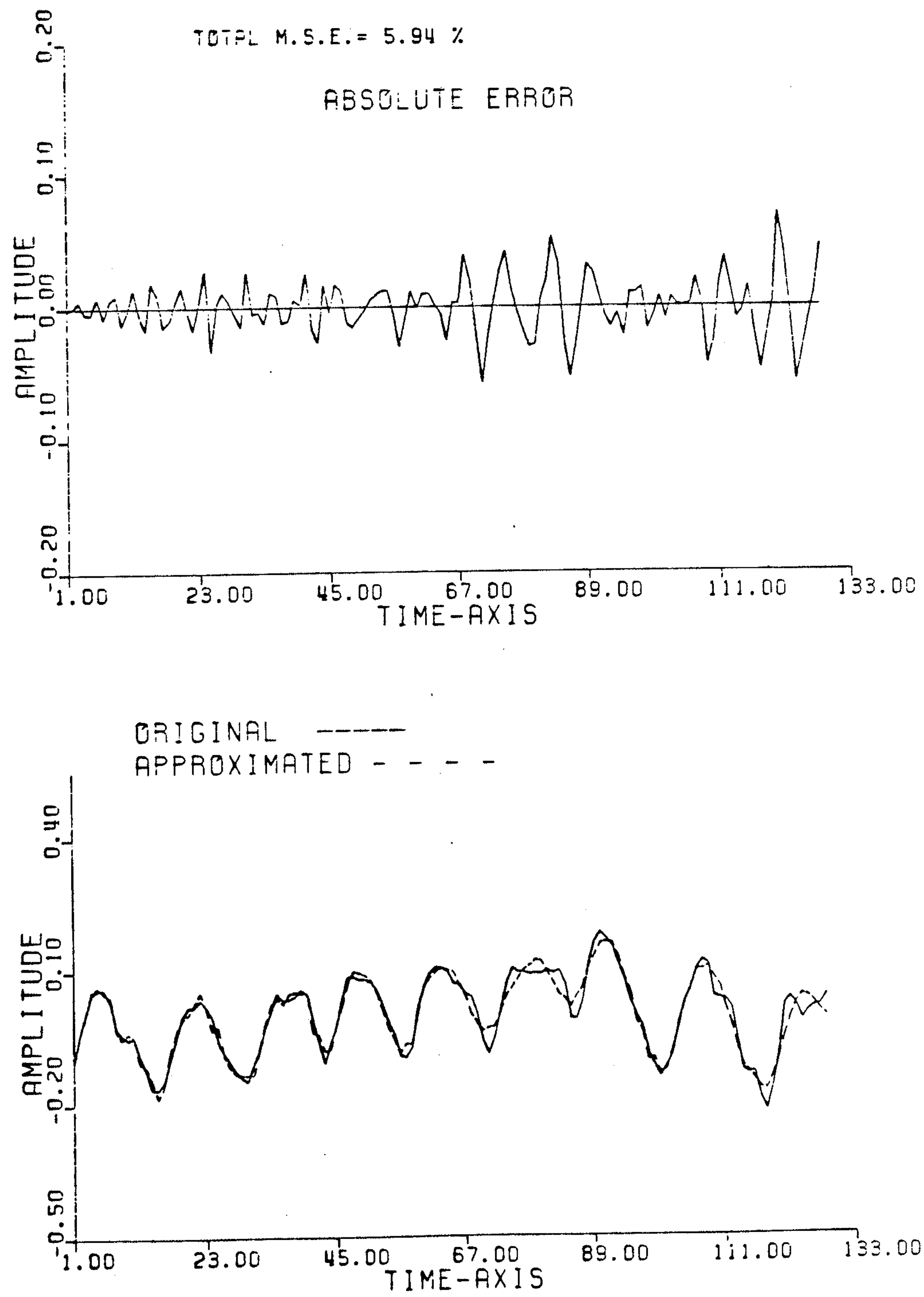


Fig.(2-11) Original and approximated normal EEG epoch together with the point-by-point error of the approximation.

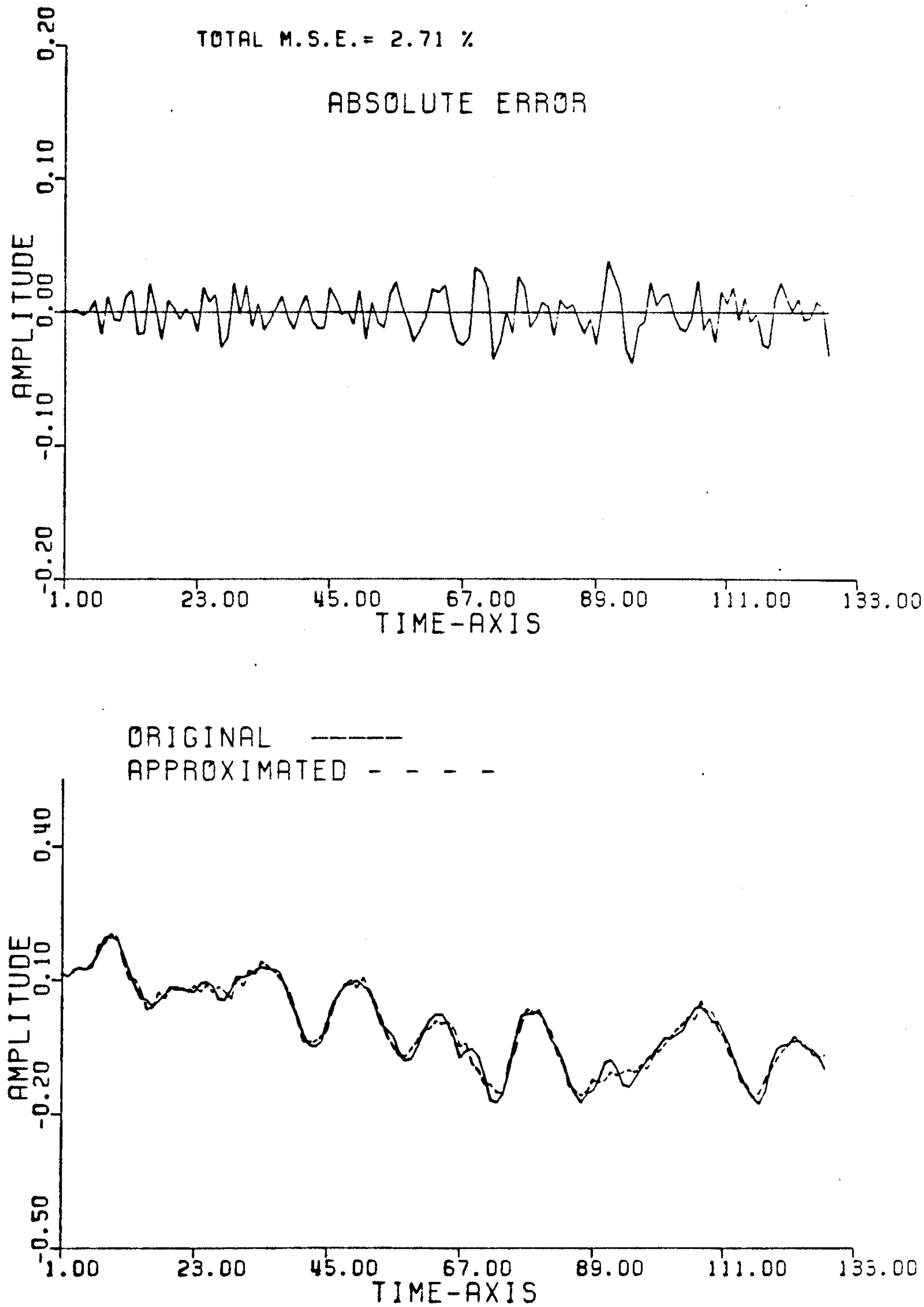
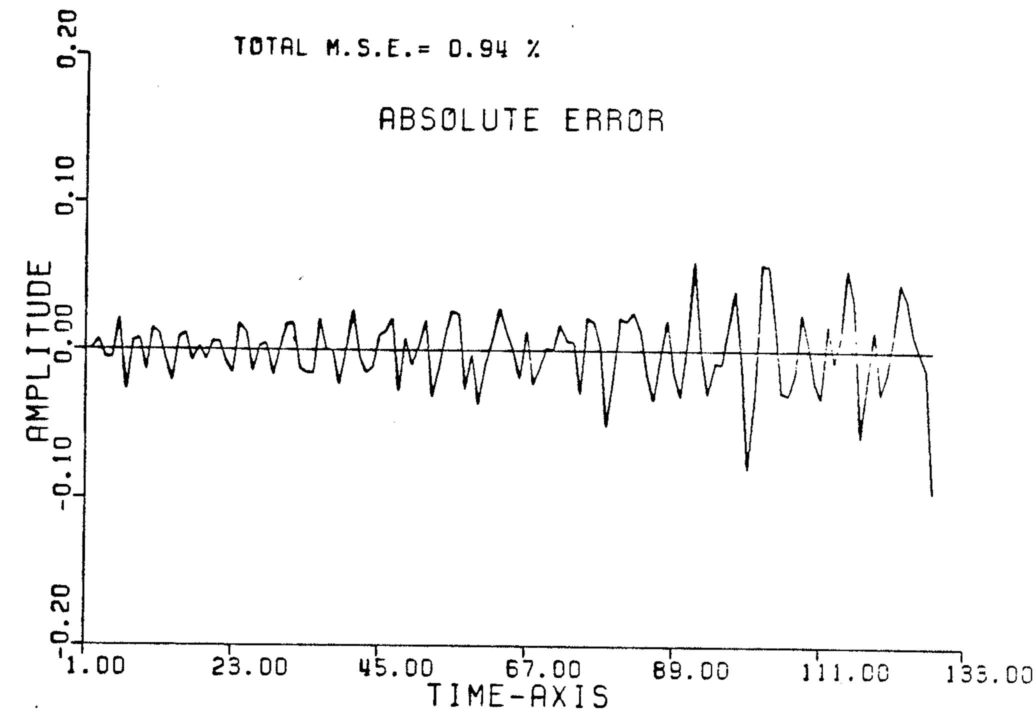


Fig. (2-12) Original and approximated normal EEG epoch together with the point-by-point error of the approximation.



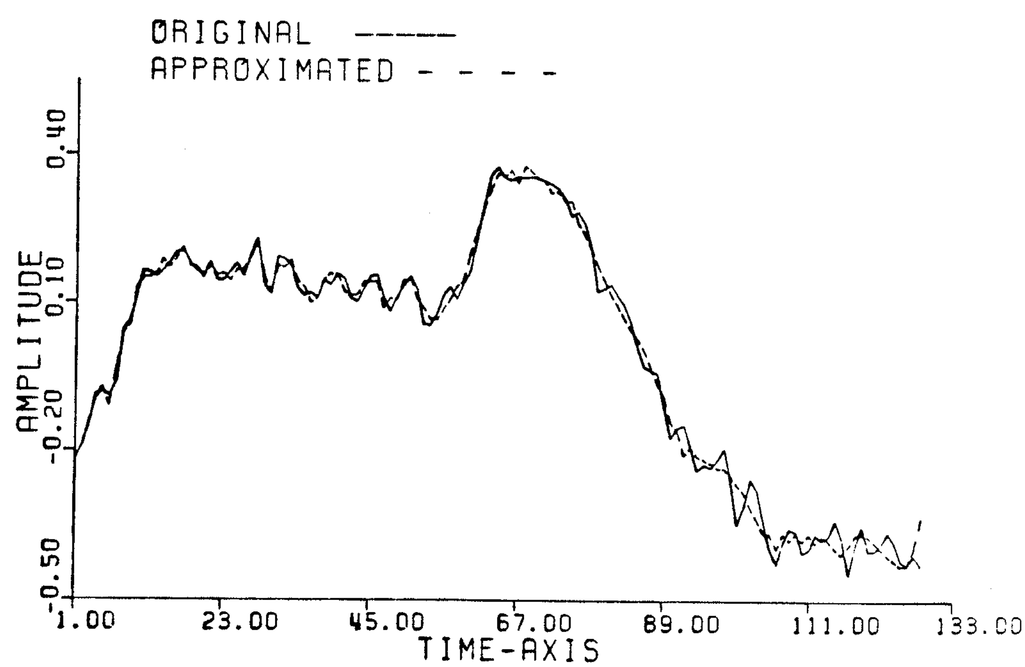


Fig. (2-13) Original and approximated artefactual EEG epoch together with the point-by- point error of the approximation.

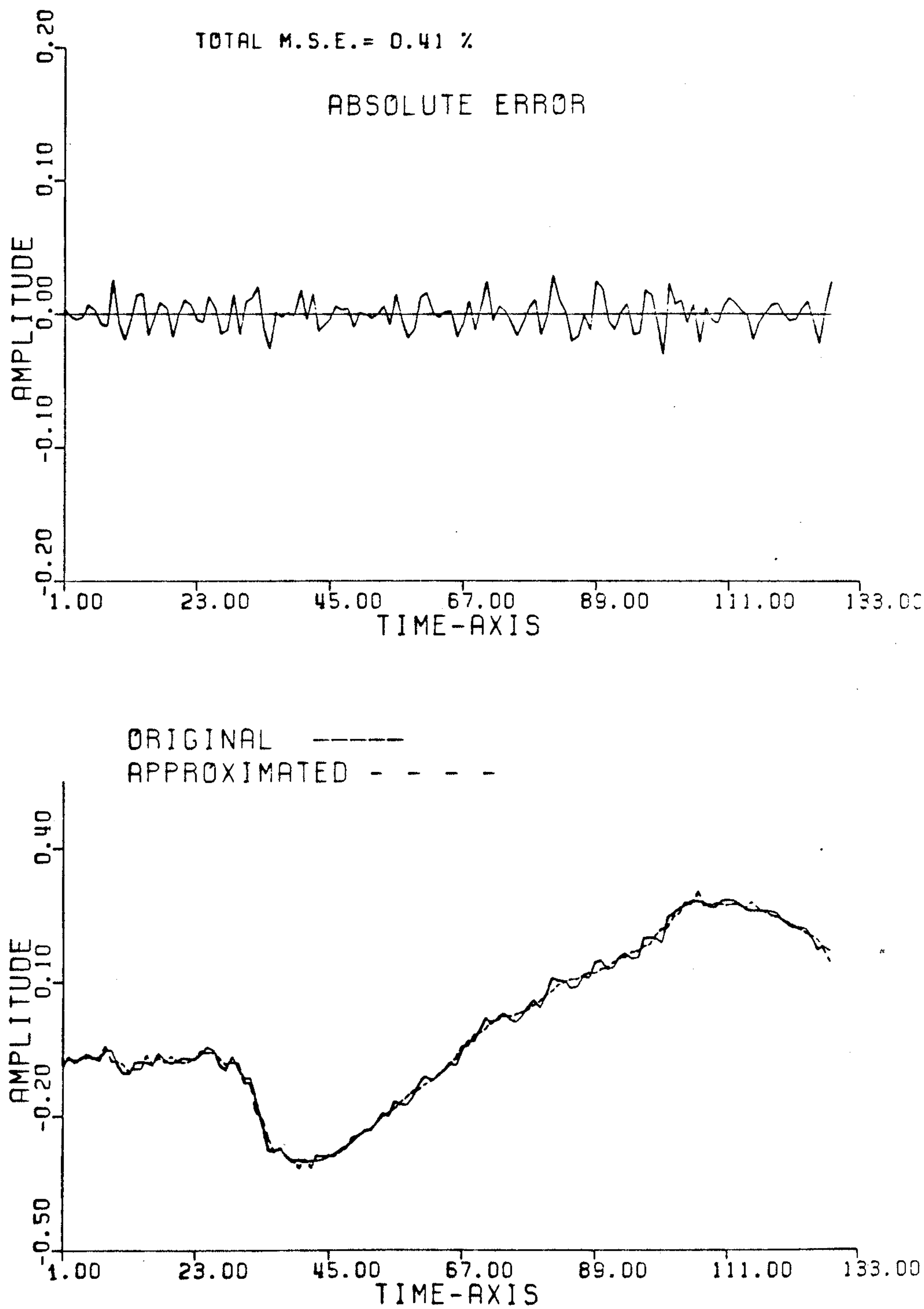
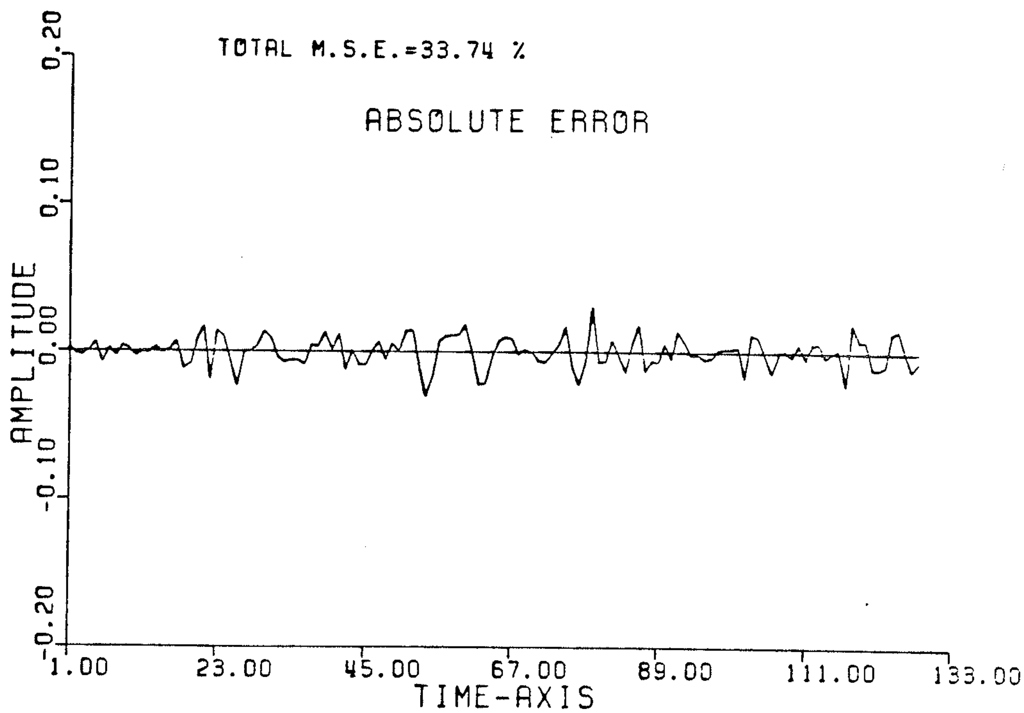


Fig.(2-14) Original and approximated artefactual EEG epoch together with the point-by-point error of the approximation.



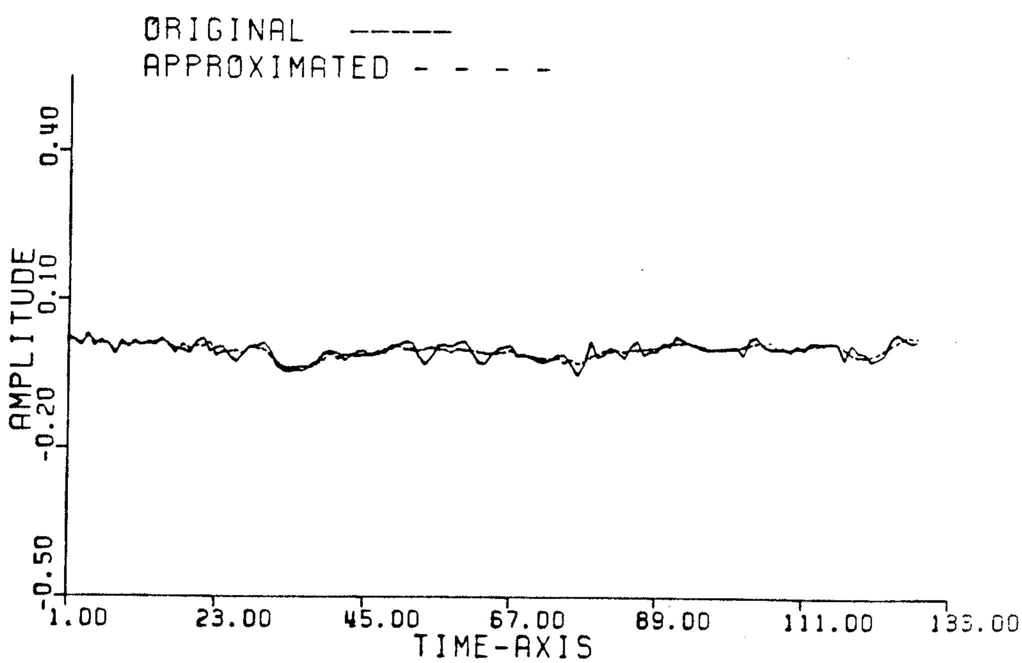
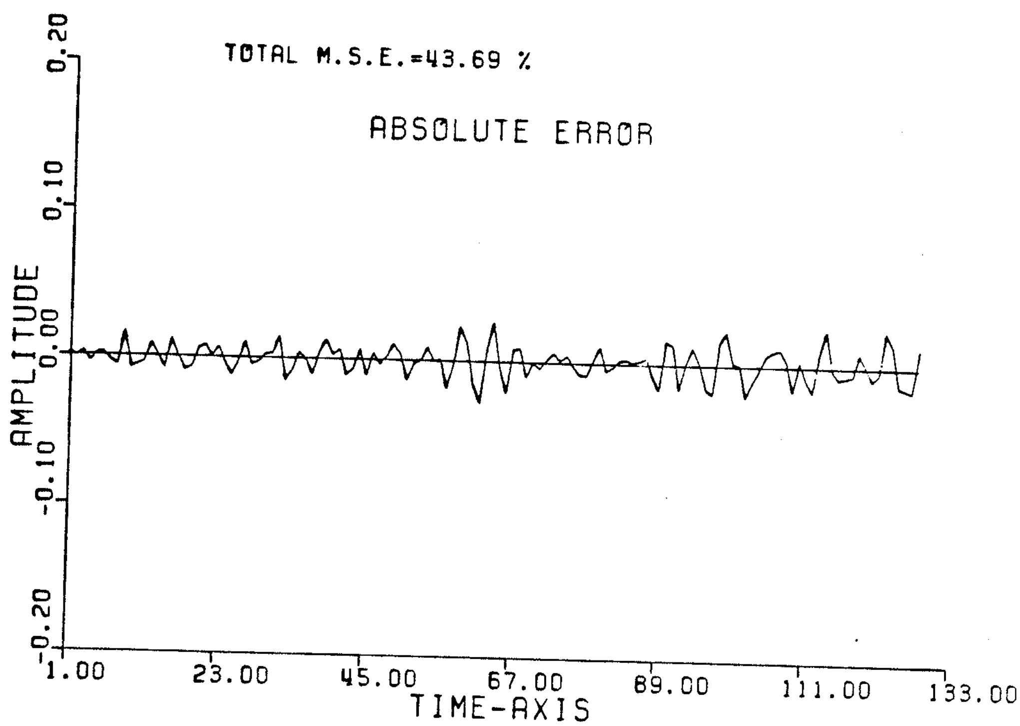


Fig.(2-15) Original and approximated fast EEG epoch together with the point-by-point error of the approximation.



