ELSEVIER

PII: S1350-4533(97)00026-X

Med. Eng. Phys. Vol. 19, No. 7, pp. 605–617, 1997

Printed in Great Britain
1350-4533/97 $17.00 + 0.00

Iterative function system and genetic algorithm-based EEG compression

Suman K. Mitra and S. N. Sarbadhikari

Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700035, India

Received 10 December 1996, accepted 28 May 1997

ABSTRACT

A method for EEG compression is proposed, using Iterative Function System (IFS) and Genetic Algorithms (GAs) with elitist model, keeping the quality sufficiently good for clinical purposes. Compression using IFS is usually called fractal compression. The self transformability property of the EEG signals is assumed and is exploited in the fractal compression technique. To ascertain the self transformability of the EEG signal, some isometric transformations have been applied. The technique described here utilizes Genetic Algorithm that decreases the search space for finding the self similarities in the given signal. This article presents theory and implementation of the proposed method. The fidelity of the reconstructed signal obtained by the present compression algorithm has been assessed both qualitatively and quantitatively. The compression ratios, for the EEG signals in various states, are found to be comparable to the other available techniques for EEG compression. In our method at least 85% data reduction has been achieved.

Keywords: EEG compression, Iterated function system (IFS), Genetic algorithm (GA), Isometry, Compression ratio, Fidelity

Med. Eng. Phys., 1997, Vol. **19**, 605–617, October

1. INTRODUCTION

EEG (Electroencephalography) reflects the electrical activity of the brain during the various states of sleep and wakefulness. The EEG signals are complex in nature. During sleep, from EEG, two distinct patterns, namely, REM (Rapid Eye Movement) or paradoxical sleep and SWS (slow wave sleep) or deep sleep, can be easily distinguished visually (*Figure 1*). The SWS recordings of EEG is quite distinct from those of the wakeful state, as well as the REM state. However, the REM sleep recordings resemble the wakeful state recordings very much, and are difficult to identify by EEG alone[1].

The sheer volume of EEG recordings for diagnostic purposes can be frightening. The amount of space required for storage too is forbidding. Each second of digitized EEG data (two channels sampled at 256 Hz) takes a kilobyte of space for storage. Therefore, the need for compression of the data is of the utmost importance. Moreover, EEG compression can help (a) to augment the storage capacity of collected EEG data for later evaluation or comparison, (b) to facilitate transmitting real-time EEG signals to distant places and (c) to transmit rapidly and economically off-line EEG data over telecommunication networks to remote interpretation centers. Unfortunately, this area has been very much neglected and only a few articles[2–8] have come to the notice of the authors.

Recently, an approach for one-dimensional and two-dimensional signal compression has begun to emerge, based on the theory of Iterative function system (IFS). Compression of signals using IFS is a very promising technique. The theory of coding using IFS and Collage theorem was first proposed by Barnsley[9]. Since then, this technique has been used successfully in image compression by several researchers[10–13]. The method proposed in Ref. [13] (for image compression) is suitably modified here for signal (2-D) compression by choosing the possible set of transformations and resetting the search parameters for finding the self transformability.

The basic idea of this technique is to approximate the given signal from a set of affine (linear) contractive transformations called IFS. The set of affine contractive transformations, through an iterative process produces a signal called the attractor or the fixed point, which is very close to the target signal. Thus, it is sufficient to store the relevant parameters of the transformations in order to code the signal. The main task is to find

Corresponding author.
This work has been carried out while Dr S. N. Sarbadhikari held a Research Associateship in a Project (No. 22(235)/93/EMR-II) of the Council for Scientific and Industrial Research, India.

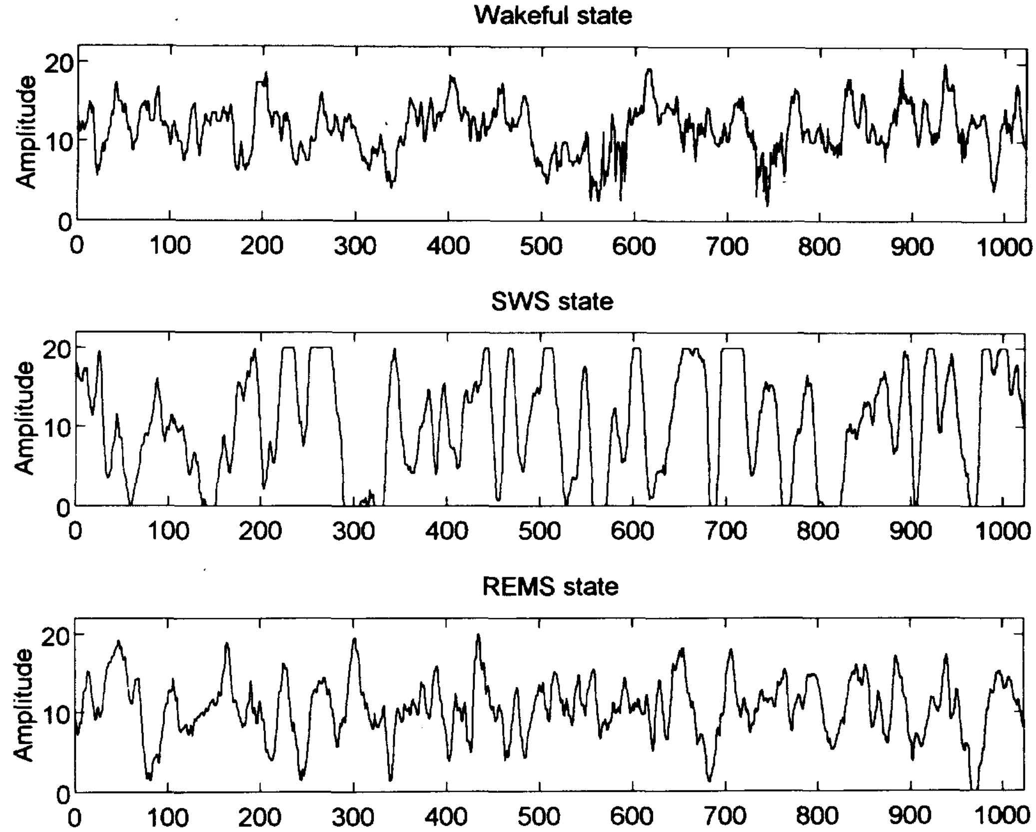

Figure 1 EEG recordings in the three states.

the appropriate set of transformations whose attractor approximates the given signal. The aforesaid transformations are obtained using the self-similarities present in the given signal. Here self-similarity implies that the waveform of a particular segment, called the range segment, of the signal is a scaled and transformed version of another segment, called domain segment, of the same signal. Out of the several possible transformations of domain segments to range segments, in the present article, we have used only a few affine transformations. Thus, the whole problem can be viewed as a search problem where the appropriate domain segment as well as the appropriate transformation are found for a range segment under consideration.

The encoding technique described here also utilizes Genetic Algorithms (GAs) as a search process for finding the self-similarity present in the signal[13]. GAs[14,15] are mathematically modeled algorithms which emulate natural selection mechanism to solve the optimization problems. GAs attempt to find near-optimal solutions without going through an exhaustive search mechanism. Thus, GAs have an advantage of minimizing the search space and hence the search time. Implementation of GAs as a search mechanism provides fast encoding of EEG signals through IFS.

The objective of the present article is to investigate whether the fractal and GA-based technique proposed by us can lead to efficient compression of EEG signals which look very irregular. The results obtained by the present technique are compared with those of the existing techniques. The theoretical foundation for the present work is described in the next section. In Section 3, the methodology used for encoding is presented in detail. This is followed by implementation details, and results in Section 4.

