On a New Approach to Time Series Tracking

Karinne Ramírez-Amaro and Juan Carlos Chimal-Eguía

Centro de Investigación en Computación, Av. Juan de Dios Bátiz s/n Unidad Profesional Adolfo López Mateos Col. Nueva Industrial Vallejo, Mexico City, 07738, Mexico,
kramireza@ipn.mx, chimal@cic.ipn.mx

Abstract. In this paper we present a new approach to time series tracking using some main ideas from the paper published by Donald Michie in 1963. One of those ideas stands that is easier to solve a complex problem by dividing it into many easy subproblems that are sequentially linked. Using this idea we divide the time series in small parts, this divisions will provide information of the amplitude behavior in order to "track" the time series. The obtained results indicate that using this new approach it is possible to track the time series with high accuracy and also deduce more information about the time series. This approach could be applied to any natural phenomena which can be represented as time series.

1 Introduction

In 1963, Donald Michie published a paper that describes a trial and error machine which learns to play the game "Noughts and Crosses" [1]. This device was initially constructed from matchboxes and colored beads as shown in Fig. 1. This machine is a perfect example of the game theory application. The game theory is an interesting topic since the games provided a microcosm of intellectual activity. Those thought processes which we regard as being specifically human accomplishments, such as learn from experience, inductive reasoning, argument for analogy, the formation and testing of hypotheses, are brought to into play even in the simple games of mental skill.

As an example of this, the matchbox machine was used for a particular mental activity of trial and error learning, and the mental task used, was the game of "Noughts and Crosses", sometimes known as "tic-tac-toe", this game represent a sequential decision process. Michie argued that a computer program could be enabled to improve its performance through its own accumulating experience [1].

In 1968 the matchbox machine was known as "Boxes" [2], first this project was undertaken as a "fun project", but there were more serious intentions to demonstrate the principle that states: "it may be easier to learn to play many easy games than a difficult one" [1]. Consequently, it may be advantageous to decompose a game into a number of mutually independent sub-games even when relevant information could be put out of reach in this process. The principle is related to the method of subgoals in problem-solving [3] but different in one

