NEURAL CONTROL OF CHAOS AND APLICATIONS

Cristina Hernández, Juan Castellanos, Rafael Gonzalo, Valentín Palencia

Abstract: Signal processing is an important topic in technological research today. In the areas of nonlinear dynamics search, the endeavor to control or order chaos is an issue that has received increasing attention over the last few years. Increasing interest in neural networks composed of simple processing elements (neurons) has led to widespread use of such networks to control dynamic systems learning. This paper presents backpropagation-based neural network architecture that can be used as a controller to stabilize unsteady periodic orbits. It also presents a neural network-based method for transferring the dynamics among attractors, leading to more efficient system control. The procedure can be applied to every point of the basin, no matter how far away from the attractor they are. Finally, this paper shows how two mixed chaotic signals can be controlled using a backpropagation neural network as a filter to separate and control both signals at the same time. The neural network provides more effective control, overcoming the problems that arise with control feedback methods. Control is more effective because it can be applied to the system at any point, even if it is moving away from the target state, which prevents waiting times. Also control can be applied even if there is little information about the system and remains stable longer even in the presence of random dynamic noise.

Keywords: Neural Network, Backpropagation, Chaotic Dynamic Systems, Control Feedback Methods.

ACM Classification Keywords: F.1.1 Models of Computation: Self-modifying machines (neural networks); F.1.2 Modes of Computation: Alternation and nondeterminism; G.1.7 Ordinary Differential Equations: Chaotic systems; G.3 Probability and Statistics: Stochastic processes

Introduction

In spite of all the achievements of classical physics and mathematics, they have failed to touch upon compete areas of the natural world. Mathematicians had managed to specify, at least, some order in the universe, and the reasons behind this order, but they were still living in an untidy world.

Over the last few decades, physicists, astronomers and economists came up with a way of comprehending the development of complexity in nature. The new science, called chaos theory, provides a method for observing order and rules where once there was only chance, irregularity and, ultimately, chaos. Chaos goes beyond traditional scientific disciplines. Being the science of the global nature of systems, it has brought together thinkers from far apart fields: biology, weather turbulences, the complicated rhythms of the human heart…

Nonlinear and chaotic systems are difficult to control because they are unstable and sensitive to initial conditions. Two close-by trajectories rapidly diverge in phase space and quickly become uncorrelated. Therefore, forcing a system to follow a predetermined orbit is a far from straightforward task. Recently, there have been many attempts at controlling nonlinear and chaotic dynamic systems to get desired phase space trajectories. Ott, Gregogi and Yorke [Ott 1990] proposed one method: a natural unstable periodic system orbit is stabilized by making small time-dependent perturbations of some set of available system parameters. The so-called entrainment and migration control methods proposed by Jackson and Hübler are another approach to controlling chaos [Hübler 1989]. The generalized formulation described by Jackson [Jackson 1990] is based on the existence of some convergent regions in the phase space of a multi-attractor system. In each one of these convergent regions, all the close orbits locally converge to each other. Based on this observation and on many well-researched examples, Jackson stated that every multi-attractor system has at least one convergent region in each basin of attraction. Besides, he described a method for finding such convergent regions. The purpose of the so-called migration goal control is to transfer the dynamics of the system from one convergent region to another. There are many reasons for this. For example, of all the attractors of a complicated system, some can have different types of dynamics (periodic, chaotic, etc.), and one attractor could be more useful for one particular system behavior [Chen 1993].

However, the application of the above control methods has several drawbacks. There must be enough data, control is applied only when the state of the system is very close to the target state, leading to great transitory times closely together before control is activated [Barreto 1995], control is only effective at points adjoining the target state and, after a time, the controlled orbit is destabilized due to the accumulated computational error.

In this article, neural networks are designed to be used as controllers for chaotic dynamic systems, overcoming the problems that appear when using other controller types.