2. THEORETICAL FOUNDATION

2.1. Iterative function systems

The central idea of IFS revolves around a set of functions (called transformations) which transform a given set of points to another set of points. Under the restriction of contractivity, these transformations can take any arbitrary set of points to a fixed set after an infinite (practically finite but large) number of iterations (applications of the transformations). The detailed mathematical description of the IFS theory, Collage theorem and other relevant results are available in Ref. [9]. Only the salient features of signal coding through IFS are given below.

Let f be a contractive affine map defined on $\mathbb{R}^2$ such that $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where f has the general linear form $f(u, v) = (au + bv + e, cu + dv + f)$. Here a, b, c and d are called scaling parameters and e and f are called shift parameters. Also, $d(f(x_1), f(x_2)) \leq s\, d(x_1, x_2); \forall x_1, x_2 \in X$, where $0 \leq s < 1$ is called the contractivity factor of the transformation f. Here d measures the distance between two sets. Usually, the Hausdorff distance is used for this purpose. However, we have used another measure which will be discussed in section 3.2. For any large positive number N, $\lim_{N \rightarrow \infty} f^N(x) = a, \forall x \in X$, and also $f(a) = a$. "a" is called the fixed point (attractor) of f. Here $f^N(x)$ is defined as

$$f^N(x) = f(f^{N-1}(x)), \text{ with } f^1(x) = f(x), \forall x \in X.$$

Now, let I be a given signal which is a subset of $\mathbb{R}^2$. This is because the signal contains different magnitude values at different time points and both the magnitude and the time points are taking values from the positive part of the real line. Our intention is to find a set $\mathcal{F}$ of affine contractive transformations for which the given signal I is an approximate fixed point. $\mathcal{F}$ is constructed in such a way that the distance between the given signal and the fixed point (attractor) of $\mathcal{F}$ is very small. The attractor A of the set of transformations $\mathcal{F}$ is defined as follows:

$$\lim_{N\to\infty} \mathcal{F}^N(J) = A, \forall J \in \mathbb{R}^2,$$

and $\mathcal{F}(A) = A$, where $\mathcal{F}^N(J)$ is defined as $\mathcal{F}^N(J) = \mathcal{F}(\mathcal{F}^{N-1}(J))$, with

$$\mathcal{F}^1(J) = \mathcal{F}(J), \forall J \in \mathbb{R}^2$$

Also the set of maps $\mathcal{F}$ is defined as follows:

$$d(\mathcal{F}(J_1), \mathcal{F}(J_2)) \leq s\, d(J_1, J_2); \forall J_1, J_2 \in \mathbb{R}^2 \quad (1)$$

and $0 \leq s < 1$.

s is called the contractivity factor of $\mathcal{F}$.

$$\text{Let } d(I, \mathcal{F}(I)) \leq \epsilon \quad (2)$$

where ϵ is a small positive quantity. Now, by Collage theorem[9], it can be shown that

$$d(I,A) \leq \frac{\epsilon}{1-s} \quad (3)$$

where A is the attractor of $\mathcal{F}$.

From Equation (3) it is clear that, after a sufficiently large number (N) of iterations, the set of affine contractive transformations $\mathcal{F}$ produces a set (A) which is a subset of $\mathbb{R}^2$ and is very close to the given original signal I. Here, $(X, \mathcal{F})$ is called the iterative function system and $\mathcal{F}$ is called the set of fractal codes for the given signal I.

2.2. Genetic algorithms

Genetic algorithms (GAs) are highly adaptive search and machine learning processes based on natural selection mechanism of a biological genetic system. GAs help to find the global near-optimal solution without getting stuck at local optima as they deal with multiple points (called, chromosomes) simultaneously. To solve the optimization problem, GAs start with the chromosomal (structural) representation of a parameter set. The parameter set is coded as a string of finite length called a chromosome or simply a string. Usually, the chromosomes are strings of 0's and 1's. If the length of a string is l, then total number of strings is 2^l.

Usually, a function "fit" is defined on the set of strings which represents the function to be optimized. This function "fit", also known as the "fitness function" denotes the fitness value of a string. Note that, more than one string may possess the same fitness value. In GAs, a string which provides optimal fitness value is found without an exhaustive search. In this article we are dealing with a minimization problem. To find a near-optimal solution, three basic genetic operators: (i) selection; (ii) crossover; and (iii) mutation are exploited in GAs.

Out of all possible 2^l strings, initially a few strings [say S number of strings] are selected randomly and this set of strings is called the initial population. In this article, we have taken S to be an even number, though S can be an odd number too. Starting with the initial population the three genetic operators are used one by one to form a new population. The process of creating a new population is called an iteration and is executed for a fixed number of times. Here also we have used the elitist model of GAs. In the elitist model of GAs the worst string in the present population is replaced by the best string of the previous population in each iteration to keep track with the best string obtained in each iteration.

In the selection procedure, S number of strings are selected from the current population to form a mating pool. The selection of each individual string, as a string in the mating pool, is directly or inversely proportional to its fitness function as the problem is either a maximization or a minimization problem, respectively. This type of selection scheme is known as proportional selection strategy. The crossover operation is then applied on the mating pool.

The most commonly used crossover operation is a single point crossover[14] operation on a pair of strings which is described here. An integer position k is selected randomly between 1 and $l - 1$ ($l > 1$), l being the string length. Two new strings are then created by swapping all the characters from position $k + 1$ to l of the old strings. For example, let $a = \alpha_1 \alpha_2 \alpha_3 \ldots \alpha_k \alpha_{k+1} \ldots \alpha_l$ and $b = \beta_1 \beta_2 \beta_3 \ldots \beta_k \beta_{k+1} \ldots \beta_l$ be two strings (called parents) forming a pair for crossover operation. The strings (called children or offspring) generated after the crossover operation are $a' = \alpha_1 \alpha_2 \alpha_3 \ldots \alpha_k \beta_{k+1} \ldots \beta_l$ and $b' = \beta_1 \beta_2 \beta_3 \ldots \beta_k \alpha_{k+1} \ldots \alpha_l$

The occurrence of a crossover operation on a pair of strings is guided by crossover probability, say, P_{cross}. A random number (≤ 1) is drawn for each pair of strings. The drawn random number less than P_{cross} indicates the occurrence of crossover for that pair and the non-occurrence if the reverse is true. Usually a high value is assigned for the crossover probability. The mutation operation is then applied.

In a mutation operation every bit of every string is replaced by the reverse character (i.e. 0 by 1 and 1 by 0) with some probability. One of the commonly used conventions[14] is to assign a very small value to the mutation probability P_{mut} and keep the P_{mut} fixed for all the iterations. But here we have prefixed the number of iterations of GA *a priori* and varied the mutation probability with the number of iterations. This varying mutation probability scheme has already been applied successfully in connection with an application of GAs to pattern recognition problem[16].

The process is executed for a fixed number of times (iterations) and the best string, obtained so far, is taken to be the near-optimal one. In the present article, the number of iterations, say T, is fixed *a priori* for the termination of GAs.

2.2.1. Description of the algorithm. The genetic algorithm is implemented using the following steps.

1. Generate an initial population Q of size S and calculate fitness value of each string S of Q.
2. Find the best string S_{bst} of Q. If the best string is not unique, then call any one of the best strings of Q as S_{bst}.
3. Construct a mating pool using proportional selection strategy (S_{bst} belongs to Q). Perform crossover and mutation operations on the strings in the mating pool and obtain a population Q_{tmp}.
4. Calculate the fitness value of each string S of Q_{tmp} and replace the worst string of Q_{tmp} by S_{bst}. Rename Q_{tmp} as Q.
5. If T iterations are completed then stop, otherwise go to step 2.

Note that steps 2, 3 and 4 together make an iteration.

3. METHODOLOGY FOR ENCODING EEG SIGNAL

In this section, initially, the procedure of collecting EEG data used in this article is described. Then the encoding and decoding mechanisms are described.