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 205-215

Received 03/03/07
Accepted 09/04/07
Final version 22/04/07

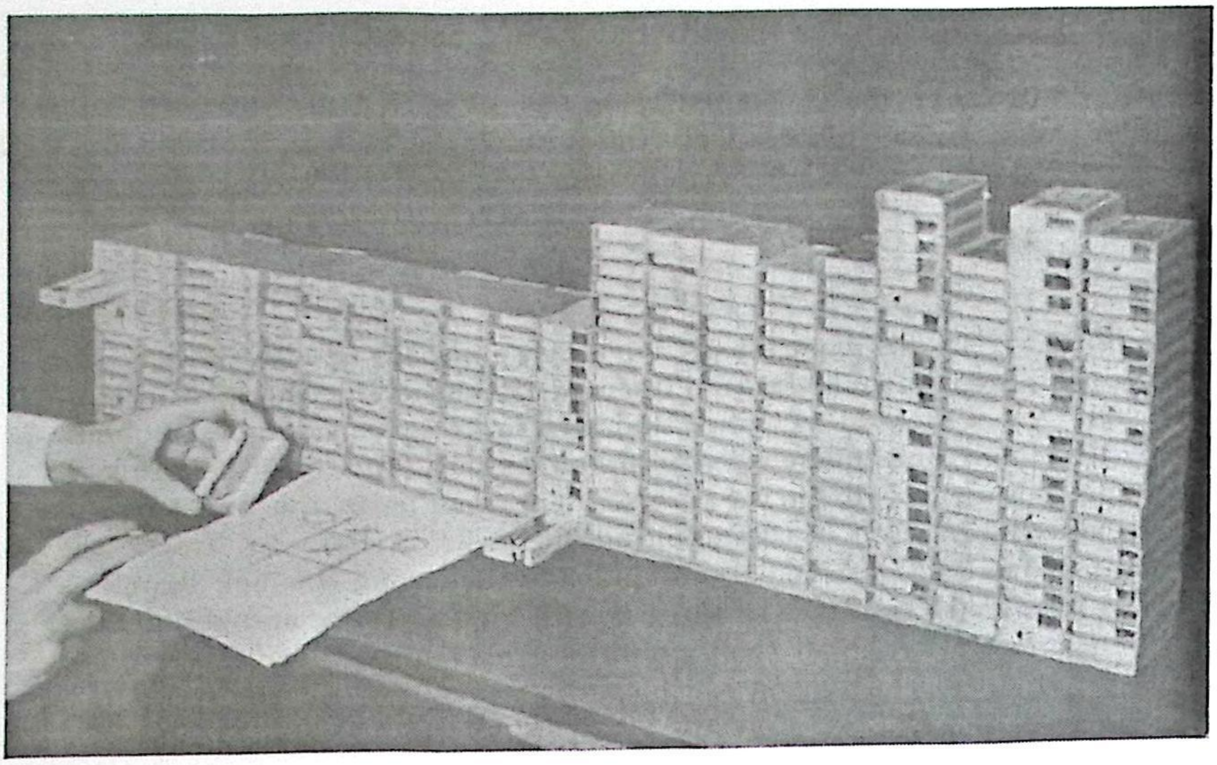

Fig. 1. The original matchbox machine known as "MENACE".

fundamental aspect: subgoals are linked in series, while sub-games are played in parallel.

The Boxes algorithm also was used in adaptive control problem in particularly the double pendulum [2]. In the adaptive control situation, where the states variables are real numbers, the large game is infinitely large, and then the sacrifice of information entailed in boxes approach is correspondingly extreme.

In this article, we used the one of the main ideas of the Boxes algorithm such as; learn to play many easy problems than a difficult one. This means that would be easier to learn the behavior of small parts of the time series than learn the whole behavior of the time series. Each small part is contained in the so-called "boxes" and they provide information about the amplitude behavior. In this work, we use this idea in order to track the dynamics of some time series with different behaviors.

The outline of this article is the following: firstly in section 2, we define the concept we used for time series and the approach of the general ideas of Boxes algorithm for time series tracking, in section 3, we explain the logic used and the implemented methodology; in section 4 we explain the obtained results and finally in section 5 some conclusions and future work are given.

2 Implementing the Boxes Algorithm to Tracking Time Series

We used the main idea from the Boxes algorithm explained before of divide the problem into small parts. Instead of matchboxes we used intervals which contain information about the amplitudes of the time series in order to track it. We are interested in the study of the time series with new techniques to learn its dynamics.

A time series is a sequence of data points, measured typically at successive times. The time series have information about the independent variables of a system which determines its dynamics. In other words, a time series is a sequence of values over the time of a system $x(t)$ which registers a sequence of experimental values [4][6][5]

$$x(t_1), x(t_2), x(t_3), ..., x(t_n) \tag{1}$$

for some interval $t = n$ with $t_0 < t_1 < ... < t_n$ [4], see Fig. 2 to observe an example of a time series.