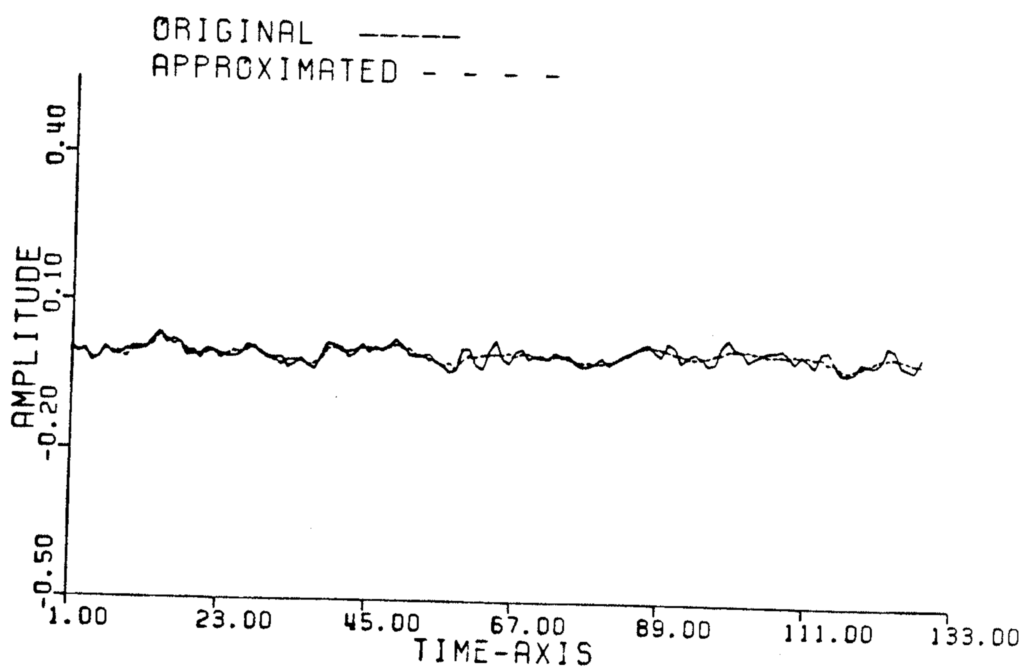


Fig. (2-16) Original and approximated fast EEG epoch together with the point-by-point error of the approximation.

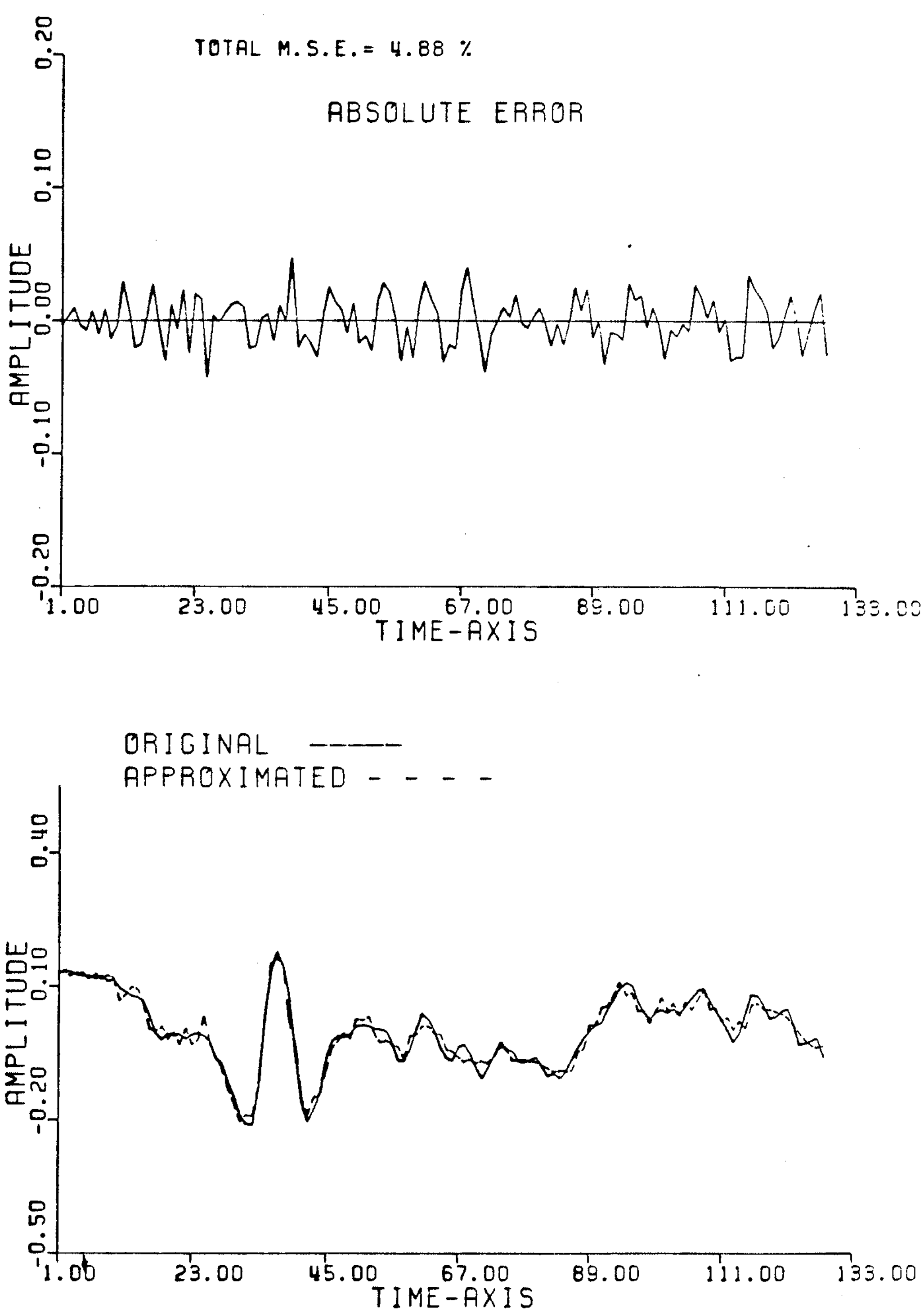


Fig.(2-17) Original and approximated isolated spike EEG epoch together with the point-by-point error of the approximation.

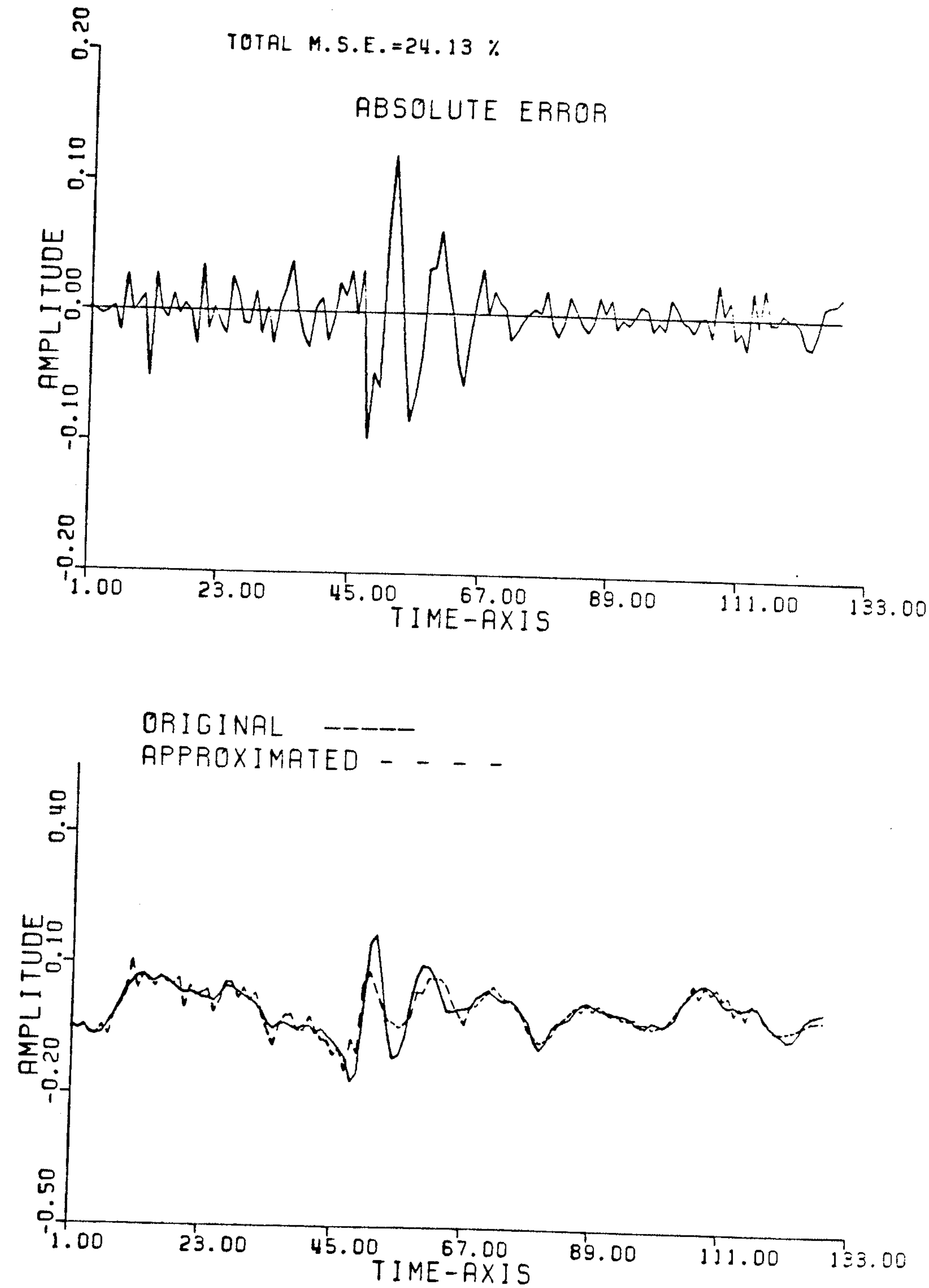


Fig.(2-18) Original and approximated isolated spike EEG epoch together with the point-by-point error of the approximation.

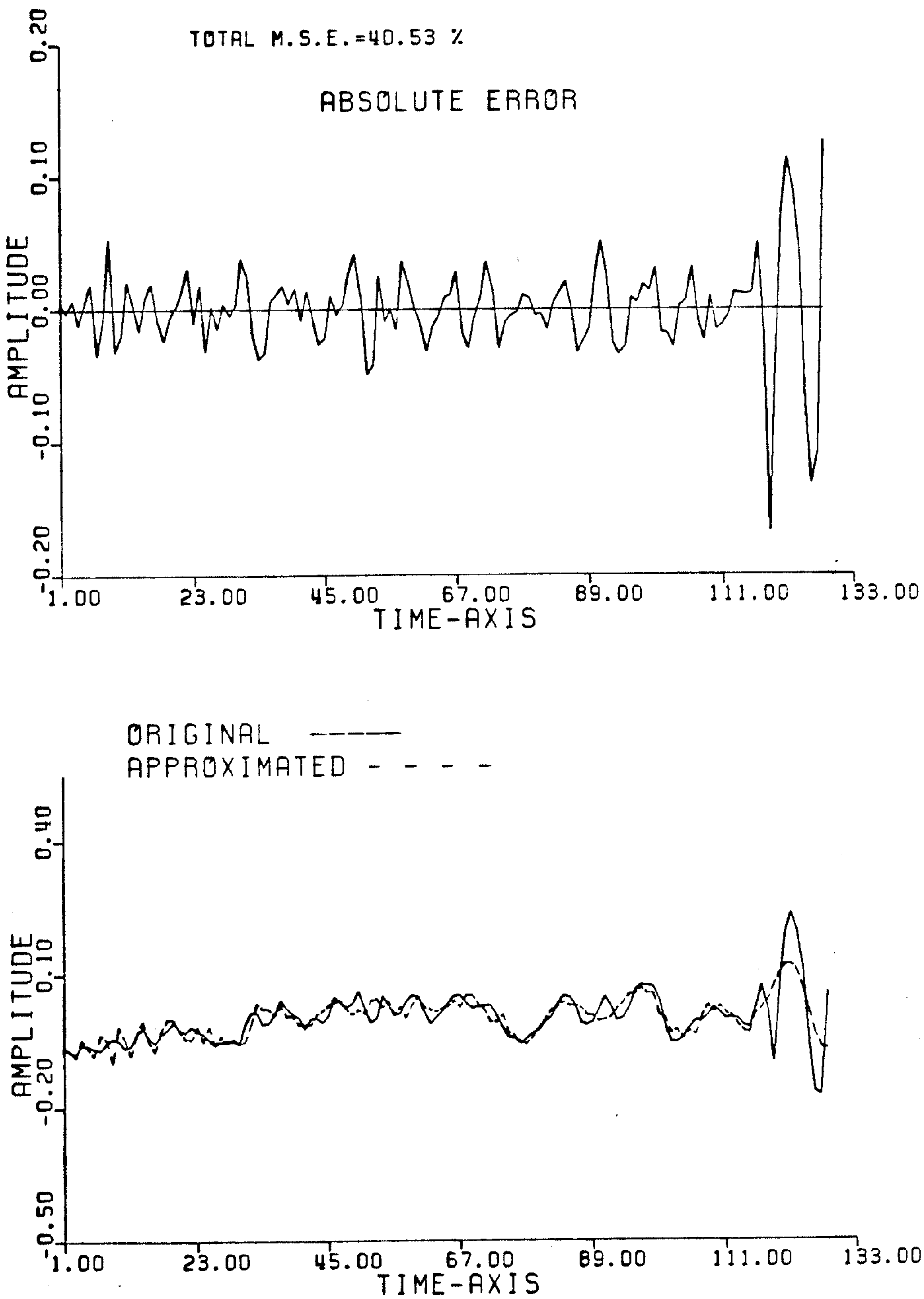


Fig. (2-19) Original and approximated multi-spikes EEG epoch together with the point-by-point error of the approximation.

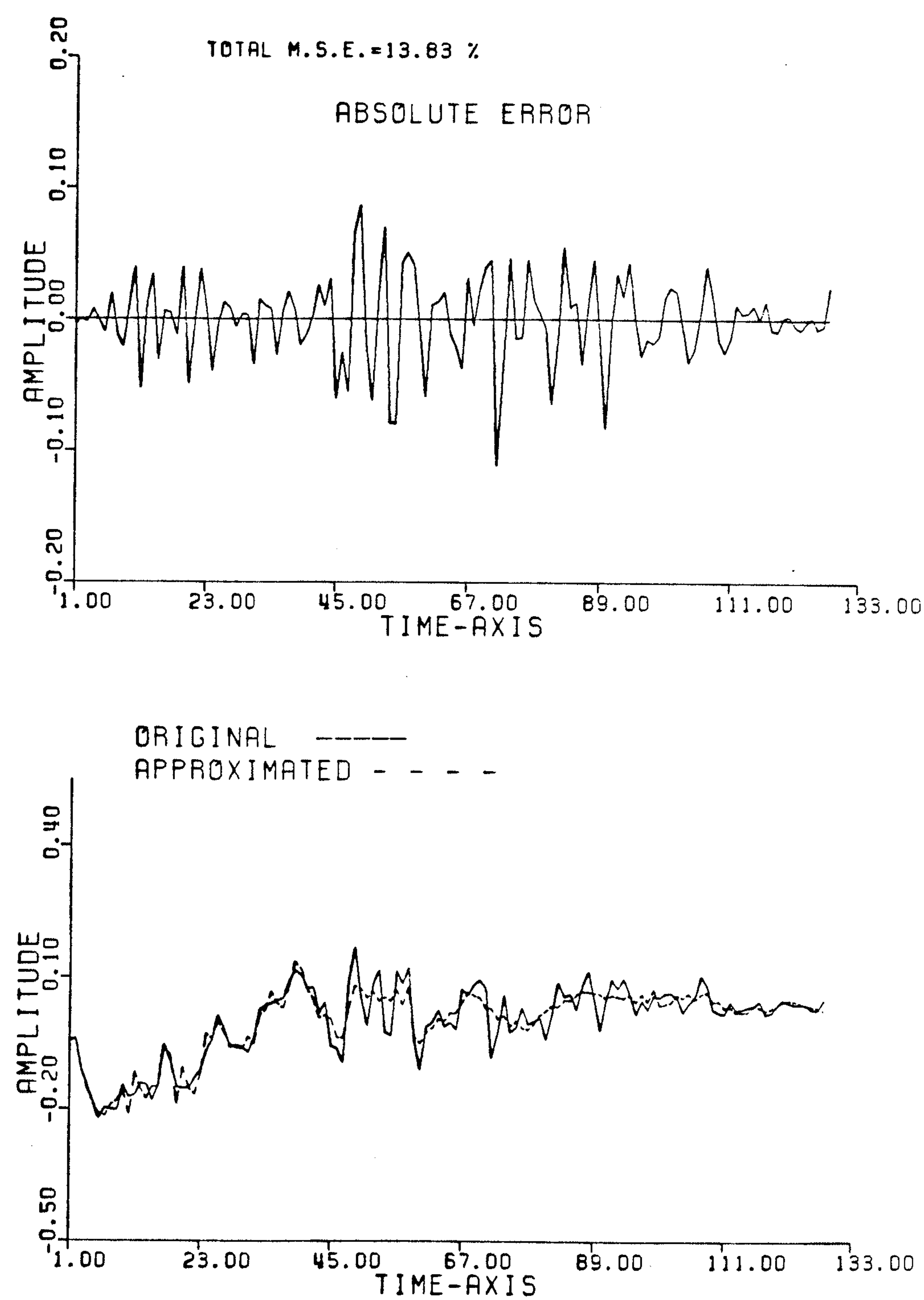
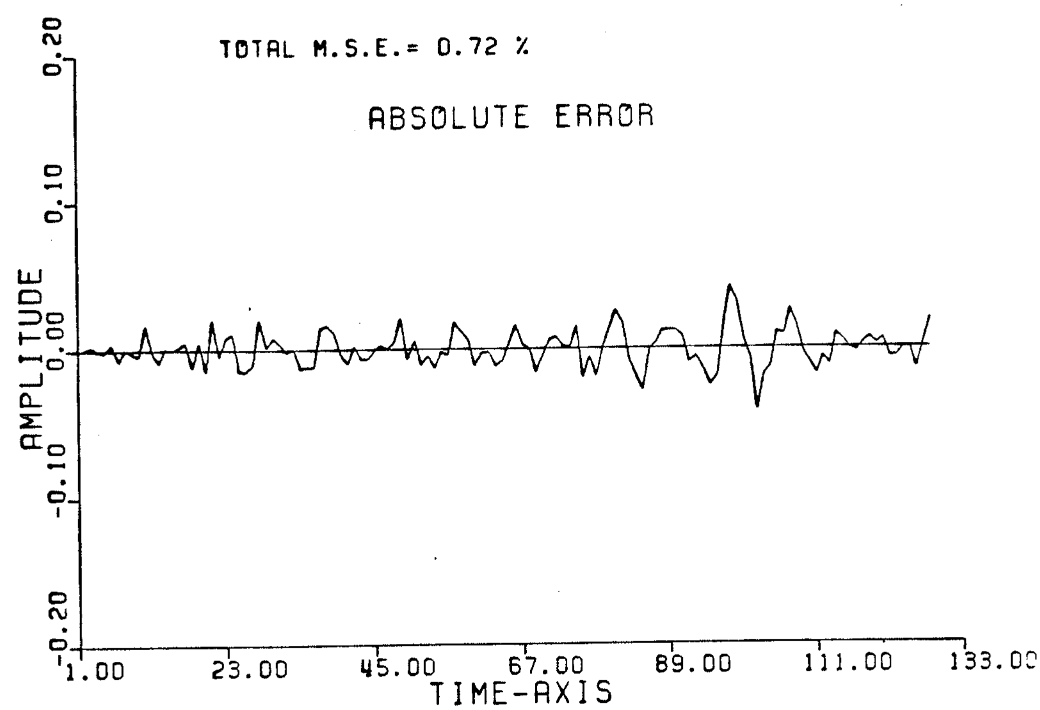


Fig.(2-20) Original and approximated multi-spikes EEG epoch together with the point-by-point error of the approximation.