3.1. Electroencephalogram

The EEG data have been collected from male rats (*Rattus norvegicus var. albinus*) of Charles Foster strains. The EEG has been recorded through a 8-channel polygraph (Medicare, India) on paper and sampled at 256 Hz through a 12-bit analog–digital converter (Micronics, India) into the hard-disk of a PC-AT (HCL, India).

3.2. Generation of fractal codes using affine transformations

Let, I be a given signal having w points with a range of amplitude values $[0,g]$. Thus, the given signal I is a subset of $\mathbb{R}^2$ as described in section 2.1. The signal is first partitioned into n non-overlapping segments having say b number of points, and let this partition be represented by $\mathcal{R} = \{\mathcal{R}_1,\mathcal{R}_2,\ldots,\mathcal{R}_n\}$. Each $\mathcal{R}_i$ is called as a range segment. Note that $n = w/b$. Let $\mathcal{D}$ be the collection of all possible segments having $2b$ number of points. Let $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2,\ldots,\mathcal{D}_m\}$. Each $\mathcal{D}_j$ is called as a domain segment with $m = (w - 2b)$.

Let $\mathcal{F}_j = \{f: \mathcal{D}_j \rightarrow \mathbb{R}^2$; f is an affine contractive map$\}$.

Now, for a given range segment $\mathcal{R}_i$, let, $f_{i|j} \in \mathcal{F}_j$ be such that

$$d(\mathcal{R}_i, f_{i|j}(\mathcal{D}_j)) \leqslant d(\mathcal{R}_i, f(\mathcal{D}_j)); \forall f \in \mathcal{F}_j, \forall_j$$

Here d measures the distance between two sets of points.

Now let k be such that

$$d(\mathcal{R}_i, f_{i|k}(\mathcal{D}_k)) = \min_j\{d(\mathcal{R}_i, f_{i|j}(\mathcal{D}_j))\} \qquad (4)$$

Also, let $f_{i|k}(\mathcal{D}_k) = \hat{\mathcal{R}}_{i|k}$. Our aim is to find $f_{i|k}(\mathcal{D}_k)$ for each $i \in \{1, 2,\ldots,n\}$. In other words, for every range segment $\mathcal{R}_i$, one needs to find an appropriately matched domain segment $\mathcal{D}_k$, as well as an appropriate transformation $f_{i|\bullet}$. The set of transformations $\mathcal{F} = \{f_{1|\cdot}, f_{2|\cdot},\ldots, f_{n|\cdot}\}$ thus obtained is called the fractal code of the given signal I[10,11,13].

To find the best matched domain segment, as well as the best matched transformation, we are to search all possible domain segments as well as all possible transformations with the help of Equation (4). The affine contractive transformation $f_{i|\bullet}$ is constructed using the fact that the points of the range segment are a scaled and shifted version of the points of domain segment. Thus, the affine transformation has two parts. The first part indicates which point of the range segment corresponds to which point of the domain segment. The second part is to find the scaling and shift parameters.

The first part is shuffling the points of the domain segment and can be achieved by using any one of the eight possible transformations (isometry) on the domain segments as described in section 3.2.1. Once the first part is obtained, the second part is estimation of a set of values (amplitude) of range segments from the set of values of the transformed domain segments. These estimates can be obtained by using the least square analysis of two sets of values.

The distance measure d [used in Equation (4)] is taken to be the simple mean square error (MSE) between the original set of amplitude values and the obtained set of amplitude values of the concerned range segment. Let $\mathcal{R}_i(p)$ and $\hat{\mathcal{R}}_{i|k}(p)$ be respectively the original and the obtained values of the p^{th} point of the range segment $\mathcal{R}_i$. Thus, the expression for MSE will be

$$d(\mathcal{R}_i, \mathcal{R}_{i|k}) = \sum_{p=1}^{b} (\mathcal{R}_i(p) - \hat{\mathcal{R}}_{i|k}(p))^2. \qquad (5)$$

The MSE is not a metric, though it serves the purpose of a distance measure. As the selection of fractal code for a range segment is dependent only on the estimation of amplitude values of that segment, it is enough to calculate only the distortion of the original and estimated amplitude values of the segment. Thus, MSE is taken as the distance measure. Note that the same measure had been used in connection with the image compression using IFS[10,11,13].

3.2.1. Class of transformations. We have stated that a contractive affine transformation $f_{i|j}$ defined on $\mathbb{R}^2$ is such that $f_{i|j}(\mathcal{D}_j) \rightarrow \mathcal{R}_i$. Also $f_{i|j}$ consists of two parts, one for spatial information and the other for information of amplitude values. The second part is obtained using least square analysis of two sets of points once the first part is fixed. Moreover, the size of the domain segment is double that of the range segment. However, the least square (straight line fitting) needs point to point correspondence. To overcome this, the amplitude values of a range segment correspond to the average values of two consecutive points in the domain segment, thus making the contracted domain segment correspond to the range segment. Note that the size of the contracted domain segment is equal to the size of the range segment.

The first part of the transformation f_{ij} indicates which point of the contracted domain segment corresponds to which point of the range segment. The following eight transformations (isometries) simply shuffle the points within a contracted domain segment so that it can correspond to the range segment in eight different ways.

Note that we have considered the size of the range segment as well as the contracted domain segment to be even (≥ 4) though it can be an odd number too. In such a case, the transformations which are described below for an even number have to be changed accordingly.

Let s_d be the starting point of the contracted domain segment $\mathcal{D}_j = \mathcal{D}_j(p)$ having b number of points. Here $p \in \mathcal{A}$ where,

$$\mathcal{A} = s_d, s_d + 1, \ldots, s_d + \frac{b}{2} - 2, s_d + \frac{b}{2} - 1, s_d + \frac{b}{2},$$

$$s_d + \frac{b}{2} + 1, \ldots, s_d + b - 2, s_d + b - 1.$$

The list of transformations (isometries) (l_j) is presented below.

1. Identity:

$$l_1(\mathcal{D}_j(p)) = \mathcal{D}_j(p) \tag{6}$$

2. Second half-reflection:

$$l_2(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(p); & if\, p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j\left(2s_d + \frac{3b}{2} - 1 - p\right); & if\, p > s_d + \frac{b}{2} - 1 \end{cases} \tag{7}$$

3. First half-reflection:

$$l_3(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j\left(2s_d + \frac{b}{2} - 1 - p\right); & if\, p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(p); & if\, p > s_d + \frac{b}{2} - 1 \end{cases} \tag{8}$$

4. First half-reflection and second half-swapping:

$$l_4(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j(2s_d + b - 1 - p); & if\, p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j\left(p - \frac{b}{2}\right); & if\, p > s_d + \frac{b}{2} - 1 \end{cases} \tag{9}$$

5. First half-swapping and second half-reflection:

$$l_5(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j\left(p + \frac{b}{2}\right); & if\, p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j(2s_d + b - 1 - p); & if\, p > s_d + \frac{b}{2} - 1 \end{cases} \tag{10}$$

6. Reflection about mid-point:

$$l_6(\mathcal{D}_j(p)) = \mathcal{D}_j(2s_d + b - 1 - p) \tag{11}$$

7. First–second and third–fourth quarter reflection:

$$l_7(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j\left(2s_d + \frac{b}{2} - 1 - p\right); & if\, p \leq s_d + \frac{b}{2} - 1 \\ \mathcal{D}_j\left(2s_d + \frac{3b}{2} - 1 - p\right); & if\, p > s_d + \frac{b}{2} - 1 \end{cases} \tag{12}$$

8. Second–third quarter reflection:

$$l_8(\mathcal{D}_j(p)) = \begin{cases} \mathcal{D}_j\left(s_d + \frac{b}{4} - 1 - p\right); & if\, s_d + \frac{b}{4} - 1 < p < s_d = \frac{3b}{4} \\ \mathcal{D}_j(p); & otherwise \end{cases} \tag{13}$$

3.3. Fractal codes using GA

The main aspect of fractal-based coding is to find a suitable domain segment and a transformation for a range segment. Thus, the whole problem can be looked upon as a search problem. Instead of a global search mechanism we have introduced GAs to find the near-optimal solution.