Model of Neural Control

The capacity of neural networks to generalize and adapt efficiently makes them excellent candidates for the control of both linear and non-linear dynamic systems. The objective of a neural network-based controller is to generate a correct control of the signal to direct the dynamics from the initial state to the final target state. The located execution and ease of building a network-based controller depend mainly on the chosen learning algorithm, as well as on the architecture used for control. Backpropagation is used as the learning algorithm in most designs.

The objective of this work is to use neural networks as the structure of a generic model for identifying and controlling chaotic dynamic systems. The procedure that must be executed to control a chaotic dynamic system is shown in Figure 1.

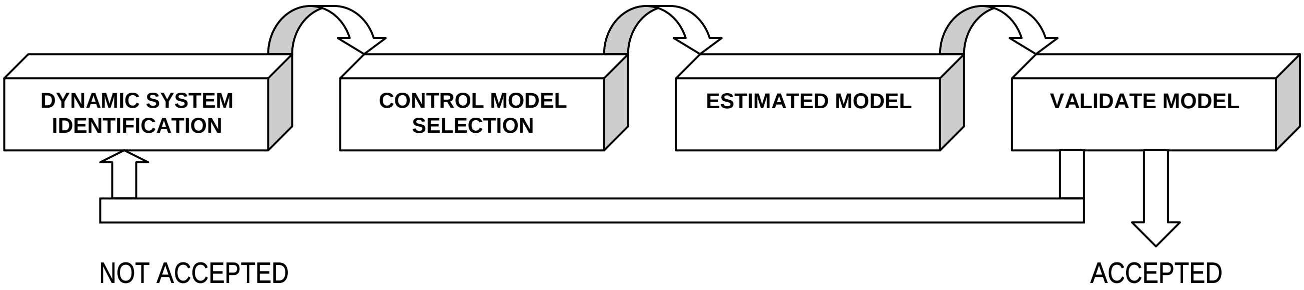


Figure 1: Design procedure of a control model

- <u>Identification of the Chaotic Dynamic System:</u>

This phase involves identifying the system, describing the fundamental data, the operational region, and pattern selection.

$$Zn = \{[\ u(t), y(t)\ ]/t = 1... N\}$$

where {u(t)} is the set of inputs, that is, the signal that is to be controlled, {y(t)} represents the output signal, t represents the pattern time. If the system in question has more than one input/output, u (t), and (t) are vectors.

- <u>Selection of the Control Model</u>

Once the data set has been obtained, the next step is to select a structure for the control model. A set of input patterns needs to be chosen, but the architecture of the neural network is also required. After defining the structure, the next step is to decide which and how many input patterns are to be used to train the network.

- <u>Estimated Model</u>

The next stage is to investigate what steps are necessary to make the control effective and to guarantee the convergence of the trajectories towards the target orbit. Control is effective if there is some $\delta>0$ and t_0 such that, for $t > t_0$, the distance between the trajectory and the stabilized periodic orbit is less than δ.

- <u>Validated Model</u>

When training a network, the network has to be evaluated to analyze the final errors. The most common validation method is to investigate residuals (error prediction) by means of crossed validations of a set of tests. The visual inspection of the prediction graph compared with the target output is probably the most important tool.

Identification of the Chaotic Dynamic System

The systems that are going to be controlled are nonlinear and chaotic dynamic systems that depend on a system of parameters, p. The basic function is: $\frac{dx(t)}{dt} = F(x(t), p)$, where F: $\Re^n \to \Re^n$ is a continuous function.

The other type of systems investigated is discrete dynamic systems, represented by an equation of nonlinear differences. Such systems are described as a function f: X $\to$ X that determines the behavior or evolution of the set when time moves forward. The control system inputs are the orbits of the elements. The orbit of $x \in X$ is defined as the succession $x_0, x_1, x_2, \ldots, x_n \ldots$, achieved by means of the rule: $x_{n+1} = f(x_n)$ with $x_0 = x$

The points of the orbit obtained are:

$x_1 = f(x_0) = f(x)$; $x_2 = f(x_1) = f(f(x)) = f^2(x)$; $x_3 = f(x_2) = f(f^2(x)) = f^3(x)$;… $x_n = f(f(\ldots f(x)\ldots)) = f^n(x)$ n times

The behavior of the orbits can vary widely, depending on the dynamics of the system.

The objective is to control the dynamic system in some unstable periodic orbit or limit cycle that is within the chaotic attractor. Therefore, the output of the system will be the limit cycle of period-1 or greater in which the system must be controlled. To find the outputs, it is necessary to consider that:

- A point α is an attractor for the function f(x) if there is a neighborhood around α such that the point orbits in the neighborhood converge to a. In other words, if the values are near to α, the orbits will converge to α.
- The simplest attractor is the fixed point. A point α is a fixed point for the function f(x) if f(α) = α.
- A point α is periodic if a positive integer number τ exists such that $f^\tau(\alpha) = \alpha$ and $f^t(\alpha) \neq \alpha$ for 0<t<n. The integer τ is known as the period of α.
- If α is a fixed point attractor, the set of initial values x_0 whose orbits converge to α form the basin of attraction of α.

The discrete systems that have been controlled are:

Discrete Systems:

Systems $f : R^2 \to R^2$ are second-order controlled, nonlinear discrete systems. All the trajectories are directed towards the stable point $x_{n+1} = f(x_n)$, where $x_n \in R^2$. The chosen systems are the Henon [Martin 1995], Lozi [Chen 1992], Ikeda [Casdagli 1989], and Tinkerbell [Nusse 1997] systems. The same type of discrete systems are controlled in unstable periodic orbits (limit cycle). The systems used in this case are Ikeda and Tinkerbell.

Another approach to controlling chaos is the so-called entrainment and migration control methods proposed by Jackson [Jackson 1991] and Hübler. Gumowski and Mira's discrete system is used to achieve migration control. This system has several attractors and several bounded convergent regions in the basin of attraction.

Figure 2. Ikeda system trajectory and Tinkerbell map

Continuous Systems: Continuous systems $f : R^3 \to R^3$ are controlled in an orbit of period 1, that is, at an equilibrium point. We look at Lorenz's [Gulick 1992] and Rössler's systems.

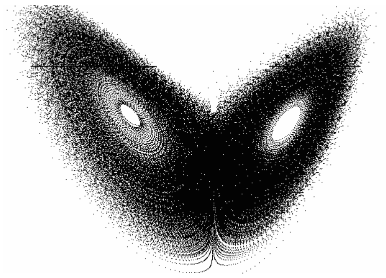
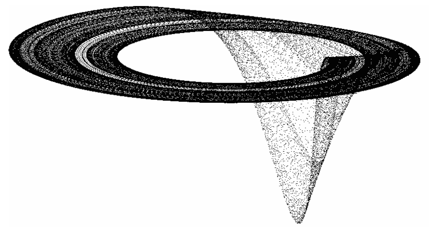


Figure 3. Lorenz's Attractor



Rössler's Attractor

Selection of the Control Model

The structure of the model has been divided to address two sub-problems:

- design the network architecture
- choose the structure of the input patterns

Architecture of the Neural Network

According to Lippmann [Lippman 1987], a model of neural network is characterized by specifying:

- The transference function of each node
- The network topology, which is defined by the number of nodes and the set of interconnections between these nodes.
- The learning rules, which are the rules that regulate how the weights associated with the connections are found.

The transference function. The neuron activation function is the sigmoid.

The neural network topology. The neural network employed as the main controller is composed of three layers of neurons (input layer, hidden layer and output layer).

Input layer:

- When control is effected at an equilibrium point, the input layer has two neurons, one for each of the variables of the function f that is going to be controlled for the discrete functions $f : R^2 \rightarrow R^2$, plus three neurons, one for each of the variables of the function f that is going to be controlled for the continuous functions $f : R^3 \rightarrow R^3$.
- When control is effected in a limit cycle, the components of the vector are separated from the function f, which will be applied first to the first and then to the second component of the function. The input layer will have as many neurons as the period of the limit cycle. If control is effected in a limit cycle of period 7, the input layer has 7 neurons; if the period is 5, the input layer has 5 neurons.

Output layer:

- When control is effected at an equilibrium point, the output layer has two or three neurons corresponding to the coordinates of the stable point.
- When control is effected in a limit cycle, the output layer will have as many neurons as the period of the limit cycle. If the period of the output layer is 7, it will have 7 neurons that correspond to the first coordinates of the period-7 point.

The number of neurons in the hidden layer plays an important role in the learning performance and generalization capability of the network:

- For the discrete functions $f: R^2 \to R^2$, the hidden layer will have one hidden neuron when control is effected at an equilibrium point.
- For the continuous functions $f: R^3 \to R^3$, several simulations have been run in order to ascertain how the number of hidden neurons affects the mean square error in finding the stable point.

The learning rules. The algorithm that is going to be used to adjust the weights is backpropagation, which descends according to the gradient that minimizes the error function.

Structure of the input patterns

Input patterns are necessary to define appropriately it, to avoid one of the biggest problems: trapping in a local minimum. The patterns for training the network will be formed by system orbits, obtained starting from a point that is within the basin of attraction of the limit cycle chosen for controlling the system.

1. Input Patterns

The input patterns are obtained by taking a starting point (x_0, y_0) and finding the time series from the components of the function by iterating $x_{n+1} = f(x_n)$. The input file is constructed from one point, using 500 patterns to search the time series.

When control is effected in a limit cycle, for example, a period-5 limit cycle, input patterns will be constructed as follows. The first pattern will be constructed from a point A= (x_0, y_0): {x_0, $f(x_0)$, $f^2(x_0)$, $f^3(x_0)$, $f^4(x_0)$ }, the following one from the next five iterations, and so on.

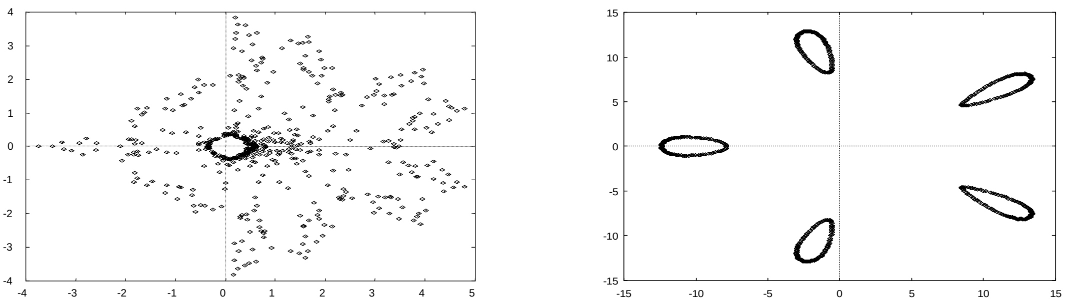


Fig. 5. a) Input for the network init point (0.5, 0) b) Input for the network init point (-2.37, 12.83)

Figure 5 shows the network input, that is, the input file consisting of 500 points from the iterative Gumowski and Mira function starting from points of different basins of attraction. The number of iterations is 10, and the final mean squared error after the learning process is 0.009397477, a good threshold.

2. Output Patterns

The input pattern is the equilibrium point at which the control function is going to be controlled.

When control is effected in a limit cycle, for example, a limit cycle of period 5, the output layer will have five neurons that correspond to the first coordinates of the period-5 point. If the point is Q = (q_1, q_2), the output will be constructed as {q_1, $f(q_1)$, $f^2(q_1)$, $f^3(q_1)$, $f^4(q_1)$}, which is the orbit of period 5.

Fig. 6. 5-period orbits of the Gumowski and Mira function located in different basins of attraction

Estimated Model

Once the control model has been designed, the following steps are taken:

1. Network weights must be fixed and always have same values to assure deterministic behavior.
2. An input pattern is presented and the output is calculated. To finish the learning phase of the network, another input pattern set is output starting from a point and finding the time series with 500 function patterns. The number of iterations to be learned is 10, and the mean square error is acceptable. Therefore, the network has found a good solution.
3. Number of input patterns. The error variance has been studied across the number of input patterns, including files with different patterns. However, the error decreases considerably as the number of sweeps increases. This is due to the fact that the network is forced to learn the patterns across more iterations. Accordingly, the error is approximately the same if the total number of patterns used to train the network ties in. Therefore, if there are few data, the number of network iterations in the learning phase needs to be increased for the error to be considered acceptable.

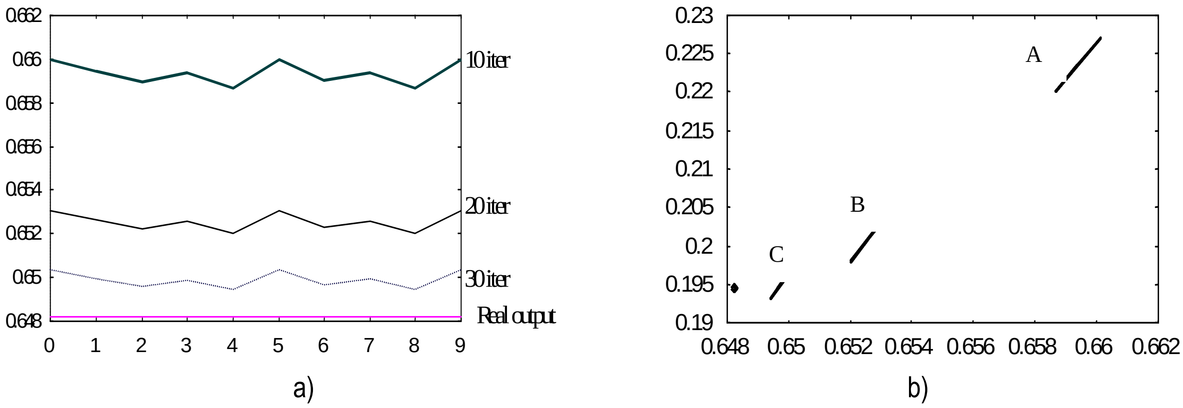


Figure 7. Real outputs from the Henon system network

Figure 7 a) shows that when the number of iterations increases, then the output of the network will better approximate the target output, that is, the stable point 0.648. One of the more important advantages of this technique is that the controllers obtained are very stable even with low random dynamic noise or with few data. Figure 7 b) shows the behavior of the Henon map with randomly added dynamic noise, controlled at the stable point and with 30 iterations in the learning phase.

Validate Model

Once the learning phase is complete, it is necessary to check if the network is able to control the function at the stable point.

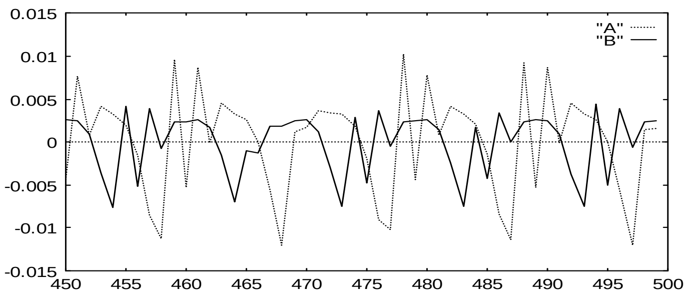


Figure.8. Real outputs of the network of the first component of the controlled Tinkerbell function around each point of the period-5 orbit (initial points A and B)

The error will be checked for acceptability, because the network needs to be validated with different orbits and the outputs and the errors need to be examined. It is also necessary to verify that the control model is robust, that is, to study the network errors when there is any sort of noise in the patterns. The fastest way of verifying the network output errors is by means of a graph.

Applications

The control models have been designed according to the previously described neural networks-based procedure to control several systems in unstable periodic orbits. Second-order discrete non-linear systems have been controlled at an equilibrium point. The chosen systems are the Henon and the Lozi systems [Hernandez 1999]. Another discrete system, Ikeda, was controlled [Hernandez 1999] at equilibrium point and a period-5 point, that is, in a period-1 and a period-5 limit cycle. The Tinkerbell system [Hernandez 2000] was controlled in period-1, 5, and 7 limit cycles. Later, the continuous Lorenz system [Hernandez 2001] was controlled by means of the same procedure.

Also, the same control model has been applied to a dynamic system with multiple attractors, and a controller has been designed to transfer and control the orbits of any basin of attraction of the different system attractors at an equilibrium point and in a period-5 limit cycle. The chosen system is Gumowski and Mira's system [Hernandez1 2001]. Finally, two chaotic signals for both discrete and continuous systems were simultaneously controlled [Hernandez 2002], [Hernandez1 2002].

Conclusions

The main contributions of this work are:

- The construction of a controller model that uses a neural network which is very simple to design.
- The control method is flexible, it can be adapted to any system and can even to control and classify several systems simultaneously.
- The control can be applied to any point of the system, there being no need to wait for the system to approach the control target orbit.
- Control is effective for as long as it is applied, the accumulation of computational errors has no influence, whereas, in other methods, it destabilizes the system after a number of iterations.
- Control is robust, its behavior is satisfactory even in the presence of random dynamic noise and even if there are few data, which is very important due to chaotic systems' sensitivity to initial conditions.
- The systems use the same model both to control a system with a single attractor in an unstable periodic orbit and to control a multi-attractor system in an unstable periodic orbit of any basin of attraction, transferring the system dynamics from one attractor to another.
- The learning speed of the designed control model is high, and few errors are generated.

The networks used for the control model have supervised learning. Other types of networks, such as associative networks, might be investigated. It is also worth looking at the possibility of controlling chaotic systems in which attractors are not known by training the network with orbits of the different basins of attraction.

References

[Barreto 1995] 'Multiparameter Control of Chaos' Ernest Barreto and Celso Grebogi, Physical Review E. Volº2 nº4, pp 3553-3557, (1995)

[Casdagli 1989] "Nonlinear prediction of Chaotic time series" , Casdagli M., Physica D35, 335-356, 1989.

[Chen 1992] "On Feedback Control of Chaotic Dynamical Systems", G. Chen y X. Dong, Int. J. of Bifurcations and Chaos, 2, pp 407-411, (1992)

[Chen 1993] "From Chaos to Order", Chen G. and Dong X., *Int. J. of Bifurcations and Chaos 3*, Pp. 1363-1409, 1993.

[Gulick 1992] 'Encounters with Chaos', D. Gulick, McGraw-Hill, Inc 1st edition, (1992) [Hübler 1989] 'Adaptive control of chaotic systems', A.W. Hübler, Helvetica Physica A62, pp 343-346, (1989)

[Jackson 1990] '*The entrainment and migration controls of multiple-attractors systems'*. Jackson, E. A., Phys. Lett, A, 151, Pp. 478-484,1990.

[Jackson 1991] 'Entrainment and migration controls of two-dimensions maps', E. A. Jackson and A. Kodogeorgion, Physica D54, pp 253-265, (1991)

[Hernandez 1999] "Neural Network Control of Chaotic Systems*". Hernandez, C., Martinez, A., Castellanos, J., Computational Intelligence for Modelling, Control &Automation. Concurrent Systems Engineering Series. ISSN: 1383-7575. Vol. 54, pp. 1-8. 1999.*

[Hernandez1 1999] "Controlling Chaotic Nonlinear Dynamical Systems". *Hernández C., Martínez A., Castellanos J., Mingo L.F..: IEEE Catalogue Number 99EX357. ISBN: 0-7802-56X2-9. Piscataway N.J. USA. pp. 1231-1234. 1999.*

[Hernandez 2000] "Periodic Orbit Stabilization with Neural Networks". *Hernández C., Martínez A., Mingo L.F., Castellanos J.; Frontiers in Artificial Intelligence and Applications. Vol. 57. New Frontiers in Computational Intelligence and its Applications. IOS Press. ISSN: 0922-6389. pp. 109-118. 2000.*

[Hernandez 2001] "Migration Goal Control of Chaotic Systems with Neural Networks". *Hernandez, C., Castellanos, J., Martinez, A., Mingo L.F.; Knowledge Based Intelligent Information Engineering Systems and Allied Technologies. Frontiers in Artificial Intelligence and Applications. IOS Press Ohmsha. ISSN: 0922-6389. ISBN: 1-58603-1929. Vol.: 69. Part II. pp.: 1165-1169. 2001.*

[Hernandez1 2001] "Controlling Lorenz Chaos with Neural Networks". *Hernandez, C., Martinez, A., Mingo, L.F., Castellanos, J.; Advances in Scientific Computing, Computational Intelligence and Applications. WSES Press. ISBN: 96-8052-36-X. pp. 302-309. 2001.*

[Hernandez 2002] "Neural Control of Simultaneous Chaotic Systems". *Hernandez, C., Gonzalo, R., Castellanos, J., Martinez, A.; Frontiers in Artificial Intelligence and Applications. Vol. 82. IOSPRESS. ISSN: 0922-6389. pp. 527-531. 2002.*

[Hernandez1 2002] "Simultaneous Control of Chaotic Systems". *Hernández, C., Martinez, A., Castellanos, J., Luengo, C.; Recent Advances in Circuits, Systems and Signal Processing, WSEAS Press. ISBN: 960-8052-64-5. pp. 200-204. 2002.*

[Hübler 1989] '*Adaptive control of chaotic systems',* Hübler, A.W., Helvetica Physica A62, Pp. 343-346, 1989.

[Martín 1995] 'Iniciación al caos', Miguel Ángel Martín, Manuel Morán, Miguel Reyes; Editorial Síntesis, ISB: 84-7738-293-X, (1995)

[Nusse 1997] "Dynamics: Numerical Explorations", H.E. Nusse, J.A. Yorke, J.E. Marden and L. Sirovich (Springer-Verlag, New York). Series: Applied Mathematical Sciences V. 101, (1997)

[Lippmann 1987] 'An Introduction to Computing with Neural Nets' L. P. Lippmann, IEEE ASSP Magazine, April 1987 pp. 4-22, (1987)

[Ott 1990] '*Controling Chaos* ', Ott E., Grebogi C. & Yorke J. A.,. Phys. Rev. Lett, 64, Pp. 1196-1199, 1990.

Authors' Information

Juan Castellanos Peñuela- Departamento de Inteligencia Artificial, Facultad de Informática – Universidad Politécnica de Madrid (Campus de Montegancedo) – 28660 Boadilla de Monte – Madrid – Spain; e-mail: jcastellanos@fi.upm.es

Cristina Hernández de la Sota- Departamento de Inteligencia Artificial, Facultad de Informática – Universidad Politécnica de Madrid (Campus de Montegancedo) – 28660 Boadilla de Monte – Madrid – Spain; e-mail: cristinah@renfe.es

Rafael Gonzalo Molina- Departamento de Inteligencia Artificial, Facultad de Informática – Universidad Politécnica de Madrid (Campus de Montegancedo) – 28660 Boadilla de Monte – Madrid – Spain; e-mail: rgonzalo@fi.upm.es

Valentín Palencia Alejandro- Departamento de Arquitectura y Tecnología de Sistemas Informáticos, Facultad de Informática – Universidad Politécnica de Madrid (Campus de Montegancedo) – 28660 Boadilla de Monte – Madrid – Spain; e-mail: vpalencia@fi.upm.es