Abstract. - The last few decades have seen dramatic growth and many important theoretical advances, in the field of dynamical systems control. This work describes a method to control the deterministic chaos in two nonlinear dynamical systems simultaneously. A backpropagation neural network has been used to control chaotic trajectories in the equilibrium points. Such neural network improves results, avoiding those problems that appear in other control methods, being also efficient dealing with a relatively small random dynamical noise. The Hénon and Lozi chaotic systems are controlled using a neural network, obtaining good results with a low mean square error.

Key-Words: - Chaos, Control, Signal, Artificial Neural Networks, Backpropagation, Filters

1 Introduction.
Nowadays, with the electronic communications growth, signals processing has become a technology of multiple facets. It has passed from implementation of tuned circuits to digital processors of signals. The base of the industry continues being the design and the realization of filters to carry out noise elimination on carrier signals of information.

The chaotic phenomena take place everywhere, so much in natural systems as in mechanisms built by the man. Previous works have been mainly focused in describing and characterizing the chaotic behaviour in situations where there is not any intervention. The control of chaotic signals is one of the most relevant research areas that have appeared in last years, taking the attention of computer scientists [6]. Recently they have being proposed ideas and techniques to transform chaotic orbits into desired periodic orbits, using temporarily programmed controls [1]. In 1990, Ott, Gegogi Yorke [5] developed methods to control two nonlinear system stabilizing one of the nonstable periodic orbits embedded in its chaotic attractor.

This paper shows how two mixed chaotic signals can be controlled using a backpropagation neural net as filter, in order to separate and to control both of them at the same time.

The neural network provides a more effective control, it can be applied to the system at any point, even being too far from the desired state, avoiding long transient times. The control can be applied if there are only a few data of the system, and it will remain stable much more time even with small random dynamical noise.

2 Controlling Discrete-Time Chaotic Systems.
A discrete dynamical function is going to be controlled, all the trajectories are focused towards the stable point $x_{n+1} = f(x_n)$, where $x_n \in \mathbb{R}^2$; $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.

Two systems are employed, the Hénon and Lozi systems. These ones are the easiest nonlinear discrete systems of second order. The equations of Hénon map [4] are the followings:
\[ x_{k+1} = h_1(x_k, y_k, a) = 1 - ax_k^2 + y_k \]
\[ y_{k+1} = h_2(x_k, y_k, b) = bx_k \]

being \( a, b \) free parameters.

Lozi system [2] is described by the following differential equations:
\[ x_{k+1} = l_1(x_k, y_k, p) = -p|x_k| + y_k + 1 \]
\[ y_{k+1} = l_2(x_k, y_k, q) = qx_k \]

where \( p \) and \( q \) are two real parameters

The parameter values are \( b=0.3 \), this is the most common value taken in many studies of the Henon function, and \( a=1.3 \). So, there are two stable points and one initial point \((-0.1, 0.1)\). The stable points of the systems are given by:

\[
P^+ = (p_1, p_2) = \left(\frac{b - 1 + \sqrt{(b - 1)^2 + 4a}}{2a}, bx_1\right)
\]

\[
P^- = (p_3, p_4) = \left(\frac{b - 1 - \sqrt{(b - 1)^2 + 4a}}{2a}, bx_2\right)
\]

3 A Neural Network Model.
The neural network employed as the main controller is a backpropagation one, consisting of three layers of neurons (input layer, hidden layer and output layer). The input layer has 4 neurons, one for each one of the variables of each function \( f \) that is going to be controlled. Output layer has 4 neurons corresponding to the coordinates of the stable points. The number of neurons in the hidden layer plays an important role in the learning performance and generalization capability of the network. The activation function of the neurons is the sigmoid.

3.1 Learning Procedure.

1.- The weights of the network are always initialized to the same values.