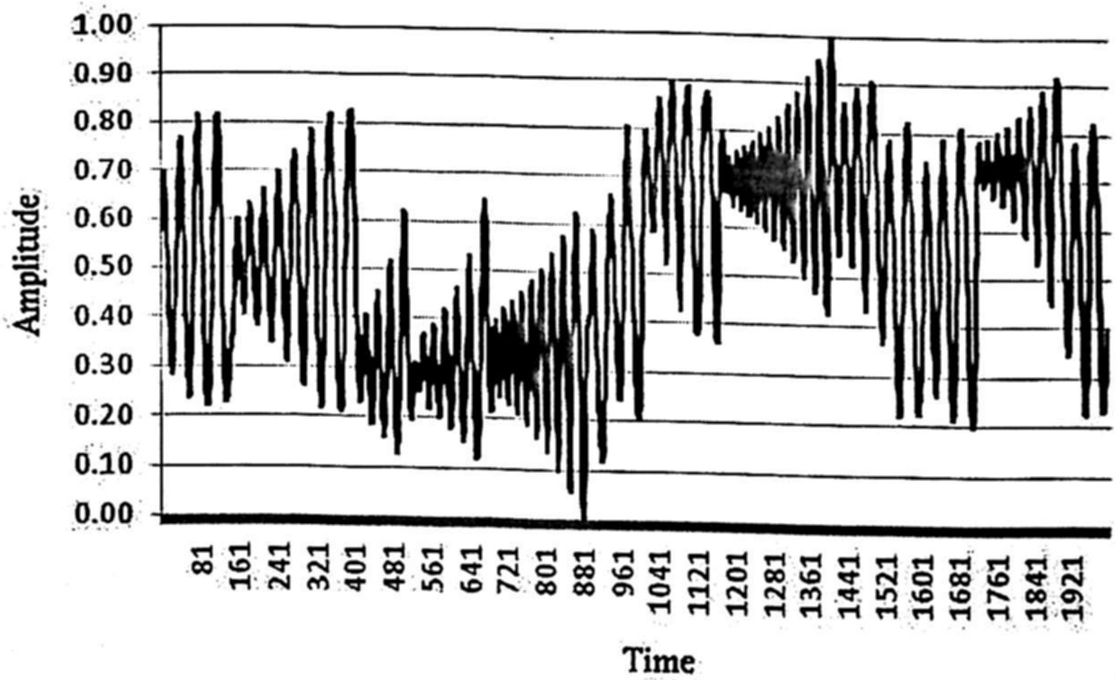

Fig. 2. Example of Lovaina time series.

In this work, we demonstrate that it is easier to learn the behavior of small amplitude intervals of the time series than trying to learn the behavior of the whole time series. That is because, the small intervals provided information from some specific amplitude of the whole time series and that allows a better learning of the behavior of it. We use this idea to generate a time series that tracks the behavior of the original one and that is what we call time series tracking, i.e., we designed an algorithm that imitate the dynamics of a desired time series. The difference between our approach and the divide and conquer algorithm is that the last one works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem [8].

During the implementation of this approach we use a pseudo-random number generator therefore, this is a randomized algorithm or probabilistic algorithm. This algoritm employs a degree of randomness as part of its logic. In common practice, this means that the machine implementing the algorithm has access to a pseudo-random number generator. The algorithm typically uses the random bits as an auxiliary input to guide its behavior, in the hope of achieving good performance in the "average case". Formally, the algorithm's performance will be a random variable determined by the random bits, with (hopefully) good expected value; this expected value is called the expected runtime. The "worst case" is typically so unlikely to occur that it can be ignored [7].

3 On the Approach for Tracking a Time Series

As we mention before the idea behind this approach is related with a principle of subgoals in order to solve a complex problem by dividing the main problem into many subproblems that are sequentially linked but at the same time these subproblems are solved in parallel. Therefore, in this article we are using that principle to analyze the amplitude from the time series dividing it into intervals as is shown in the Fig. 3.

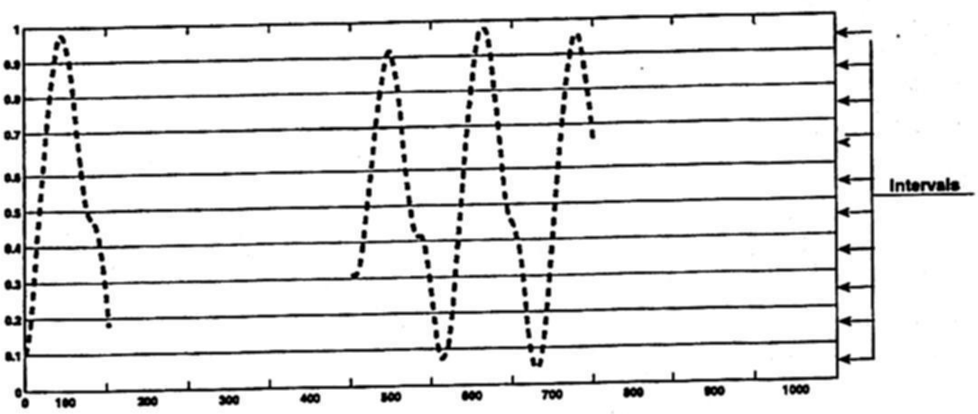

Fig. 3. Example of how we obtain new information from the time series. Additionally we can observe the division of the time series into ten intervals with the same width

The methodology used in this work follows the next steps:

– Normalize the original time series into the interval [0, 1].

