Genetic Estimation of Iterated Function System for Fractal modeling

Alfredo Cuzzocrea[1] and Enzo Mumolo[2]

[1] DIA Dept., University of Trieste and ICAR-CNR, Italy,
alfredo.cuzzocrea@dia.units.it
[2] DIA Dept., University of Trieste, Italy,
mumolo@units.it

Abstract. In this paper, we describe an algorithm to estimate the parameters of Iterated Function System (IFS) fractal models. We use IFS to model Speech and Electroencephalographics signals and compare the results. The IFS parameters estimation is performed by means of a genetic optimization approach. We show that the estimation algorithm has a very good convergence to the global minimum. However, the set-up of the genetic algorithm should be properly tuned. In this paper, besides the optimal set-up description, we describe also the best tradeoff between performance and computational complexity. To simplify the optimization problem some constraints are introduced. A comparison with suboptimal algorithms is reported. The performance of IFS modeling of the considered signals are in accordance with known measures of the fractal dimension.

Keywords: Iterated Function Systems, Fractal, Genetic Optimization, Speech, EEG

1 Introduction

The interest in fractal models of signals arises from the observation that many signals, like images and speech, possess some degree of fractal properties [1]. A popular tool for describing and generating fractal objects is Iterated Function System (IFS) [2]. Very complex objects with fractal properties, such as self-affinity, are easily generated with IFS. However, the types of data an IFS can represent well are limited. Thus different IFS models, such as the piecewise self-affine [3] or the hidden-variable [4] models, has been developed in order to represent signals that are neither self-affine nor self-similar. In this paper, we use self-affine and piecewise self-affine IFS fractal models. The estimation of the parameters of an IFS model that reconstructs a given function, which is called inverse problem of the IFS, can be carried out in various ways. In [5] an approach is proposed through wavelet decomposition and the moment method. Alternatively, the inverse problem can be formulated through least squares [6], and therefore a function minimization algorithm can be used. In [4], an optimization approach for the determination of the parameters of IFS models is

used. However, the optimization problem turns out to be quite complex, because the function to be minimized is nonlinear with many local minima and because many parameters are involved. In [7] a heuristic approach based on simulated annealing is used, but some drawbacks are pointed out, such as the very slow convergence to the global optimum and the critical tuning of the optimization algorithm. If constraints are imposed on the variables, however, efficient suboptimal solutions can be obtained and the definition of the best trade-off between complexity and performance becomes the major concern. In [8] a computationally efficient suboptimal inverse algorithm is described, but the search space is greatly limited. The main question would then be whether the imposition of different constraints could lead to better results. The answer to this question is not trivial because it requires extensive experimentation. The main goal of this paper is therefore to evaluate the trade-offs between performance and computational requirements related to different constraints in order to find the best one. The estimation is carried out with Genetic Algorithms (GAs) based optimization, which offers greater robustness with respect to other heuristic optimization algorithms, better convergence to the global minimum, and ease of programming, code maintenance, and updating. Moreover, GA's lead quite naturally to parallel implementations [9].

This paper is organized as follows: section 2 reviews some concepts of fractal geometry and describes the self-affine and piecewise self-affine fractal models, useful to this work. Section 3 gives a brief description of GAs. Sections 4 and 5 describe, respectively, the tuning of the optimization algorithm based on GAs and the experimental results.

2 Fractal Geometry, IFS, Linear and Piecewise Fractal Interpolation

Fractal geometry extends Euclidean geometry and describes objects characterized by a non-integer dimension. If we define $H(R^n)$ as the set of compact subsets of R^n, we can say that a fractal object is an element of $H(R^n)$. The usual distance between two compact subsets is the Hausdorff distance [10], which introduces a metric in $H(R^n)$. Let us consider the bidimensional case, that is, $n = 2$. A succession in the metric space $H(R^2)$ is obtained by an iterative application of affine transformations. An affine transformation is a deformation of elements of $H(R^2)$ realized by means of suitable maps, which can be represented in matrix form as shown in Equation 1.

$$Y = \begin{bmatrix} a & b \\ c & d \end{bmatrix} X + \begin{bmatrix} e \\ f \end{bmatrix} \tag{1}$$