The number of possible domain segments to be searched are $(w - 2b)$ (section 3.2). The number of transformations to be searched for each domain block is 8 (section 3.2.1). Thus, the space to be searched consists of M elements. M is called cardinality of the search space. Here $M = 8 \times (w - 2b)$. Let the space to be searched be represented by $\mathcal{P}$ where $\mathcal{P} = \{1, 2, \ldots, (w - 2b)\} \times \{1, 2, \ldots, 8\}$.

Binary strings are introduced to represent the elements of $\mathcal{P}$. The set of 2^l binary strings, each of length l, are constructed in such a way that the set exhausts the whole parametric space. The value of l depends on the values of w and b. The fitness value of a string is taken to be the MSE between the given range segment and the estimated range segment. Note that the problem under consideration is a minimization problem. Here we are to minimize the MSE of estimated range segment with respect to the given range segment.

Let S be the population size and T be the maximum number of iterations for the GA. Initially, S strings are selected randomly from 2^l strings, to result in an initial population for GA. The various steps of the GA, as mentioned in section 2.2, are implemented repeatedly up to T iterations. Note that the total number of strings searched up to T iterations is $S \times T$. Hence, $M/(ST)$ provides the search space reduction ratio for each range segment, and $(M - S\,T)$ provides the actual reduction in the search space for each range segment. Thus, for n number of range segments, $(n \times (M - (S\,T)))$ would provide the total search space reduced for finding the set of transformations $\mathcal{F}$ for fractal signal compression.

A two-level partition scheme, described below, is also adopted for the specific implementation of the present methodology.

3.3.1. Two-Level partition scheme. A two-level partition scheme has been used for the implementation of the GA-based fractal compression methodology. Range segments of two different sizes, b and $b/2$, have been taken to encode the whole signal. The range segments of smaller size, called child range segments are easy to encode and they take care of finer details of a very small portion of the signal. On the other hand, large segments, called parent range segments, lead to a higher compression ratio. Note that there are two child range segments of each parent range segment.

The GA-based fractal compression is implemented on a parent range segment to obtain a good estimate of it. The estimated range segment is then split to provide two children. The original parent range segment is also split into two child range segments. The distortion measure values between the estimated child range segments and the corresponding original child range segments are calculated. If the distortion measure value for any one of the child range segments is above a prefixed threshold value, then new transformations (using the GA-based fractal compression method) are obtained for both the child range segments. On the other hand, the transformation for the parent range segment is stored if the distortion measure values for both the child range segments are below the threshold value. This scheme is repeated for each parent range segment. Note that the threshold value for comparing the distortion measure for the child range segments has been fixed heuristically.

3.4. Decoding

The decoding scheme simply consists of iterating the fractal code $\mathcal{F}$ on any initial signal I_0, until convergence to a stable decoded signal is obtained. The transformation of a signal under the fractal code is done sequentially. The transformation $f_{i|k}$ is applied on the kth domain segment ($\mathcal{D}_k$) of the current signal to produce the ith range segment ($\mathcal{R}_i$) of the next signal. In our case, all the EEG signals have been reconstructed, from the respective fractal codes, starting from an arbitrary signal having all the amplitude values zero. Also the process of iterating the fractal codes has been stopped after 20 repetitions.

3.5. Compression

Compression techniques practically aim at obtaining maximum data volume reduction, while preserving the significant signal features upon reconstruction. Conceptually, data compression is the process of detecting and eliminating redundancies in a given data set. Redundancy may be defined 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.

Whenever the term compression is used, we need to estimate its quantitative extent. The compression ratio and redundancy are two such commonly used measures. If N_o is the size of the original file in bits and N_c is the size of the compressed signal, then the compression ratio C_R is $C_R = \frac{N_o}{N_c}$ and redundancy R_D is $R_D = (1 - \frac{1}{C_R}) \times 100\%$. Clearly, $R_D \rightarrow 100\%$ for $N_c < N_0$ and that indicates good compression.

We have used EEG signals of 1024 points (4-s epochs) each. The amplitude value of each point is a real number ranging from -10.0 to $+10.0$, where 10.0 denotes the brain electrical activity of about 200 microvolts. The sign + ve denotes electrical activity towards the recording electrodes, while the sign − ve denotes electrical activity away from the electrodes. For the purpose of implementation, we have converted the amplitude range to 0.0 to 20.0. Thus, 9 bits are sufficient for storing the amplitude values of each point.

In the encoding process, for each range segment the matched domain segment and the matched transformation have been obtained. Thus, we have to store: (1) the location of the domain segment; (2) the orientation (isometry) of the domain segment; (3) the scaling factor; and (4) the shift factor (section 3.2). So, the number of bits stored, for a range segment, depends on the range of values of the four above-mentioned parameters as well as the size of the range segment (b). Note that the values of first two parameters range from 1 to $(1024 - b)$ and 1 to 8, respectively. Also we have considered discretized values for scaling and shift parameters. Moreover, in a two-level partition scheme we have parent as well as child range segments. Thus, to indicate the type of range segment we have to store another bit.

In our case, the range segment size b has been chosen to be 32 (parent) and 16 (child). The number of bits stored for both parent and child range segments becomes 21 bits. In single-level scheme, N_r becomes $N_r = N_R \times 21$, where N_R is the number of range segments. Also, in a two-level scheme, N_r turns out to be $N_r = N_{PR} + (N_{PR} + N_{CR}) \times 21$, where N_{PR} and N_{CR} are number of parent range segments and child range segments, respectively.

3.6. Fidelity

To validate the reliability of the compression method, the fidelity (quality) of the reconstructed signal has to be assessed. Mainly, two types of performance measures, quantitative and qualitative, are used for this purpose.

The most commonly used quantitative measures are root mean square (rms) signal-to-noise ratio (SNR_{rms}) of a signal and the peak-signal-to-noise ratio ($PSNR$). The SNR is defined as the ratio of the rms signal power to the rms noise power, (where noise is defined as the difference between the original and reconstructed signals). Similarly, PSNR is defined as the function of rms signal power. If $I(x)$ is the original signal and $\hat{I}(x)$ is the reconstructed signal then

$$SNR_{rms} = \sqrt{\frac{\sum_x I(x)^2}{\sum_x (I(x) - \hat{I}(x))^2}},$$

and

$$PSNR = -20\log_{10}\left(\frac{\sqrt{\sum_x (I(x) - \hat{I}(x))^2}}{2^n - 1}\right),$$

where, n is the number of bits per point (pixel in the case of images) in the signal.

We have also checked the fidelity of the reconstructed signal by: (a) using Cross correlation (CC) between the actual and estimated signal; and (b) examining the power spectra of both the signals.

Cross correlation denotes the statistical correlation between two signals. We have measured the correlation between the original and estimated EEG signals. The cross correlation is estimated as

$$CC = \frac{cov(I(x),\hat{I}(x))}{\sqrt{var(I(x))\,var(\hat{I}(x))}},$$

where $cov(I(x), \hat{I}(x))$ is the covariance between $I(x)$ and $\hat{I}(x)$; $var\ (I(x))$ and $var\ (\hat{I}(x))$ are the variances of $I(x)$ and $\hat{I}(x)$, respectively. Cross correlation plays an important role in judging the resemblance of two signals. We can conclude that the reconstructed signal is very close to the original one as $CC \rightarrow 1$.

Power spectra of both the signals have been taken as the other objective quantitative measure. Apart from comparing the frequency and amplitude changes, power spectra by FFT of EEG signals often convey more information[17]. This is also known as quantitative EEG or quantified EEG or qEEG. Therefore, power spectra by FFT of both the original and the reconstructed signals have been computed for comparison. Similarities in both the power spectra imply that similar clinical conclusions can be drawn from both the signals. The EEG power spectra has been calculated by an FFT routine from the digitized signals[17].