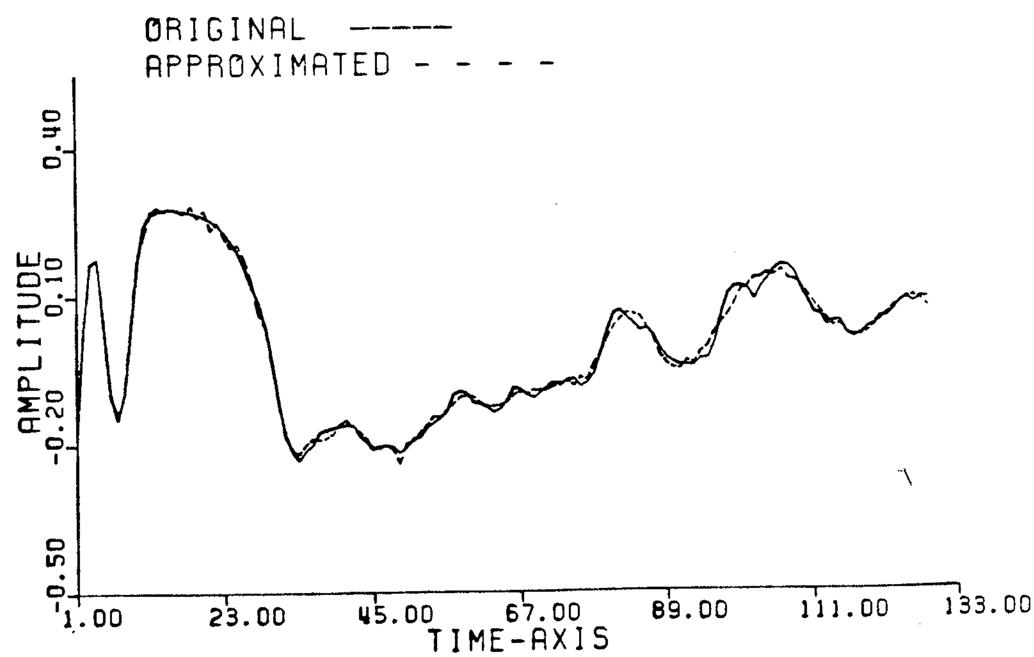


Fig.(2-22) Original and approximated slow wave EEG epoch together with the point-by-point error of the approximation.

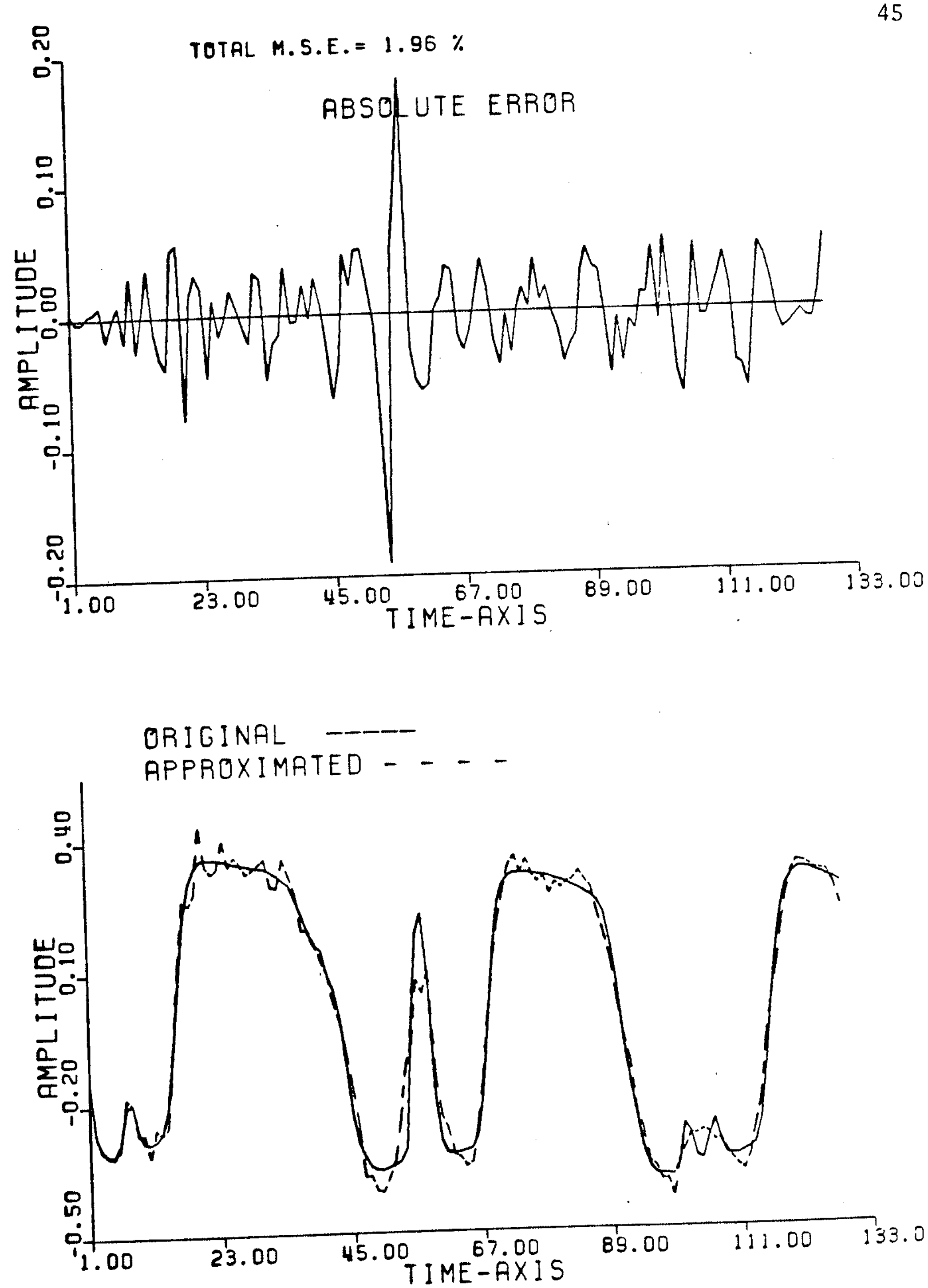


Fig.(2-21) Original and approximated sharp/spike wave EEG epoch together with the point-by-point error of the approximation.

CHAPTER 3

EEG DATA COMPRESSION USING STRAIGHT LINE INTERPOLATION

3-1- Introduction

One of the well known category of data compression is what is called redundancy reduction. Redundancy reduction is a technique for eliminating data samples that can be implied by examination of preceding or succeeding samples, or by comparison with arbitrary reference patterns. Shannon has defined redundancy as "that fraction of a message or datum which is unnecessary and hence repetitive in the sense that if it were missing the message would still be essentially complete, or at least could be completed"[8]. The choice of reference patterns used to detect redundancy is virtually unlimited. Polynomials, exponentials, and sine waves are good examples of reference patterns by which real data can often be approximated. The process of redundancy reduction can be achieved by means of "prediction" from a priori knowledge of previous samples, or by a posteriori "interpolation" from future samples.

For redundancy reduction to achieve reasonable compression efficiencies, it is often necessary to introduce certain errors. However redundancy reduction is designed such that the original waveforms can be reconstructed with a guaranteed fidelity. This fidelity can be established to supply the data within the accuracy requirements of the user.

Many techniques for redundancy reduction are possible, however, one simple method relies on the approximation of the signal by polynomial segments or "Interpolators". The parameters of each polynomial are stored in place of the original data. First order polynomials, i.e., straight line interpolation, are considered here specifically although the idea can be readily generalized to any other polynomial. At the end of this chapter, a comparison is given between this technique and the KLT technique with respect to their efficiency in compressing EEG signal.

3-2- Description of First-order Polynomial Interpolation Techniques

As the name implies, the first-order polynomial interpolator algorithm approximates the data with a linear curve. These algorithms operate by fitting the longest straight line to the signal data and storing signal information only when the signal deviates by more than a preset error limit from the straight line. This error limit is adjusted according to the error criterion of the algorithm. The non redundant samples are selected according to this criterion. The signal can be reconstructed, with the resulting sample point error always less than a fixed limit [10-20]. Several methods exist for representing the signal by a straight line, the following is a description of some of these methods.

3-2-1 First-order Interpolator - Two Degrees of Freedom [9-10]

The first-order interpolator - two degrees of freedom (Fan Interpolator) operates by fitting the longest straight line between two sample points say X_o and X_n. This is done such that all the intermediate sample points fall within the slanting aperture defined by the two straight lines between $X_o + \Delta X$ and $X_n + \Delta X$, and $X_o - \Delta X$ and $X_n - \Delta X$ as shown in Figure (3-1). It is obvious that this algorithm uses the absolute value error criteria in fitting the straight line. In this algorithm, the first point X_o is stored. A line is drawn between X_o and the second sampled point after X_o, i.e., X_2. If the first point after X_o is within a tolerance ΔX of the interpolated value, then a straight line is drawn between X_o and the third point after X_o, i.e., X_3. The interpolated values of the first and the second points are now checked to see if they are within a tolerance ΔX of the actual values. If at the n^{th} sample point after the last stored one, a line is drawn,

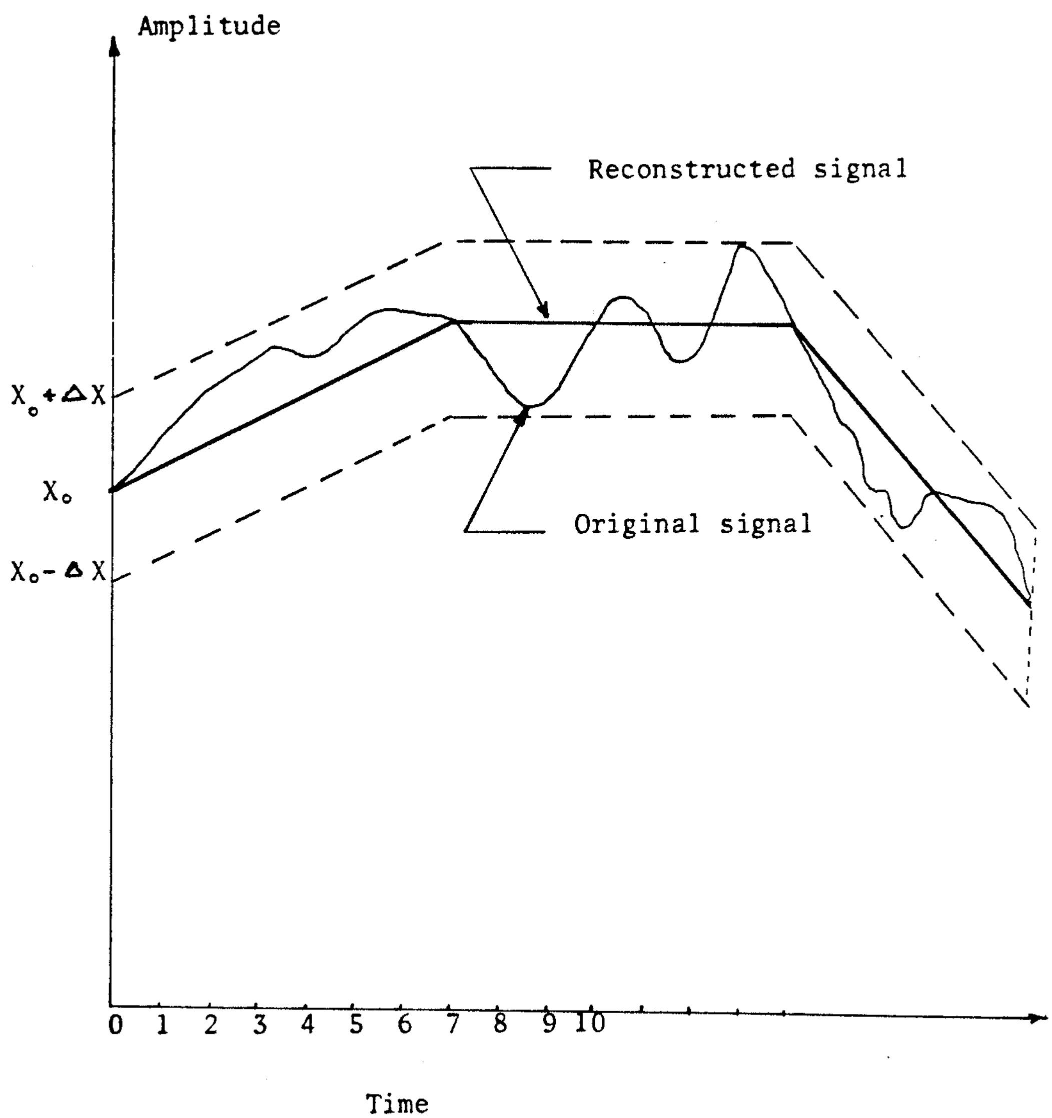


Fig. (3-1) Operation of the FAN INTERPOLATOR and the actual value differs from the interpolated value by a quantity greater than the tolerance, then the (n-1)th sample is stored as well as the total number of samples between the start and the end points of the line, i.e., the number (n-1) is stored. The process is then repeated using X_{n-1} as the starting point of the new line segment. The signal is reconstructed by drawing straight lines between the points $\{\chi_i, i\}$.

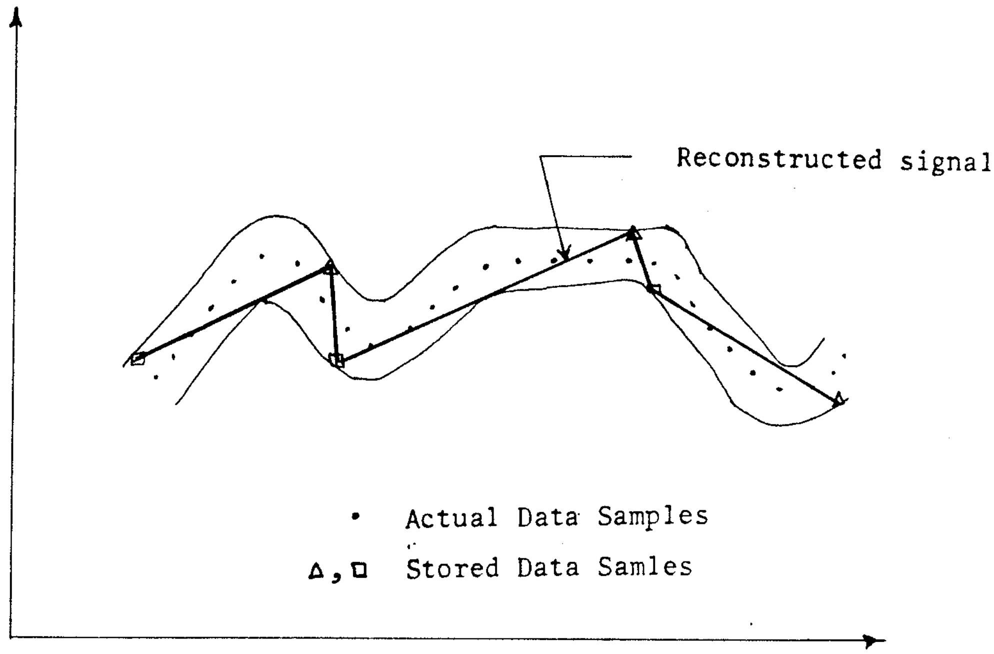
A Fortran program is given in Appendix B, which has been written to implement the algorithm.

3-2-2- First-order Interpolator - Four Degrees of Freedom

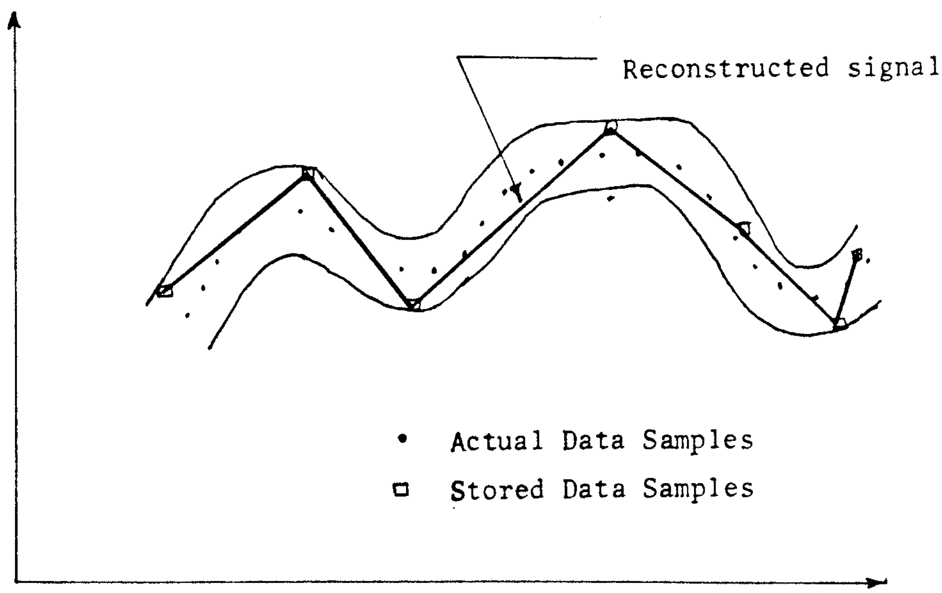
To achieve the largest compression ratio, using a linear interpolator, it is necessary to select a line segment that is within K percent of as many samples as possible. This optimum first-order algorithm requires freedom of both the starting and end points of the straight line, resulting in four degrees of freedom.

The performance of this optimum process is shown in Figure (3-2(a)). It is seen that both the starting and end points of each line are computed values such that the most positive error and the most negative error between the sample value and the interpolated value are equal and within prescribed tolerance. The computed values of both ends of the line are stored. The next straight line is started from the next sample value after the last stored point. Also, the end point of the segment can be connected with a straight line to the beginning of the following line segment. This method is also known as the min-max straight line fitting or the equal error line approximation [22].

Since the four-degrees-of-freedom linear interpolator is a complex process to implement, it is reasonable to seek less than optimum approximations that can be mechanized more easily. Anchoring the starting point of the line segment to an actual or computed sample greatly simplifies the solution. Figure (3-2(b)) shows one solution whereby the starting point of a new line is common to the end point of the previous line. This algorithm is referred to as the "Joined line segment" first-order interpolator. This technique was applied to EEG data using the least-squares error criteria in fitting the straight line (the regression line fitting), combined with the equal-error property. Such an almost-equal-error is desirable since it implies that our approximation has almost uniform accuracy along the entire signal [22-23]. Both the parameters which completely define the straight line $p(x) = Mx + B$, and the number of data points between the ends of the line, are stored. These parameters are given by the following equations:



(a)



(b)

Fig.(3-2) First order interpolator

(a) General, Four degrees of freedom .
(b) Joined line segments.

$$M = \frac{S_0 t_1 - S_1 t_0}{S_0 S_2 - S_1^2} \quad , \quad B = \frac{S_2 t_0 - S_1 t_1}{S_0 S_2 - S_1^2} \qquad (3\text{-}1)$$

where

$$S_0 = (N + 1) \quad , \quad S_1 = \Sigma x_i \quad , \quad S_2 = \Sigma x_i^2 \qquad (3\text{-}2)$$

$$t_0 = \Sigma y_i \quad , \quad t_1 = \Sigma x_i y_i$$

The detail of this criteria is fully explained in Appendix D. In addition, a Fortran program has been written to describe how this technique was implemented to operate on EEG data, and is listed in Appendix B.

3-3- Results and Comparison

Applying the straight line interpolation techniques, shows that their performance is not satisfactory. To illustrate this conclusion, Table (2) is given, which contains the results of the techniques as applied to six different selected epochs. From the table, it is apparent that the compression ratio varies considerably from one epoch to another for the same amount of tolerance, which simply means that the compression ratio is not constant along the entire record. Also, it shows that the compression ratio is too dependent on the predefined tolerance, which is found to be strongly dependent on the epoch in question. Therefore, it has to take different values depending on the epoch characteristic. In spite of this dependence, if the tolerance is adjusted to change in agreement with the epoch characteristic, the compression ratio would still vary effectively from an epoch to another. The table also shows that the resultant mean-square error of the approximation is high, even for a reasonable compression ratio. Besides, the reconstructed signal is found to be too distorted from the original signal.

On the other hand, Karunen-Loève transform performs much better than these interpolation techniques, in that, it results in a constant compression ratio 2.56:1 which does not depend on the signal characteristic at all. Also, the mean-square error of the approximation is fairly acceptable for different epochs (range 1% - 20% in most cases). The reconstructed signal using the KLE, is very close to the original signal in most cases.

It is clear from the above discussion that, given sufficient a priori knowledge of the signal process, there is little to be gained by using these redundancy removal techniques with EEG data. Also, it is obvious that the class of compression methods considered here may or may not perform well depending on the data.

Table (2) Results of the straight line interpolation techniques as applied to six different EEG epochs

epoch No.	Fan Interpolator Two degrees of freedom				Four degrees of freedom Interpolator			
	$\Delta=80.0 \times 10^{-3}$		$\Delta=40.0 \times 10^{-3}$		$\Delta=80.0 \times 10^{-3}$		$\Delta=40.0 \times 10^{-3}$	
	D.C.R.	M.S.E. %	D.C.R.	M.S.E. %	D.C.R.	M.S.E. %	D.C.R.	M.S.E. %
1	2.03	72	0.81	6.2	3.3	40	1.07	74.6
2	3.12	49.3	1.11	19.1	8.53	55.4	2.56	48
3	18.3	121.6	2.51	33.9	42.6	62.4	14.2	31.6
4	25.6	132.8	5.56	60.5	42.6	99.8	14.2	55.8
5	1.58	23.8	0.83	6.0	5.33	69	1.94	98.1
6	1.85	40.94	0.82	4.5	3.88	90.2	1.94	92.6

S.

D.C.R. ≡ Data Compression ratio m (m = N/M)

M.S.E. ≡ Mean-Square error

Δ ≡ error tolerance.

CHAPTER 4

DESIGN OF A TERMINAL CLASSIFIER

4-1 Introduction

Designing a terminal classifier is found to be necessary in order to detect the abnormalities associated with epilepsy. Besides, the terminal classifier could be used as an alternative way of testing how efficient the KLE data compression technique is, in terms of preserving the essential features in the signal after approximation. In this chapter we concentrate in recognizing the abnormalities associated with epilepsy of both original and approximated EEG signal. Therefore, a brief description of some features of the epileptic EEG is provided below.

The epileptic EEG is described in terms of 3 types of waveforms - spikes, sharp waves, and slow waves. A spike is defined as a transient, clearly distinguished from background activity, with a pointed peak and duration from 1/50 to 1/14 seconds. A sharp wave is a transient similar to a spike, but is of longer duration (1/14 to 1/5 seconds). A slow wave is any wave of duration greater than 1/8 seconds.

