Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 91–95, April 2–5, 2001 - Rio de Janeiro - RJ - Brazil

Forecasting Chaotic Time Series and Plasma Disruption Instabilities by Using Artificial Neural Networks

Kenya Andrésia de Oliveira[1,2], Álvaro Vannucci[1]
and Elton Cesar da Silva[1]

[1]Instituto de Física da Universidade de São Paulo,
caixa postal: 66318, CEP: 05315-970, São Paulo, SP, Brazil
e-mails: kenya@if.usp.br, vannucci@if.usp.br, elton@fap01.if.usp.br
[2]Faculdades Domus,
av. Higienópolis, 996, CEP: 01238-910, São Paulo, SP, Brazil
e-mail: kenya@facdomus.br

Abstract

Two-layer feedforward neural network was used to forecast chaotic time series and disruptive instabilities observed in the TEXT tokamak plasma discharges with very promising results. In both cases it was verified that a neural network with an architecture of the type m:2m:m:1, *where* m *is the embedding dimension of the attractor of the dynamical system in consideration, is a very good initial guess for the process of finding the ideal architecture for the neural network, which is usually hard to achieve. A* 15:30:15:1 *(*m = 15*) neural network was capable, for example, to forecast the disruptive instabilities in X-rays signals up to* $4\ ms$ *in advance, period of time about fourfold larger than the one obtained previously, when experimental magnetic signals from Mirnov coils were used. These very good forecasting results and those obtained by using chaotic temporal series like Lorenz system clearly suggest that there is an interplay between the architecture of a multilayer network and the embedding dimension* m *of the time series used. They are quite significant and opens up to the possibility of using neural networks for making predictions over the evolution of nonlinear systems, such as confined plasmas, for example.*

1. Introduction

The controlled thermonuclear fusion can be a hope for the mankind of a cheap and inexhaustible source of energy. Toward this goal, intensive efforts have already been made, during the last decades, in search for processes that could maintain the plasma magnetically confined while its temperature is strongly increased to allow nuclear reactions that could give the energy desired [1]. Unfortunately, during the plasma confinement some instabilities named *disruption instabilities* occur and frequently provoke the loss of the confinement inducing strong eletric tensions in many parts of the tokamak and many times damaging them. This kind of instabilities results from disturbances in the macroscopic parameters of plasmas confined in tokamaks propagates by waves and is usually studied by magnetohidrodynamic equations (MHD theory), that describe the plasma as a system of great scale where microscopic aspects in the media are not relevant [2]. In fact, disruption instabilities are still considered as real obstacles for the operation of future fusion reactors, such as tokamaks, and despite the several attempts at understanding the responsible mechanisms for the occurence of disruption instabilities, the right way to control or avoid them has not yet been found. In this way, because of the relevance of the problem, one has worked with artificial neural networks, by using X-ray experimental signals measured from confined plasma with the objetive of trying to forecast the instance of occurence of this kind of instabilities and we have obtained very promising results. Another kind of work we have also done with very good results was on predicting the behavior of chaotic time series such as Lorenz system [3], as it will be discussed next.

2. Forecasting Chaotic Time Series

In many situations in science and technology we usually face the necessity of predicting the future evolution of a system from past measurements of it. Mathematical models of physical systems are generally investigated by writing down the equations of motion and by trying to integrate them, forward in time, to predict the future state of the system. Mathematically speaking, this dynamics is described by the motion of a point $\vec{v}$, which represents the state of the system in a multi-dimensional space Γ. However, in nonlinear systems with many degrees of freedom, it is just impraticable to solve all the equations without making some sort of assumptions and simplifications. Dissipations can reduce the number of the effectively relevant degrees of freedom in apparently chaotic dynamical systems. Thus the motion of the system becomes confined to a subspace Γ_A, of Γ, known as *atractor* with lower dimension d [4]. Within this scenario, neural networks may be considered an important forecasting tool to be used in such situations [4].

According to *Takens* [5], there exists a smooth function of at most $2d + 1$ past measurements of a temporal series that allows the correct prediction of its future value, and the prediction is just as good as the one it would be obtained if we had been able to solve the complete system with its all degrees of freedom [4, 5]. What the theorem of Takens does not provide is the explicit form of the function which would contain the desired extrapolation and it is in this context that neural networks can be successfully used. By setting as the input pattern $\{x_i^\mu\}$ the delay coordinates of the temporal series $x(t)$: $\vec{v}_D = (x(t), x(t-\tau), \ldots, x(t-(m-1)\tau))$, and choosing $x(t+\Lambda)$ as the known target, the network can be trained to predict the future state of the system at a time Λ, which corresponds to a certain number of iterations, or time steps. Mathematically:

$$\{x(t), x(t-\tau), ..., x(t-(m-1)\tau)\} \longrightarrow x(t+\Lambda), \quad (1)$$

where τ is the time delay.