An important class of affine transformations is the class of contractive affine transformations [11]. An IFS is given by a complete metric space, in our case R^2, a distance measure (the Hausdorff distance), and a number N of affine contractive transformations on the metric space $w_n : R^2 \to R^2$, each with a contraction factor s_n. The contraction factor of the IFS is given by $s = \max\{s_n, n =$

$1, 2, \ldots, N\}$. Let us define a function $W : H(R^2) \to H(R^2)$ as reported in Equation 2.

$$W(B) = \bigcup_{n=1}^{N} w_n(B) \tag{2}$$

Then W is itself a contractive transformation on the complete metric space $H(R^2)$. That is, calling $h(\ldots, \ldots)$ the Hausdorf distance, $\exists s,\ 0 < s < 1$, such that Equation 3 holds.

$$h(W(B)) \leq s \cdot h(B, C) \tag{3}$$

for each $B, C \in H(R^2)$. Therefore the succession $A_n = W(A_{n-1})$ converges to a point of $H(R^2)$, called attractor of the IFS or fractal. The fractal $A = W(A)$ can be represented as shown in Equation 4.

$$A = \bigcup_{n=1}^{N} w_n(A) \tag{4}$$

Two algorithms can be used for the generation of a fractal by means of IFS: the deterministic and the random iteration algorithm [3]. The deterministic algorithm is a direct application of the definition of attractor mentioned above. Assume that an initial set A_0 is given. Then the succession A_n, computed as reported in Equation 5, converges to the attractor of the IFS.

$$A_n = \bigcup_{j=1}^{N} w_j(A_{n-1}) \ \ n = 1, 2, \ldots \tag{5}$$

The random iteration algorithm is the following: let us take a given IFS, together with a set of probabilities p_i, one for each transformation. Starting from an initial point x_0, we then select one transformation w_i with probability p_i. A succession of points is then generated as reported in Equation 6.

$$x_n = w_i(x_{n-1}) \tag{6}$$

The succession of points x_n converges to the attractor of the IFS.

For the work described in this paper, a very important result is the so-called Collage Theorem [12, 13], described in (7).

$$\text{given } L \in H(R^2),\ \epsilon \geq 0 \text{ and an IFS such that } h(L, W(L)) \leq \epsilon \to h(L, A) \leq \frac{\epsilon}{1s} \tag{7}$$

where A is the attractor of the IFS and s the contraction factor of the IFS. The importance of this theorem is that it provides a way to test an IFS without the need to compute the attractor. Because we want to represent a function, the affine transformations suitable for us are the so-called shear transformations, reported in Equation 8.

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & 0 \\ c_i & d_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \tag{8}$$

A shear transformation maps vertical lines into vertical lines, and therefore it represents a single-valued function.

The fractal interpolation approach [3] is applicable to self-affine curves. Let us assume that a number of data points $(u_n, v_n),\ n = 0 \dots N$ and $u_n < u_{n+1}$ are given. A set of interpolation points $(x_i, y_i),\ i = 0 \dots M$ and $M < N$ are also given. The interpolation points are a sub-set of the data points with $(x_0, y_0) = (u_0, v_0), (x_M, y_M) = (u_N, v_N)$. Using fractal interpolation, an IFS can be built according to the constraints reported in Equation 9.

$$w_i \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix};\ \ w_i \begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix};\ \ i = 1, \dots M \tag{9}$$

For each interpolation point the system of four equations into five unknowns described in Equation 10 can therefore be written.

$$\begin{array}{l|r} a_i \cdot x_0 + e_i = x_{i-1} & a_i \cdot x_m + e_i = x_i \\ c_i \cdot x_0 + d_i \cdot y_0 + f_i = y_{i-1} & c_i \cdot x_0 + d_i \cdot y_0 + f_i = y_{i-1} \end{array} \tag{10}$$

Equation 10 can be solved fixing one of the coefficients, usually the so-called contraction factors d_i. The are two ways of fixing these coefficients. The first one [14] consists in imposing a given fractal dimension to the final curve. In the second one [8], the contraction factor is computed through a least-squares minimization of the error given by the difference between the original data and the collage.

Once all the five parameters are determined, an IFS describing the data points is derived. We can state that this IFS is a fractal model of the signal $(u_n, v_n),\ n = 0 \dots N$. As stated in the introduction, for many signal a piecewise self-affine fractal model [8] is more appropriate. The piecewise self-affine model, which is merely an extension of the linear self-affine model, contains many degrees of freedom and is therefore extremely flexible. In the following, a brief description of the fractal piecewise model is given.