For the purpose of recognizing epileptic abnormalities, the EEG could be adequately described in terms of ten terminal classes: (a terminal is an epoch of one second length)

1) Normal activity (predominantly alpha activity). [NORM]
2) Low amplitude activity ($< 20\mu v$) [LAMP]

3) Fast activity (> 14 HZ, but not including activity related to abnormalities) [FAST]

4) Low frequency artefact (characterized by large shifts in the baseline of the recording). [ARTEF]

5) Slow wave activity of low amplitude (< 50 μv). [SLOLO]

6) Slow wave activity of significant amplitude (> 50 μv). [SLOHI]

7) Spike/Sharp wave of low amplitude (< 75 μv). [SSWLO]

8) Spike/Sharp wave of significant amplitude (> 75 μv). [SSWHI]

9) Isolated spike activity. [ISOSP]

10) Multiple spike activity. [PLYSP]

A design of a terminal classifier was part of a B.Sc. Thesis submitted to the Department of Electrical Engineering at the University of Manitoba, by R. Thorne. The same technique has been used here, and some modifications have been added to design another classifier. But since Thorne's thesis is not published, a detailed description of the used technique is provided in Appendix E.

For the purpose of our study, 615 one-second EEG epochs were obtained from six patients between the ages of ten and sixteen years, all were suspected of having some form of epileptic abnormality. These epochs were used in their original and approximated forms to evaluate both the classification and the data compression efficiencies. Table (3) gives the number of epochs belonging to each of the above mentioned classes.

4-2 Terminal Classifier Design Alternatives and Results

4-2-1 Parallel Classification

It is noted from Table (3) that none of the available 615 one-second epochs, as classified manually, have been classified as SSWLO (spike/sharp wave of low amplitude). Therefore, the step wise discriminant analysis (SWDA) was adjusted to discriminate only between the previously mentioned nine classes. Appendix E contains more information about the SWDA. A training set of 350 EEG epochs, each represented by 48 features, formed the entry data to the SWDA. Only 14 features were selected by the SWDA as they proved to be the most significant in discriminating between the nine classes. To test the classifier performance, a Fortran program has been written and listed in Appendix B. Figure (4-1) illustrates the parallel classification approach, and Table (4) displays the results of this classification technique.

The success rate of this classifier in recognizing epileptic abnormalities was 95.3%, and the false detection rate was 43.2% (considering the abnormalities as every activity except those of normal and artefactual type). Due

Table (3) Number of epochs as classified manually

Class	No. of epochs
NORM	112
LAMP	20
FAST	89
ARTEF	43
SLOLO	111
SLOHI	83
SSWLO	0
SSWHI	120
ISOSP	17
PLYSP	20
TOTAL	615

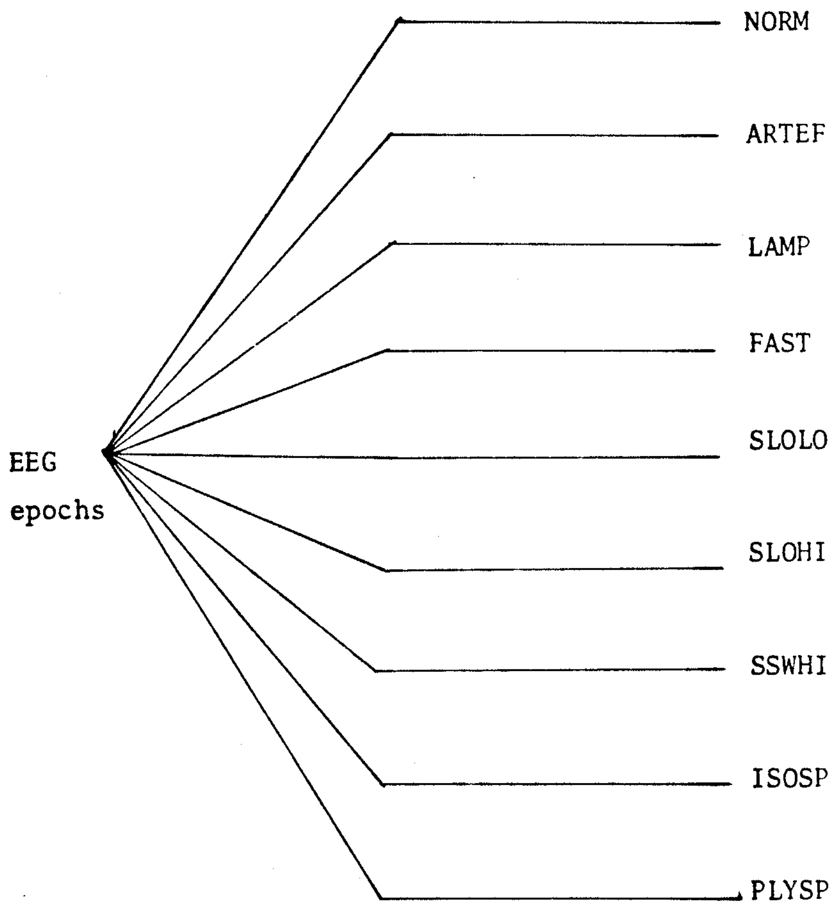


Fig.(4-1) Tree diagram of the parallel classifier.

Table 4: Jackknifed Classification performance of the decision functions generated by using SWDA, in case of parallel classification.

	Correct Original Classification in (%)	Correct Approximated Classification in (%)
NORM	47.3	20.5
ARTIF	79.1	83.7
LAMP	90	95
FAST	91	32.6
SLOLO	55.5	34.5
SLOHI	38.6	73.5
SSWHI	80.8	27.5
ISOSP	70.6	5.9
PLYSP	45	15

to this high false detection rate, it is required to seek other classifiers.

4-2-2 Sequential Classification

As the name indicates, this technique consists of a sequence of steps. At each step an epoch is classified into a subset of classes. At the end of the sequence, each epoch will be classified into one distinct class out of the nine classes previously mentioned.

Four different methods have been implemented using this technique, each one differing slightly from the other. A description of these methods is given below with their performance in classifying epileptic abnormalities.

The first method consists of four steps; in the first step, normal activity epochs are separated from all other epochs. Then, the second step separates the artefactual epochs from the remaining epochs. The rest of the epochs are then classified into seven abnormal classes in two steps. Figure (4-2(a)) illustrates these steps by means of a tree diagram, and Table (5) displays the results.

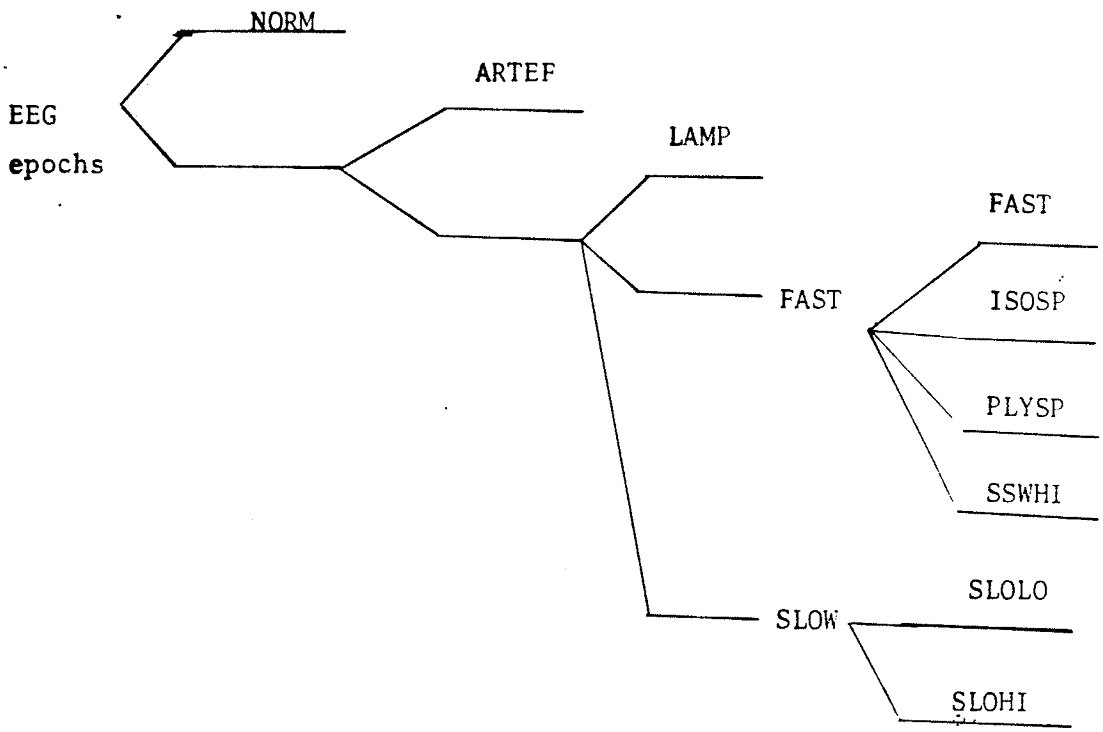
The only difference between the first and the second method is that the abnormalities are classified into the seven classes in one step only. Figure (4-2(b)) illustrates this method via a tree diagram, and Table (6) contains the results.

For these previously described methods, the success rate in recognizing EEG abnormalities was 76.2% and the false detection rate was 21.3%.

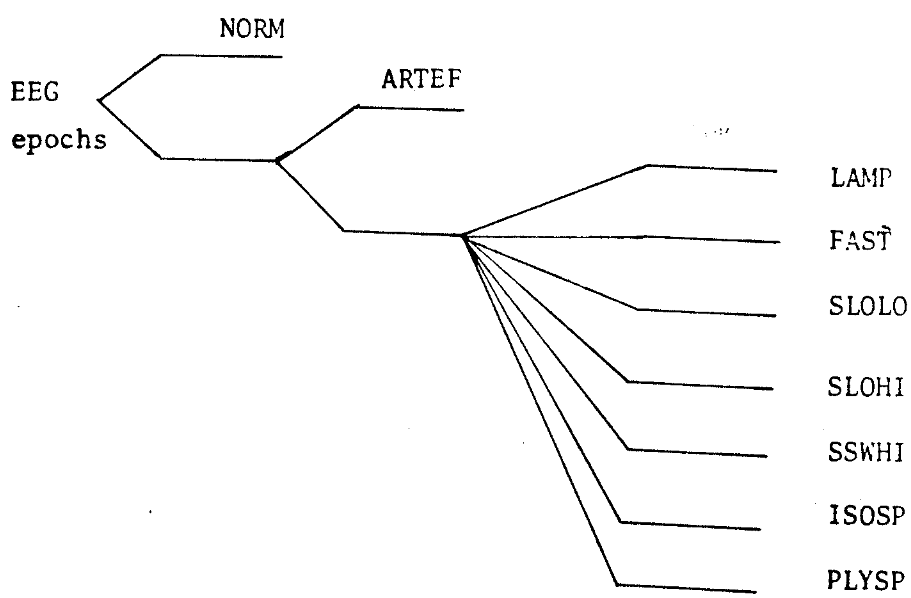
The third and fourth methods are based on the same idea of the first and the second methods, respectively, with the exception that in their first step the epochs are separated into three classes, i.e., normal, artefact and abnormal activities. The tree diagrams for these methods is shown in Figure (4-3), and Tables (7) and (8) are given to illustrate their performance in classifying EEG data. The success rate in recognizing the abnormalities using these methods was 81.1%, and the false detection rate was 24.5%.

From the classification results obtained, several deficiencies in the terminal classifier performance were noted. Of these, the most serious was the classifier's inability to detect very short duration spikes, especially after approximation. The probable cause of this deficiency is that the chosen feature set does not accurately represent the spikes. A second problem encountered was the excessive detection of low amplitude slow waves, which are seldom of clinical significance. This problem likely arose due to deficiencies in the training set.

It is also observed that, in general, the correct classification drops significantly in cases of fast, spike

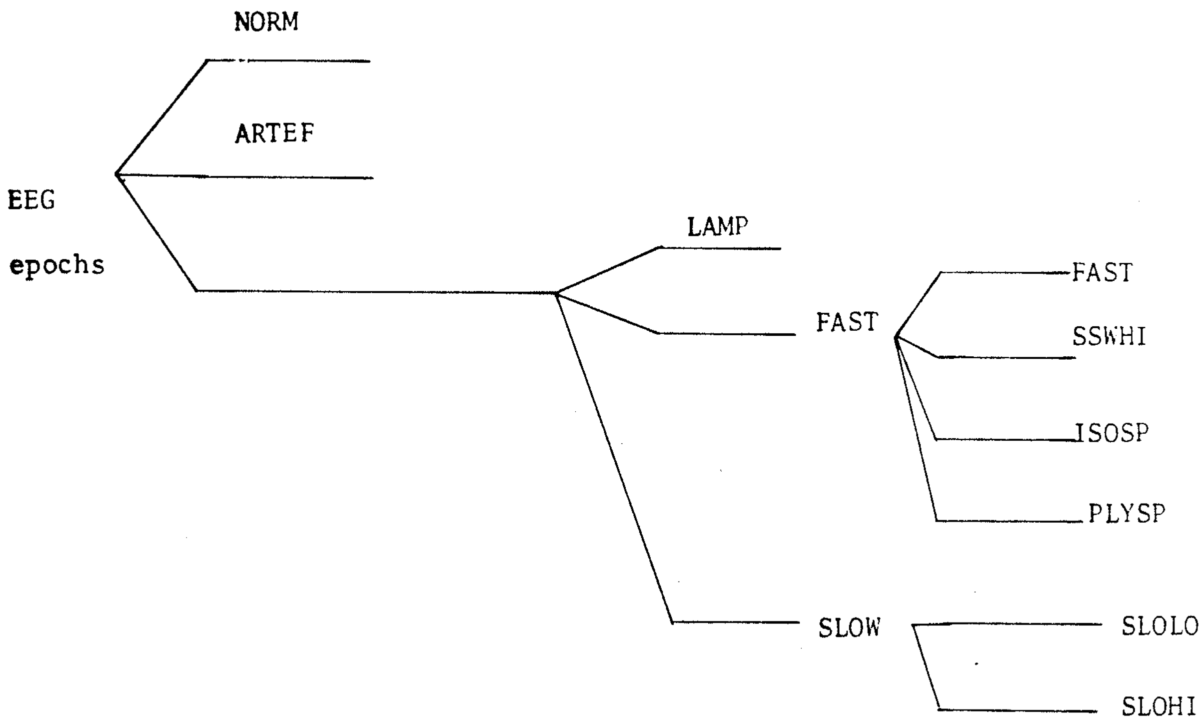


(a) Method [1]

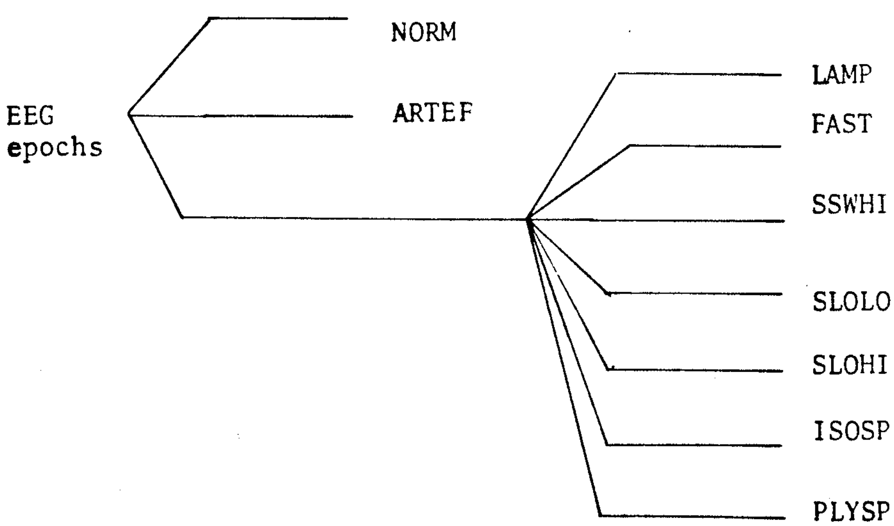


(b) Method [2]

Fig.(4-2) Tree diagram of the sequential classifier



(a) Method [3]



(b) Method [4]

Fig.(4-3) Tree diagram of the sequential classifier.

Table 5: Jackknifed Classification performance of the decision functions generated by using SWDA, in case of method 1, sequential classification

	Correct Original Classification in (%)	Correct Approximated Classification in (%)
NORM	75.9	51.8
ARTIF	86.05	86.05
LAMP	85	75
FAST	60.7	5.6
SLOLO	39.6	47.8
SLOHI	34.9	37.4
SSWHI	40	34.2
ISOSP	29.4	5.9
PLYSP	35	5

Table 6: Jackknifed Classification performance of the decision functions generated by using SWDA, in case of method 2, sequential classification

	Correct Original Classification in (%)	Correct Approximated Classification in (%)
NORM	75.9	51.8
ARTIF	86.05	86.05
LAMP	85	75
FAST	84.3	12.4
SLOLO	42.3	36
SLOHI	34.9	26.5
SSWHI	24.2	16.7
ISOSP	47.1	5.9
PLYSP	17.7	0.0

Table 7: Jackknifed Classification performance of the decision functions generated by using SWDA, in case of method 3, sequential classification

	Correct Original Classification in (%)	Correct Approximated Classification in (%)
NORM	71.4	48.2
ARTIF	86	95.4
LAMP	95	85
FAST	62.9	7.9
SLOLO	46	61.3
SLOHI	34.9	37.4
SSWHI	34.2	20
ISOSP	29.4	0.0
PLYSP	40	5

Table 8: Jackknifed Classification performance of the decision functions generated by using SWDA, in case of method 4, sequential classification

	Correct Original Classification in (%)	Correct Approximated Classification in (%)
NORM	71.4	48.2
ARTIF	86	95.4
LAMP	95	85
FAST	86.5	16.9
SLOLO	48.7	48.7
SLOHI	34.9	26.5
SSWHI	20.8	11.7
ISOSP	58.8	5.9
PLYSP	20.0	0.0

and spike/sharp wave activities after approximation. This is because the approximated signal suffers severely from losses of its high frequency components. Also, it is noted that mean-square error of the approximation does not have, nearly, any effect on the classification performance of the classifier in classifying the approximated signal.

4-3- Random Coefficients of KLT As a Feature Set

Our major concern, here, is to examine the application of Karhunen-Loeve expansion Transform (KLT) as a pattern recognition technique. Since KLT is a linear feature extraction operator, its random coeffients were used as a feature set to represent the EEG data.

The training set, supplied to the SWDA, consisted of 315 one-second EEG epochs represented by their 48 random coefficients (projections) as a feature set. The third method, discussed in the previous section, was used as the classification technique using this feature set. The results obtained using this feature set were not encouraging, even with readjusting of the SWDA parameters. This implies that the feature set suggested by R. Thorne was better in representing the EEG signal. However, the KLT coefficients could be used successfully to represent the EEG signal, if some modifications are introduced in the SWDA control parameters. Also, it is noted that the choice of the training set epochs is critical and has great influence on the results. Therefore, attention must be paid in selecting the elements of the training set. Figure (4-4) shows the results of this method as applied to EEG.

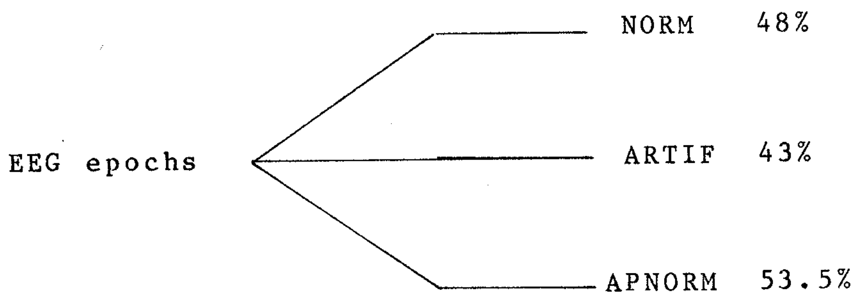


Fig. (4-4) KLE Random Coefficients as a feature set with method (3) of the sequential classifier. The percentage of the correct classification is written beside each class.

CHAPTER 5

CONCLUSIONS

The overall goal of this thesis was the study of the effectiveness of various data compression techniques in reducing the storage requirements of the EEG recording. Of the techniques considered, the Karhunen-Loéve transform proved to be superior to the others. A data compression ratio of 2.56:1 is feasible, using the KLT technique, for a mean-square error less than 15%. It was found that 17% of the epochs were approximated with a mean-square error (M.S.E.) greater than 15%. Most of these were of low amplitude nature, consequently the absolute error was of the same order of magnitude as the signal which resulted in a high M.S.E. In spite of this, the approximated epochs agreed qualitatively with the original data, in most of these cases.

The straight line interpolation techniques were investigated for the EEG. They proved to be of less significance as compared to the KLT for three reasons. First, the mean-square error of the approximation was fairly high. Second, even with low mean-square error, the approximated signal appeared to have a distorted waveform as compared with the original data. Third, the data compression ratio was not uniform over the whole signal, i.e., it varies considerably from one epoch to another. In spite of this, the straight line techniques were able to approximate the isolated spike epochs, better than the KLT.

An alternative approach to measure the effectiveness of the KLT, was to compare the original and approximated epochs from a pattern recognition point of view. This was done by using different forms of a terminal classifier to classify both the original and the approximated epochs. From the obtained results, it is noted that the classifier failed to detect the spikes in most of the original epochs (only 37 epochs were available). Also, it is noted that the correct classification drops significantly in cases of fast, spike and spike/sharp wave activities. This indicated that the approximation, using KLE, causes high frequency information losses. However, misclassification, in many cases, was due to deficiencies in the terminal classifier itself. The choice of the training set elements, which is used to obtain the decision functions, had a great influence on the classifier performance. Also, the improper definition of some EEG classes, e.g. slow and fast, was another reason for the classifier deficiencies. Therefore, a critical and careful choice of the training set elements is required to improve the classifier performance. Besides, a better definition of a new set of classes, suitably describing the EEG activities, may, as well, improve the classifier performance.

REFERENCES

[1] Alan S. Gevins and et. al., "Automated Analysis of the Electrical Activity of the Human Brain (EEG): A Progress Report", Proc. of IEEE, Vol 63, No. 10, Oct. 1975, pp. 1382-1399.

[2] J.S. Barlow, "Computerized Clinical Electroencephalography in Prospective", IEEE Transactions on Biomedical Eng., Vol. BME - 26, No. 7, July 1979, pp. 377-391.