In order to investigate the capability of neural networks in predicting the future state of the chaotic systems, the Lorenz system was initially used. The time series was obtained from the numerical solutions yielded by the three ordinary differential equations [6]:

$$\begin{aligned} \dot{x} &= \sigma x + \sigma y \\ \dot{y} &= -xz + rx - y \\ \dot{z} &= xy - bz \end{aligned} \quad (2)$$

For this work, it has been used a two-layer feedforward network (fig. 1) and the time series chosen to feed this network with were the ones related to the solutions in x with $\sigma = 3.0$ (*Prandtl number*), $r = 26.5$ (*Rayleigh number*) and $b = 1.0$. Starting from a given initial configuration for the m input units $\{x_1^\mu, x_2^\mu, ..., x_m^\mu\}$, the dynamics of the two-layer feedforward network is defined as follows:

Given the data training pattern μ, each hidden unit j in the first hidden layer receives a net input

$$\Upsilon_j^\mu = \sum_i W_{ji} x_i^\mu, \quad (3)$$

and produces the output

$$V1_j^\mu = \tanh(\Upsilon_j^\mu) = \tanh\left(\sum_i W_{ji} x_i^\mu\right), \quad (4)$$

where W_{ji} represents the connection weight between the ith input unit and the jth hidden unit in the first layer. Following the same procedure for the other units in the next layers, the final output is then given by:

$$\hat{O}_s^\mu = \sum_l W_{sl}\left\{\tanh\left[\sum_j W_{lj}\tanh\left(\sum_i W_{ji} x_i^\mu\right)\right]\right\}, \quad (5)$$

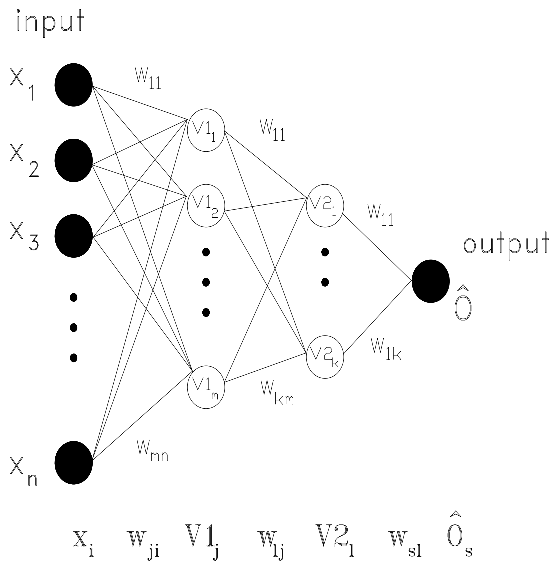

Figure 1: Diagram of a feedforward neural network with two hidden layers and architecture *n:m:k:1*.

where the *hyperbolic tangent* activation function was chosen for all hidden units, and the *linear function* for the final output unit.

Supervised feedforward networks learn from examples. The weights of the conections are determined by presenting the network with a set of actual input-output values (the training set) and comparing, by means of some *error* or *cost function* $E(W)$, the output of the network with the real value of the time series. For this work, the cost function was chosen to be the mean square error:

$$E(W) = \frac{1}{M}\sum_{\mu s}^{M}(O_s^\mu - \hat{O}_s^\mu)^2, \quad (6)$$

where M is the size of the training data set, $\hat{O}_s^\mu$ is the output data yielded by the unit of the last layer and O_s^μ is the actual data. The error is minimized by adjusting the weights according to the backpropagation algorithm, which corresponds to the gradient descent recipe with the inclusion of an inertial term to accelerate the convergence [4]. In particular,

$$W_{qt}^{new} = \alpha W_{qt}^{old} - \eta \Delta W_{qt}, \quad (7)$$

where:
$\Delta W_{qt} = \frac{\partial E(W_{qt})}{\partial W_{qt}}$; q and t identify each connection
$0 < \eta \leq 1 \equiv$ learn rate, and
$0 < \alpha < 1 \equiv$ inertial term.