At each interpolation point $(x_i, y_i), i = 0 \dots M$ an affine map w_i and two points, called addresses in [8], described as $(\tilde{x}_i^1, \tilde{y}_i^1)$ and $(\tilde{x}_i^2, \tilde{y}_i^2)$ are associated. The affine maps w_i map the function between the addresses and the interpolation points. Therefore we can write the condition reported in Equation 11.

$$w_i \begin{bmatrix} \tilde{x}_i^1 \\ \tilde{y}_i^1 \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} \quad w_i \begin{bmatrix} \tilde{x}_i^2 \\ \tilde{y}_i^2 \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad i = 1, \dots M \tag{11}$$

Also in this case according to (7) a set of four equation into five unknown (the IFS's parameters) can be written [8], if the interpolation points and the addresses are known.

The generation of a fractal function by means of the piecewise model must be done only through the deterministic algorithm, because the random iteration algorithm does not guarantee that the points within a given interpolation section are necessarily contained within any address interval.

3 Genetic Algorithms

Genetic algorithms (GAs) are well-known population-based heuristic algorithms for the optimization of complex problems. In contrast with other optimization methods, GAs adopt a parallel approach in the search of the global optimum because during the search a "population" of candidates to the optimum is maintained. Each candidate solution is characterized by its fitness, which is a measure of the goodness of the solution. Some operators, which exploit an analogy with the processes of natural selection and sexual reproduction, are applied to the candidates during the process. Each element of the population is given a "genetic code," usually a string built with characters taken from a given alphabet $V = a_0, a_l, \ldots, a_{n-1}$ called "chromosomes" (this is a first analogy with nature). During the elaboration the population is updated, old individuals being replaced with new ones. At each process iteration, called a "generation," the new solutions, forming the new population, are created applying suitable procedures (operators) to the chromosomes. These procedures are selection (the analogue of natural selection), cross-over (sexual reproduction) and mutation. A basic concept of GAs is the "scheme," which describes the configurations evaluated during each iteration of the algorithm. Note that during the evaluation of a configuration's fitness, GAs also gain some information about all the hyperplanes to which that configuration belongs [15]. The techniques used in the choice of the parents of a new solution is of fundamental importance. The parents are choosen between the best scoring elements of the population, that is, the ones with the extreme values of the objective function. In the simplest procedure, the roulette wheel selection [15], parents are choosen according to a probability distribution based on the fitness of the population's elements. This technique, although very simple and immediate, is strongly inconvenient: it tends to lead to premature convergence owing to the lack of restorative pressure, which in terms of GAs means that the population loses its genotypic diversity. In order to maintain good genotypic diversity we have adopted a simple strategy, which we call in this paper *modified roulette wheel selection (MRWS)* and is described as follows. The elements of the population are chosen as in RWS, but the parents are selected only if the genotypic difference is greater than a given threshold. The genotypic difference between coded strings is computed with the Euclidean distance. We have found that another procedure, the so-called local selection [16], gives better results. In this procedure the elements are arranged on a bidimensional grid and two parents are choosen through a random walk over the grid, starting from the element to be replaced and selecting the elements with the best fitness. This kind of selection, while giving the chance of mixing the genetic patrimony, mantains a good genetic diversity, thus preserving the process from premature convergence. In the context of GAs, the *cross-over operator* realizes the genetic recombination, which is fundamental because it allows for an efficient sampling in the space of the solutions. If the cross-over is too low, its effect is negligible and a premature convergence is eventually introduced. If it is too high, it introduces a schemes disruption that eventually rejects the high fitness schemes. The chromosome is generated concatenating the coded strings