[3] C.D. Binnie et al., "Pattern Recognition in Electroencephalography", in Pattern Recognition: Ideas in Practice, B. Batchelor, Ed., New York: Plenum Press, 1978, pp. 399-426.

[4] D. Michael and J. Houchin, "Automated EEG Analysis: A Segmentation Procedure Based on the Autocorrelation function", Electronic and Clinical Neurophysiology 1979, 46, pp. 232-235.

[5] C. Bodenstein and H. Michael Praetorius, "Feature Extraction from the Electroencephalogram By Adaptive Segmentation", Proc of IEEE, Vol. 65, No. 5, May 1977, pp. 642-652.

[6] Anders Isaksson et. al., "Computer Analysis of EEG Signals with Parametric Models", Proc. of IEEE, Vol. 69, No. 4, April 1981.

[7] Lars H. Zetterberg, Kjell Ahlin, "Analogue Simulator of EEG Signals Based on Spectral Components", Medical and Biological Engineering, March 1975, pp. 272-278.

[8] C.M. Kortman, "Redundancy Reduction - A Practical Method of Data Compression", Proc. of IEEE, Vol. 55, No. 3, March 1967, pp. 253-263.

[9] C.A. Andrews and et. al., "Adaptive Data Compression", Proc. of IEEE, Vol. 55, No. 3, March 1967, pp. 278-286.

[10] L. Ehrman, "Analysis of Some Redundancy Removal Bandwidth Compression Techniques", Proc. of IEEE, Vol. 55, No. 3, March 1967, pp. 278-286.

[11] M. Edward Womble and et.al., "Data Compression For Storing and Transmitting ECG's/VCG's", Proc. of IEEE, Vol. 65, No. 5, May 1977.

[12] Nasir Ahmed and et. al., "Electrocardiographic Data Compression via Orthogonal Transforms", IEEE Trans. on Biomedical Engineering, Vol. BME-22, No. 6, Nov. 1975.

[13] S.J. Campanella and Guner S. Robinson, "A Comparison of Orthogonal Transformations for Digital Speech Processing", IEEE Trans. Commun. Tech., Vol. COM-19, No. 6, pp. 1045-1049, (1971).

[14] J. Tou and R. Gonzalez, "Pattern Recognition Principles", Don Mills: Addison Wesley, 1974.

[15] L. Otterstran, "Automatic Monitoring of Cardiac Arrhythmias Using Template Matching", Proc. of the IV Nordic Meeting on Medical and Biological Engineering, June 1977, pp. 66-1 - 66-2.

[16] D.F. Sprecht, "Vectorcardiographic Diagnosis Using the Polynomial Disriminant Method of Pattern Recognition", IEEE trans. on Biomedical Engineering, January 1977, pp. 63-67.

[17] Kao, R., "Morphological Feature Extraction for Arrhythmia Monitoring", B.Sc. and M.Sc. Thesis, MIT, June 1980.

[18] Ali Habibi, "Image Coding by Linear Transformation and Block Quantization", IEEE Trans. Commun. Tech., Vol. Com-19, No. 1, pp. 50-62, (1971).

[19] Brice Carnahan, H.A. Luther, James O. Wilkes, "Applied Numerical Methods", Pub. John Wiley & Sons, Inc., 1969.

[20] Lee D. Davisson, "Data Compression Using Straight Line Interpolation", IEEE Trans. Inform. Theory, IT-14(3), pp. 390-394, (1968).

[21] Lee D. Davisson, "Theoretical Analysis of Data Compression Systems", Proc. IEEE - Vol. 56, No. 2, pp. 176-186, (1968).

[22] F. Scheid, "Numerical Analysis", Schoum's Outline Series, McGraw-Hill Book Co., 1968.

[23] Kreysziy, "Advanced Engineering Mathematics", John Wiley & Sons, Inc., 1972.

[24] Torsten Bohlin, "Analysis of EEG Signals with Changing Spectra Using a Short-Word Kalman Estimator", Math. Biosciences-35-, pp. 221-259, (1977).

[25] R. Gonzalez and M. Thomason, "Syntactic Pattern Recognition: An Introduction", Don Mills: Addison-Wesley, 1978.

[26] L.H. Zetterberg, "Means and Methods for Processing of Physiological Signals with Emphasis on EEG Analysis", in Advances in Biological and Medical Physics, Vol. 16, J. Lawrence et.al., Eds., New York: Academic Press, 1977, pp. 41-97.

[27] B. Hjorth, "EEG Analysis Based on Time Domain Properties", Electroncephalography and Clinical Neurophysiology, Vol. 29, pp. 306-310.

[28] R. Jennrich and P. Sampson, "BMDP7M - Stepwise Discriminant Analysis", in Biomedical Computer Programs, San Francisco: University of California Press, 1977, pp. 711-740.

[29] Harry L. Van-Trees, "Detection, Estimation and Modulation Theory", John Wiley & Sons, Inc., 1968.

APPENDICES

Appendix

APPENDIX - A -

Karhunen-Loéve Expansion of Continuous EEG Signal

i) EEG Signal Model

Several methods attempt to describe the spectral properties of the EEG, e.g. by calculating the signal power for narrow frequency bands, or expressed more precisely by estimating the spectral power density. Very few and mild assumptions need to be made about the process in order to make these methods applicable, but in return the accuracy in estimated spectral density tends to be fairly low. It is characteristic of these methods that the spectral density will be approximated by a rational function of low order and actually this is the essential assumption being made about the model.

The idea with one of the methods, called spectral parameter analysis, is to partition the EEG spectrum into one, two or more spectral components such that each contributes to one peak in the total spectrum. Such a component will be characterised by a few parameters, essentially the location of the spectral peak, the width of the peak and the contribution it gives to the total power.

A model, which describes the spectral properties directly and in terms that come close to the usual clinical practice using the spectral EEG parameters, is based on a partition of the spectral density into three spectral components called delta, alpha and beta components and denoted $S_{\delta}(f)$, $S_{\alpha}(f)$ and $S_{\beta}(f)$. In a typical case the total spectral density may be written as a sum of these components

$$S(f) = S_{\delta}(f) + S_{\alpha}(f) + S_{\beta}(f) \tag{A-1}$$

This is shown in Figure (A-1), where two parameters describe each component, specifically, the resonance frequency and the bandwidth. Apart from the frequency parameters, information about the power parameters G_{δ}, G_{α}, G_{β} is also given in the representation. They indicate how much power each component represents. In the Figure, G corresponds to the surface under each curve. Furthermore, the power parameters H_{α} and H_{β} are defined and they measure the skewness of each spectral component around resonance frequency.

We call $S_{\delta}(f)$ a low-frequency component while the other components are called rhythmic. These names have been chosen to indicate the close relationship between these and the activities in terms of which an EEG recording is commonly described. The delta activity is defined to be the power below 3.5 HZ, alpha activity between 7.5 and 12.5 HZ and beta activity above 12.5 HZ. However, the difference between activity and spectral component should be observed.

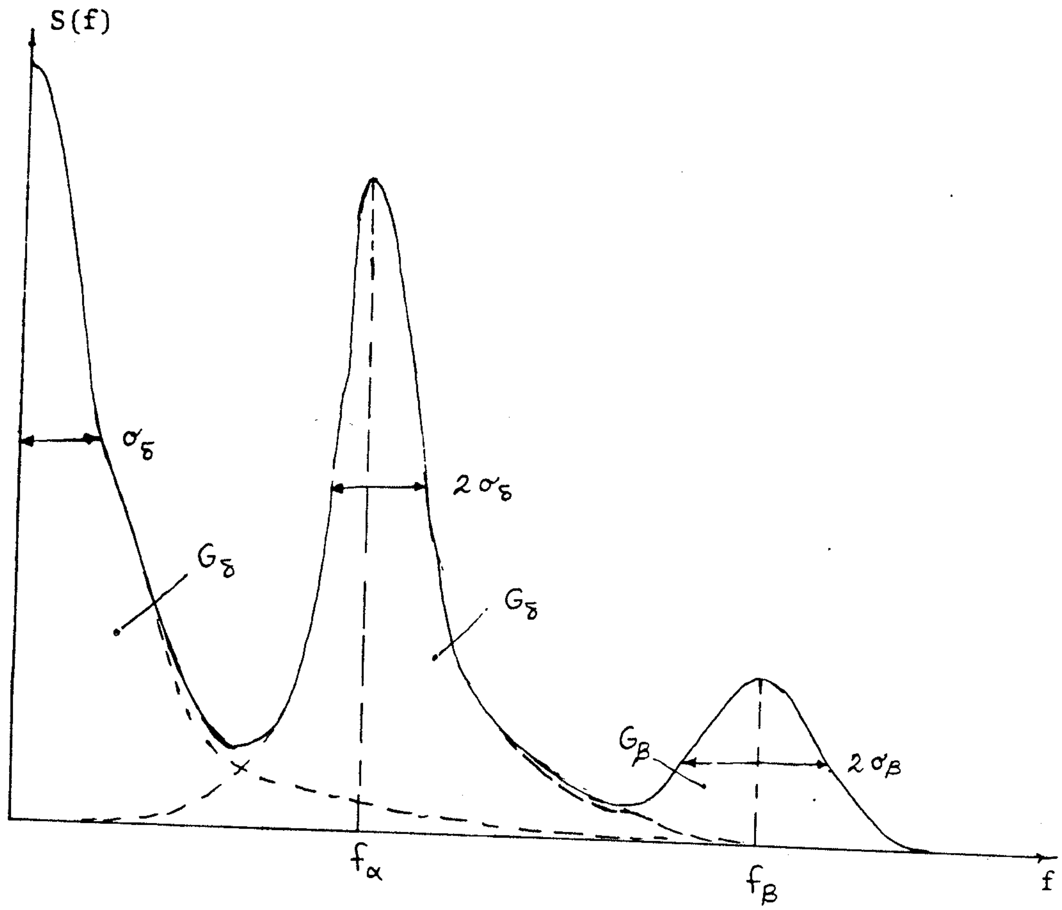


Fig. (A-1) Spectral components of the EEG signal.

Whereas, for example, the alpha activity is confined to the frequency range 7-7-12.5 HZ, the alpha component will have a spectral peak and contain most of the power in the same range, but will give a certain contribution to the power for all frequencies of interest. Hence, the partition of power between spectral activity and spectral components is different.

Exact expressions of the spectral components are needed, for the δ component the following holds true

$$S_\delta(f) = \frac{1}{\pi} \frac{G_\delta \sigma_\delta}{\sigma_\delta^2 + f^2} \tag{A-2}$$

and for the rhythmic components

$$S_i(f) = \frac{1}{\pi} \frac{A_i(f_i^2 + \sigma_i^2) + B_i f^2}{(f_i^2 + \sigma_i^2 - f^2)^2 + (2f\sigma_i)^2} \tag{A-3}$$

where i can take any notation α or β. The quantities A_i and B_i may be expressed in the spectral parameters as follows:

$$A_i = \sigma_i G_i - f_i H_i \tag{A-4}$$

$$B_i = \sigma_i G_i + f_i H_i$$

Eqns. 3 and 4 may appear rather complicated but hopefully the interrelation between the parameters and their interpretation appear more lucid when expressions are given for the autocorrelation functions that correspond to eqns. (A-2) and (A-3):

$$R_\delta(\tau) = G_\delta \exp\,(-2\pi\, \sigma_\delta |\tau|) \tag{A-5}$$

$$R_i(\tau) = (G_i \cos 2\pi\, f_i \tau - H_i \sin 2\pi f_i |\tau|) \exp\,(-2\pi\sigma_i |\tau|) \tag{A-6}$$

The autocorrelation for the entire signal may in the typical case be written as the following sum:

$$R(\tau) = R_\delta(\tau) + R_\alpha(\tau) + R_\beta(\tau) \tag{A-7}$$

Hence, the model implies that the autocorrelation function is described as a sum of damped sinusoids.

The following table (A-1) gives a typical list of these spectral parameters.

Component	Bandwidth	Resonance Frequency
δ	σ_δ: 0 - 10 HZ	f_δ: 0 HZ
α	σ_α: 0 - 10 HZ	f_α: 4 - 14 HZ
β	σ_β: 0 - 20 HZ	f_β: 8 - 28 HZ

There is also a fundamental limit imposed by the following inequality

$$|H_i / G_i| \leqslant \sigma_i / f_i \tag{A-8}$$

which corresponds to the physical requirement that the spectral density for each component can not be negative.

ii) Karhunen-Loéve expansion of Random Processes

To represent any random process with finite energy in terms of a series expansion, we start off by choosing an arbitrary complete orthonormal set: $\phi_1(t)$, $\phi_2(t)$, ...

To expand x(t) we write

$$x(t) = \underset{N \to \infty}{\text{L.i.m}} \sum_{i=1}^{N} x_i \, \phi_i(t), \quad 0 \leqslant t \leqslant T \tag{A-9}$$

where

$$x_i = \int_0^T x(t) \, \phi_i(t) \, dt \tag{A-10}$$

and the limit converges in the mean-square sense, which is defined as

$$\underset{N \to \infty}{\text{Lim}} \; E\left[\left(x_t - \sum_{i=1}^{N} x_i \, \phi_i(t)\right)^2\right] = 0, \quad 0 \leqslant t \leqslant T \tag{A-11}$$

The set $\phi_i(t)$ has to be chosen such that, the coefficients x_i are uncorrelated, i.e.,

$$\text{if} \qquad E[x_i] = m_i \tag{A-12}$$

we would like

$$E[(x_i - m_i)(x_j - m_j)] = \lambda_i \delta_{ij} \tag{A-13}$$

For simplicity we assume that $m_i = 0$ for all i substituting (A-10) into (A-13) and bringing the expectation inside the integral, we obtain

$$\lambda_i \, \delta_{ij} = E[x_i x_j] = E[\int_o^T x(t) \; \phi_i(t)dt \int_o^T x(u) \; \phi_i(u)du]$$

$$= \int_o^T \phi_i(t)dt \int_o^T R_x(t-u) \; \phi_j(u)du$$

$$\text{For all i and j} \qquad \text{(A-14)}$$

In order that (A-14) may hold for all choices of i and a particular j, it is necessary and sufficient that the inner integral equal $\lambda_i \phi_i(t)$ or

$$\lambda_i \phi_j(t) = \int_o^T R_x(t - u) \; \phi_j(u)du, \quad 0 \leqslant t \leqslant T \qquad \text{(A-15)}$$

The functions $\phi_i(t)$ are called eigenfunctions and the numbers λ_i are called eigenvalues. Several observations are worthwhile

1)- the value x_i^2 has a simple physical interpretation. It corresponds to the energy along the coordinate function $\phi_i(t)$. Similarly, $E[x_i^2] = \lambda_i$ corresponds to the expected value of energy along $\phi_i(t)$.

2)- if $R_x(t-u)$ is positive definite, the eigenfunctions form a complete orthonormal set, and vice-versa.

3)- if $R_x(t-u)$ is positive definite, every λ_i is greater than zero.

4)- if $\phi_1(t)$ and $\phi_2(t)$ are eigenfunctions associated with the same eigenvalue λ, then $a_1\phi_1(t) + a_2\phi_2(t)$ is also eigenfunction associated with λ.

5)- the eigenfunctions corresponding to different eigenvalues are orthogonal, and there is at most a countably infinite set of eigenvalues and all are bounded.

iii - <u>Karhunen-Loéve Expansion of EEG signal</u>

EEG signal is assumed to be a stationary random signal in the interval of interest and have a spectra that can be written as a ratio of two polynomials in w^2.

$$S(w) = S_\delta(w) + S_\alpha(w) + S_\beta(w)$$

$$= \frac{N(w^2)}{D(w^2)} \qquad \text{(A-16)}$$

taking only the delta, alpha and beta components into account. Where $N(w^2)$ is a polynomial of order $q = 4$ in w^2 and $D(w^2)$ is a polynomial of order $p = 5$ in w^2. There are two different approaches to solve the problem [29]. The first is to convert the integral equation (A-15) to a differential equation whose solution can be found. The solution is then substituted back into the integral equation to satisfy the boundary conditions. This method lead to a differential equation of the 10th order which is very tedious to solve.

Knowing that the differential equation does not depend explicitly on T. This independence is true whenever the signal has a rational spectrum. Therefore the second method to solve for the eigenfunctions is to start with the integral equation

$$\lambda\ \phi(t) = \int_{-\infty}^{\infty} R_x(t - u)\ \phi(u)du, \qquad -\infty<t<+\infty \qquad \text{(A-17)}$$

By use of Fourier transforms a formal solution to this equation follows immediately:

$$\lambda\ \phi(jw) = S(w)\ \Phi(jw) = \frac{N(w^2)}{D(w^2)}\ \Phi(jw)$$

or (A-18)

$$\left[\lambda\ D(w^2) - N(w^2)\right]\ \Phi(jw) = 0$$

There are 2P = 10 homogeneous solutions to the differential equation corresponding to [A-18] for every value of λ (corresponding to the roots of the polynomial in the bracket). We denote them as $\phi_{hi}(t,\lambda)$, $i = 1, \ldots, 2P = 10$.

To find the solution, we substitute

$$\phi(t) = \sum_{i=1}^{10} a_i\ \phi_{hi}(t,\lambda) \qquad \text{(A-19)}$$

into the integral equation and solve for those values of λ and a_i that lead to a solution. In fact, there is no conceptual difficulties, but the procedure is tedious. This was the major reason which directed us to seek a solution using the real data in its discrete form.

It is intended here to shed some light on what form these solutions may take, which will be done by investigating the solution for any individual components, i.e., considering the EEG signal consists of only one component.

For any rhythmic component, Eqn. (A-18) could be rewritten as

$$w^4 - bw^2 + c = 0 \tag{A-20}$$

where the values of b and c depends upon the spectral parameters.

The roots of equation (A-20) are

$$w^2 = \frac{b \pm \sqrt{b^2 - 4c}}{2}$$
$$w = \sqrt{\frac{b \pm \sqrt{b^2 - 4c}}{2}} \tag{A-21}$$

we have 3 different possibilities according to the value of the eigenvalue λ in terms of the spectral parameters

(a) $\lambda = \dfrac{\sigma_i G_i - f_i H_i}{\pi(f_i^2 + \sigma_i^2)} \rightarrow \quad c = \text{Zero}$

$$\therefore w^2 = \frac{b \pm b}{2} = \text{Zero},\ b$$

$$\therefore w_{1,2} = \text{Zero} \quad , \quad w_{3,4} = \pm\sqrt{b}$$

$$\therefore \phi(t) = a_1 + a_2 e^{\sqrt{b}t} + a_3 e^{-\sqrt{b}t} \tag{A-22}$$

(b) $\lambda < \dfrac{\sigma_i G_i - f_i H_i}{\pi (f_i^2 + \sigma_i^2)} \rightarrow c < 0$

therefore we have one positive root for w^2 and another negative root, which is given by equation (A-21), this leads to the following solution

$$\phi(t) = a_1 e^{w_1 t} + a_2 e^{-w_1 t} + a_3 e^{j w_2 t} + a_4 e^{-j w_2 t} \qquad \text{(A-23)}$$

$$w_1 = \sqrt{\frac{b + \sqrt{b^2 - 4c}}{2}} \qquad w_2 = \sqrt{\frac{b - \sqrt{b^2 - 4c}}{2}}$$

(c) $\lambda > \dfrac{\sigma_i G_i - f_i H_i}{\pi (f_i^2 + \sigma_i^2)} \rightarrow c > 0$

in this case we have three possibilities

i) $b^2 > 4c$

Therefore the two roots for w^2 are real positive, which leads to the solution

$$\phi(t) = a_1 e^{w_1 t} + a_2 e^{-w_1 t} + a_3 e^{w_2 t} + a_4 e^{-w_2 t} \qquad \text{(A-24)}$$

ii) $b^2 < 4c$

Therefore equation (A-21) could be rewritten as follows

$$w^2 = \frac{b \pm j \sqrt{4c - b^2}}{2}$$

$$\therefore w_{1,2} = \pm c^{1/4} e^{+j\theta/2}$$

$$, \quad \theta = \tan^{-1} \frac{\sqrt{4c - b^2}}{b}$$

$$w_{3,4} = \pm c^{1/4} e^{-j\theta/2}$$

and $\phi(t)$ will be composed of damped sinusoids components.

iii) $4c = b^2$

$$\therefore w = \pm \sqrt{b}/2$$

$$\therefore \phi(t) = (a_1 + a_2 t)\, e^{w_1 t} + (a_3 + a_4 t) e^{w_2 t} \tag{A-25}$$

Fig. (A-2) illustrates all these possibilities.

To get the values of λ and a's that lead to a solution, substitute $\phi(t)$ into the integral equation and apply the boundary conditions.

For the δ component, the solution takes the form of sinusoids.[29]

So, we can imagine that the solution could take the form of sinusoids, damped sinusoids, exponentials, hyperpolic functions, damped linear function or a sum of these mentioned solutions such that the solution is a stable one and satisfies the boundary conditions. This agrees to some extent with the solutions we obtained through using the discrete real data, which are shown in the Chapter two Figures.

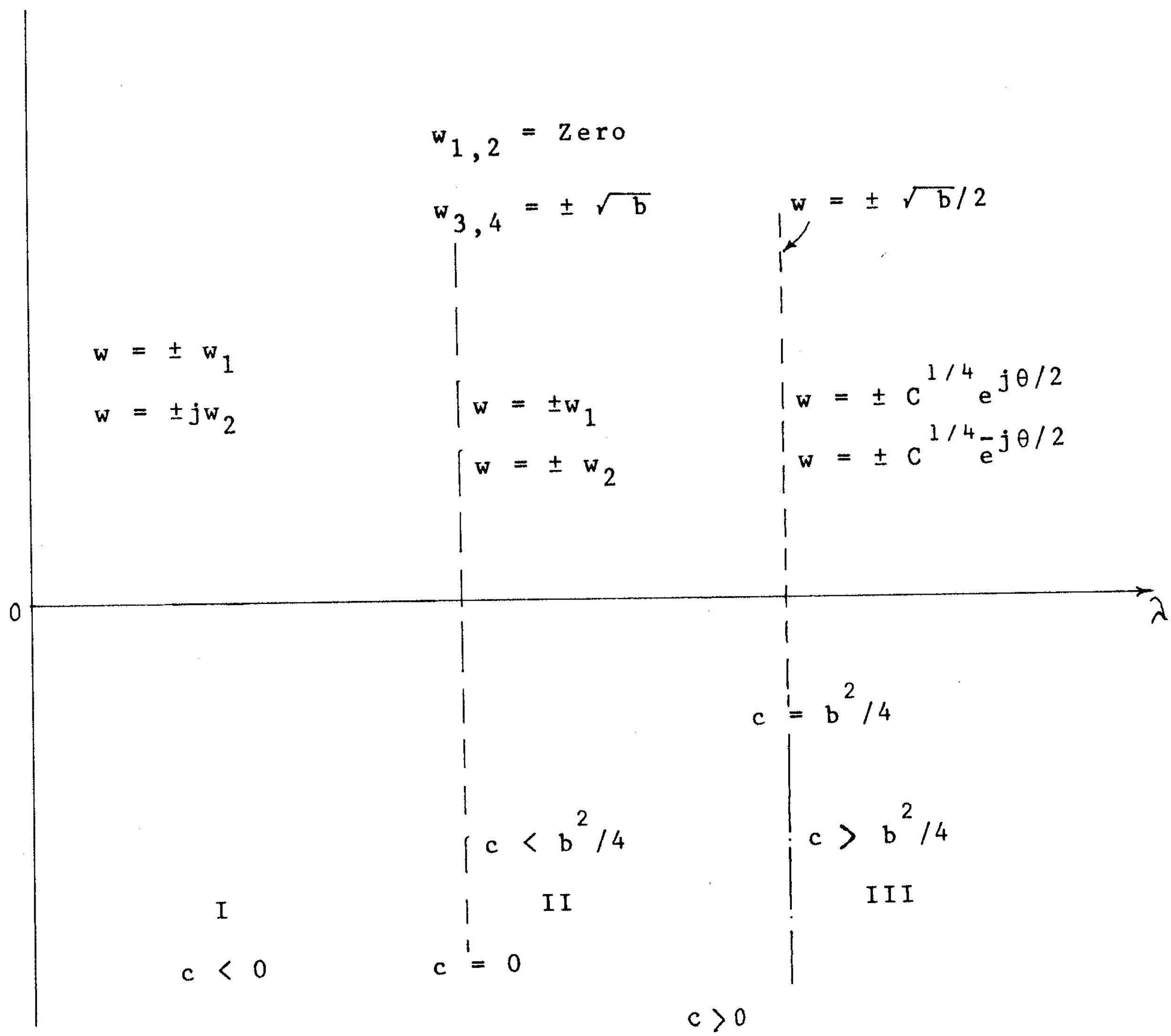


Figure (A-2) The different solution possibilities for the rhythmic component.

APPENDIX-B-

Computer Programs

1)- Program for solving the eigenvalue,eigenvector problem.

```
$JOB WATFIV
C PROGRAM TO SOLVE THE EIGENVALUE,EIGENVECTOR PROBLEM
C OF AN EEG OUTOCORRELATION MATRIX R(128X128).FIRST
C THE OUTOCORRELATION MATRIX HAS TO BE COMPUTED FOR
C 1 SEC. EEG EPOCH.
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,CHAN,UNIT,K,N,M,MK,IZ,ITMAX
      REAL*8 EEG(4,128),PEEG(4,128),WEEG(4,128)
      REAL*8 MEAN(4),VAR(4),ACF(4,10),AR(4,10)
      REAL*8 ACF1(4,10),A(128,128),T(128,128),EIGEN(128)
      REAL*8 ACF2(4,128),GSQR(4),PDS(4,61)
C
C READ THE RAW EEG DATA,SCALE IT, AND SEPARATE IT INTO 2 EPOCHS
C (THE TRAINING SET EPOCH AND THE PRECEEDING EPOCH) AND 4 CHANNELS
C
      CALL SCALSP(EEG,PEEG)
C
C SCALE THE RAW EEG DATA, CALCULATE THE MEAN AND VARIANCE,
C AND SUBTRACT THE MEAN FROM THE DATA
C
      CALL MEANVR(EEG,WEEG,MEAN,VAR)
C
C CALCULATION OF THE AUTOCORRELATION COEFFICIENTS (THE FIRST 10)
C
      CALL AUTCOR(WEEG,ACF,ACF1)
C
C CALCULATION OF THE 10TH ORDER AUTOREGRESSIVE MODEL
C
      CALL AUTREG(ACF,VAR,AR,GSQR)
C
C CALCULATION OF THE POWER DENSITY SPECTRUM USING THE
C AUTOREGRESSIVE MODEL
C
      CALL SPCDEN(AR,GSQR,PDS)
C
C CALCULATION OF THE 128 OUTOCORRELATION COEFFICIENTS BY
C TAKING THE INVERSE FOURIER TRANSFORM OF THE POWER DENSITY
C SPECTRUM
C
      CALL AOTCF(PDS,ACF2)
C
C FORM A SYMMETRIC OUTOCORRELATION MATRIX FILLING
C ONLY THE UPPER HALF USING THE 128 COEFFICIENTS
C
      CALL STORAG(ACF2,A)
C
C DEFINING THE PARAMETERS REQUIRED FOR SOLVING THE
C EIGENVALUE,EIGENVECTOR PROBLEM
C N=THE MARTIX RAW DIMENSION=THE COLUMN DIMENSION
C ITMAX=THE MAXIMUM NUMBER OF ITERATIONS
C EPS1,EPS2 ARE TOLERANCES CONTROLLING THE SOLUTION METHOD
C EPS3= TOLERANCE DETERMINING  THE TERMINATION
C OF THE ITERATIONS
C
      N=128
      ITMAX=50
      EPS1=1.00E-10
      EPS2=1.00E-10
      EPS3=1.00E-5
      CALL SALAM(A,N,ITMAX,EPS1,EPS2,EPS3,T,EIGEN)
C
C STORING THE EIGENVECTORS IN DATA SET NO. 15 IF IT
C IS NEEDED
C
```

```
      DO 5 J=1,128
      WRITE (15,4) (T(I,J),I=1,N)
4     FORMAT (7X,6F11.7)
5     CONTINUE
      STOP
      END
C
C SUBROUTINE SCALSP
C SUBROUTINE FOR SCALING THE EEG, AND SEPARATING IT INTO
C 2 EPOCHS AND 4 CHANNELS
C RAWDAT - RAW EEG FROM THE AD CONVERTERS
C SEEG - EEG DATA FOR THE TRAINING SET
C SPEEG - EEG DATA FOR THE PRECEEDING EPOCH
      SUBROUTINE SCALSP(SEEG,SPEEG)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 SEEG(4,128),SPEEG(4,128)
      INTEGER RAWDAT(1024),I,J,CHAN
      READ(9,10) (RAWDAT(I),I=1,1024)
10    FORMAT(16I4)
      J=1
      DO 30 I=1,128
         DO 20 CHAN=1,4
            SPEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
            J=J+1
20       CONTINUE
30    CONTINUE
      J=513
      DO 50 I=1,128
         DO 40 CHAN=1,4
            SEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
            J=J+1
40       CONTINUE
50    CONTINUE
      RETURN
      END
C
C SUBROUTINE MEANVR
C SUBROUTINE FOR CALCULATING THE MEAN AND VARIANCE OF THE EEG
C FROM A SINGLE EPOCH
C THE EEG DATA ARE ALSO SCALED AND THE MEAN SUBTRACTED OUT
C SMEAN - MEAN OF THE EEG
C SVAR - VARIANCE OF THE EEG
C SEEG - ZERO-MEAN EEG DATA
C
      SUBROUTINE MEANVR(SEEG,SWEEG,SMEAN,SVAR)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,CHAN
      REAL*8 SEEG(4,128),SWEEG(4,128),SMEAN(4),SVAR(4),HAM
      DO 5 CHAN=1,4
         SMEAN(CHAN)=0
5     CONTINUE
      DO 20 I=1,128
         DO 10 CHAN=1,4
            SMEAN(CHAN)=SMEAN(CHAN)+SEEG(CHAN,I)
10       CONTINUE
20    CONTINUE
C
C SUBTRACTION OF THE MEAN AND HAMMING WINDOWING OF THE EEG DATA
C
      DO 40 CHAN=1,4
         SMEAN(CHAN)=SMEAN(CHAN)/128
         SVAR(CHAN)=0.0
         DO 30 I=1,128
            SEEG(CHAN,I)=SEEG(CHAN,I)-SMEAN(CHAN)
```

```
             HAM=.54-.46*COS(2*3.14157*(I-1)/127)
             SWEEG(CHAN,I)=SEEG(CHAN,I)*HAM
             SVAR(CHAN)=SVAR(CHAN)+SWEEG(CHAN,I)**2
30        CONTINUE
40      CONTINUE
        RETURN
        END
C
C SUBROUTINE AUTCOR
C SUBROUTINE FOR CALCULATING THE AUTOCORRELATION COEFFICIENTS
C R(1) THROUGH R(10) FOR THE EEG DATA.
C THE CALCULATION IS PERFORMED USING KENDALL'S METHOD
C SACF - AUTOCORRELATION COEFFICIENT VECTOR
C
        SUBROUTINE AUTCOR(SEEG,SACF,SACF1)
        IMPLICIT REAL*8 (A-H,O-Z)
        INTEGER I,J,K,M,Y,Z,CHAN,EVOD
        REAL*8 SEEG(4,128),SACF(4,10),SACF1(4,10)
        REAL*8 A(128),B(128)
        DO 90 CHAN=1,4
C
C CALCULATION OF THE VALUES OF A(K) AND B(K)
C
           A(128)=0
           A(127)=0
           B(128)=0
           B(127)=0
           EVOD=0
           DO 20 I=1,126
              K=127-I
              IF (EVOD .EQ. 0) GOTO 10
C
C CALCULATION OF THE COEFFICIENTS FOR ODD K
C
              A(K)=A(K+1)
              B(K)=B(K+2)+SEEG(CHAN,K+1)*SEEG(CHAN,K+2)
              EVOD=0
              GOTO 20
10            CONTINUE
C
C CALCULATION OF THE COEFFICIENTS FOR EVEN K
C
              A(K)=A(K+2)+SEEG(CHAN,127-K)*SEEG(CHAN,128-K)
              B(K)=B(K+2)+SEEG(CHAN,K+1)*SEEG(CHAN,K+2)
              EVOD=1
20         CONTINUE
           EVOD=1
           DO 60 K=1,10
              SACF(CHAN,K)=0
              IF (EVOD .EQ. 0) GOTO 40
C
C CALCULATION FOR ODD K
C
              Y=(128-K-1)/2
              DO 30 I=1,Y
                 M=I-1
                 Z=2*M+1
                 SACF(CHAN,K)=SACF(CHAN,K)+(SEEG(CHAN,Z)+SEEG(CHAN,Z+K+1)
     *           )*(SEEG(CHAN,Z+1)+SEEG(CHAN,Z+K))
30            CONTINUE
              SACF(CHAN,K)=SACF(CHAN,K)-A(K)-B(K)+SEEG(CHAN,128-K)*
     *        SEEG(CHAN,128)
              EVOD=0
              GOTO 60
```

```
C
C CALCULATION FOR EVEN K
C
40            CONTINUE
              Y=(128-K)/2
              DO 50 I=1,Y
                 M=I-1
                 Z=2*M+1
                 SACF(CHAN,K)=SACF(CHAN,K)+(SEEG(CHAN,Z)+SEEG(CHAN,Z+K+1)
     *           )*(SEEG(CHAN,Z+1)+SEEG(CHAN,Z+K))
50            CONTINUE
              SACF(CHAN,K)=SACF(CHAN,K)-A(K)-B(K)
              EVOD=1
60         CONTINUE
C
C CALCULATION OF ACF HAS BEEN COMPLETED
C
C CALCULATION OF THE ACF USING THE PREVIOUS METHOD
           DO 80 K=1,10
              SACF1(CHAN,K)=0
              Z=128-K
              DO 70 J=1,Z
                 SACF1(CHAN,K)=SACF1(CHAN,K)+SEEG(CHAN,J)* SEEG(CHAN,J
     *           +K)
70            CONTINUE
80         CONTINUE
90      CONTINUE
        RETURN
        END
C
C SUBROUTINE AUTREG
C SUBROUTINE FOR CALCULATING THE 10TH ORDER AUTOREGRESSIVE MODEL
C COEFFICIENTS FOR THE 1 SECOND EEG EPOCHS
C THE CALCULATION IS MADE USING THE LEVINSON-DURBIN PROCEEDURE
C SAR - AUTOREGRESSIVE COEFFICIENT VECTOR
C SGSQR - SQUARED MAGNITUDE OF THE GAIN G
C
        SUBROUTINE AUTREG(SACF,SVAR,SAR,SGSQR)
        IMPLICIT REAL*8 (A-H,O-Z)
        INTEGER I,J,K,CHAN,M
        REAL*8 KI(10),AI(10,10),EI(10)
        REAL*8 SACF(4,10),SVAR(4),SAR(4,10),SGSQR(4)
        DO 60 CHAN=1,4
           KI(1)=-SACF(CHAN,1)/SVAR(CHAN)
           AI(1,1)=KI(1)
           EI(1)=(1-KI(1)*KI(1))*SVAR(CHAN)
           DO 30 I=2,10
              KI(I)=0
              M=I-1
              DO 10 J=1,M
                 KI(I)=KI(I)+AI(J,I-1)*SACF(CHAN,I-J)
10            CONTINUE
              KI(I)=-(SACF(CHAN,I)+KI(I))/EI(I-1)
              AI(I,I)=KI(I)
              DO 20  J=1,M
                 AI(J,I)=AI(J,I-1)+KI(I)*AI(I-J,I-1)
20            CONTINUE
              EI(I)=(1-KI(I)*KI(I))*EI(I-1)
30         CONTINUE
           DO 40 J=1,10
              SAR(CHAN,J)=AI(J,10)
40         CONTINUE
C
C CALCULATION OF THE GAIN G FOR THE AR MODEL
```

```
C
          SGSQR(CHAN)=SVAR(CHAN)
          DO 50 K=1,10
             SGSQR(CHAN)=SGSQR(CHAN)+SAR(CHAN,K)*SACF(CHAN,K)
50        CONTINUE
60     CONTINUE
       RETURN
       END
C
C  SUBROUTINE SPCDEN
C  SUBROUTINE FOR CALCULATING THE POWER DENSITY SPECTRUM
C  ESTIMATE FOR THE 1 SECOND EEG EPOCHS
C  THE SPECTRUM IS CALCULATED FOR FREQUENCIES FROM F=0 HZ TO 30 HZ
C  IN STEP OF .5 HZ USING THE AUTOREGRESSIVE SPECTRAL ESTIMATOR
C  SPDS - POWER DENSITY SPECTRUM ESTIMATE
C
       SUBROUTINE SPCDEN(SAR,SGSQR,SPDS)
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER I,J,CHAN
       REAL*8 W,F,RMAG,IMAG
       REAL*8 SAR(4,10),SGSQR(4),SPDS(4,61)
       W=2*3.14159/128
       DO 30 CHAN=1,4
          F=0.0
          DO 20 J=1,61
             RMAG=1
             IMAG=0
             DO 10 I=1,10
                RMAG=RMAG+SAR(CHAN,I)*DCOS(W*I*F)
                IMAG=IMAG+SAR(CHAN,I)*DSIN(W*I*F)
10           CONTINUE
             SPDS(CHAN,J)=SGSQR(CHAN)/(RMAG*RMAG+IMAG*IMAG)
             F=F+.5
20        CONTINUE
30     CONTINUE
       RETURN
       END
C
C  SUBROUTINE ADTCF TO CALCULATE THE ACF.FROM THE PSD
C  USING INVERSE FOURIER TRANSFORM
C
       SUBROUTINE ADTCF(PDS,ACF2)
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 ACF2(4,128),PDS(4,61),ACF11(4,128)
       CHAN=1
       NF=127
       DELTA=1.0
       DO 20 I=1,128
          I1=I-1
          C=COS(3.14157*I1/NF)
          AF0=0.0
          AF1=0.0
          DO 10 K=2,61
             KK=63-K
             AF2=2*C*AF1-AF0+PDS(CHAN,KK)
             AF0=AF1
             AF1=AF2
10        CONTINUE
          ACF11(CHAN,I)=2*DELTA*(PDS(CHAN,1)+2*(AF1*C-AF0))
          ACF2(CHAN,I)=ACF11(CHAN,I)/ACF11(CHAN,1)
20     CONTINUE
       WRITE (6,4) (ACF2(CHAN,I),I=1,128)
4      FORMAT (8(F12.6,1X))
       RETURN
```

```
      END
C
C SUBROUTINE STORAG
C THIS SUBROUTINE IS FORMING A SYMMETRIC AUTOCORRELATION
C MATRIX FILLING ONLY ITS UPPER HALF USING IN THAT THE
C 128 COEFFICIENTS
C
      SUBROUTINE STORAG(ACF2,A)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 ACF2(4,128),A(128,128)
      INTEGER CHAN,I,K,J,N
      CHAN=1
      DO 10 I=1,128
         DO 15 J=1,128
            A(I,J)=0.0
15       CONTINUE
10    CONTINUE
      DO 30 I=1,128
         DO 20 J=I,128
            K=J-I+1
            A(I,J)=ACF2(CHAN,K)
20       CONTINUE
30    CONTINUE
      RETURN
      END
C
C SUBROUTINE SALAM
C TO COMPUTE THE EIGENVALUES AND THE EIGENVECTORS OF
C A SYMMETRIC OUTOCORRELATION MATRIX BY JACOBI'S METHOD.
C ONLY THE UPPER TRIANGULAR PART OF THE MATRIX IS USED.
C T IS AN NXN ORTHOGONAL MATRIX,BEING THE PRODUCT OF THE
C SEQUENCE OF TRANSFORMATION MATRECIES USED TO ANNIHILATE
C SUCCESSIVELY THE OFF-DIAGONAL ELEMENTS OF A ,AND
C CONSEQUENTLY TO REDUCE A TITERATIVELY TO NEAR-DIAGONAL
C FORM.  SIGMA1 AND SIGMA2 ARE THE VALUES OF SIGNAL A(I,I)**2
C BEFORE AND AFTER ONE COMPLETE ITERATION. SINCE ANNIHILTION
C OF AN ELEMENT MAY CAUSE ANOTHER ALREADY-ANNIHILATED ELEMENT
C TO ASSUME A NON-ZERO VALUE.THE ITERATIVE PROCESS IS REPEATED
C UNTIL THE CONVERGENCE TEST IS PASSED(1-SIGMA1/SIGMA2) LESS
C THAN (EPS3) OR ITMAX IS EXCEEDED. EPS1 IS COMPARED WITH
C (A(I,I)-A(J,J)) TO DETERMINE THE METHODE OF COMUTING THE
C COSINE AND SINE OF THE ROTATION ANGLE. WHEN A HAS BEEN REDUCED
C TO DIAGONAL FORM, THE EIGENVALUES WILL BE FOUND IN THE
C DIAGONAL AND ARE SAVED IN EIGEN(I) ARRAY. THE ASSOCIATED
C EIGENVECTORS ARE IN CORRESPONDING COLUMNS OF THE FINAL
C TRANSFORMATION MATRIX T.
C
      SUBROUTINE SALAM(A,N,ITMAX,EPS1,EPS2,EPS3,T,EIGEN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(128,128),T(128,128),AIK(128),EIGEN(128)
C
C SET UP INITIAL MATRIX T,COMPUTE SIGMA1 AND S
C
      DO 4 I=1,N
         DO 2 J=1,N
            T(I,J)=0.0
2        CONTINUE
4     CONTINUE
      NM1=N-1
      WRITE (6,200) N,ITMAX,EPS1,EPS2,EPS3
      SIGMA1= 0.0
      OFFDSQ= 0.0
      DO 5 I=1,N
         SIGMA1=SIGMA1+A(I,I)**2
```

```
         T(I,I)=1.0
         IP1=I+1
         IF (I.GE.N) GO TO 6
         DO 7 J=IP1,N
            OFFDSQ=OFFDSQ+A(I,J)**2
7        CONTINUE
5     CONTINUE
6     S=2.0*OFFDSQ+SIGMA1
C
C BEGIN JACOBI ITERATION
C
      DO 26 ITER =1,ITMAX
         DO 20 I=1,NM1
            IP1=I+1
            DO 22 J=IP1,N
               Q=DABS(A(I,I)-A(J,J))
C
C COMPUTE SINE AND COSINE OF ROTATION ANGLE ..
C
               IF (Q .LE. EPS1) GO TO 9
               IF (DABS(A(I,J)) .LE.EPS2) GO TO 22
               P =2.0*A(I,J)*Q/(A(I,I)-A(J,J))
               SPQ =DSQRT(P*P+Q*Q)
               CSA =DSQRT((1.0+Q/SPQ)/2.0)
               SNA =P/(2.0*CSA*SPQ)
               GO TO 10
9              CSA =1.0/DSQRT(2.0D0)
               SNA =CSA
10             CONTINUE
C
C UPDATE COLUMNS I AND J OF T -
C
               DO 11 K=1,N
                  HOLDKI =T(K,I)
                  T(K,I)=HOLDKI*CSA+T(K,J)*SNA
                  T(K,J)=HOLDKI*SNA-T(K,J)*CSA
11             CONTINUE
C
C COMPUTE NEW ELEMENTS OF A IN COLUMNS I,J
C
               DO 16 K =I,N
                  IF (K .GT.J) GO TO 15
                  AIK(K)=A(I,K)
                  A(I,K)=CSA*AIK(K)+SNA*A(K,J)
                  IF (K.NE.J) GO TO 14
                  A(J,K) =SNA*AIK(K)-CSA*A(J,K)
14                GO TO 16
15                HOLDIK =A(I,K)
                  A(I,K)=CSA*HOLDIK+SNA*A(J,K)
                  A(J,K)=SNA*HOLDIK-CSA*A(J,K)
16             CONTINUE
C
C COMPUTE NEW ELEMENTS OF A IN COLUMNS I,J
C
               AIK(J)=SNA*AIK(I)-CSA*AIK(J)
C
C WHEN K IS LARGER THAN I
C
               DO 19 K=1,J
                  IF (K.LE.I) GO TO 18
                  A(K,J)=SNA*AIK(K)-CSA*A(K,J)
                  GO TO 19
18                HOLDKI =A(K,I)
                  A(K,I)=CSA*HOLDKI+SNA*A(K,J)
```

```
                  A(K,J)=SNA*HOLDKI-CSA*A(K,J)
19             CONTINUE
               A(I,J)=0.0
22            CONTINUE
20         CONTINUE
C
C FIND SIGMA2 FOR TRANSFORMED A AND TEST FOR CONVER-
C GENCE.....
C
           SIGMA2=0.0
           DO 21 I=1,N
              EIGEN(I) =A(I,I)
              SIGMA2=SIGMA2+EIGEN(I)**2
21         CONTINUE
           IF (1.0-SIGMA1/SIGMA2.GE.EPS3) GO TO 25
           CALL REARR (T,EIGEN,T,EIGEN)
           WRITE (6,204) ITER,SIGMA2,S,N
           WRITE (6,201) (EIGEN(I),I=1,N)
           WRITE (6,206)
           DO 24 J=1,N
           WRITE (6,201) (T(I,J),I=1,N)
24         CONTINUE
           GO TO 208
25         WRITE (6,202) ITER,SIGMA1,SIGMA2
           SIGMA1=SIGMA2
26      CONTINUE
C
C IF ITER EXCEEDS ITMAX NO CONVERGENCE...
C
        WRITE (6,203) ITER,S,SIGMA1,SIGMA2
        DO 27 I=1,N
           WRITE (6,201) (A(I,J),J=1,N)
27      CONTINUE
        WRITE (6,207)
        DO 28 I=1,N
           WRITE (6,201) (T(I,J),J=1,N)
28      CONTINUE
        GO TO 208
C
C   FORMATS
C
200          FORMAT (1H1,4X,65H DETERMINATION OF EIGENVALUES BY
      *   JACOBI'S METHOD,WITH/1H0,6X,10H N          =,I4/
      *   7X,10H ITMAX   = ,I4/ 7X,10H EPS1      =,E12.3/
      *   7X,10H EPS2      =,E12.3/ 7X,10H EPS3       =,E12.3/1H )
201        FORMAT (7X, 10F11.7)
202        FORMAT (1H0,6X,10H ITER       =,I5,10X,10H SIGMA1   =,F10.5,
      *    10X,10H SIGMA2   =,F10.5)
203        FORMAT (1H0,4X,21H NO CONVERGENCE, WITH// 7X,
      *    10H ITER       =,I5,5X,10H  S         =,F10.5,5X,10H SIGMA1   =,
      *    F10.5,5X,10H SIGMA2   =,F10.5/1H )
204        FORMAT (1H0,4X, 31H CONVERGENCE HAS OCCURRED, WITH/1H0,
      *   6X,10H ITER       =,I5,5X,10H SIGMA2   =,F10.5,5X,
      *   10H  S       =,F10.5,5X,10H  N        =,I5/1H0,
      *   4X,36H EIGENVALUES EIGEN(1)...EIGEN(N) ARE/1H )
206       FORMAT (1H0,4X, 71H EIGENVECTORS ARE IN CORRESPONDING RAWS
      *  OF THE FOLLOWING T MATRIX/1H )
207       FORMAT (1H0,4X, 24H THE CURRENT T MATRIX IS/1H )
C
208       RETURN
          END
C SUBROUTINE REARRANGE
C TO ARRANGE THE EIGENVALUES AND THE CORRESPONDING
C EIGENVALUES IN DESCENDING ORDER
```

```
C
      SUBROUTINE REARR(TT,EIG,T,EIGEN)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(128,128),EIGEN(128),TT(128,128),EIG(128)
      N=128
      X=0.0
      Y=0.0
      Z=0.0
      S=0.0
      NN=N-1
      DO 70 J=1,NN
         NNN=N-J
         DO 60 I=1,NNN
            IF(EIG(J).GT.EIG(J+I)) GO TO 60
            X=EIG(J+I)
            Y=EIG(J)
            EIGEN(J)=X
            EIGEN(J+I)=Y
            DO 50 II=1,N
               Z=TT(II,J)
               S=TT(II,J+I)
               T(II,J)=S
               T(II,J+I)=Z
50          CONTINUE
60       CONTINUE
70    CONTINUE
      RETURN
      END
$ENTRY
/*
```