Visual inspection plays an important role in medical diagnosis. Thus, apart from the above, the reconstructed EEG signals have been compared (qualitatively) with the corresponding original one by visual inspection.

4. IMPLEMENTATION AND RESULTS

We have tested our method on three states of EEG (wakeful, REM and slow wave sleep). In each state a datapoint (of 4-s duration) consists of 1024 (w) points. We have examined the performance of the proposed algorithm for two-level (parent and child) partition scheme (section 3.3.1) as well as single-level partition scheme (parent only) using eight transformations (section 3.2.1). The other parameters used are: (i) parent range segment size (b) = 32; (ii) mating pool size for each iteration (S) = 4; (iii) number of iterations for each range segment (T) = 390. The search space reduction ratio ($M/(ST)$) in each range segment is found to be about 5. The original and decoded signals are shown in *Figures 1* and *2*, respectively. In *Figures 3* and *4*, the power spectra by FFT of the original and decoded EEG signals, respectively, are shown.

The cross correlation, PSNR and SNR values of the original and decoded signals (*Table 1*) indicate that the performance of the coding methodology is good in all the three states. The compression ratios (*Table 1*) indicate that at least 85% reduction is achieved in all the data sets.

The algorithm is also tested in single-level scheme, where the parent range segments are not subdivided into its child in any situation. The range segment of two different sizes (namely 16 and 32) have been selected to examine the performance of the algorithm. The other parameters are as above. The results are shown in *Table 2*. The decoded signals using single-level scheme, having range segment sizes 16 and 32 are shown in *Figures 5* and *6*. *Figures 7* and *8* show the power spectra by FFT of the decoded EEG signals in a single-level scheme having range segment size 16 and 32, respectively.

5. DISCUSSION AND CONCLUSIONS

Hinrichs[2] has used an adaptive pulse code modulation scheme and achieved up to 75% of data reduction.

Toraichi *et al.*[3] have reduced the data volume by storing the coefficients of the functions approximating the EEG waveforms. They have also divided the waveforms into different segments but of similar frequencies.

The fidelity of the reconstructed signals, in the proposed method is comparable with that of the other existing methods. On the other hand, the

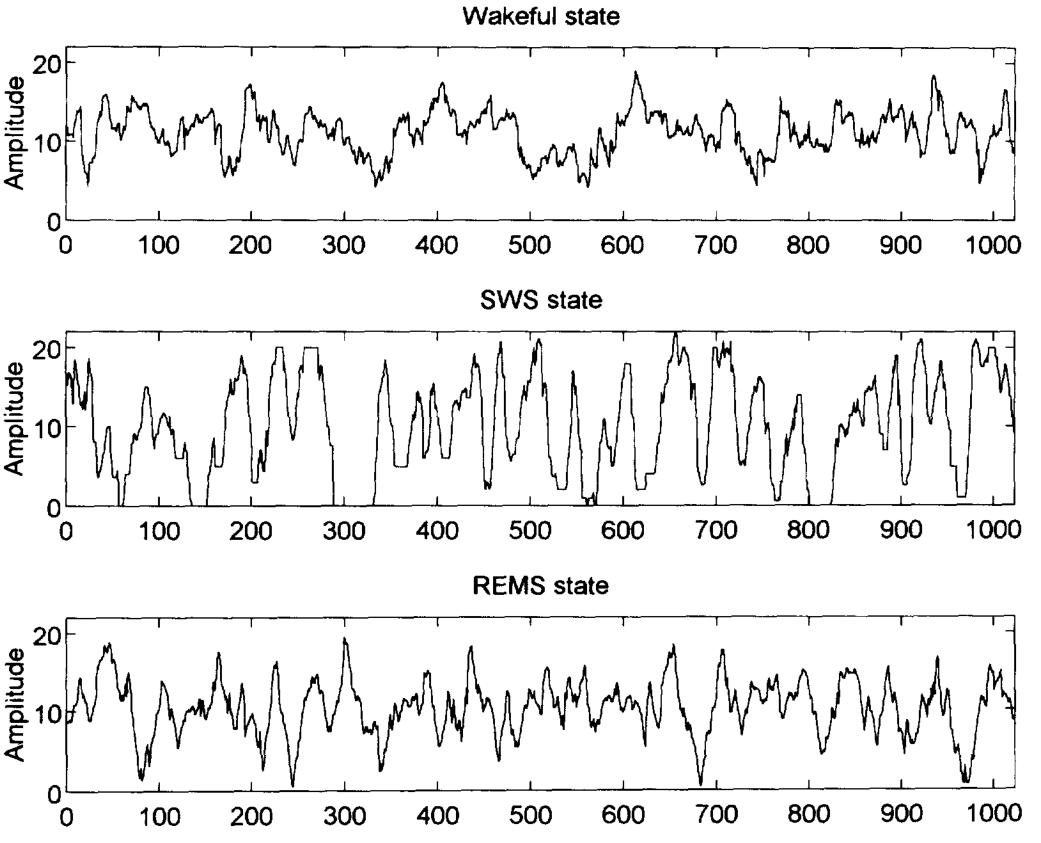

Figure 2 Decoded EEG, using the two-level scheme, in the three states.

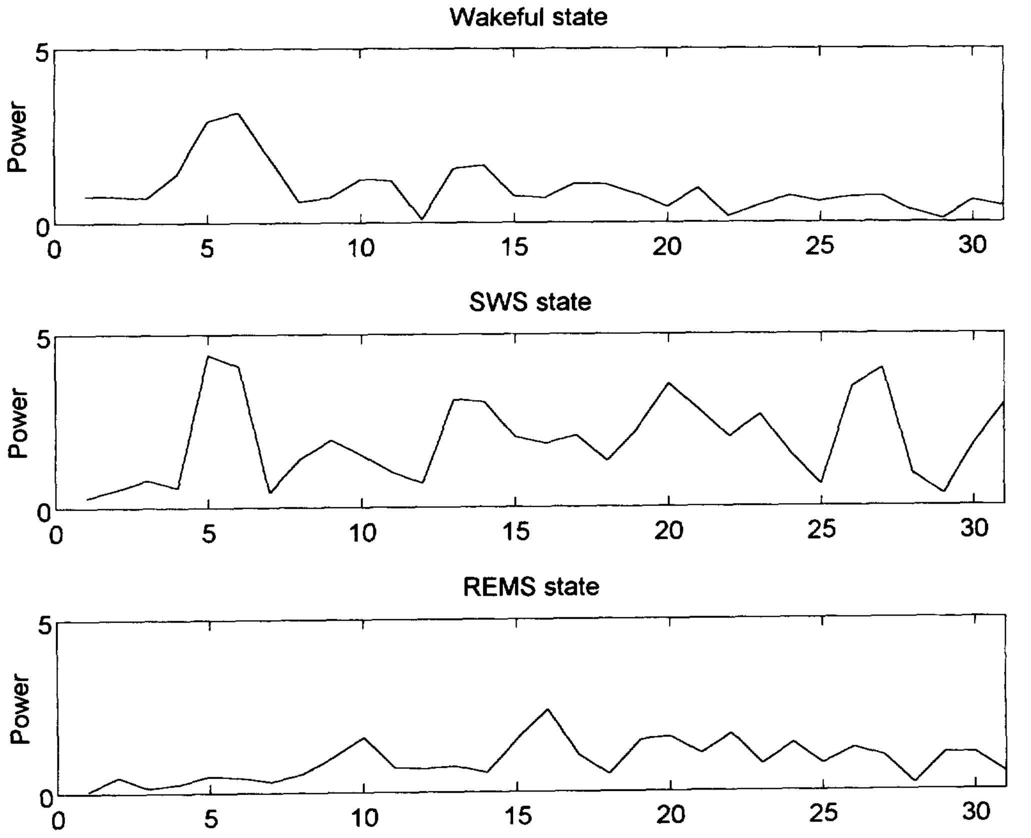

Figure 3 Power spectra by FFT of original EEG in the three states.