of the parameters. In the classical *1-point cross-over operator*, the cross-over can take place anywhere inside the genetic string, whereas in our *1-point modified cross-over* operator it can happen only at the boundary between groups of genes that represent an IFS transformation. This is like working with a high cardinality alphabet, since the disruptive effects are reduced and the convergence is speeded up. The continuous cross-over is an attempt to balance the two main phenomena of genetic computation, that is, exploration of the solution space and exploitation of solutions that have already been found. Another important GA operator is mutation, which is a change in an hereditary character. In this work, three approaches were considered: the soft, hard, and dynamic ones. *Soft mutation* is performed by incrementing (or decrementing) the integer value of one parameter, chosen with a given probability. *Hard mutation* was performed by changing the value of a parameter with a random one. Finally, *dynamic mutation* [17] is performed by modifying the value of a given parameter by a value that decreases, according to a given perturbation function, as the generations go on. The result of this approach is to make the exploration in the solutions space more and more local as the algorithm proceeds. The last issue concerning GAs is the way to handle constraints. In the case that a given solution does not satisfy a constraint, we added a penalty to the solution itself. Our coding performs a very complex mapping between genotypes and phenotypes, because some parameters are optimized in a genetic way and others are computed in closed form. Therefore, the constraints cannot be applied directly to the values of the genes.

4 Optimal Determination of the IFS Parameters

The parameters involved in linear self-affine models are interpolation points and contraction factors, whereas the piecewise self-affine models are controlled by interpolation points, contraction factors, and the addresses associated with each section of the discrete sequence. In order to reduce the number of variables, the contraction factors are computed in a closed form [8]. All the other parameters are estimated through the minimization of the fitness function reported in Equation (12).

$$E = \sum_{n=1}^{N} [s(n) - c(n)]^2 \tag{12}$$

where $s(n)$ is the input signal and $c(n)$ the collage $W(L)$, defined in Equation 12. Clearly, L is the input frame and N its dimension. It is worth noticing that, in virtue of the Collage theorem, described in (7), the Collage instead of the attractor is used. As the parameters are not correlated to each other, a complete genetic optimization of all the parameters should be performed. However, this is the most critical situation for the GAs. In order to reduce the number of parameters to be optimized, constraints should be imposed on the parameters. The estimated solution is sub-optimal because the constraints limit the search

space. In order to solve the minimization problem described in Equation 12, algorithms have been developed [8] that perform an exhaustive search over the solution space, although they are subject to a number of constraints that keep the complexity low. For an exhaustive algorithm to be computationally affordable, the constraints on the search space must be quite heavy. for example, the algorithm described in [8] explores only 32 out of $1.32E^36$ possible points (in the hypothesis of eight interpolation sections and four address intervals, with eight bit parameters). The introduction of GAs allows us to relax the constraints keeping the computational time sufficiently low. In conclusion, depending on what parameters are included in the optimization process and on the type of constraints, a number of sub-optimal solutions can be obtained. Each solution is therefore a trade-off between required computational power and performances. For example, if the interpolation points are distributed in some known way and we want to determine the optimum d_{i^s} and address positions $(\tilde{x}_i^1, \tilde{x}_i^2)$, the chromosome must be set as reported in Equation 13.

$$\{d_0, \tilde{x}_0^1, \tilde{x}_0^2, \ldots, d_n, \tilde{x}_n^1, \tilde{x}_n^2, \tag{13}$$

Then the genetic algorithm described in Section 3 is used to minimize Equation 12.

In the following, we describe the tuning of the GAs used for the optimal determination of IFS parameters, that is, the optimal setup for coding, selection, mutation, and cross-over. Several experiments are performed to obtain the optimum set-up. Data for experiments is obtained for two class of signals, namely speech and electroencephalography signals. Speech data is extracted from the Artic speech dataset from CMU [18] and electroencephalography data is extracted from the KDD Archive from UC at Irvine, Ca., [19]. Speech data in Artic is sampled at 16 KHz. The signal is divided in 18.75 ms frames or 300 samples long. On the other hand, electroencephalography data is sampled at 256 Hz and the signal is divided in 400 ms frames or 102 samples. One hundred of speech sentences from Artic and one hundred of electroencephalography measurements from the KDD Archive are selected and the results are averaged among them. In Fig. 1 the convergence behaviour is shown. Three selection procedures roulette wheel, modified roulette wheel, and local selection are reported in this figure.

The best selection turns out to be the local selection, in terms of better convergence with smaller difference between minimum and average values. We have implemented and tested a number of variants of the classical genetic algorithm, including non binary coding and cross-over. Fig. 2 reports the average value of the population versus cross-over probability, showing different kinds of cross-over procedures.

The procedure that achieves the lowest values is a variant of the classical 1-point cross-over procedure, namely the *1-point modified operator*, with a probability of 90%. Finally, we also tested several mutation operators. In Fig. 3 the average value of the population versus mutation rate for dynamic and hard mutation is shown.