THIS program is associated with the previous one,

-program to reconstruct the EEG signal.

```
$JOB WATFIV
C
C PROGRAM TO RECONSTRUCT THE SIGNAL USING THE KLE RANDONM COEFFI.
C AND TO COMPUTE THE ERROR RESULTED FROM THE APPROXIMATION
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,CHAN
      REAL*8 EEG(4,128),PEEG(4,128),WEEG(4,128),MEAN(4),VAR(4)
      REAL*8 T(128,128),ERROR(128)
C
C READ THE RAW EEG DATA,SCALE IT, AND SEPARATE IT INTO 2 EPOCHS
C (THE TRAINING SET EPOCH AND THE PRECEEDING EPOCH) AND 4 CHANNELS
C
      CALL SCALSP(EEG,PEEG)
C
C SCALE THE RAW EEG DATA, CALCULATE THE MEAN AND VARIANCE,
C AND SUBTRACT THE MEAN FROM THE DATA
C
      CALL MEANVR(EEG,WEEG,MEAN,VAR)
      N=128
      DO 5 J=1,128
      READ (11,4) (T(I,J),I=1,N)
4     FORMAT (7X,6F11.7)
5     CONTINUE
C RECONSTRUCTING THE EEG SIGNAL AND COMPUTING THE ERROR
C VECTOR AND THE TOTAL MEAN SQUARE ERROR OF THE APPROXIMATION
C
      CALL GAMIL(T,N,WEEG,ERROR)
      STOP
      END
C SUBROUTINE SCALSP
C SUBROUTINE FOR SCALING THE EEG, AND SEPARATING IT INTO
C 2 EPOCHS AND 4 CHANNELS
C RAWDAT - RAW EEG FROM THE AD CONVERTERS
C SEEG - EEG DATA FOR THE TRAINING SET
C SPEEG - EEG DATA FOR THE PRECEEDING EPOCH
      SUBROUTINE SCALSP(SEEG,SPEEG)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 SEEG(4,128),SPEEG(4,128)
      INTEGER RAWDAT(1024),I,J,CHAN
      READ(10,10) (RAWDAT(I),I=1,1024)
10    FORMAT(16I4)
      J=1
      DO 30 I=1,128
         DO 20 CHAN=1,4
            SPEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
         J=J+1
20       CONTINUE
30    CONTINUE
      J=513
      DO 50 I=1,128
         DO 40 CHAN=1,4
         SEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
         J=J+1
40       CONTINUE
50    CONTINUE
      RETURN
      END
C SUBROUTINE MEANVR
C SUBROUTINE FOR CALCULATING THE MEAN AND VARIANCE OF THE EEG
C FROM A SINGLE EPOCH
C THE EEG DATA ARE ALSO SCALED AND THE MEAN SUBTRACTED OUT
C SMEAN - MEAN OF THE EEG
C SVAR - VARIANCE OF THE EEG
```

```
C SWEEG - ZERO-MEAN EEG DATA
C
      SUBROUTINE MEANVR(SEEG,SWEEG,SMEAN,SVAR)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER I,CHAN
      REAL*8 SEEG(4,128),SWEEG(4,128),SMEAN(4),SVAR(4),HAM
      DO 5 CHAN=1,4
         SMEAN(CHAN)=0
5     CONTINUE
      DO 20 I=1,128
         DO 10 CHAN=1,4
            SMEAN(CHAN)=SMEAN(CHAN)+SEEG(CHAN,I)
10       CONTINUE
20    CONTINUE
C
C SUBTRACTION OF THE MEAN
C
      DO 40 CHAN=1,4
         SMEAN(CHAN)=SMEAN(CHAN)/128
         SVAR(CHAN)=0.0
         DO 30 I=1,128
            SWEEG(CHAN,I)=SEEG(CHAN,I)-SMEAN(CHAN)
            SVAR(CHAN)=SVAR(CHAN)+SWEEG(CHAN,I)**2
30       CONTINUE
40    CONTINUE
      RETURN
      END
C
C SUBROUTINE GAMIL FOR CALCULATING THE ERROR
C AND RECONSTRUCTING THE SIGNAL
C
      SUBROUTINE GAMIL(T,N,WEEG,ERROR)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(128,128),WEEG(4,128),ERROR(128)
      DIMENSION SOME(128),AIK(128),ERRPER(128)
      INTEGER I,K,J,CHAN,IER
      CHAN=1
      DO 10 I=1,N
         AIK(I)=0.0
10    CONTINUE
      DO 20 I=1,N
         K=I
         DO 30 J=1,N
            AIK(I)=AIK(I)+T(J,K)*WEEG(CHAN,J)
30       CONTINUE
20    CONTINUE
      PRINT *,(AIK(I),I=1,N)
C
C GENERATE THE APPROXIMATED VECTOR DATA
C
      DO 40 J=1,N
         SOME(J)=0.0
40    CONTINUE
      DO 60 I=1,N
         DO 50 J=1,50
            SOME(I)=SOME(I)+T(I,J)*AIK(J)
50       CONTINUE
60    CONTINUE
C
C CALCULATE THE ERROR VECTOR
C
        ERRMS=0.0
        TENG =0.0
        DO 70 I=1,N
```

```
          ERROR(I)=WEEG(CHAN,I)-SOME(I)
          ERRPER(I)=ERROR(I)/WEEG(CHAN,I)
          ERRMS =ERRMS+ERROR(I)**2
          TENG =TENG+WEEG(CHAN,I)** 2
70      CONTINUE
        ERMSPC=(ERRMS/TENG)*100
        WRITE (6,100)
        WRITE (6,80) (ERROR(I),I=1,N)
        WRITE (6,110)
        WRITE (6,80) (ERRPER(I),I=1,N)
        WRITE (6,90) ERRMS,TENG,ERMSPC
80      FORMAT (4(E14.6,1X))
90      FORMAT (1H0,6X,10H ERRMS    =,E14.6/
     *7X,10H TENG     =,E14.6/ 7X,10H ERMSPC   =,F12.6/1H )
100     FORMAT (1H1,4X,20H THE ERROR VECTOR IS/1H )
110     FORMAT (1H1,4X,31H THE PERCENTAGE ERROR VECTOR IS/1H )
        RETURN
        END
/*
```

2)- Program to implement the FAN INTERPOLATOR.

```
C PROGRAM TO IMPLEMENT THE FAN INTERPOLATOR,  TO
C RECONSTRUCT THE SIGNAL FROM THE STORED PARAMETERS
C AND TO CALCULATE THE RESULTED ERROR WITH THE TOTAL
C MEAN SQUARE ERROR OF THE APPROXIMATION .
C FIRST ,THE SIGNAL HAS TO BE SCALED AND TO BE
C OF A ZERO MEAN
      INTEGER I,CHAN,UNIT,K,N,M(128),MM
      REAL8 EEG(4,128),PEEG(4,128),WEEG(4,128),ERRPER(128)
      REAL8 MEAN(4),VAR(4),A(128),APP(128),ERROR(128)
C
C READ THE RAW EEG DATA,SCALE IT, AND SEPARATE IT INTO 2 EPOCHS
C (THE TRAINING SET EPOCH AND THE PRECEEDING EPOCH) AND 4 CHANNELS
C
      CALL SCALSP(EEG,PEEG)
C
C SCALE THE RAW EEG DATA, CALCULATE THE MEAN AND VARIANCE,
C AND SUBTRACT THE MEAN FROM THE DATA
C
      CALL MEANVR(EEG,WEEG,MEAN,VAR)
C
C CALCULATE THE STRAIGHT LINE PARMETERES TO BE STORED.
C
      CALL STRINT(WEEG,A,M)
C
C RECONSTRUCT THE SIGNAL,FIND THE APPROXIMATED SIGNAL.
C
      CALL RECONS(A,M,APP)
C
C CALCULATE THE ERROR VECTOR ,AND THE MEAN SQUARE ERROR
C
      N=128
      CHAN=1
      ERRMS=0.0
      TENG =0.0
      DO 70 I=1,N
         ERROR(I)=WEEG(CHAN,I)-APP(I)
         ERRPER(I)=ERROR(I)/WEEG(CHAN,I)
         ERRMS =ERRMS+ERROR(I)**2
         TENG =TENG+WEEG(CHAN,I)**2
70    CONTINUE
      ERMSPC=(ERRMS/TENG)*100
      WRITE (6,100)
      WRITE (6,80) (ERROR(I),I=1,N)
      WRITE (6,110)
      WRITE (6,80) (ERRPER(I),I=1,N)
      WRITE (6,90) ERRMS,TENG,ERMSPC
80    FORMAT (8(E14.6,1X))
90    FORMAT (1H0,6X,10H ERRMS    =,E14.6/
     *7X,10H TENG     =,E14.6/ 7X,10H ERMSPC   =,F12.6/1H )
100   FORMAT (1H1,4X,20H THE ERROR VECTOR IS/1H )
110   FORMAT (1H1,4X,31H THE PERCENTAGE ERROR VECTOR IS/1H )
      STOP
      END
C
C SUBROUTINE SCALSP
C SUBROUTINE FOR SCALING THE EEG, AND SEPARATING IT INTO
C 2 EPOCHS AND 4 CHANNELS
C RAWDAT - RAW EEG FROM THE AD CONVERTERS
C SEEG - EEG DATA FOR THE TRAINING SET
C SPEEG - EEG DATA FOR THE PRECEEDING EPOCH
      SUBROUTINE SCALSP(SEEG,SPEEG)
      REAL8 SEEG(4,128),SPEEG(4,128)
      INTEGER RAWDAT(1024),I,J,CHAN
      READ(10,10) (RAWDAT(I),I=1,1024)
```

```
10     FORMAT(16I4)
       J=1
       DO 30 I=1,128
           DO 20 CHAN=1,4
               SPEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
             J=J+1
20         CONTINUE
30     CONTINUE
       J=513
       DO 50 I=1,128
           DO 40 CHAN=1,4
             SEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
             J=J+1
40         CONTINUE
50     CONTINUE
       RETURN
       END
C
C SUBROUTINE MEANVR
C SUBROUTINE FOR CALCULATING THE MEAN AND VARIANCE OF THE EEG
C FROM A SINGLE EPOCH
C THE EEG DATA ARE ALSO SCALED AND THE MEAN SUBTRACTED OUT
C SMEAN - MEAN OF THE EEG
C SVAR - VARIANCE OF THE EEG
C SWEEG - ZERO-MEAN EEG DATA
C
       SUBROUTINE MEANVR(SEEG,SWEEG,SMEAN,SVAR)
       INTEGER I,CHAN
       REAL8 SEEG(4,128),SWEEG(4,128),SMEAN(4),SVAR(4)
       DO 5 CHAN=1,4
           SMEAN(CHAN)=0
5      CONTINUE
       DO 20 I=1,128
           DO 10 CHAN=1,4
               SMEAN(CHAN)=SMEAN(CHAN)+SEEG(CHAN,I)
10         CONTINUE
20     CONTINUE
C
C SUBTRACTION OF THE MEAN
C
       DO 40 CHAN=1,4
           SMEAN(CHAN)=SMEAN(CHAN)/128
           SVAR(CHAN)=0.0
           DO 30 I=1,128
             SWEEG(CHAN,I)=SEEG(CHAN,I)-SMEAN(CHAN)
              SVAR(CHAN)=SVAR(CHAN)+SWEEG(CHAN,I)**2
30         CONTINUE
40     CONTINUE
       RETURN
       END
C
C SUBROUTINE STRINT
C TO CALCULATE THE PARMETERES OF THE LINE SEGMENTS
C WHICH APPROXIMATES THE EEG SIGNAL EPOCH.
C A= A VECTOR USED TO STORE THE END POINTS OF THE LINE.
C M= A VECTOR USED TO STORE THE NUMBER OF SAMPLES FOR EACH INDVIDUA
C LINE
C
       SUBROUTINE STRINT(WEEG,A,M)
       DIMENSION A(128),M(128),WEEG(4,128)
       CHAN=1
C AS AN EXAMPLE THE TOLERANCE IS SET TO HAVE CERTAIN VALUE
       DELTA=80.00E-03
C
```

```
      J=1
      J1=J+1
      J2=J
      J3=J-1
      II=J+1
      N=128
      DO 20 I=2,N
          II=II+1
          IF(II.GT.N) GO TO 11
          S=WEEG(CHAN,II)-WEEG(CHAN,J)
          K=J+1
          DO 10 K1=K,I
              K2=K1-J
              K3=II-J
              SS=WEEG(CHAN,K1)-((K2*S)/K3)
              IF(DABS(SS).LE.DELTA) GO TO 10
              IF(I.EQ.N) M(1)=2*J2
              A(J2)=WEEG(CHAN,J)
              A(J2+1)=WEEG(CHAN,I)
              M(J1)=I-J3
              J=I
              J1=J1+1
              J2=J2+1
              II=I+1
              J3=I
              GO TO 20
10        CONTINUE
11        IF(I.EQ.N)  GO TO 21
20    CONTINUE
21    M(1)=2*J1
      A(J2)=WEEG(CHAN,J)
      A(J2+1)=WEEG(CHAN,I)
      M(J1)=I-J3
      WRITE (6,30)    (M(I),I=1,J1)
      WRITE (6,31)    (A(I),I=1,J1),(WEEG(CHAN,I),I=1,N)
      WRITE (6,32) DELTA
30    FORMAT (1X,10(I3,1X))
31    FORMAT (8(F12.6,1X))
32    FORMAT (E14.6)
      RETURN
      END
C
C SUBROUTINE RECONS
C TO RECONSTRUCT THE EEG SIGNAL FROM TH_ LINE SEGMENTS
C PARAMETERS
C APP =THE APPROXIMATED (RECONSTRUCTED) SIGNAL
C
      SUBROUTINE RECONS(A,M,APP)
      DIMENSION A(128),M(128),APP(128)
      N=M(1)/2
      K2=1
      K1=0
      DO 20 I=2,N
          K=M(I)+K1
          SS=A(I-1)
          BB=A(I)
          ZZ=BB-SS
          DO 10 J=K2,K
              K3=J-K2
              K4=K-K2
              APP(J)=SS+((K3*ZZ)/K4)
10        CONTINUE
          K1=K
          K2=K
```

```
20      CONTINUE
        WRITE(6,30) (APP(J),J=1,128)
30      FORMAT (8(F12.6,1X))
        RETURN
        END
$ENTRY
/*
```