In order to characterize the precision of the training process, the *average relative prediction error variation* (ARV) and the *correlation coefficient* (ρ) are therefore introduced [7]:

$$ARV = \frac{\sum_{\mu s}(O_s^\mu - \hat{O}_s^\mu)^2}{\sum_{\mu s}(O_s^\mu - \langle O_s^\mu\rangle)^2} \quad (8)$$

$$\rho = \frac{\sum_{\mu s}(O_s^\mu - \langle O_s^\mu \rangle)(\hat{O}_s^\mu - \langle \hat{O}_s^\mu \rangle)}{\sigma_O \sigma_{\hat{O}}} \quad (9)$$

where $\langle O_s^\mu \rangle$ is the mean value of the set of M actual outputs, and σ_O and $\sigma_{\hat{O}}$ are the standard deviations of the actual and the predicted outputs respectively.

The training of the network was carried out by using one arbitrarily chosen temporal series out of a given dynamical system. In order to obtain the best set of weights $\{W\}$ we monitored the performance of the net by using a second temporal series of the same system which was different from the first one. For each *training epoch* (certain number of passes through the training set) a resulting set of weights are recorded and the value of *E* (eq. 6), *ARV* (eq. 8) and ρ (eq. 9) are then calculated as an evaluation of the neural network's performance. The best results of the training process are those which give the lowest value of the *E*, the smallest value of the *ARV* and the biggest ρ.

Initially, an extensive set of tests were carried out by using many different architectures and also different conditions for training the net and forecasting. For the training process, the series (1,0,0)[1] was used, and the validation process was carried out by using the (0,2,0) series with the following parameters: $\eta = 0.001$, $\alpha = 0.0$ and $\tau = 6$, over 100 training epochs. Finally, the prediciton process was done over the series (1,1,1).

Basically, two different approaches, based on different types of architecture, were chosen to test the prediction power of neural networks. Firstly, we tried using the same architecture (*15:9:3:1*) that had been successfully used to forecast plasma disruptions in tokamaks [7]. However, the neural network did not allow good results to the chaotic Lorenz series with this type of architecture. Some other different architectures were also tried but the best results were obtained when the embedding dimensiom *m* of the dynamical system (in the proprotion *m:2m:m:1*) was taken into account. Therefore, as an initial guess, we started using the architecture *3:6:3:1*, since the embedding dimension for this type of dynamical system is known to be 3 [4, 6].

With this type of architecture we were then able to successfully predict the future state of the system for a wide range of different time steps. For up to time step 15 ($\Lambda = 0.3$), for example, the result of the forecasting process is practically perfect when compared to the real data, as shown in figure 2-a. For time steps up to 25 ($\Lambda = 0.5$) the match of the neural network output with the actual data can also be considered to be very good, as shown in figure 2-b. Although the fitting is not so perfect, in this case, the chaotic pattern of the Lorenz series was still very well predicted by the neural network, and our results can be considered much better than the ones reported elsewhere. *Koga*, for instance, obtained good predictions only up to 3 time steps, when neural networks were used with a type of architecture different from ours [8]. On the other hand, *Diambra* and *Plastino* were able to obtain good predictions for up to 10 time steps, by using another method of time series predictions [9].

[1]The set (x_0,y_0,z_0) represents the initial conditions chosen to solve the system given by equation 2.

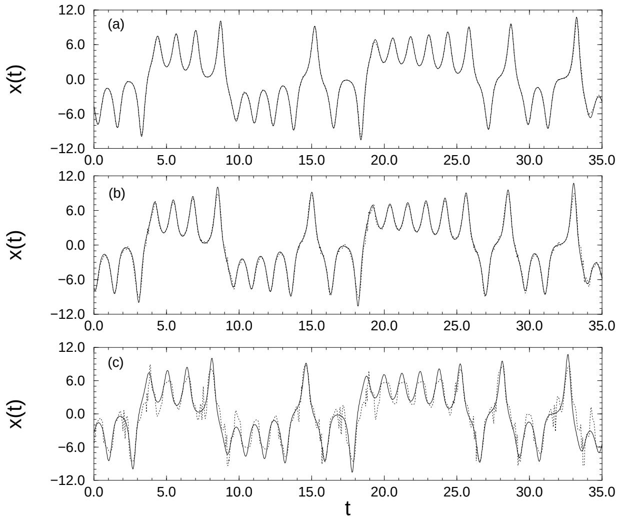

Figure 2: Prediction of the Lorenz series using the architecture *3:6:3:1* (dotted line) as compared with the real data (solid line) for (a) 15 time steps, (b) 25 time steps and (c) 45 time steps. As it can be observed, the prediction is pratically perfect for 15 time steps, which indicates the network has successfully learned the dynamic of the system.