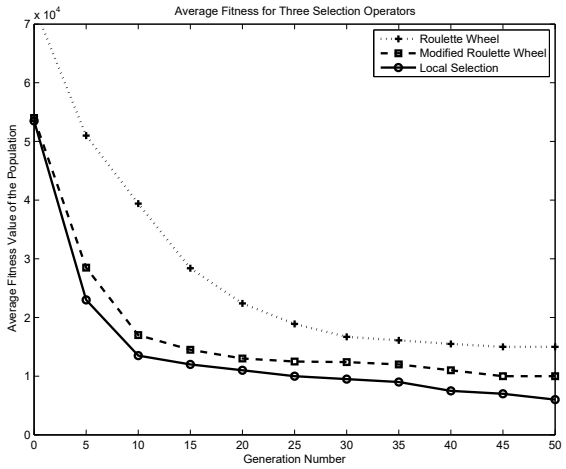

Fig. 1. Average value of the population for the three selection operators as in the figure.

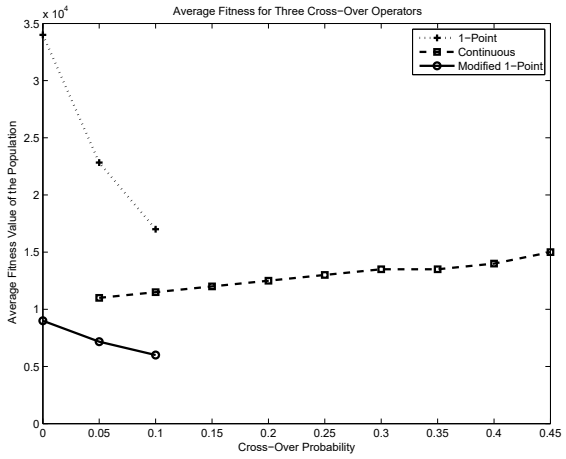

Fig. 2. Average value of the population for the three cross-over operators as in the figure.

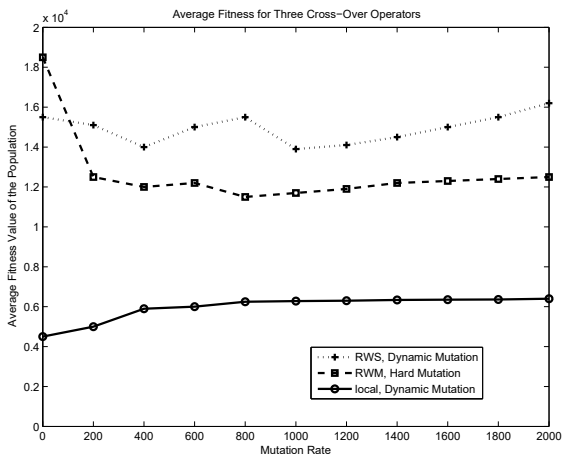

Fig. 3. Average value of the population for the three mutation operators as in the figure.

The best operator we found is a dynamical mutation at a 1/100 rate, where the amount by which a "gene" is changed is a function of time, starting with higher values at the beginnning of the elaboration and decreasing towards the end.

5 Experimental Results

Experimental results are obtained with the same data described before. The segmental SNR is defined as shown in Equation (14).

$$Segmental\ SNR = \frac{10}{M}\sum_{m=0}^{M-1} log\frac{\sum_{n=N\cdot n}^{N\cdot n+N-1} x^2(n)}{sum_{n=N\cdot n}^{N\cdot n+N-1}[x(n)-\hat{x}(n))]^2} \qquad (14)$$

where $x(n)$ is the original signals and $\hat{x}(n)$ is the attractor of the IFS estimated from the original signal by means of genetic optimization. In section 4 the design of the GAs for the determination of the fractal models parameters (local selection, modified one-point cross-over with a 90% probability, dynamic mutation with 1/100 rate) has been made through experimental measurements. We first implemented the linear self-affine fractal interpolation with genetic optimization of the interpolation nodes (called method 1 in the following). For comparison purposes, moreover, the suboptimal procedure described in [4] and called the $Mazel/Hayes$ method has been implemented. In summary, we implemented the following versions of the piecewise self-affine fractal interpolation:

– method 1: linear self-affine fractal interpolation with genetic optimization of the interpolation nodes

– method 2: uniformly distributed interpolation nodes and genetic optimization of the addresses
– method 3: complete genetic optimization of the interpolation nodes and of the addresses
– method 4: uniformly distributed addresses and genetic optimization of the interpolation nodes
– method 5:uniformly distributed addresses and inter-polation nodes, with optimization of the indexing be-tween interpolation sections and addresses
– method 6: genetic optimization of the interpolation nodes and of the indexing between interpolation sections and addresses. The addresses correspond to the interpolation nodes.

Fig. 4 reports the average segmental SNR versus the number of IFS transformations for each of the above methods, and Fig. 5 shows the relative computational complexity of the described methods.

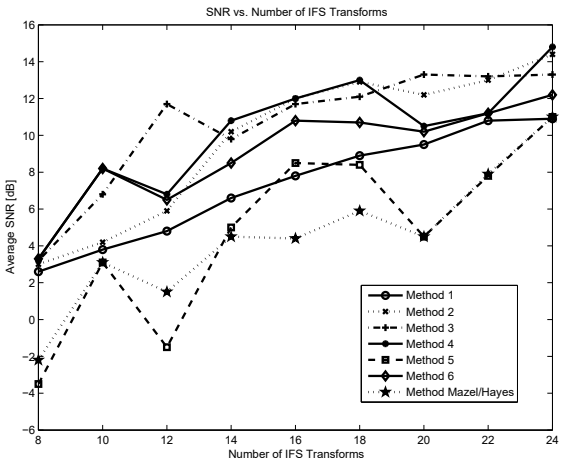

Fig. 4. Segmental SNR versus the number of IFS transformations for each method.

Let's make some considerations on Fig. 4 and Fig. 5. Methods 1 and 5 are worst in terms of SNR performance, as method 1 is not suitable for modelling sequences that are not self-similar, and method 5's constraints limit the search space too much. On the other hand, methods 2 and 4 behave quite similarly, suggesting that their setup is equivalent. The efficient $Hayes/Mazel$ method, described in [8], though better than methods 1 and 5, gives worse performances than all the other methods. Moreover, method 3, which is a complete genetic optimization of the parameters, seldom gives very high results, supporting the conclusion that convergence, in this case, is quite difficult to reach. In any case, Fig. 5 shows that method 3 is the most expensive piecewise method in terms of required computational power. In summary, comparing the different methods,

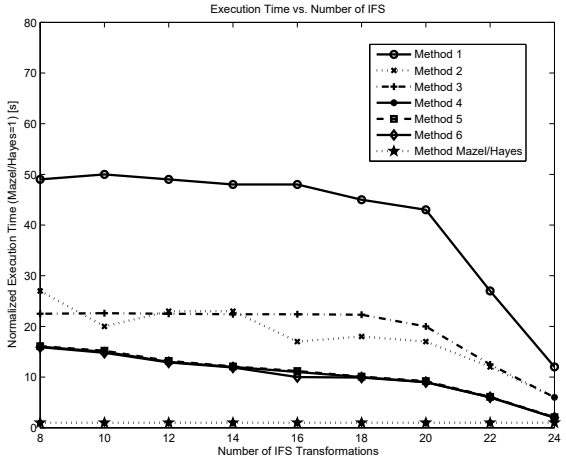

Fig. 5. Normalized execution time for each method.

method 6 offers high performances without requiring too much computation, and thus it represents the best tradeoff between SNR, compression ratio, and computational complexity. We then used method 6 to model the data sequences.

The first results concern the modelling of speech. In Fig. 6 a speech sentence from the Artic database is reported. The signal is divided into frames which are separatly modelled with IFS.

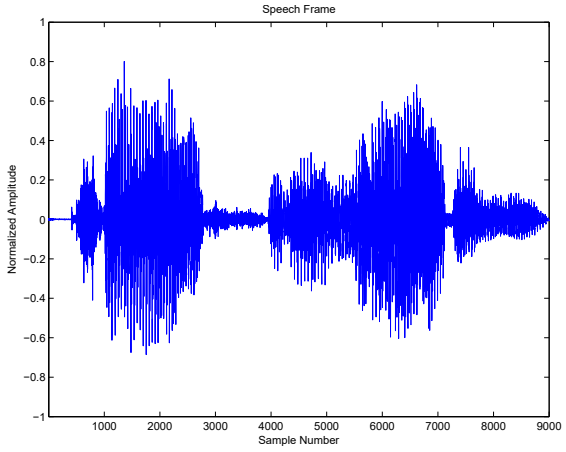

Fig. 6. Waveform of one speech sentence.

The quality of IFS modeling is shown by reconstruction the original signal and comparing the two. In this example, 24 IFS maps for each frame were used. The segmental SNR (frame by frame) computed according to Equation 14 is reported in Fig. 7.

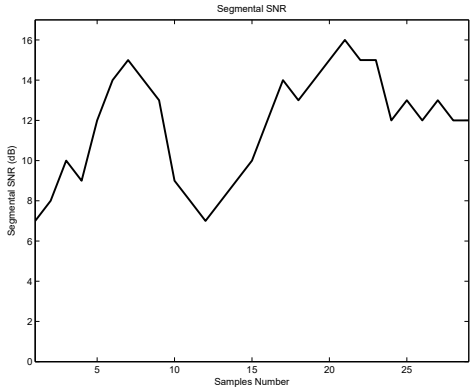

Fig. 7. Segmental SNR of the speech sencence reported in Fig.6

Piecewise fractal model is then applied to a Electroencephalography (EEG) signals [20], which is a measure of the electrical activity of the brain. Using method 6 with 16 IFS transformations per 102 samples frame and 30 generations, we obtained an average SNR, among all the extracted recordings, of about 19 dB. The waveform of a section of an Electroencephalographc recording from one channel is reported in Fig. 8.

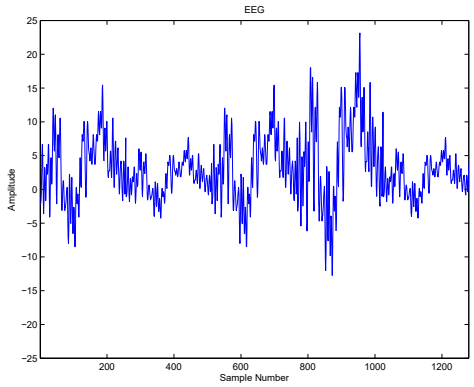

Fig. 8. Waveform of an Electroencephalographic signal extracted from the KDD dataset

Fig. 9 shows the frame-by-frame SNR results related to an analysis of an EEG signal. With the same parameters as before, that is, 16 IFS for each frame and 102 samples per frame, we obtained an average SNR of about 26 dB.

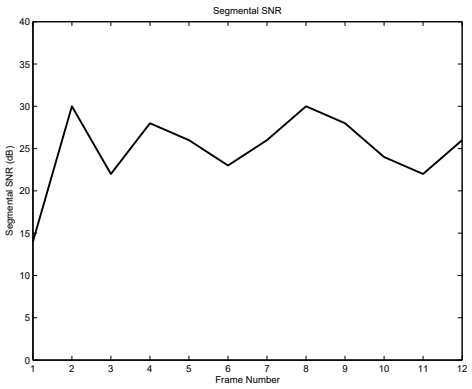

Fig. 9. Segmental SNR of the signal reported in Fig.8.

6 Final Remarks and Conclusions

We have described an optimization approach to the determination of fractal models of speech and electroencephalography signals. Very good Segmental SNR results were obtained by estimating the IFS fractal models with genetic algorithms. It is worth noting that the SNR results depend on how much the signals possess fractal properties. This can be measured by computed the fractal dimension of the signals. There are several different definitions of fractal dimension [16, 21]. In speech signals the fractal dimension depends on the phoneme type. The values of fractal dimension can range normally from 1.2 to 1.6 for vowels and from 1.6 to 1.8 for consonants [16]. For electroencephalography signals the values of fractal dimension can range normally from 1.7 to 1.9 depending on the pathological conditions of the subject [22]. Our electroencephalography data is related to alcoholic subjects and thus the fractal dimension is expected to be high. Turning to the Segmental SNR reported in Figure **??**, **??**, this is the reason why the values of SNR are higher for the electroencephalography data then speech data we considered.

A number of constraints on the search space were explored and the best tradeoff are experimentally defined. The proposed method gives much higher SNR performaces than suboptimal techniques. On the other hand, even if suboptimal approaches are the most efficient, the proposed method requires the same order of computational effort as the number of IFS transformations in-

creases. The problem of efficient coding of the estimated parameters for storing or ransmission purposes are not considered in this paper.

References

1. Mandelbrot, B.B.: The Fractal Geometry of Nature. W. H. Freeman and Company (1977)
2. Jaros, P., Maslanka, L., Strobin, F.: Algorithms generating images of attractors of generalized iterated function systems. Numerical Algorithms **73**(2) (2016) 477–499
3. DRAKOPOULOS, V., BOUBOULIS, P., THEODORIDIS, S.: Image compression using affine fractal interpolation on rectangular lattices. Fractals, World Scientific **14** (2006) 1–11
4. Akhtar, N., Prasad, M.G.P.: Graph-directed coalescence hidden variable fractal interpolation functions. Applied Mathematics **07** (2016) 1–11
5. Abenda, S.: Inverse problem for one-dimensional fractal measures via iterated function systems and the moment method. Inverse Problems **6**(6) (1990) 885
6. Sarafopoulos, A., Buxton, B.: Resolution of the inverse problem for iterated function systems using evolutionary algorithms. In: IEEE International Conference on Evolutionary Computation, CEC 2006, part of WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006. (2006) 1071–1078
7. Dimri, V.P., Srivastava, R.P., Vedanti, N.: Fractal Models in Exploration Geophysics, Volume 41: Applications to Hydrocarbon Reservoirs - Handbook of Geophysical Exploration Seismic Exploration. 1st edn. Elsevier Science, San Diego, USA (2012)
8. Mazel, D.S., Hayes, M.H.: Using iterated function systems to model discrete sequences. IEEE Trans. Signal Processing **40**(7) (1992) 1724–1734
9. López, N., Rabanal, P., Rodríguez, I., Rubio, F.: A formal method for parallel genetic algorithms[1]. In: Proceedings of the International Conference on Computational Science, ICCS 2015, Computational Science at the Gates of Nature, Reykjavík, Iceland, 1-3 June, 2015, 2014. (2015) 2698–2702
10. Arutyunov, A.V., Vartapetov, S.A., Zhukovskiy, S.E.: Some properties and applications of the hausdorff distance. J. Optimization Theory and Applications **171**(2) (2016) 527–535
11. Barnsley, M.F.: Fractals everywhere. Academic Press (1988)
12. Honda, H., Haseyama, M., Kitajima, H., Matsumoto, S.: Extension of the collage theorem. In: Proceedings 1997 International Conference on Image Processing, ICIP '97, Santa Barbara, California, USA, October 26-29, 1997. (1997) 306–309
13. Øien, G.E., Baharav, Z., Lepsøy, S., Karnin, E.D.: A new improved collage theorem with applications to multiresolution fractal image coding. In: Proceedings of ICASSP '94: IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, South Australia, Australia, April 19-22, 1994. (1994) 565–568
14. Wadströmer, N.: An automatization of barnsley's algorithm for the inverse problem of iterated function systems. IEEE Trans. Image Processing **12**(11) (2003) 1388–1397
15. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley (1989)
16. Maragos, P., Potamianos, A.: Fractal dimensions of speech sounds: Computation and application to automatic speech recognitiona. J. Acoust. Soc. Am. 105 (3), March 1999 **105**(3) (1999) 1925–1932

17. Chan, K.Y., Fogarty, T.C., Aydin, M.E., Ling, S., Iu, H.H.C.: Genetic algorithms with dynamic mutation rates and their industrial applications. International Journal of Computational Intelligence and Applications **7**(2) (2008) 103–128
18. : Cmu artic database. `http://www.festvox.org/cmu_arctic/index.html/`
19. : Uci kdd archive. `http://kdd.ics.uci.edu/databases/eeg/eeg.html`
20. Teplan, M.: Fundamentals of eeg measurement. Measurement science review **2**(12) (2002) 1–11
21. Paramanathan, P., Uthayakumar, R.: An algorithm for computing the fractal dimension of waveforms. Applied Mathematics and Computation **195**(2) (2008) 598–603
22. Truong, Q.D.K., Ha, V.Q., Toi, V.V.: Higuchi fractal properties of onset epilepsy electroencephalogram. Comp. Math. Methods in Medicine **2012** (2012) 461426:1–461426:6