- Divide the normalized time series in eight intervals.
- Generate from a random variable the time series that tracks the dynamics of the original one.
- Compute the tracking error between the original time series vs. the computed.

The above steps can be exemplified in the next Figure 4.

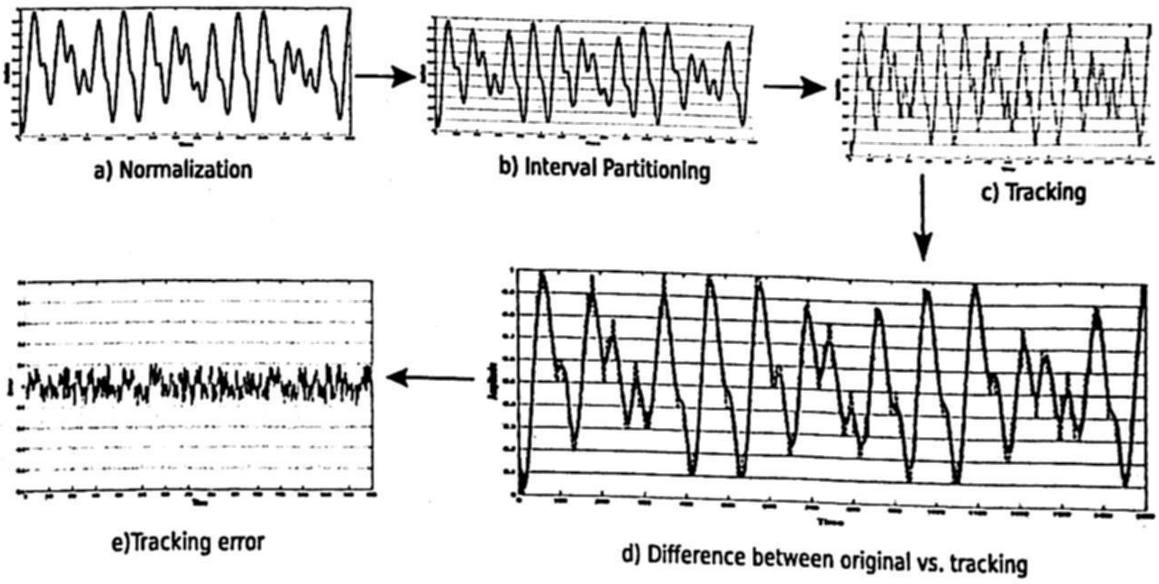

Fig. 4. Example of the metodology used for tracking time series. a) Normalization of the time series, b)Division of the original time series, c) Computation of the tracking time series, d) Difference between the both time series, e) Compute of the tracking error.

3.1 Normalization and Division

Broadly, normalization (also spelled normalisation) is any process that makes something more normal, which typically means conforming to some regularity or rule, or returning from some state of abnormality. In this approach normalization is very important due to the fact that we want to compare two or more time series from differents behaviors to prove the performance of this technique.

The following step is the division, this means the decision of how many intervals we are going to consider to divide all the time series that will analyze. From the mathematical definition an *interval* $[x]$ is a conected subset of $\mathbb{R}$. Even when the interval is not closed, we shall keep to the notation $[x]$. The *lower bound* $lb\,([x])$ of an interval $[x]$, also denoted by $\underline{x}$ is defined as

$$\underline{x} = lb\,([x]) \triangleq \sup\left\{a \in \mathbb{R} \cup \{-\infty, \infty\} \,|\forall x \in [x]\,, a \leq x\right\} \tag{2}$$

Its *upper bound* $ub([x])$, also denoted by $\overline{x}$, is defined as

$$x = ub([x]) \triangleq \inf\{b \in \mathbb{R} \cup \{-\infty, \infty\} \,|\forall x \in [x], x \le b\} \tag{3}$$

Thus, $\underline{x}$ is the largest number on the left of $[x]$ and $\overline{x}$ is the smallest number on its right. The *width* of any non empty interval $[x]$ is

$$w([x]) \triangleq \overline{x} - \underline{x} \tag{4}$$

In order to tests our approach we choose arbitrary the following eight intervals:

$$[-0.001, 0.125], [0.125, 0.25], [0.25, 0.37], [0.37, 0.5]$$

$$[0.5, 0.625], [0.625, 0.75], [0.75, 0.875], [0.875, 1] \tag{5}$$

Where the $lb([x])$ are open intervals and the $ub([x])$ are closed intervals.