As it could be observed from our simulations, if the predictions are carried out for time steps greater than 25, the output data from the net starts deviating from the real data, but the pattern of the Lorenz series is still reasonably well predicted by the net, as exemplified in figure 2-c for 45 time steps ($\Lambda = 0.9$).

By increasing the time prediction steps even further, no good results are obtained. However, by slightly changing the architecture used, i.e, by increasing the *m* number of neural units in the input and hidden layers we were able to improve these results a little. Figure 3-b, for instance, shows the forecasting results obtained from the architecture *4:8:4:1* for 45 time steps. The match between the two sets of data can be considered slightly better in this case than the results obtained with the previous architecture *3:6:3:1* (fig. 3-a). In short, from these results we can conclude that neural networks can be effectivelly used to predict with success the chaotic behavior of Lorenz series.

As an easy way to evaluate the prediction performance of the neural network we plotted the *ARV* and ρ parameters against the number of time prediction steps. The results obtained are shown in figures 4-a and 4-b, respectively, for the architecture *3:6:3:1*. From these figures it is observed that the degradation of the prediction power of the neural networks tends to exponentially increase as the number of time steps increases, as one should expect. It is important to notice, however, that the prediction results can be considered excellent up to 15 time steps. For bigger values than that, the fitting between the actual data

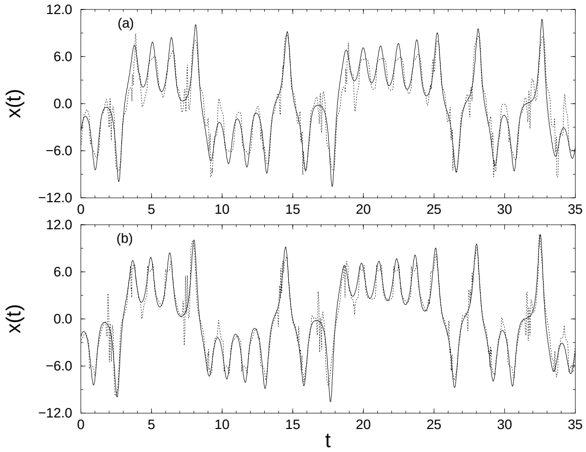

Figure 3: Prediction of the Lorenz series using (a) the architecture *3:6:3:1* (*ARV* = 0.07, $\rho = 0.96$) as compared with (b) the architecture *4:8:4:1* (*ARV* = 0.05, $\rho = 0.98$). In both figures, the solid line represents the x component of Lorenz system and the dotted line the predictions of the neural network for 45 time steps. As it can be observed, the predictions in (b) are slightly better than those obtained in (a).

and the results predicted by the net are not so perfect, but the neural networks still predict reasonably well the general chaotic pattern of the Lorenz series.

As it can be observed in figure 5 the best forecasting results are obtained when the *m:2m:m:1* type of architecture is used.

3 Forecasting Disruption Instabilities

During the tokamak plasma discharges, several diagnostics are put in operation to measure the plasma parameters and monitor the plasma conditions. In particular, the soft X-ray detection system has become an important tool to investigate the central part of the plasma column, since the emission of low energy X-rays is closely related to the electron temperature, important parameter of confined plasmas, impurities, plasma density and disruptions instabilities [7]. In figure 6 we can see some of the basic signals of one disruptive plasma discharge, such as Plasma Current (fig. 6-a), Loop Voltage (fig. 6b), one of the Soft X-ray signal used in this task (fig. 6-c), Magnetic Coils (fig. 6-d) and Eletronic density (fig. 6-e). The disruption instabilities cause a significant drop in the average electron density (figure 6-e) and electron temperature, by soft X-ray emission signal (figure 6-c). The procedure used to forecast disruption instabilities was almost the same as the Lorenz series with few diferences. The value of immersion dimension, for instance, was *m=15* when we used soft X-ray signals obtained from TEXT tokamak (Austin, Texas-USA). Then, the feed-forward architecture to make predictions for this kind of discharges was *15:30:15:1*. It is important to notice here that neural network was trained with signals obtained from certain conditions (temperature, density, impurity concetration, etc.)