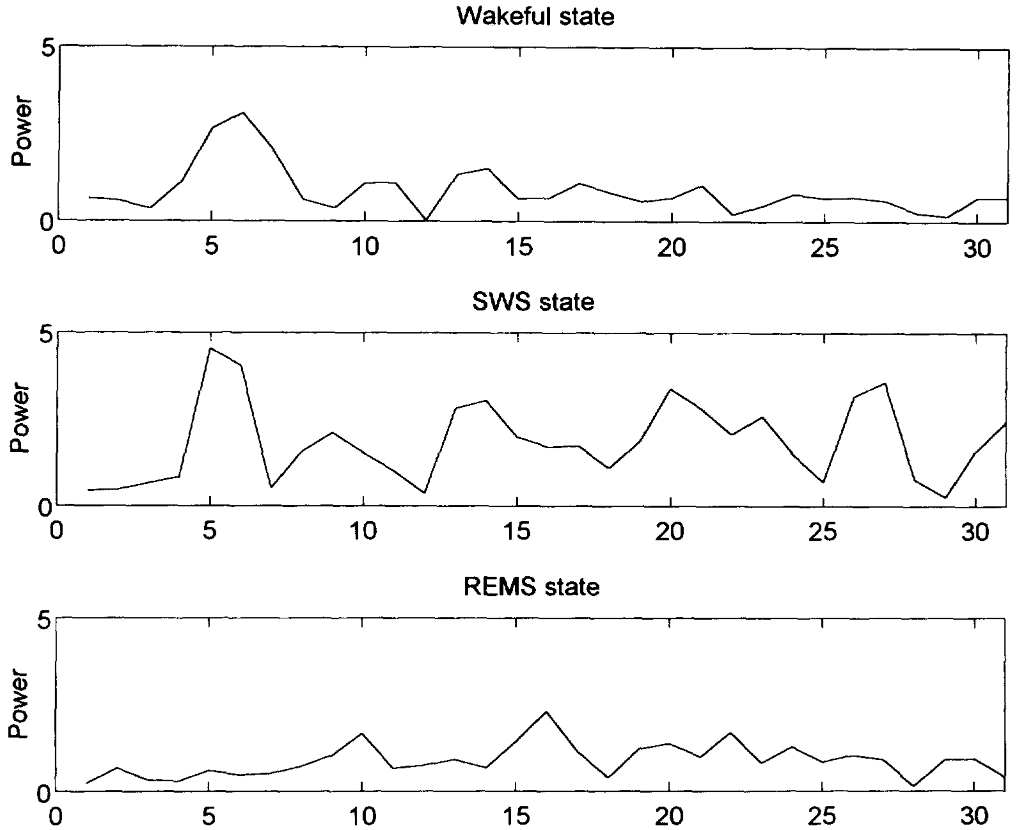

Figure 4 Power spectra by FFT of EEG using the two-level scheme in the three states.

Table 1 Results of the algorithm using the two-level partition scheme

State	b	N_{PR}	N_{CR}	C_R	R_D	*PSNR*	*SNR*	*CC*
Awake	32	13	38	8.35	88.03%	51.2	0.12	0.90
SWS	32	2	60	6.91	85.52%	51.0	0.12	0.98
REMS	32	14	36	8.52	88.26%	52.9	0.10	0.95

Table 2 Results of the algorithm using the single-level partition scheme

State	b	N_{PR}	N_{CR}	C_R	R_D	*PSNR*	*SNR*	*CC*
Awake	16	64	Nil	6.86	85.41%	52.6	0.10	0.93
SWS	16	64	Nil	6.86	85.41%	51.0	0.12	0.98
REMS	16	64	Nil	6.86	85.41%	54.1	0.09	0.96
Awake	32	32	Nil	13.71	92.71%	49.4	0.14	0.86
SWS	32	32	Nil	13.71	92.71%	45.7	0.21	0.92
REMS	32	32	Nil	13.71	92.71%	48.2	0.18	0.85

proposed method seems to perform better in terms of the compression ratios in comparison with Refs [2,3].

The effectiveness of the GA based fractal compression technique, depends upon three factors: (i) the number of points in the search space 2^l; (ii) the size of the initial population S; and (iii) the number of iterations T. The number of iterations needs to be different for different data sets to achieve the near-optimal solution using GAs. The GA-based method can provide efficient and fast encoding by adjusting the initial population S and the number of iterations T in case of large data sets. Moreover, by choosing a proper range segment size, the compression ratio can be increased for a data set containing a large number of data points. Note that by proper range segment size we mean the range segment size which can take care of the finer details of the segment.

One can achieve a high compression ratio by sacrificing the quality of the decoded signal. On the contrary, a high quality decoded signal can be obtained at the cost of the compression ratio. Thus, a trade-off has to be made to obtain a good quality decoded signal with a considerable amount of compression.

The compression ratio is likely to be raised and the search space will be decreased if the number

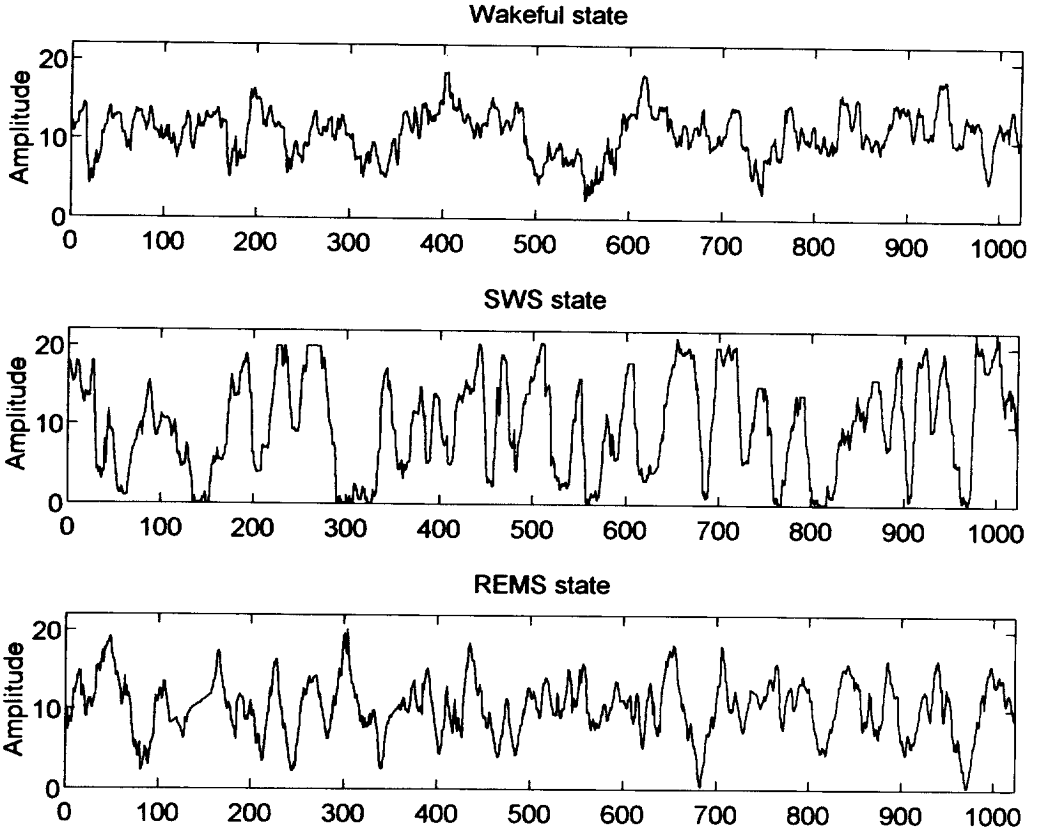

Figure 5 Decoded EEG, using the single-level scheme having range segment size 16, in the three states.