2.- Input patterns are obtained: taking an initial point \((x_0, y_0)\) and computing the time series of the Henon function components and taking other initial point \((x_1, y_1)\) and computing the time series of the Lozi function components. Once temporal series are built, input patterns are generated with the components of each function, for example: the first pattern would have four elements – the first and the second components of Henon and Lozi’s functions respectively.

\[ h_1(x_0, y_0) \]
\[ h_2(x_0, y_0) \]
\[ l_1(x_1, y_1) \]
\[ l_2(x_1, y_1) \]

being \( H_0 = (h_1(x_0, y_0), h_2(x_0, y_0)) \)
\( H_k = (h_1^k(x_0, y_0), h_2^k(x_0, y_0)) \)
and \( L_0 = (l_1(x_1, y_1), l_2(x_1, y_1)) \)
\( L_k = (l_1^k(x_1, y_1), l_2^k(x_1, y_1)) \)

3.- Output patterns are the stable points that the functions has to control. If \( P = (p_1, p_2) \) is the Henon’s function stable point and \( Q^- = (q_1, q_2) \) is the Lozi’s function stable point then the output patterns are:

\[ h_1(x_0, y_0) \]
\[ h_2(x_0, y_0) \]
\[ l_1(x_1, y_1) \]
\[ l_2(x_1, y_1) \]
4.- Number of hidden neurons.
Several simulations have been performed in order to
know how the number of hidden neurons affects the
mean square error in finding the stable point. Figure
3 shows results obtained with 1, 2, 3 and 4 hidden
neurons. Input file is made up starting from the
initial point $A_1 = (-0.1, 0.1)$ for Henon map and
$A_2=(0,0)$ for Lozi map, and computing the time
series with 500 patterns. The stable points are
$P'=(0.6482203,0.19446601)$ and
$Q'=(0.55463117,0.55296728)$, N is the number of
iterations to the whole pattern set and error 1, error 2,
error 3 and error 4 correspond to the mean square
error of the network with 1, 2, 3 or 4 hidden neurons.

As the number of hidden neurons increases, the
mean square error also increases. So, the best
network is the one which has only one hidden
neuron.

5.- Number of input patterns. The variation of error
along the number of input patterns has been studied,
among them files with {500, 1000, 2000} patterns
have been taken into account. Table 1 shows error
measure at iteration 10 for each one of the input
files.

To finish the learning phase of the network,
another input pattern set is obtained starting from
point $B_1 = (0, 0)$, finding the time series with 500
patterns de la función de Henon y punto $B_2 = (0.3, -1)$,
finding the time series with 500 patterns de la
función de Lozi. The number of iterations to learn is
10 and the mean square error is 0.000340435, so the
network has achieved a good solution.

3.2 Achieving the Control
Once the learning phase is completed, it is necessary
to check if the network is able to control the function
in the stable point. Then, the point $C_1=(-0.1, -0.5)$
that is far enough from the stable point $P'$ is the basis
for the pattern generation de la función de Henon y
point $C_2 = (1, 0)$ for the Lozi function. The pattern set
is made up of 500 patterns that are presented to the
networks 10 times. Table 3 show the error
corresponding to the two initial points, B and C.

So far, the number of patterns is 500 and the
number of iteration in the learning phase is 10, but
the network is also able to control the function when
there exist only a few data and with some kind of
noise. Starting from point $C_1 = (-0.1, -0.5)$ and $C_2 =
(1, 0)$, the pattern set will have only 50 patterns, to
achieve a similar error to the previous one is needed
to increase the number of iterations in the learning
phase. The mean square error is 0.000414981.
Next, some results will be reported corresponding to data set with just a few data and with random dynamic noise in order to check if neural networks are suitable to control chaotic signals. Starting from points C1 and C2 and applying the iterative functions, a file with 50 patterns is built and some small random dynamic noise, uniformly distributed on interval [-0.08, 0.08], is added to each pattern, then the error after 10 iterations is 0.000609192.

An important advantage of this control technique is that the obtained controllers are very stable, presenting a good behaviour even with a small random dynamical noise or with only a few data.

4 An Extension To The Control

In the previous section, a signal made up by two chaotic signals arriving together at the same time to the control device has been controlled. Now, only one of the previous chaotic signals is going to be controlled by the same neural network.