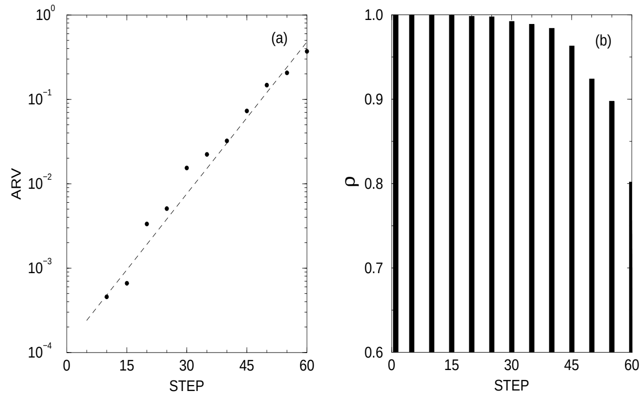

Figure 4: Plots showing (a) the predicting error ARV and (b) correlation coefficient for the Lorenz system. As noted, the performance of the network tends to degrade exponentially as the time step increases, and the result is very good up to 15 time steps prediction.

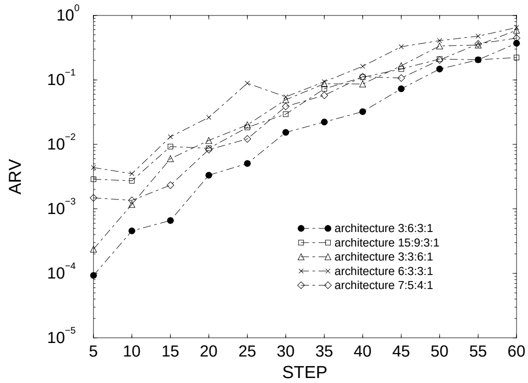

Figure 5: Different architectures used to forecast series from Lorenz system. As noted, the best configuration is *3:6:3:1*.

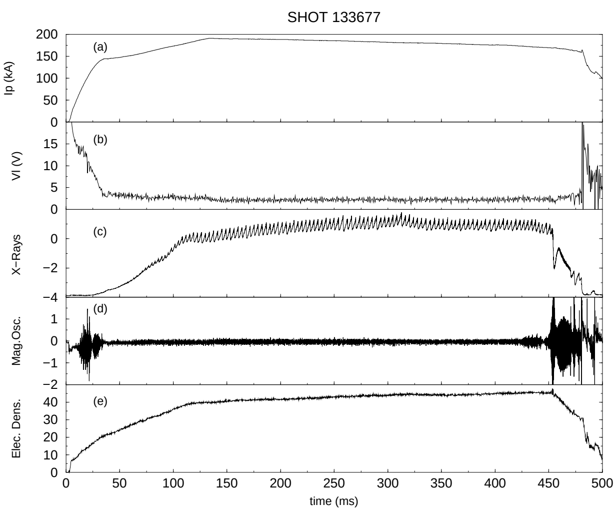

Figure 6: Some of basic signals from plasma discharge of TEXT tokamak, Austin, Texas-USA: a-) Plasma Current, b-) Loop Voltage, c-) Soft X-ray, d-) Magnetic Coils and e-) Eletronic Density, in $10^{12} cm^{-3}$.

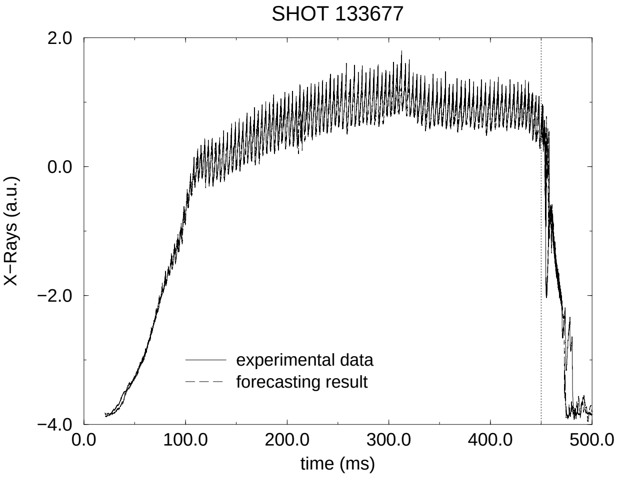

Figure 7: Soft X-ray from a disruptive discharge which was used for forecasting. As it can be observed the minor disruptions occurs at $t \sim 52ms$ and is perfectly predicted by *15:30:15:1* feedforward neural network.