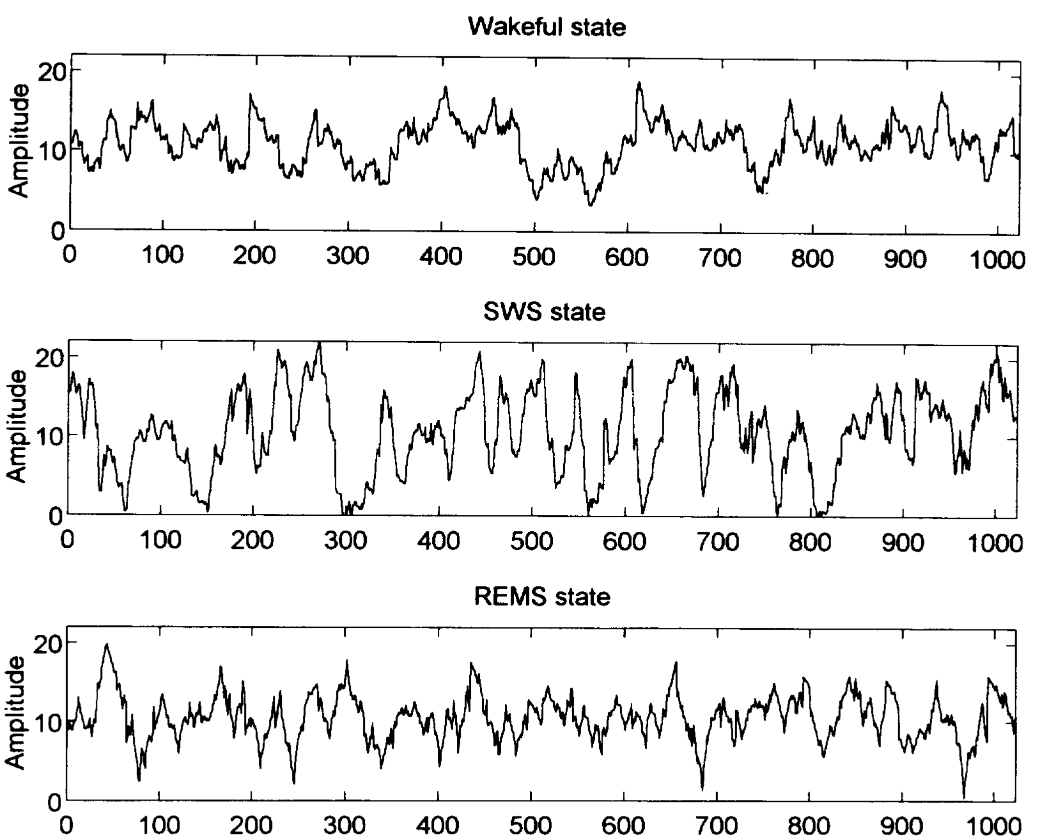

Figure 6 Decoded EEG, using the single-level scheme having range segment size 32, in the three states.

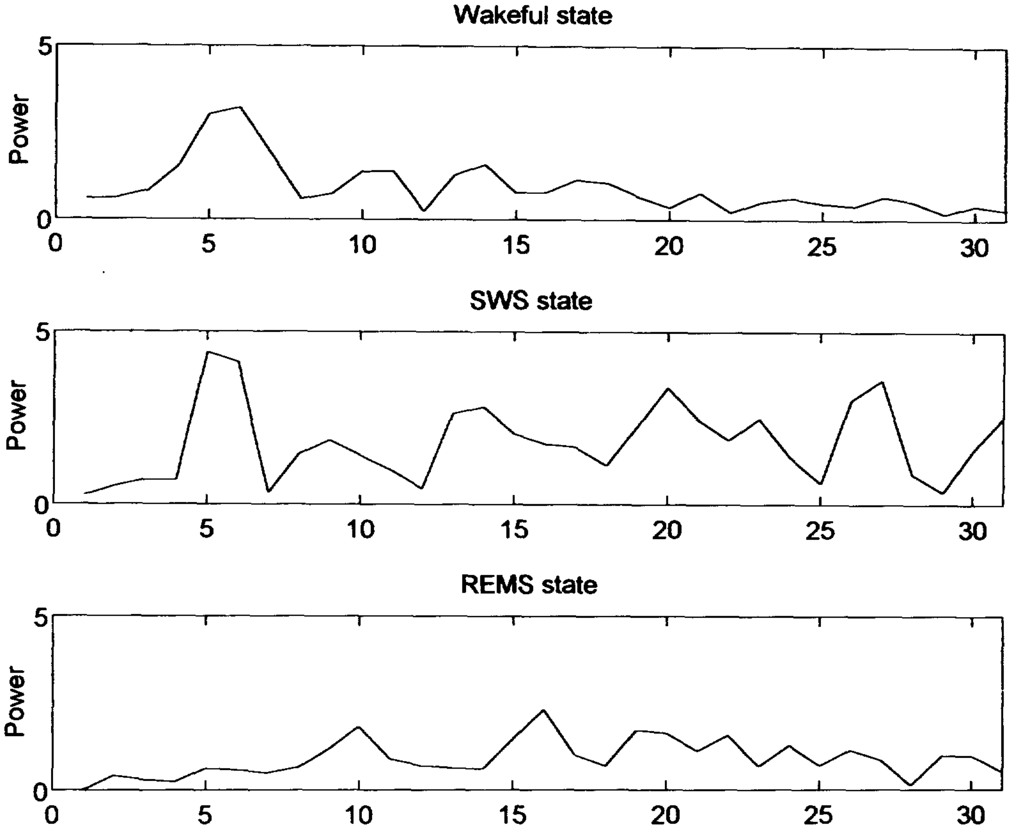

Figure 7 Power spectra by FFT of EEG, using the single-level scheme having range segment size 16, in the three states.