3.2 Generation of the Time Series Tracking and its Error

The goal of this subsection is to find and extract information from the amplitude of the time series. In order to do that, we generate a new time series that tracks the behaviour of the original time series. The procedure we implemented is briefly shown in Fig. 5.

```
1 for i=1 to lastPointTimeSeries -1
2     x(i)=actualPoint
3      xnext(i+1)=nextpoint
4      if x(i)>lb([x]) and x(i)<ub([x])
5          if this is the first time this interval is visited then
6                  compute randPoint=rand([lb([x])], [ub([x]))
7                  compute the error= x(i)-randPoint
8          else means that this interval was visited previously then
9                   previosError=error(i-1)
10                   if previosError>0.0009 then
11                          newlb([x])=randPoint
12                          newub([x])=ub([x])
13                          compute randPoint=rand([newlb([x])], [newub([x])])
14                          compute the error=x(i)-randPoint
15                  end if
16          end if
17      end if
18 end for
```

Fig. 5. Main procedure to extract information from the original time series.

Firstly, we will start to moving a pointer from the first point of the original time serie to the last point (lines 1-3). Then we do the validation of the interval and only the points from the time series that belongs to an specific intervalo ($[x]$) were considered in its analysis (lines 4-17). Thereafter the lines 5 to 8 only are visited when the first points belonging to certain interval enter for first time, because the lines 8 to 16 represent the training to get optimal rand numbers to future points. In line 9 there is an interesting validation, we notice that if we define a minimal error as in this case 0.0009 we obtaint better results than just keep it trainig without this validation and we define this error because it give us best results. If some points revisits the same interval and enter to line 11 means that we need to redefine the *lower bound* $lb([x])$ of the interval ($[x]$) in order to enclose the new interval $([newlb([x])], [newub([x])])$ and obtain a better result than the last computed.

The measure that we implemented in order to verify the performance of this approach was the root mean square error (RMSE) which is a frequently-used measure of the difference between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. In this particular case we used to measure the tracking error between the original time series and the computed one. The RMSE is defined

$$\sqrt{\frac{\sum_{i=1}^{n}(x_o - x_t)^2}{n}} \tag{6}$$

where x_o means the original points from the time series, x_t means the tracking series and n means the total number of the both time series [5].

4 Results

In order to verify the performance of this algorithm, we probed it with different time series such as: periodic, quasiperiodic, chaotic, complex and stochastic systems [5]. The data base we used to the time series tracking was taken from the reference [6] and is briefly shown in the Fig.6.

The software used in our simulations was Matlab 7.0®and the specification of the hardware we used is a CPU Pentium 4 with 3.0 GHz and 512 MB of RAM.

Some of the numerous experiments we performed are showed in the Fig. 7 to 10, and we are going to briefly explain each of the figures. Firstly in fig. 7 a) we depicted the zoom of the last 200 points of both time series, in order to see the difference between the original time series and the tracking one and in b) we can observe the errors at any time betwen the original point and the computed one obtained by the proposed algorithm and also we obtain the tracking error using the measure RMSE defined in the equation 6 and the error obtain of the tracking the behaviour from the Sine time series is equal to 0.0007 meaning that the tracking time series is resemblance to the original one as we can observe in that figure.

As we can see from Fig. 6 the time series plotted in Fig. 8 has a complex behavior which means that displays variation without being random and as can

Time Series	Dynamical Behavior
Sine Vanderpool	Periodic
Qperiodic2	Quasiperiodic
Mackey-Glass (*) Logistic Lorenz Rossler Ikeda Henon Cantor Tent	Chaotic
Laser Down Jones Kobe HIV DNA El niño Lovaina(*)	Complex
Browmian Motion White Noise(*)	Stochastic

Fig. 6. Some of the time series used to prove our technique. The time series marked with (*) are the ones used to obtain the experimental results showed in this paper.

be seen in this figure the difference between the original time series and the tracking one is minimum and this observation could be quantify by calculating the RMSE wich is equal to 0.00093. It is important to mention that the plotted points in this figure are the 300 last points in order to see the error between both series.

In the Fig. 9 we observe the last 200 points of the Mackey-Glass time series vs. the tracking one. This time series has a chaotic behaviour which means that it describes the behavior of certain nonlinear dynamical systems that under certain conditions exhibit dynamics that are sensitive to initial conditions (popularly referred to as the butterfly effect). As a result of this sensitivity, the behavior of chaotic systems appears to be random, because of an exponential growth of errors in the initial conditions. Despite of its behavior the results indicate that it is possible to track its dynamics and the RMSE obtained is equal to 0.018.

Finally, in Fig. 10 we observ the results obtained from the White Noise time series wich has a stochastic behavior. A stochastic process is one whose behavior is non-deterministic in that a state does not fully determine its next state. Therefore, in this figure we observe that the tracking is very accuracy due its behavior with a RMSE equal to 0.0024.

All the result indicates that it is possible to tracking time series from different behaviors obtaining in all the cases a very low error between the original and the time series tracking i.e., the RMSE.

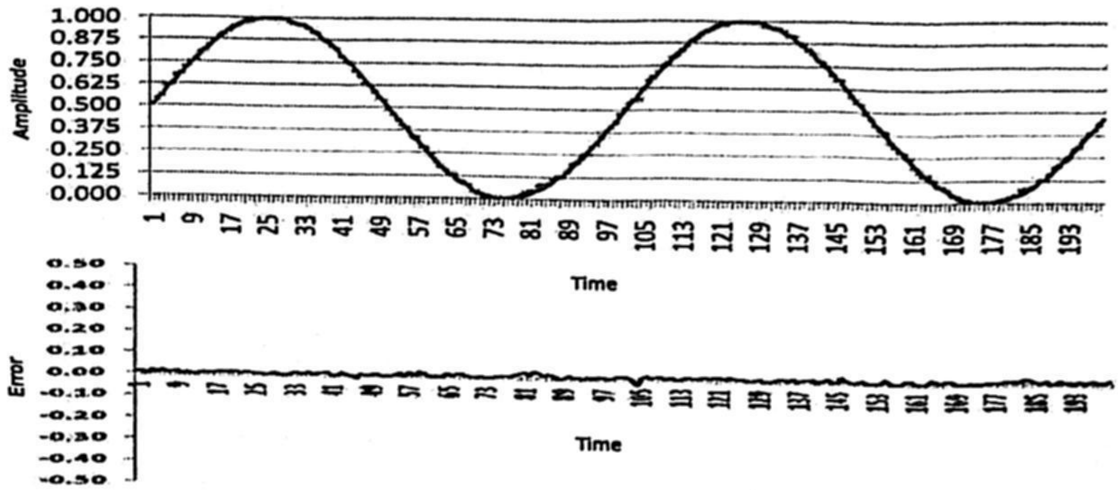

Fig. 7. A) Results of the Sine time series, which has a periodical behaviour. The continuos series (black) corresponde to the original points and the dotted series (gray) belongs to the tracking series. B)The Error tracking is showed