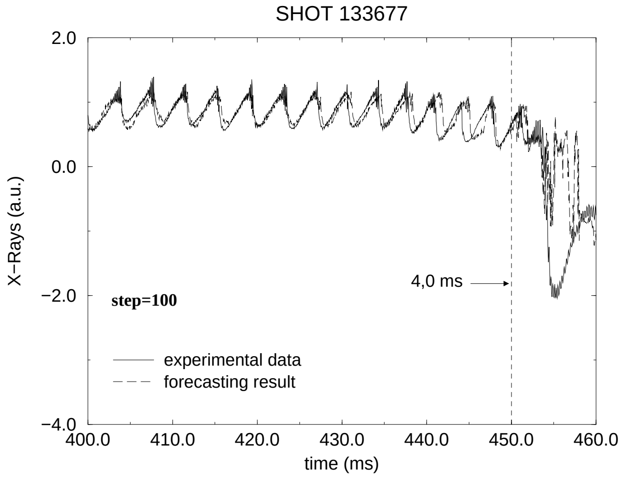

Figure 8: Shot 133677 expanded in time, where it is possible to observe the very good forecasting results from a *15:30:15:1* feedforward neural network.

of plasma confinement. Therefore, all the experimental signals for training, validation or forecasting process must belong to this sample of data. If the confinement conditions of plasma are changed, the network must be trained again with another set of data.

Initially, extensive attempts were carried out to find the best architecture for working with soft X-rays from TEXT, since the the number of immersion dimension, *m*, of plasmas confined by tokamaks is not surely known. After several tests we have found the best architecture for our purpose: *15:30:15:1*. For the training process, we have used three soft X-ray signals, all of them similar to the signal observed in figure 6-c, with the parameters: $\alpha = 0.9, \eta = 0.001$ (equation 7) and 500 training epochs. The validation was carried out with another signal and for the prediction process, a fifth signal was chosen. The patterns of training, validation and forecasting tasks were obtained randomly with $\tau = 30$ and the time prediction was $4ms$. In figures 7 and 8 very good prediction results are shown for *15:30:15:1* neural network. As it can be observed, the result obtained agrees almost perfectly with the experimental signal.

4. Conclusions

It has been shown in this work that neural network can be successfully used to predict the future state of nonlinear systems as the ones described by the Lorenz equations. For all these cases, it has been verified that the architecture *m:2m:m:1* is an initial good guess to construct the neural network architecture for predicting the future state of chaotic temporal series. The reasonable good results obtained here clearly suggests an interplay between the architecture of a multilayer network with the embedding dimension *m* of the time series that is under investigation. This conclusion can be very useful if we consider that, when dealing with neural networks, a large number of training data sets and interactions are generally required to find the right architecture and, therefore, reliable and faster convergence process for the neural networks are mostly welcome. The prediction approach used herein has shown to have considerable advantages in terms of quality of the results and flexibility over the more conventional methods reported in other published articles [8, 9]. Thus, the results obtained here are encouraging and neural networks can be considered an important tool for making predictions in time of nonlinear systems, like disruptions instabilities in confined plasmas. This kind of instabilities is one of the major obstacles to future fusion reactors and if one knows approximately the moment of its occurence it is possible to change the conditions of tokamak control or to start up safety mechanisms of the machine such as the injection of neutral particles (or pellets), and the aplication of external magnetic field, etc., to avoid these instabilities or, at least, attenuate their harmful effects.

References

[1] A. Vannucci, I. C. Nascimento, and I. L. Caldas. *Plasma Physics and Controlled Fusion*, 31:147, 1989.

[2] E. Westerhof. *Transations of Fusion Technology*, 25:152, 1994.

[3] K. A. de Oliveira, A. Vannucci, and E. C. da Silva. *Physica A*, (284):393, 2000.

[4] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, New York, 1991.

[5] F. Takens. *Lectures Notes in Mathematics*, (898), 1981.

[6] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos: An Introduction to Dynamical Systems*. Springer, New York, 1996.

[7] A. Vannucci, K. A. de Oliveira, and T. Tajima. *Nuclear Fusion*, 39(2):255, 1999.

[8] J. Koga. *Notes on Basic Science (Jaeri-Japan)*, 3(2):37, 1996.

[9] L. Diambra and A. Plastino. *Physics Letters A*, 216:278, 1996.