3)- Program for implementing the least-square error interpolator.

```
$JOB WATFIV
C PROGRAM TO IMPLEMENT A FIRST ORDER INTERPOLATOR USING
C THE LEAST SQUARE ERROR CRITERION IN FITTING THE LINES.
C THEN RECONSTRUCT THE SIGNAL FROM THE STRAIGHT LINE PARAMETERS
C AND CALCULATE THE RESULTED ERROR AND THE TOTAL MEAN SQUARE ERROR
C OF THE APPROXIMATION.
C FIRST THE DATA HAS TO BE SCALED AND TO BE OF ZERO MEAN.
      INTEGER I,CHAN,K,N,M(256),MM
      REAL8 EEG(4,128),PEEG(4,128),WEEG(4,128),ERRPER(128)
      REAL8 MEAN(4),VAR(4),A(256),APP(128),ERROR(128)
C
C READ THE RAW EEG DATA,SCALE IT, AND SEPARATE IT INTO 2 EPOCHS
C (THE TRAINING SET EPOCH AND THE PRECEEDING EPOCH) AND 4 CHANNELS
C
      CALL SCALSP(EEG,PEEG)
C
C SCALE THE RAW EEG DATA, CALCULATE THE MEAN AND VARIANCE,
C AND SUBTRACT THE MEAN FROM THE DATA
C
      CALL MEANVR(EEG,WEEG,MEAN,VAR)
C
C CALCULATE THE LINE SEGMENTS PARAMETERS.
C
      CALL STRINT(WEEG,A,M,MM)
C
C RECONSTRUCT THE EEG EPOCH USING THE STORED LINE SEGMENTS
C PARAMETERS
C
      CALL RECONS(A,M,MM,APP)
C
C  CALCULATE THE ERROR VECTOR,AND THE TOTAL MEAN SQUARE ERROR
C
      N=128
      CHAN=1
      ERRMS=0.0
      TENG =0.0
      DO 70 I=1,N
         ERROR(I)=WEEG(CHAN,I)-APP(I)
         ERRPER(I)=ERROR(I)/WEEG(CHAN,I)
         ERRMS =ERRMS+ERROR(I)**2
         TENG =TENG+WEEG(CHAN,I)**2
70    CONTINUE
      ERMSPC=(ERRMS/TENG)*100
      WRITE (6,100)
      WRITE (6,80) (ERROR(I),I=1,N)
      WRITE (6,110)
      WRITE (6,80) (ERRPER(I),I=1,N)
      WRITE (6,90) ERRMS,TENG,ERMSPC
80    FORMAT (8(E14.6,1X))
90    FORMAT (1H0,6X,10H ERRMS    =,E14.6/
     *7X,10H TENG     =,E14.6/ 7X,10H ERMSPC  =,F12.6/1H )
100   FORMAT (1H1,4X,20H THE ERROR VECTOR IS/1H )
110   FORMAT (1H1,4X,31H THE PERCENTAGE ERROR VECTOR IS/1H )
      STOP
      END
C
C SUBROUTINE SCALSP
C SUBROUTINE FOR SCALING THE EEG, AND SEPARATING IT INTO
C 2 EPOCHS AND 4 CHANNELS
C RAWDAT - RAW EEG FROM THE AC CONVERTERS
C SEEG - EEG DATA FOR THE TRAINING SET
C SPEEG - EEG DATA FOR THE PRECEEDING EPOCH
      SUBROUTINE SCALSP(SEEG,SPEEG)
      REAL8 SEEG(4,128),SPEEG(4,128)
```

```
      INTEGER RAWDAT(1024),I,J,CHAN
      READ(10,10) (RAWDAT(I),I=1,1024)
10    FORMAT(16I4)
      J=1
      DO 30 I=1,128
         DO 20 CHAN=1,4
            SPEEG(CHAN,I)=(RAWDAT(J)-2047.5)/2047.5
            J=J+1
20       CONTINUE
30    CONTINUE
      J=513
      DO 50 I=1,128
         DO 40 CHAN=1,4
            SEEG(CHAN,I)=(RAWLAT(J)-2047.5)/2047.5
            J=J+1
40       CONTINUE
50    CONTINUE
      RETURN
      END
C
C SUBROUTINE MEANVR
C SUBROUTINE FOR CALCULATING THE MEAN AND VARIANCE OF THE EEG
C FROM A SINGLE EPOCH
C THE EEG DATA ARE ALSO SCALED AND THE MEAN SUBTRACTED OUT
C SMEAN - MEAN OF THE EEG
C SVAR - VARIANCE OF THE EEG
C SWEEG - ZERO-MEAN EEG DATA
C
      SUBROUTINE MEANVR(SEEG,SWEEG,SMEAN,SVAR)
      INTEGER I,CHAN
      REAL8 SEEG(4,128),SWEEG(4,128),SMEAN(4),SVAR(4)
      DO 5 CHAN=1,4
         SMEAN(CHAN)=0
5     CONTINUE
      DO 20 I=1,128
         DO 10 CHAN=1,4
            SMEAN(CHAN)=SMEAN(CHAN)+SEEG(CHAN,I)
10       CONTINUE
20    CONTINUE
C
C SUBTRACTION OF THE MEAN
C
      DO 40 CHAN=1,4
         SMEAN(CHAN)=SMEAN(CHAN)/128
         SVAR(CHAN)=0.0
         DO 30 I=1,128
            SWEEG(CHAN,I)=SEEG(CHAN,I)-SMEAN(CHAN)
            SVAR(CHAN)=SVAR(CHAN)+SWEEG(CHAN,I)**2
30       CONTINUE
40    CONTINUE
      RETURN
      END
C
C SUBROUTINE STRINT
C SUBROUTINE TO FIT LINE SEGMENTS TO THE EEG DATA OF
C THE FORM P(X)=MX+B ACCORDING TO THE LEAST OF SQUARES ERROR
C CRITERION AND STORE THE PARAMETERS M,B FOR EACH SEGMENT
C AND THE NUMBER OF SAMPLES REPRESENTED BY EACH SEGMENT.
C A = A VECTOR USED TO STORE THE VALUES M,B
C M = A VECTOR USED TO STORE THE NUMBER OF SAMPLES OF EACH LINE
C MM= 0 OR 1 ACCORDING TO THE TOTAL NUMBER OF STORED POINTES TO
C BE ODD OR EVEN RESPECTIVELY.
C
      SUBROUTINE STRINT(WEEG,A,M,MM)
```

```
      DIMENSION A(256),M(256),WEEG(4,128),F(2),B(2)
      CHAN=1
C
C AS AN EXAMPLE THE TOLERANCE IS SET HERE TO TAKE DEFINIT VALUE
C
      DELTA=40.00E-03
      J=1
      J1=J+1
      J2=J
      J3=J-1
      F(2)=0.0
      B(2)=0.0
      N=128
      DO 20 I=1,N
         II=I+1
         IF(II.GT.N) GO TO 11
         I4=0
         I5=0
         T0=0.0
         T1=0.0
         F(1)=F(2)
         B(1)=B(2)
         DO 5 I3=J,II
            I4=I4+(I3-J3)
            I5=I5+(I3-J3)**2
            T0=T0+WEEG(CHAN,I3)
            T1=T1+WEEG(CHAN,I3)*(I3-J3)
5        CONTINUE
         F(2)=(((II-J3)*T1)-(I4*T0))/(((II-J3)*I5)-(I4**2))
         B(2)=    ((I5*T0)-(I4*T1))/(((II-J3)*I5)-(I4**2))
         K=J
         DO 10 K1=K,II
C AS ANOTHER EXAMPLE THE TOLERANCE COULD BE ADJUSTED
C TO TAKE VALUES DEPENDING ON THE EPOCH CHARACTERISTICS
C           DEL=5.000*WEEG(CHAN,K1)
C
            SS=WEEG(CHAN,K1)-((F(2)*(K1-K))+B(2))
            IF(DABS(SS).LE.DELTA) GO TO 10
            A(J2)=F(1)
            A(J2+1)=B(1)
            M(J1)=I-J3
            J=II
            J1=J1+1
            J2=J2+2
            J3=I
            IF (I.EQ.(N-1)) GO TO 9
            GO TO 20
9           A(J2)=WEEG(CHAN,N)
            M(J1)=1
            M(1)=(3*J1)-3
            MM=1
            WRITE (6,31) (A(KK),KK=1,J2)
            GO TO 22
10       CONTINUE
11       IF(I.EQ.N)  GO TO 21
20    CONTINUE
21    M(1)=(3*J1)-2
      A(J2)=F(2)
      A(J2+1)=B(2)
      M(J1)=I-J3
      MM=0
      KKK=J2+1
      WRITE(6,31) (A(I),I=1,KKK)
22    WRITE (6,30)    (M(I),I=1,J1)
```

```
      WRITE (6,31)    (WEEG(CHAN,I),I=1,N)
      WRITE (6,32) MM
30    FORMAT (1X,10(I3,1X))
31    FORMAT (8(F12.6,1X))
32    FORMAT(I4)
      RETURN
      END
C
C SUBROUTINE RECONS
C SUBROUTINE TO FIND THE APPROXIMATED EEG EPOCH USING THE
C STORED PARAMETERS
C APP= THE POINTS OF THE APPROXIMATED SIGNAL
C
      SUBROUTINE RECONS(A,M,MM,APP)
      DIMENSION A(256),M(256),APP(128)
      N=128
      CHAN=1
      IF(MM.EQ.1) GO TO 5
      J1=(M(1)+2)/3
      J2=(2*J1)-2
      K2=J2/2
      GO TO 10
5     J1=(M(1)+3)/3
      J2=(2*J1)-3
      K2=(J2-1)/2
      APP(N)=A(K2+1)
10    K1=0
      K3=0
      DO 25 I=1,K2
         K3=M(I+1)+K3
         JJ=2*I
         SS=A(JJ-1)
         BB=A(JJ)
         K=K1+1
         KK=K3+1
         IF(KK.GT.128) KK=128
         DO 15 J=K,KK
            APP(J)=(SS*(J-K))+BB
            WRITE (6,8) APP(J)
8           FORMAT (F12.6)
15       CONTINUE
         K1=K3+1
25    CONTINUE
      WRITE(6,30) (APP(J),J=1,128)
30    FORMAT (8(F12.6,1X))
      RETURN
      END
$ENTRY
/*
```

4) - Program for implementing the classifiers.

```
$JOB WATFIV
C
C PROGRAM TO ASSIGNE EACH APPROXIMATED SIGNAL TO A CLASS
C I.E..TO CLASSIFY THE EPOCHS INTO A NUMBER OF CLASSES
C USING THE DIC.FUNCTIONS OBTAIND FROM THE SWDA.THE PROGRAM
C IS WRITTEN HERE TO BE A GENERAL ONE
C FIRST READ THE SIGNIFICANT FEATURES USED TO DISCR. BETWEEN
C THE CLASSES .AND THEN READ THE PARAMETERES TO FORM THE
C DIC. FUNCTIONS.
C
      INTEGER I,K,J,CHAN
      REAL EQ(M),EEQ(M),S(4,MM),EA,EE,F(M,MM+1)
C
C READ THE PARAMETERS TO FORM THE EQUATIONS
C M = THE NUMBER OF EQUATIONS =THE NUMBER OF CLASSES
C MM = THE NUMBER OF FEATURES
C MM+1 = THE NUMBER OF CONSTANTS IN EACH DIC. FUNCTION.
C MMM = THE TOTAL NUMBER OF EPOCHS TO BE CLASSIFIED.
C
      DO 15 J=1,M
         READ(11,5) (F(J,I),I=1,MM+1)
15    CONTINUE
      DO 60 JJJ=1,MMM
         DO 10 CHAN=1,4
            READ(10,5) (S(CHAN,I),I=1,MM)
5           FORMAT (7(F9.5,1X))
10       CONTINUE
C
C FORMING THE EQUATIONS
C
         DO 50 CHAN=1,4
            DO 25 J=1,M
               EQ(J)=0.0
               DO 20 I=1,MM
                  EQ(J)=EQ(J)+F(J,I)*S(CHAN,I)
20             CONTINUE
               EQ(J)=EQ(J)+F(J,MM+1)
25          CONTINUE
            K=1
            DO 35 I=1,M-1
               IF(EQ(K).GT.EQ(I+1)) GO TO 30
               EE=EQ(K)
               EA=EQ(I+1)
               IF (K.GT.1) K=1
               EEQ(K)=EA
               EEQ(I+1)=EE
               K=I+1
               GO TO 35
30             EEQ(I+1)=EQ(I+1)
               IF (K.GT.1) GO TO 35
               EEQ(K)=EQ(K)
35          CONTINUE
            WRITE (6,6) (EEQ(I),I=1,M),(EQ(I),I=1,M)
6           FORMAT (6X,6(E14.6,1X))
            DO 45 I=1,M
               IF(EEQ(I).EQ.EQ(I)) GO TO 40
               GO TO 45
40             PRINT 41, I
41             FORMAT (I4)
               GO TO 50
45          CONTINUE
50       CONTINUE
60    CONTINUE
      STOP
      END
$ENTRY
/*
```

APPENDIX C

Comments On KLE Results

In this Appendix, the KLE results as applied to EEG data are illustrated via tables. Each table contains information about the classification of some epochs (using the parallel classifier) in both original and approximated forms. Also, the mean-square error of the approximation is given. The epochs are chosen such that the range of the mean-square error will be covered.

1) - Normal (NORM)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	NORM	SLOLO	17.5
2	NORM	NORM	2.3
3	NORM	NORM	7.3
4	NORM	SLOHI	2.1
5	NORM	NORM	16.2
6	FAST	LAMP	23.1
7	SLOHI	SLOHI	0.08
8	NORM	NORM	7.5
9	NORM	SLOHI	7.96
10	NORM	NORM	4.7

M.S.E. range for normal epochs is 0.08% - 23.1%

2) - Artefact (ARTEF)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	ARTEF	ARTEF	0.06
2	ARTEF	ARTEF	11
3	ARTEF	ARTEF	0.14
4	ARTEF	SLOHI	3.1
5	PLYSP	SLOHI	11.6
6	ARTEF	ARTEF	2.7
7	ARTEF	ARTEF	0.4
8	ARTEF	ARTEF	0.25
9	ARTEF	ARTEF	2.8
10	PLYSP	SLOHI	1.7

M.S.E. range for artefactual epochs is 0.06% - 11.6%

3) - Low Amplitude (LAMP)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	LAMP	LAMP	36.9
2	LAMP	LAMP	22
3	LAMP	LAMP	4.5
4	LAMP	LAMP	1.6
5	LAMP	LAMP	14.6

M.S.E. range for Low amplitude epochs is 1.6% - 36.9%

4) - Fast (FAST)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	FAST	FAST	49.8
2	FAST	FAST	30.4
3	FAST	FAST	50.4
4	FAST	FAST	19.9
5	FAST	FAST	7.7
6	FAST	LAMP	15.6
7	FAST	SLOHI	5.7
8	FAST	SLOHI	0.8
9	FAST	FAST	4.4
10	FAST	LAMP	22.5

M.S.E. range for fast epochs is 0.8% - 50.4%

5) - Slow - Low Amplitude (SLOLO)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	FAST	LAMP	0.4
2	SLOLO	SLOLO	2.7
3	SLOLO	SLOHI	2.1
4	SLOLO	SLOLO	10
5	SLOLO	SLOLO	22.5
6	SLOLO	SLOLO	8.3
7	SLOLO	SLOHI	4.5
8	SLOLO	SLOLO	4.9
9	SLOLO	SLOHI	9.7
10	SLOLO	SLOLO	6.3

M.S.E. range for slow-low amplitude epochs is 0.4% - 22.5%

6) - Slow - High Amplitude (SLOHI)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	SLOHI	SLOHI	5.1
2	SLOLO	SLOLO	18.8
3	SLOHI	SLOHI	0.15
4	SLOHI	SLOHI	3.1
5	SLOHI	SLOHI	11.7
6	PLYSP	SLOHI	12.6
7	SLOHI	ARTIF	0.8
8	SLOHI	SLOHI	1.1
9	SLOHI	SLOHI	0.24
10	NORM	SLOHI	2.9

M.S.E. range for slow-high amplitude epochs is 0.15% - 18.8%

7) - Sharp/Spike Wave - High Amplitude (SSWHI)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	SSWHI	SSWHI	11.2
2	SSWHI	SLOHI	5.2
3	SLOLO	SLOLO	38.5
4	SSWHI	NORM	12.7
5	SSWHI	ARTEF	0.96
6	SSWHI	PLYSP	30.1
7	SSWHI	SSWHI	7.8
8	SSWHI	SLOHI	10.9
9	SSWHI	SSWHI	2.7
10	SSWHI	ARTEF	0.5

M.S.E. range for sharp/spike wave-high amplitude epochs is 0.5% - 38.5%

8) - Isolated Spike (ISOSP)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	ISOSP	NORM	4.9
2	ISOSP	ISOSP	24.1
3	ISOSP	NORM	13.9
4	PLYSP	SLOLO	32.5
5	ISOSP	SLOHI	5.3

M.S.E. range for isolated spike epochs is 4.9% - 32.5%

9) - Multi-Spikes (PLYSP)

Epoch No.	Original Classif.	Approx. Classif.	M.S.E. %
1	PLYSP	SLOLO	40.5
2	PLYSP	SLOHI	43.6
3	PLYSP	PLYSP	13.8
4	SLOLO	PLYSP	37
5	PLYSP	SLOLO	16.5

M.S.E. range for multi-spikes epochs is 7.2% - 43.6%

APPENDIX - D -

Least - Squares Line (Discrete Data)

The method of last squares, was developed first by Gauss. First of all, when the data are discrete we may minimize the sum

$$S = \sum_{i=0}^{N} [y_i - a_o - a_1 x_i - \ldots\ldots a_m x_i^m]^2 \qquad [D-1]$$

for given data x_i, y_i and $m < N$.

The condition $m < N$ makes it unlikely that the polynomial

$$p(x) = a_o + a_1 x + a_2 x^2 + \ldots\ldots + a_m x^m \qquad [D-2]$$

can collocate at all N data points. So S probably can not be made zero.

For the case of a linear polynomial

$$p(x) = Mx + B \qquad [D-3]$$

the sum S has the form

$$S = \sum_{i=0}^{N} (y_i - Mx_i - B)^2 \qquad [D-4]$$

to minimize S, we have to equate the 1st derivatives to zero.

$$\frac{\partial S}{\partial B} = -2 \sum_{i=0}^{N} (y_i - M x_i - B) = 0$$

$$\frac{\partial S}{\partial M} = -2 \sum_{i=0}^{N} x_i (y_i - M x_i - B) = 0 \qquad [D-5]$$

Rewriting we have

$$(N + 1)\ B + (\Sigma\ x_i)\ M = \Sigma\ y_i \qquad [D-6]$$

$$(\Sigma\ x_i)\ B + (\Sigma\ x_i^2)\ M = \Sigma\ x_i y_i \qquad [D-7]$$

introducing the symbols

$$S_o = (N + 1),\ S_1 = \Sigma\ x_i\ ,\quad S_2 = \Sigma\ x_i^2$$

$$t_o = \Sigma\ y_i\ ,\ t_1 = \Sigma\ x_i y_i \qquad [D-8]$$

equations (D-6), (D-7) may be solved in the form

$$M = \frac{S_o t_1 - S_1 t_o}{S_o S_2 - S_1^2} \qquad B = \frac{S_2 t_o - S_1 t_1}{S_o S_2 - S_1^2} \qquad [D-9]$$

which completely define the straight line $p(x) = Mx + B$.

To show that $S_o S_2 - S_1^2$ is not zero, we may notice that squaring and adding terms such as $(x_o - x_1)^2$ leads to

$$0 < \sum_{i<j} (x_i - x_j)^2 = N.\ \Sigma\ x_i^2 - 2 \sum_{i-j} x_i x_j \qquad [D-10]$$

but also $(\Sigma\ x_i)^2 = \Sigma\ x_i^2 + 2 \sum_{i<j} x_i x_j$ [D-11]

So that $S_o S_2 - S_1^2$ becomes

$$(N + 1)\ \Sigma\ x_i^2 - (\Sigma\ x_i)^2 = N.\ \Sigma\ x_i^2 - 2 \sum_{i<j} x_i x_j > 0 \qquad [D-12]$$

Here we have assumed that the x_i are not all the same, which is surely reasonable. This last inequality also helps to prove that M and B chosen, actually produce a minimum.

Calculating second derivatives we find

$$\frac{\partial^2 S}{\partial B^2} = 2\ S_o \quad , \frac{\partial^2 S}{\partial M^2} = 2\ S_2\ , \quad \frac{\partial^2 S}{\partial B \partial M} = 2\ S_1 \qquad [D-13]$$

Since the first two are positive and since

$$(2S_1)^2 - 2\ (N + 1)(2S_2) = 4(S_1^2 - S_o S_2) < 0$$

The second derivative test for a minimum of a function of two arguments B and M is satisfied. The fact that the first derivatives can vanish together only once, shows that our minimum is an absolute minimum.

APPENDIX - E -

The Technique used to Design the Terminal Classifier

In the past decade, several systems for automatic recognition of abnormalities associated with epilepsy have been developed. A complete discussion of automated analysis techniques are given by Gevins [1] and Barlow [2].

i - Pattern Recognition Background - The Decision - Theoretic Approach

Pattern recognition can be defined as "the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background or irrelevant detail" [25]. Mathematical pattern recognition deals with underlying theory and practical techniques for machine implementation of a given recognition task. Recognition of patterns using the decision - theoretic approach involves three basic steps: (1) pattern Preprocessing (2) Feature extraction and (3) pattern classification. The functions of the preprocessing include pattern encoding and approximation and the filtering, restoration and enhancement of the approximated patterns. For example, in the case of the EEG, filtering to remove insignificant high frequencies and digital encoding to facilitate further processing. After the preprocessing stage, each pattern typically is represented by a large number of values. The purpose of feature extraction is to determine from these values minimal set of features which adequately describes differences between patterns of various classes. The final step in the decision-theoretic approach is to classify the input patterns, based on the values of their features, into predetermined classes or groups. This classification is accomplished using discriminant or decision functions (hence the name decision-theoretic). These functions are derived in what is known as a training procedure, in which sample patterns representing each of the classes are used to calculate the decision function coefficients. An excellent discussion of the decision theoretic approach is given by Tou and Gonzales [14].

ii Feature Extraction and Terminal Classification

All EEG data used in this research, was recorded using a four channel cassette recorder. Eight 24-hour recordings were obtained from six patients between ages of ten and sixteen years. All six patients were suspected of having some form of epileptic abnormality. Portions of record suitable for use in training the terminal classifier as well as recorders for evaluating system performance were selected. For the purpose of our analysis, each of the EEG channels was segmented into terminals consisting of one second duration epochs of EEG activity. These terminals were selected because they satisfied the requirement of representing the simplest subpatterns of significance of EEG.

As discussed previously, decision-theoretic technique was chosen to identify terminals. This identification requires that a set of features be extracted from the EEG which adequately describes differences between the terminal classes.

Several feature extraction techniques have been employed in the analysis of EEG signals. A partial listing of these techniques includes spectral analysis using Fourier - transformed autocorrelation functions or the FFT, time domain descriptions based on moments of the power spectrum [26-27], auto and cross-spectra, filtering and interval analysis [26]. Rather than use any of the above techniques, spectral estimation based on autoregressive modelling was chosen for extracting features from the EEG.

There were several reasons for choosing autoregressive spectral estimation for extracting features from the EEG. First, features derived from spectral estimates generally provide a better description of the EEG and have more physical significance than do those obtained using other techniques. Second, autoregressive methods provide better spectral estimates for short pieces of record, an important feature given the duration of the terminal used. Another interesting property of the autoregressive estimate is that it models peaks in the signal spectrum before the valleys. An additional advantage of this technique is that the spectral estimates obtained are smooth, and thus do not require windowing.

In order to determine which of the used features were most relevant to classifying the one-second EEG epochs into the terminal classes, and to generate the decision function, a procedure known as stepwise discriminant analysis was used. This was accomplished using the program BMDP7M of the Biomedical Computer Programs Package [28], available on the University of Manitoba's computer system.

iii Stepwise Discriminant Analysis as a Classification Technique

Stepwise discriminant analysis (SWDA) combines a process for generating linear decision functions for classification purposes with a process for selecting from a large number of features a small set of important ones. Initially, the program must be provided with a training set, i.e., a set of sample patterns or cases, represented by their features, and the class to which each of the patterns belongs. As suggested by its title, the procedure proceeds in steps. At each one, a number (related to the F-ratio, used to test the statistical significance of the differences between means of two sets of observations) is calculated for each feature to express the amount of discriminatory power it provides. The feature which provides the most discriminatory power is then entered into the decision functions. This procedure continues until the features added no longer contribute significantly to discriminating between classes. The program provides information describing how well different classes are discriminated. In addition, several options are provided to allow the user to specify the entering of variables into the decision functions.

The decision functions calculated by P7M are those for a Bayes Maximum Likelihood Classifier [14]. To simplify the calculations and produce linear decision functions, the algorithm assumes that all pattern classes have normal distributions with equal covariance matrices. In the final output of the program, the decision functions, calculated from those features which have been entered, as well as a pseudojackknifed classification matrix (which gives the classification accuracy of the decision functions when used on the training set data) are provided.

Stepwise discriminant analysis was chosen for feature selection and generation of decision functions for two reasons. First, SWDA provided a simple way of eliminating useless features, reducing the computational load involved in recognizing terminals and improving classification performance. Second, SWDA simplifies the generation of their performance.

One disadvantage to using SWDA, arises from the assumptions used in generating the classification functions. These assumptions (normally distributed classes, equal covariance matrices) are likely invalid, and undoubtably result in less than optimum classification performance.