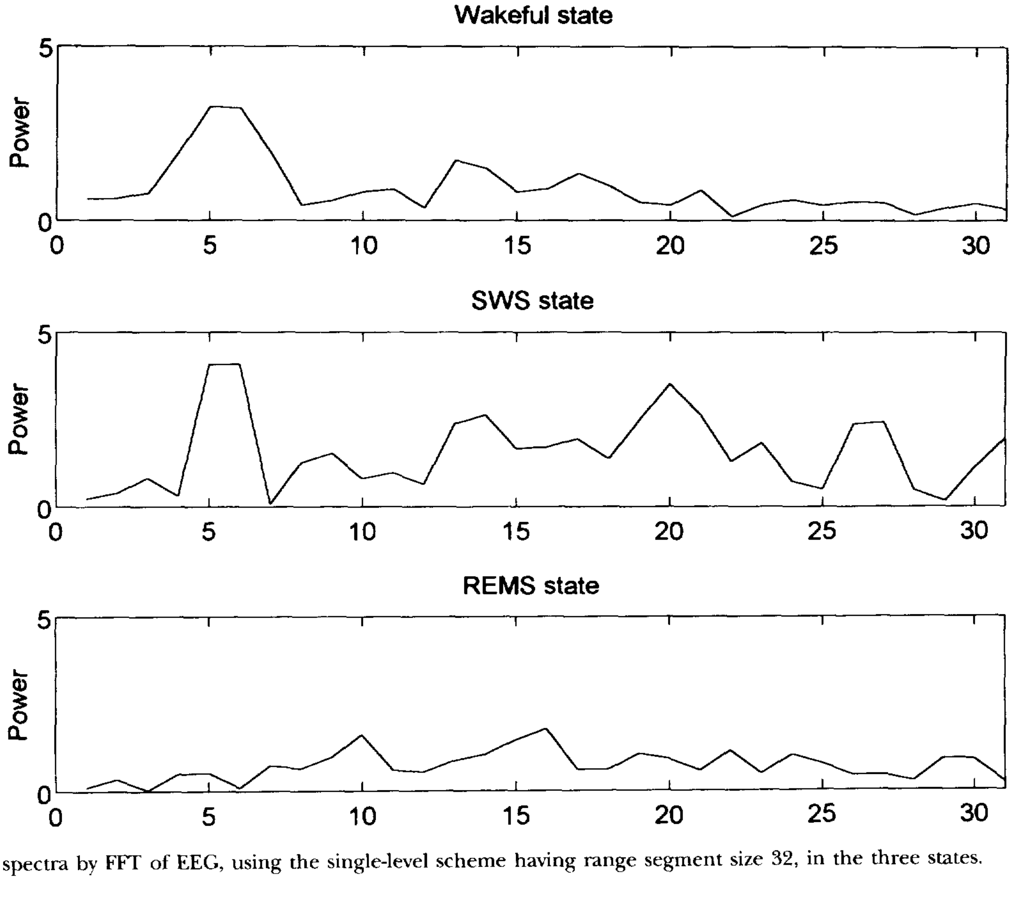

Figure 8 Power spectra by FFT of EEG, using the single-level scheme having range segment size 32, in the three states.

of isometric transformations is reduced. Despite this, it has to be assured that the fidelity of the reconstructed signals should not be affected. One may use fewer than eight transformations for this purpose. In such a case, the selection of transformations will play a major role.

Here, only normal EEG during the three stages of sleep and wakefulness has been tested. This technique may be applied to the EEG in other (abnormal) states, such as epilepsy and depression. Moreover, evoked EEG responses too may be compressed using this method.

In conclusion, using IFS with GA can lead to efficient and reliable compression of EEG signals.

6. SUMMARY

EEG (Electroencephalography) reflects the electrical activity of the brain during the various states of sleep and wakefulness. The EEG signals are quite complex in nature.

EEG compression can help: (a) to augment the storage capacity of collected EEG data for later evaluation or comparison; (b) to facilitate transmitting real-time EEG signals to distant places; and (c) to transmit rapidly and economically offline EEG data over telecommunication networks to remote interpretation centers. Unfortunately, this area has been very much neglected.

Recently, an approach for one-dimensional and two-dimensional signal compression has begun to emerge, based on the theory of Iterative function system (IFS). Compression of signals using IFS is a very promising technique. The theory of coding using IFS and Collage theorem was first proposed by Barnsley[9]. Since then, this technique has been used successfully in image compression by several researchers[10–13]. The method proposed in Ref. [13] (for image compression) is suitably modified here for signal (2-D) compression.

The basic idea of this technique is to approximate the given signal from a set of affine (linear) contractive transformations called IFS. The set of affine contractive transformations, through an iterative process produces a signal called the attractor or the fixed point, which is very close to the target signal. Thus, it is sufficient to store the relevant parameters of the transformations in order to code the signal. The main task is to find the appropriate set of transformations whose attractor approximates the given signal. The aforesaid transformations are obtained using the self-similarities present in the given signal. Here self-similarity implies that the waveform of a particular segment, called the range segment, of the signal is a scaled and shifted version of another segment, called the domain segment, of the same signal. Out of the several possible transformations of domain segments to range segments, in the present article, we have used only a few affine transformations. Thus, the whole problem can be viewed as a search problem where the appropriate domain segment as well as the appropriate transformation are found for a range segment under consideration.

The encoding technique described here also utilizes Genetic Algorithms (GAs) as a search process for finding the self-similarity present in the signal[13]. GAs[14,15] are mathematically modeled algorithms which emulate biological evolutionary theories to solve the optimization problems. GAs attempt to find near-optimal solutions without going through an exhaustive search mechanism. Thus, GAs have an advantage of minimizing the search space and hence the search time. Implementation of GAs as a search mechanism provides fast encoding of EEG signals through IFS.

We have tested our method on three states of EEG (wakeful, REM and slow wave sleep). In each state a datapoint (of 4-s duration) consists of 1024 (w) points. We have examined the performance of the proposed algorithm for a two-level (parent and child) partition scheme as well as a single-level partition scheme using eight isometric transformations.

The cross correlation, PSNR and SNR values of the original and decoded signals indicate that the performance of the coding methodology is good in all three states. The compression ratios indicate that at least 85% reduction is achieved in all the data sets.

In conclusion, using IFS with GA can lead to efficient and reliable compression of EEG signals.

ACKNOWLEDGEMENTS

The authors are glad to acknowledge Dr C. A. Murthy and Dr M. K. Kundu for their helpful suggestions and encouragement during the course of the work.

REFERENCES

1. Dement, W. C. and Kleitman, N., Cyclic variations in EEG during sleep and their relation to eye movements, body motility and dreaming. *Electroenceph. clin. Neurophysiol.*, 1957, **9**, 673–690.
2. Hinrichs, H., EEG data compression with source coding techniques. *J. Biomed. Eng.*, 1991, **13**, 417–423.
3. Toraichi, K., Ohtaki, Y., Ishiyama, Y. and Haruki, R., Compressing EEG data using adaptive function approximation. *Trans. Inst. Electrical Eng. Japan, Part-C*, 1993, **113-C**, 14–20.
4. Tizard, J. and Clark, C. R. Neurocognitive pattern transmission by identity mapping networks. In *Proceedings of the third Australian Conference on Neural Networks (ACNN'92)*. Sydney University, Sydney, Australia, 1992, pp. 248–251.
5. Kalayci, T., Ozdamar, O. and Erdol, N., The use of wavelet transforms as a preprocessor for the neural network detection of EEG spikes. In *Proceedings of 1994 IEEE SOUTHEASTCON '94*. IEEE Press, New York, 1994, pp. 1–3.
6. Pradhan, N. and Dutt, D. N., Use of running fractal dimensions for the analysis of changing patterns in electroencephalograms. *Comput. Biol. Med.*, 1993, **23**, 381–388.
7. Pradhan, N. and Dutt, D. N., Data compression by linear prediction for storage and transmission of EEG signals. *Int. J. Biomed. Comput.*, 1994, **35**, 207–217.
8. Bartolini, F., Cappellini, V., Nerozzi, S. and Mecocci, A., Recurrent neural network predictors for EEG signal compression. In *Proceedings of 1995, 20th International Conference on Acoustic, Speech & Signal Processing, Part-5*. IEEE Press, Piscataway, NJ, 1995, pp. 3395–3398.

9. Barnsley, M. F., *Fractals Everywhere*. Academic Press, New York, 1988.
10. Jacquin, A. E., Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Process.*, 1992, **1**(1), 18–30.
11. Fisher, Y., Jacbos, E. W. and Boss, R. D., Fractal image compression using iterated transforms. In *Image and Text Compression*, ed. J. A. Storer. Kluwer, The Netherlands, 1992, pp. 36-61.
12. Thomas, L. and Deravi, F., Region-based fractal image compression using heuristic search. *IEEE Trans. Image Process.*, 1995, **4**(6), 832–838.
13. Mitra, S. K., Murthy, C. A. and Kundu, M. K., Fractal based image coding using genetic algorithm. In *Pattern Recognition, Image Processing and Computer Vision. Recent Advances*, eds P. P. Das and B. N. Chatterji. Narosa, New Delhi, 1995, pp. 86–91.
14. Goldberg, D. E., *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, 1989.
15. Davis, L. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
16. Bandyopadhyay, S., Murthy, C. A. and Pal, S. K., Pattern classification with genetic algorithms. *Pattern Recognition Letters*, 1995, **16**(8), 801–808.
17. Sarbadhikari, S. N., A neural network confirms that physical exercise reverses EEG changes in depressed rats. *Med. Eng. Phys.*, 1995, **17**(8), 579–582.