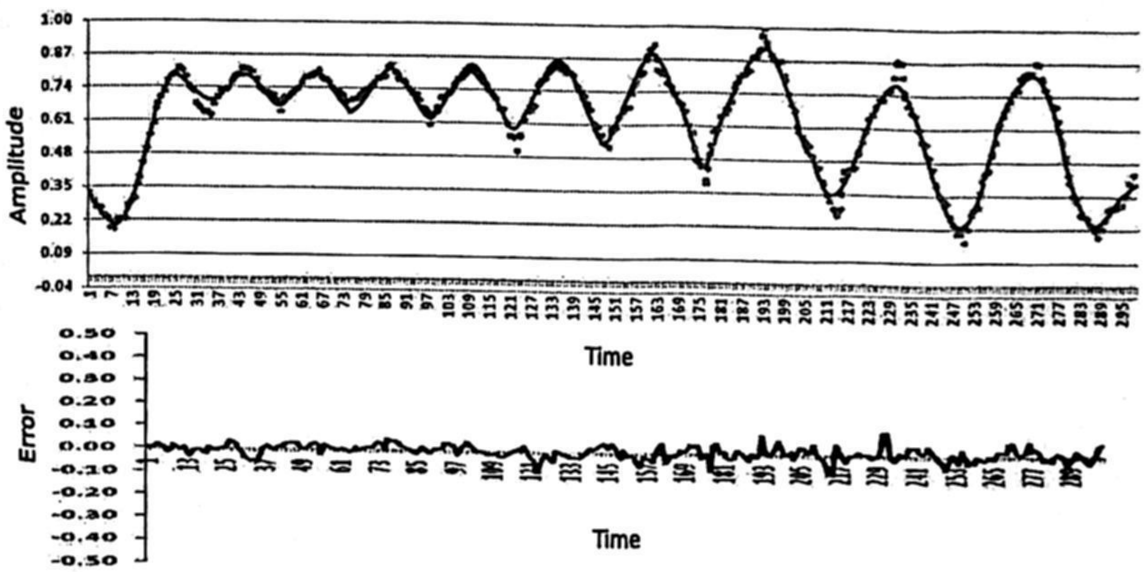

Fig. 8. A) Results of the Lovaina time series which is characteristic of complex behaviour. The original series es the continous one (black)and its tracking is the dotted (gray). B) The tracking error.

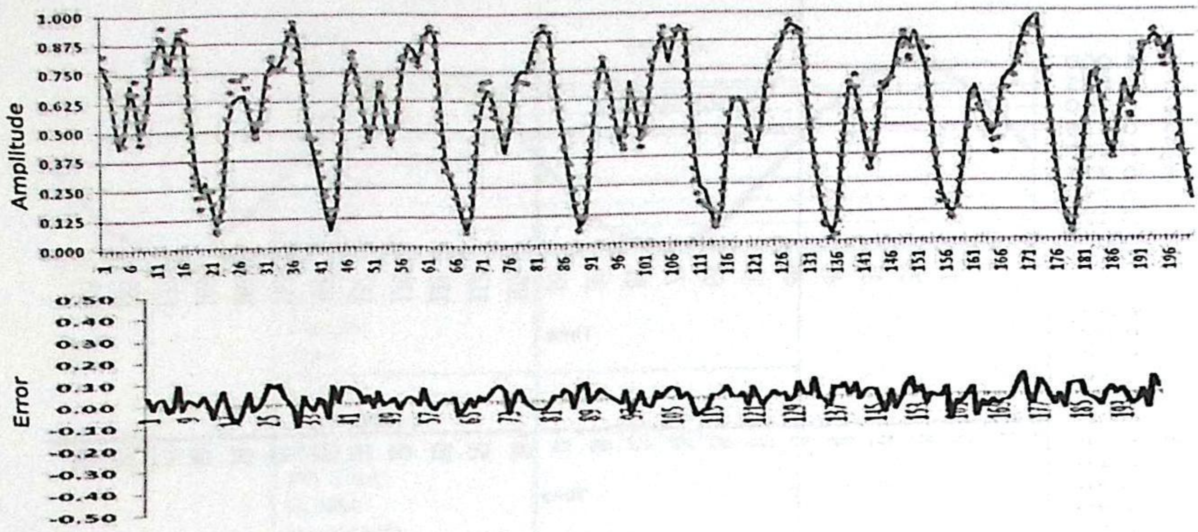

Fig. 9. A) Results of Mackey-Glass time seires, which has a chaotic behaviour. The continous series (black) correspond to the original points and dotted series (gray) belongs to the time series tracking. B) The tracking error is showed.

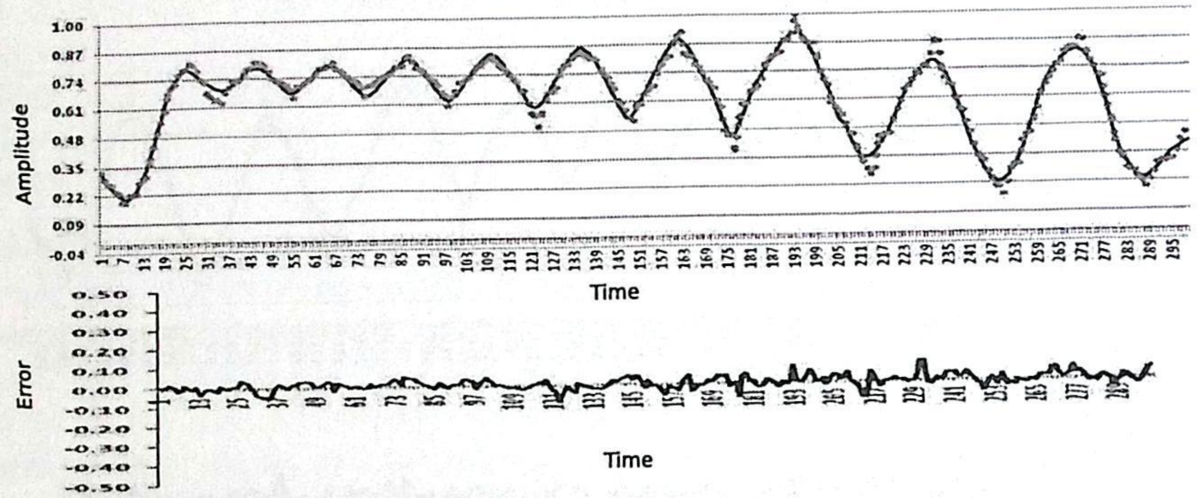

Fig. 10. A) Results of White Noise time series, which has a stochastic behaviour. The continous series (black) correspond to the original points and the dotted series (gray) belongs to the computed series. B) The tracking error is showed.

5 Conclusions

We have presented in this paper a new approach for time series tracking of different behaviors from an easiest time series (periodical) to a more complex behavior (chaotic). The tracking errors obtained with our technique demonstrated that the time series tracking has a high accuracy.

In the literature we found very few algorithms for time series tracking which makes a little difficult to compare our technique to others. Nevertheless, we consider this new approach a very accuracy one, because of the tracking errors that this technique obtains.

We think that some of the applications of the time series tracking are: tracking the trajectory of airplanes, reproduce music from the known music sheet among others.

At this moment we are working on the formalization of some aspects such as how many intervals is the best for any dynamic behavior of time series because in this work we used eight intervals determining it in an arbitrary way.

Using the information obtained with the time series tracking we can expand these results to a more difficult problem of time series namely, prediction. It is important to keep in mind that any natural phenomena can be able to be represented as a time series, and then it is possible to apply this approach in several time series tracking.

References

1. Michie, D.: Experiments on the mechanization of game-learning Part I. Characterization of the model and its parameters. Computer Journal **6** (1963) 232-236
2. Michie, D., Chambers, R. A.: Boxes: An Experiment in Adaptive Control. eds. E. Dale and D. Michie. Machine Intelligence 2. (1968) 125-133
3. Newell, A., Shaw, J.C., Simon, H.A.: A variety of intelligent learning. eds. C Marshall, Yovits and Scott Cameron. London: Pergamon : in a general problem solver in Self-organizing Systems. (1960). 153-189
4. Ramírez-Amaro, K., Figueroa-Nazuno, J.G.: Empleo de la Técnica Mapa Recurrente en el Análisis de los Índices de Teleconexión Atmosféricos. México, D.F. Centro de Investigación en Computación, I.P.N. (2006). Reporte Técnico. ISBN: 970-36-0330-0
5. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge : Cambridge University Press (1997)
6. Bautista-Thompson, E.F.: Medición de la Predictibilidad de las Series de Tiempo: un Estudio Experimental. México, D.F. Centro de Investigación en Computación. (2004). Ph. D. En Ciencias de la Computación.
7. Motwani, R., Raghavan P.: Randomized Algorithms. Cambridge University Press, New York (NY) (1995)
8. Frigo M., Leiserson C.E., Prokop H.: Cache-oblivious algorithm. Proc. 40th Symp. on the Foundations of Computer Science (1999).