4.1 Henon Map Control

The first step is to control only the chaotic signal that arrives to the Henon’ system. The neural network used as a controller has the same architecture than in the case above, it is a network with three layers with sigmoid neurons. The pattern set is made up starting from the point A1=(-0.1,0.1) and computing the time series for Henon map with 500 patterns. The stable point is P+= (p1,p2) = (0.6482203, 0.19446601)

\[ h_1(x_0,y_0) \to \circ \]
\[ h_2(x_0,y_0) \to \circ \]
\[ 0 \to \circ \]
\[ 0 \to \circ \]

being \( H_0 = (h_1 (x_0,y_0) , h_2 (x_0,y_0) ) \) \( (7) \)
\( H_k = (h_1^k (x_0,y_0) , h_2^k (x_0,y_0) ) \) \( (8) \)

The output pattern is the stable points that the function has to control. If \( P = (p_1, p_2) \) is the Henon function stable point and \( Q = (01,0) \), is the output pattern is:

\[ \circ \to p_1 \]
\[ \circ \to p_2 \]
\[ \circ \to 0 \]
\[ \circ \to 0 \]

The number of iterations to learn is 10 and the mean square error is 0.0027636 the network has achieve a good solution. It is necessary to check if the network is able to control the function in the stable point. Then, the points \( B_1 = (0, 0) \) and \( C_1 =(-0.1, -0.5) \) that is far enough from the stable point \( P+ \) is the basis for the pattern generation de la función de Henon. The pattern sets are made up of 500 patterns as in the previous case. The pattern set is presented to the networks 10 times. Table 4 shows the error concerning the two initial points, \( B_1 \) (training phase) and \( C_1 \) (performance phase).

<table>
<thead>
<tr>
<th>N</th>
<th>ERROR B</th>
<th>ERROR C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00234254</td>
<td>0.002548</td>
</tr>
<tr>
<td>5</td>
<td>0.00124739</td>
<td>0.001036</td>
</tr>
<tr>
<td>10</td>
<td>0.000582923</td>
<td>0.000591</td>
</tr>
</tbody>
</table>

4.2 Lozi Map Control

Secondly, the chaotic signal arriving to the Lozi map will be controlled. The pattern set is made up starting from the point \( A_2 = (0, 0) \) and computing the time series for Lozi map with 500 patterns.

\[ 0 \to \circ \]
\[ 0 \to \circ \]
\[ l_1(x_1,y_1) \to \circ \]
\[ l_2 (x_1,y_1) \to \circ \]

being \( L_0 = (l_1 (x_1,y_1) , l_2 (x_1,y_1) ) \) \( (9) \)
\( L_k = (l_1^k (x_1,y_1) , l_2^k (x_1,y_1) ) \) \( (10) \)

The output pattern is the stable points that the function has to control. The stable point is \( Q+= (q_1,q_2)= (0.55463117, 0.55296728) \), the output pattern is:

\[ \circ \to 0 \]
\[ \circ \to 0 \]
\[ \circ \to q_1 \]
\[ \circ \to q_2 \]

The network is trained with the same procedure as the previous one with the following points: \( A_2=(0,0), B_2=(0.3, -1) \) and \( C_2=(1, 0) \) and similar results are obtained. Table 5 show the error concerning the two initial points, \( B_2 \) (training phase) and \( C_2 \) (performance phase).
Table 5. Mean square error (Lozi map).

<table>
<thead>
<tr>
<th>N</th>
<th>ERROR B</th>
<th>ERROR C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00234254</td>
<td>0.00743567</td>
</tr>
<tr>
<td>5</td>
<td>0.00124739</td>
<td>0.00595792</td>
</tr>
<tr>
<td>10</td>
<td>0.00058292</td>
<td>0.00478354</td>
</tr>
</tbody>
</table>

Once the network is able to control the system in the stable point, the obtained real outputs have been studied. In all the tests the outputs are stable after or before.

5 Discussion

This paper states that it is possible to control and to make stable chaotic dynamical systems simultaneous, driving their trajectories until they reach the equilibrium point, with no complicated methods. The control of chaos can be applied without paying attention to the fact that the trajectory must be close enough to the equilibrium point.

Control can be applied although there exists a few data; even it is very effective with noise in the training patterns. Besides, once the network is applied as a system controller, the system remains stable in a periodic way for a long period of time, without being affected by the error in the calculus process.

References:


