Paper Number:
DOE/METC/C-96/7237

Title:
Iterative Prediction of Chaotic Time Series Using a Recurrent Neural Network

Authors:
M.A. Essawy (Tennessee State Unversity)
M. Bodruzzaman (Tennessee State University)
A. Shamsi (METC)
S. Noel (METC)

RECEIVED
NOV 05 1996
OSTI

Conference:
Artificial Neural Network in Engineering (ANNIG) Conference

Conference Location:
Rolla, Missouri

Conference Dates:
October 10, 1996

Conference Sponsor:
University of Missouri at Rolla

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CONF-9610220--1

ITERATIVE PREDICTION OF CHAOTIC TIME SERIES USING A RECURRENT NEURAL NETWORK

DOE/METC/C--96/7237

MAGDI A. ESSAWY, MOHAMMAD BODRUZZAMAN
Department of Electrical and Computer Engineering
Tennessee State University, Nashville, TN 37209

ABOLGHASEM SHAMSI, STEPHEN NOEL
Morgantown Energy Technology Center (METC)
Morgantown, WV 26505

MASTER

ABSTRACT:
Chaotic systems are known for their unpredictability due to their sensitive dependence on initial conditions. When only time series measurements from such systems are available, neural network based models are preferred due to their simplicity, availability, and robustness. However, the type of neural network used should be capable of modeling the highly non-linear behavior and the multi-attractor nature of such systems. In this paper we use a special type of recurrent neural network called the "Dynamic System Imitator (DSI)", that has been proven to be capable of modeling very complex dynamic behaviors. The DSI is a fully recurrent neural network that is specially designed to model a wide variety of dynamic systems. The prediction method presented in this paper is based upon predicting one step ahead in the time series, and using that predicted value to iteratively predict the following steps. This method was applied to chaotic time series generated from the logistic, Henon, and the cubic equations, in addition to experimental pressure drop time series measured from a Fluidized Bed Reactor (FBR), which is known to exhibit chaotic behavior. The time behavior and state space attractor of the actual and network synthetic chaotic time series were analyzed and compared. The correlation dimension and the Kolmogorov entropy for both the original and network synthetic data were computed. They were found to resemble each other, confirming the success of the DSI based chaotic system modeling.

INTRODUCTION

Chaotic systems are known for their unpredictability, due to their sensitive dependence on initial conditions which is measured by positive Lyapunov exponents. [1] In other words, even when the exact model of a chaotic system is available, it is impossible to predict a chaotic system behavior for a long period of time. [2,3] The reason is that our measurements and calculations are never perfect and are susceptible to errors. Similar errors contribute to the non-exact determination of initial conditions. Any minute error in the initial conditions for a chaotic system will

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

turn, with time, into great differences in the results. However, short term predictions of chaotic systems are still possible. [2,4] How short the time duration is for valid prediction depends on the system average loss of information represented by its Lyapunov exponents and Kolmogorov entropy. Some systems are less predictable than others due to faster loss of information with time, represented by larger positive Lyapunov exponents and larger positive Kolmogorov entropy. Since exact predictions are not possible for such systems, approximate models may produce results as satisfactory as those produced by exact models. This makes neural network based models very good candidates for such applications. Neural network based models are known not to be exact models, but they are easy to implement, robust, fast and data driven. Dynamic neural network models are preferable for such applications, due to their ability to capture time behaviors. [5]

In this work we used a special type of dynamic neural network called the Dynamic System Imitator (DSI). [6] We developed the DSI a few years ago and have used it for several modeling and control applications. [6,7,8] The DSI is biologically motivated and is specially designed to model a wide variety of dynamic systems. It has both short term and long term memory mechanisms that enable the modeling of a system's transient and steady state behavior. In addition, the DSI behavior depends on its initial conditions the same as any differential equation model does, even though no explicit differential equation solving is incorporated in this case. What we know of the DSI characteristics encourages us to recommend it for modeling non-linear systems in general and chaotic systems in particular. More details about the DSI will be discussed below. Since the dynamics of most real systems are accessed via time series measurements, the focus in this paper will be on modeling chaotic time series. The way the chaotic time series model is implemented in this paper is through a one step predictor model. The dynamics of a chaotic time series are modeled through training the DSI to perform a one step prediction. However, at any point of time, the DSI response depends on the initial conditions at time zero, the history of inputs and network state variables, and the current network input. Assuming the network was able to capture the dynamics in the time series, we can start the trained network with any set of initial conditions, use a number of initial data points to put the network on track, and iteratively feed the output of the network back to compute next predicted values. Even though we applied this methodology to several theoretical systems, the current motive is to use it in a strategy to identify certain chaotic behavior modes encountered in a Fluidized Bed Reactor (FBR) system. [9] This identification can be achieved by comparing the actual measurement from the chaotic system with the time series predicted by the DSI iterative predictor model, starting from a short time history of the actual data. In this paper, results from the DSI iterative predictor are discussed for chaotic time series generated using the logistic, Henon and the cubic equations, in addition to one experimental time series measured from the FBR experiment at the Morgantown Energy Technology Center. [9] The DSI network model was evaluated based on comparison made on the time series, phase space trajectories, and chaotic parameters computed from these trajectories. However, in this case, time series similarities are not as important as similar phase space trajectories and similar chaotic parameters. [5] The actual combination of DSI iterative predictor and FBR system is published in other domains. [9]

THE DYNAMIC SYSTEM IMITATOR (DSI) NEURAL NETWORK

The neural network used for the chaotic time series prediction in this paper is a dynamic neural network called Dynamic System Imitator (DSI). The DSI is a fully recurrent neural network that is specially designed to model a wide variety of dynamic systems. [6,7] As shown in Figure.1, the DSI has a three layer structure: input, hidden, and output layer. Connections have both weights and integrators in parallel to model short term and long term memory mechanisms that handle modeling of time behaviors and time lags in real systems. Every node in the input layer has one input, $x_k(t)$, and two outputs defined by: [6]

$$\begin{aligned} o_{1j}^{i}(t) &= x_j(t)\ , \\ o_{2j}^{i}(t) &= \int_0^t x_j(t)\, dt\ . \end{aligned} \qquad (1)$$

The input layer is fully connected to the hidden and output layers. Every node in the hidden layer is connected to every other node in the hidden and output layers and to itself. The two outputs of every neuron are computed according to the relationship: [6]

$$\begin{aligned} o_{1j}^{h}(t) &= A_j\psi_j(B_j^{-1}[\sum_{k=0}^{m}(w_{1jk}^{hi}o_{1k}^{i}(t) + w_{2jk}^{hi}o_{2k}^{i}(t)) + \sum_{k=0}^{n}(w_{1jk}^{hh}o_{1k}^{h}(t) + w_{2jk}^{hh}o_{2k}^{h}(t))]\,) \\ o_{2j}^{h}(t) &= C_j\psi_j(D_j^{-1}\int_0^t o_{1j}^{h}(t)\, dt) \end{aligned} \qquad (2)$$

where ψ_j is a nonlinear transformation function, and A_j, B_j, C_j, and D_j are adjustable weights associated with the hidden neuron j, which are used to shape the transfer function for every node. B and D are used to adjust the steepness of the function, while A and C are used to adjust its min-max value. Also m and n are the number of processing nodes in the input and hidden layers respectively; w_1 and w_2 refer to weights associated with direct and delayed outputs, respectively. The superscript h refers to the hidden layer, i refers to the input layer, hi refers to weights from the input to hidden layer, and hh refers to weights from the hidden to hidden layer. The integrators and feedback connections promote enough asynchrony and interaction in the network to model several system state variables as a function of time. When enough intermediate state variables are generated, the network output can be a function of those state variables. The output layer has only one output per node, which is computed according to the following equation: [6]

$$o_j^{o}(t) = E_j\psi_j(F_j^{-1}[\sum_{k=0}^{m}(w_{1jk}^{oi}o_{1k}^{i}(t) + w_{2jk}^{oi}o_{2k}^{i}(t)) + \sum_{k=0}^{n}(w_{1jk}^{oh}o_{1k}^{h}(t) + w_{2jk}^{oh}o_{2k}^{h}(t))]) \qquad (3)$$

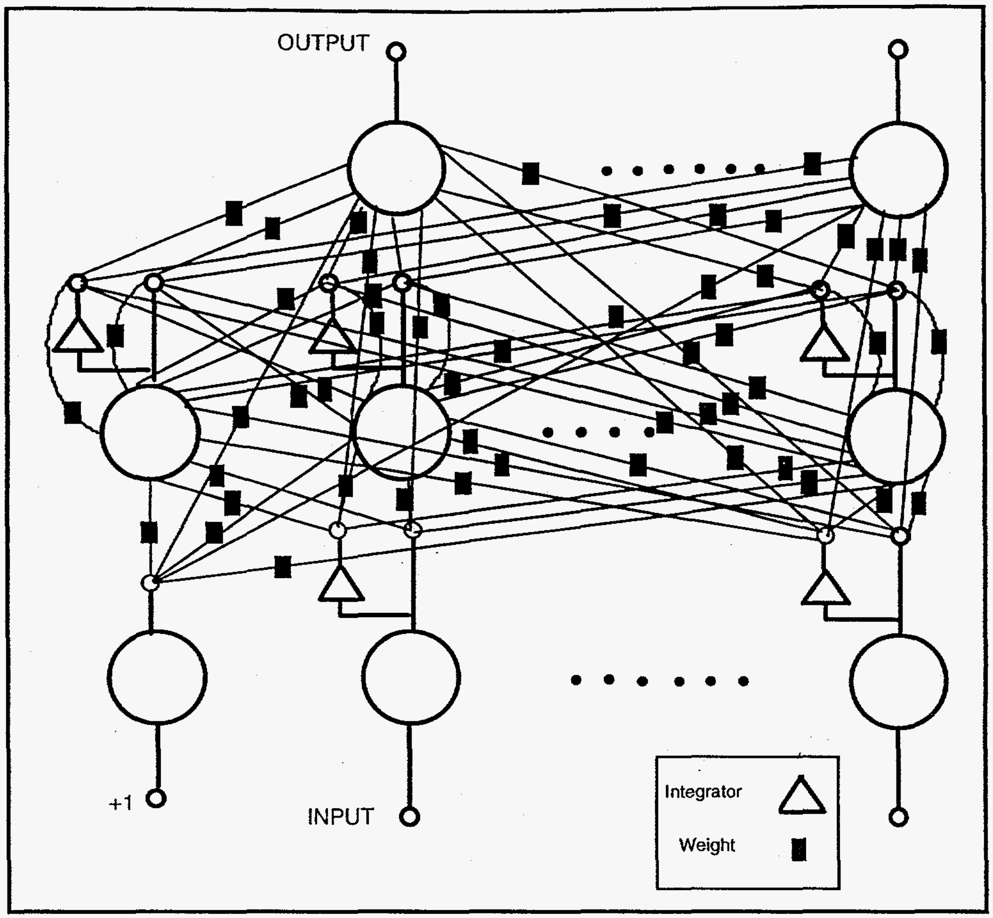

Figure 1: Schematic diagram of the Dynamic System Imitator (DSI) Network.

where m and n are the number of nodes in the input and hidden layers respectively, and E_j and F_j are two constants associated with each node in the output layer to shape its own transfer function when needed. The superscript o refers to the output layer, i refers to the input layer, h refers to the hidden layer, oi refers to weights from the input to output layer, and oh refers to weights from the hidden to output layer.

By looking at the complete DSI network design, it is easy to observe that the node interaction, information feedback, and action transfer time lags generate an activity in the network that is similar to the internal activity in real dynamic systems. Even with a simple configuration, the DSI has a complex structure, which makes it very difficult to train. A multi-dimensional optimization technique that adopts the simplex method is used to train the DSI. [6] There are two other difficulties in the training of such a network. One is that a very long time series cannot be introduced to the network at one time and must be divided into reasonably sized sections. The other is that the behavior of the network is dependent on the initial conditions of its state variables, and a certain set of initial conditions has to be found in conjunction with every network design. In other words, whenever the network is updated, a new set of initial conditions must be found. The first problem was overcome by using a moving time window that cascades the introduction of the time series segments to the network during training. The network final conditions at the final training step of every segment is taken as the initial conditions for the next segment, to keep the physical association between the consequent segments of the time series. The second problem was overcome by adding initial conditions search method that runs after every training iterate to find updated initial conditions for every modified version of the network. An arbitrary set of initial conditions can be used at the start of the training process.

FLUIDIZED BED REACTOR (FBR) PRESSURE DATA COLLECTION

Morgantown Energy Technology Center has built and operated a cold flow model to emulate fluid dynamics in a Fluidized Bed Reactor (FBR). The cold flow verification test facility consists of a ten foot high jetting fluidized bed made of clear acrylic and configured as a half cylinder vessel to facilitate jet observation. A central nozzle, made up of concentric pipes, continuously fed solids at 0 to 8 psig pressures. Separate flow loops controlled the conveyance of solids (inner pipe), the make-up air flow (middle pipe), sparger flow (outer pipe), and six air jets on the sloping conical grid. The half round fluid bed model provided useful information to study fluidization and design issues including jet penetration, chaotic pressure fluctuations, and mass flow rates of particles in various regions of the jetting fluid bed. The fluid bed tests were conducted using cork particles to simulate the relative density of gases to scale for a high pressure coal conversion reactor. As expected, the test generated chaotic pressure fluctuations. The differential pressures were measured at two location with each location consisting of two pressure taps spaced four inches apart. The lower pair of pressure taps were placed at a height just above the nozzle and the upper pair of pressure taps were placed at a height where the jet becomes evenly distributed across the diameter of the reactor. Differential pressure data collected at the higher sensor served as the primary data for the investigation of chaos. It clearly indicated the fluidization regime of the bed supported by visual observations. Data were collected on a data acquisition card at a rate of 50 Hz.

USING THE DSI FOR ITERATIVE PREDICTION OF CHAOTIC TIME SERIES

A simple configuration of the DSI neural network was used for the iterative prediction of a chaotic time series. This configuration has one node in the input layer, three nodes in the hidden layer, and one node in the output layer. The network was trained to predict one point ahead of the time series, using a set of previous values. These values are not explicitly used for prediction, but are implicitly used by adjusting the state of the network from which the prediction is performed. The prediction method is based upon the idea that once the network is trained to predict one point ahead with good accuracy, this same point can be used as an input to the network to predict the next point. This process can be repeated iteratively to predict many points in the time series. Naturally, the accuracy of prediction will deteriorate over time. During training, a time window of 200 points was used to cascade the time series to the network. The algorithm was applied to three types of simulated chaotic time series generated from the logistic, Henon, and cubic equations, in addition to one experimental time series measurement taken from a Fluidized Bed Reactor (FBR). FBR systems are known for their chaotic behavior, as discussed in several references. [9,10,11,12] The network was able to learn simple one step prediction in a reasonable number of training iterations. After training, the output of the DSI was used iteratively to generate the time series. However, the training initial conditions together with the first actual 25 points of the training time series should be used to start the DSI, in case the time behavior of the training time series must be generated. If not, any network initial conditions and any starting points can be used to generate the state space behavior of the system to which the training time series belongs.

Comparing the predicted time series to the actual time series, we found that the DSI was able to track the training time series time behavior for a short period of time (around 30 points), when started from the training initial conditions , and activated by the first 25 points of the training time series. However, it was able to track the state space attractor to which the training time series belongs, starting from any initial conditions, activated by any arbitrary set of starting points. The only case that fails is zero initial conditions together with zero starting points, which leads to zero solution The actual and predicted time series for all cases are shown in Figures 2-9, while the actual and predicted state space attractors are shown in Figures 10-17. To quantitatively compare these attractors, the correlation dimension and Kolmogorov entropy for the actual and predicted attractors were computed. The results of the correlation dimension and Kolmogorov entropy of the different cases are summarized in Table.1. The correlation dimension is computed, according to the box-counting method, from the slope of the lines representing the correlation integral versus ε (the size of a computing box) on log-log curves for different embedding dimensions. The correlation integral was computed according to the following equation: [13]

$$C(N,\varepsilon) = \frac{2}{N(N-1)} \sum_{j=1}^{N} \sum_{i=j+1}^{N} \theta(\varepsilon - |x_i - x_j|) \tag{4}$$

where $\theta(x)=1$ for $x>0$ and $\theta(x)=0$ for $x<0$. The Kolmogorov entropy was computed according to the equation: [14]

$$K_{m,d} = \frac{1}{\tau m} \ln \frac{C_d(\varepsilon)}{C_{d+m}(\varepsilon)} \tag{5}$$

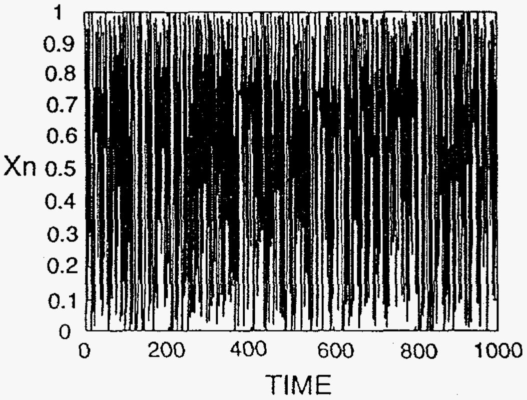

Figure 2: Actual logistic time series.

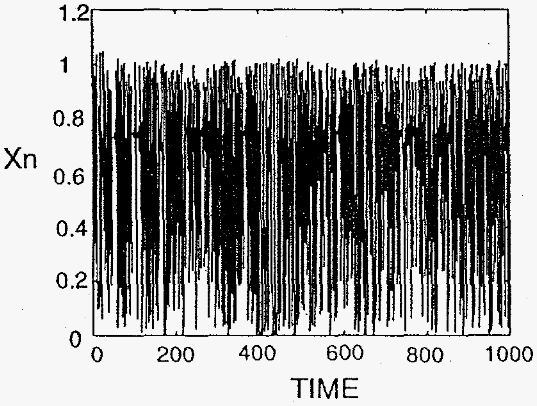

Figure 3: Synthetic logistic time series.

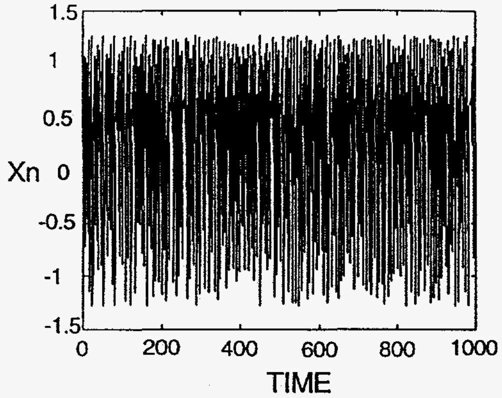

Figure 4: Actual Henon time series.

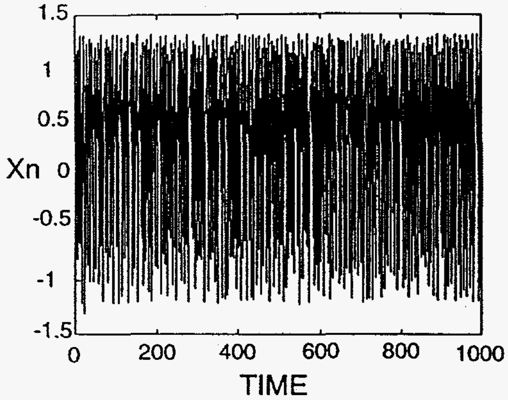

Figure 5: Synthetic Henon time series.

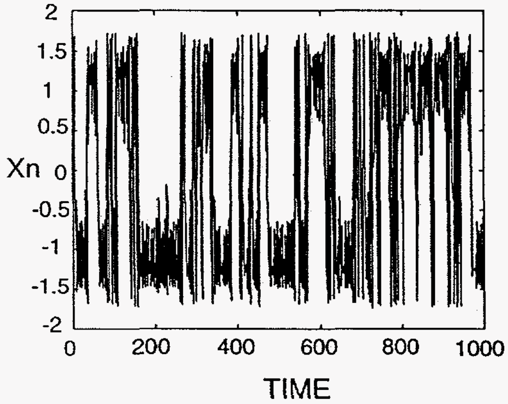

Figure 6: Actual cubic time series.

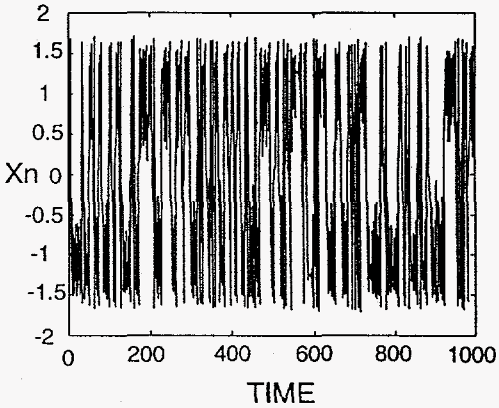

Figure 7: Synthetic cubic time series.

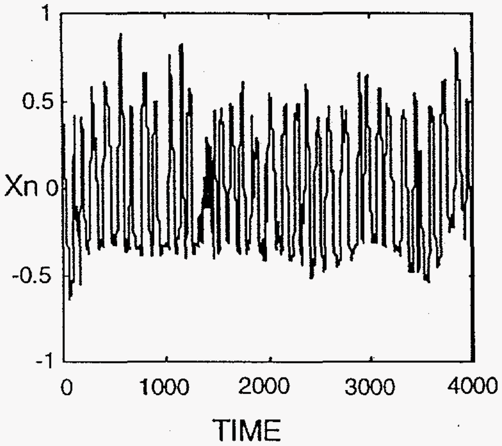

Figure 8: Actual normal FBC time series.

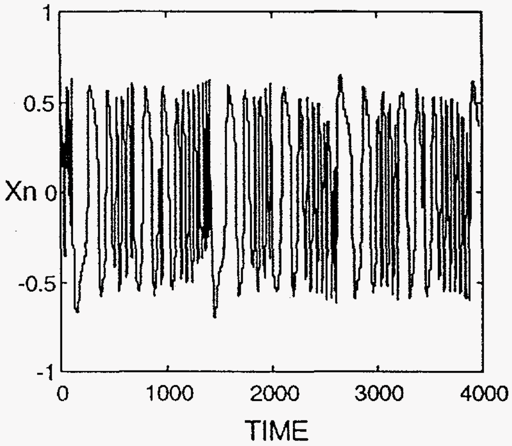

Figure 9: Synthetic normal FBC time series.

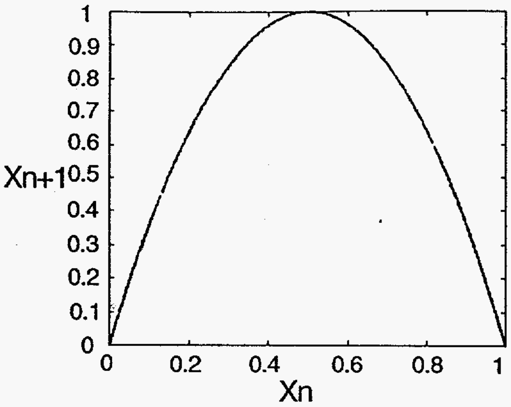

Figure 10: Actual logistic attractor.

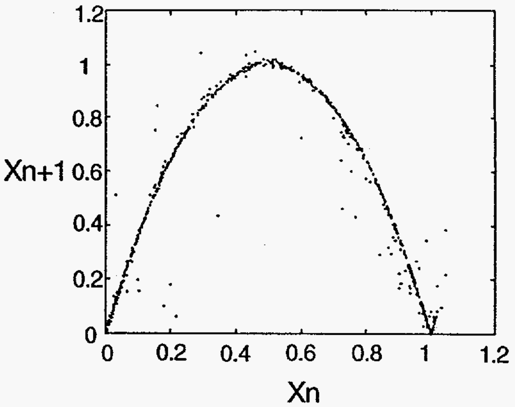

Figure 11: Synthetic logistic attractor.

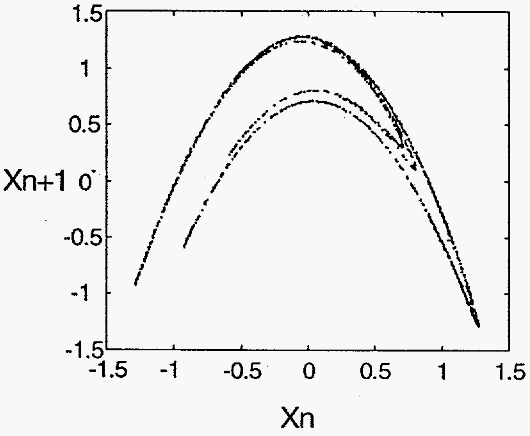

Figure 12: Actual Henon time attractor.

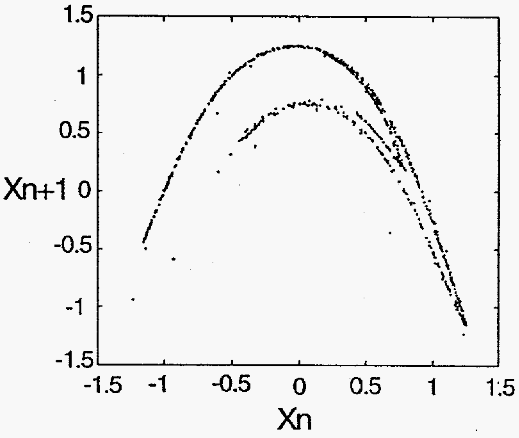

Figure 13: Synthetic Henon attractor.

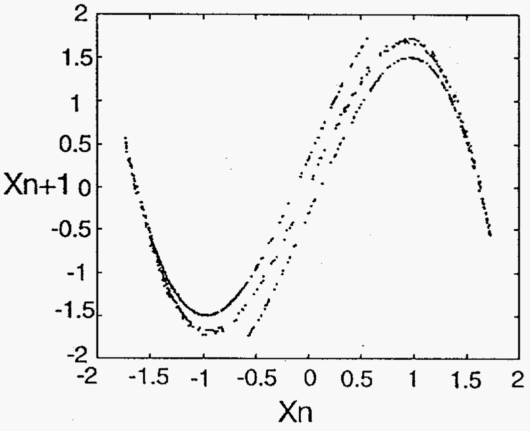

Figure 14: Actual cubic attractor.

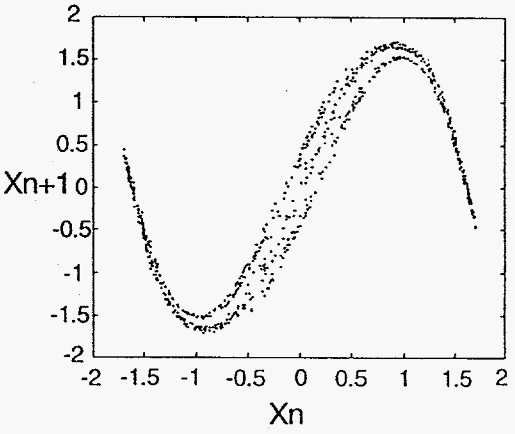

Figure 15: Synthetic cubic attractor.

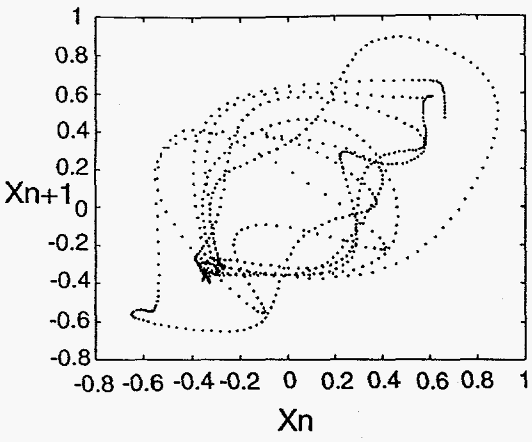

Figure 16: Actual normal FBC attractor.

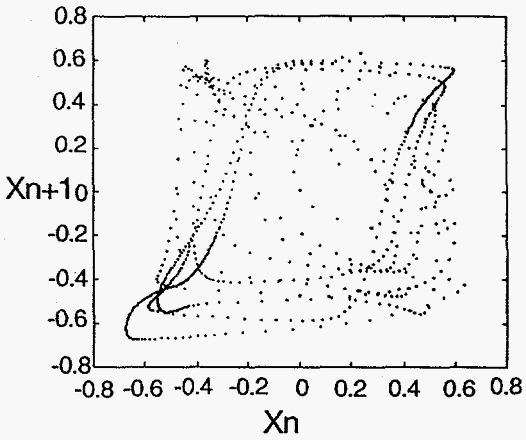

Figure 17: Synthetic normal FBC attractor.

TABLE 1: COMPARISON OF THE ACTUAL AND DSI SYNTHETIC ATTRACTORS PARAMETERS

Time Series	Correlation Dimension	Kolmogorov Entropy
Logistic Map (actual)	1.0457±0.0057	0.6872±0.0198
Logistic map (synthetic)	1.2316±0.0374	0.6013±0.0435
Henon Map (actual)	1.2607±0.0331	0.3267±0.0135
Henon map (synthetic)	1.3171±0.0725	0.2962±0.0239
Cubic Map (actual)	1.2248±0.0090	0.4245±0.0183
Cubic Map (synthetic)	1.8171±0.01365	0.5340±0.0134
FBC normal (actual)	2.934±.065	5.034±.095
FBC normal (synthetic)	2.12±.07	6.3764±1.26

CONCLUSION

In this paper a dynamic neural network based model for chaotic time series has been developed. A one step predictor model was used to iteratively generate chaotic time series. A dynamic neural network called the Dynamic system Imitator (DSI) was utilized. The DSI has distinguishable dynamic features due to its special architecture. The DSI time behavior depends on its initial conditions. After training, the DSI was able to generate the chaotic time series in all test cases. For a short period of time, it was able to generate the same time behavior of the training time series if started with the same initial conditions and the first initial points of the time series. Furthermore, it was able to track the system attractor to which the training time series belongs, for any period of time, for any initial conditions and any initial points, in all test cases. The only case that fails is the zero initial conditions together with zero starting points, which leads to a zero solution. This methodology was applied to three known chaotic models, the logistic, Henon and cubic maps, in addition to one experimental time series taken from differential pressure measurement of a Fluidized Bed Reactor (FBR). The correlation dimension and the Kolmogorov entropy for the actual and DSI network synthetic data were computed and compared. There is a very good match between the actual and synthetic time series parameters in all cases which indicates that the DSI was able to learn the dynamics in those chaotic time series to a very good extent.

ACKNOWLEDGMENT

This work has been supported by the DOE Grant No. DE-FG22-94MT94015. We wish to thank the DOE for their support of this research.

REFERENCES

1. E. Ott, "Controlling chaos," Physics Today, pp 34-40, May 1995.
2. J.D. Farmer, J. Sidorowich, "Predicting Chaotic Time Series, " Phys. Rev. Lett. 59, pp 845, 1987.
3. M. Casdagli, "Non-linear Prediction of Chaotic Time Series," Physica D 35, pp 335, 1989.
4. G. Sugihara, R. M. May, "Non-linear Forecasting as a way of Distinguishing Chaos from Measurement Error in Time Series," Nature 344, pp 734, 1990.
5. J. C.Principe, J-M Kuo, "Dynamic Modeling of Chaotic time Series with Neural Networks," Advances in Neural Information Processing Systems 7, Ed. Tesauro, Touretzky, Leen, pp 311-318, 1995.
6. Magdi A. Essawy, "Artificial Neural Networks for System Modeling Monitoring and Control," Ph.D. Dissertation, University of Tennessee, Knoxville, July 1995.
7. Magdi A. Essawy, Robert E. Uhrig, "The Dynamic System Imitator: A New Approach for a Dynamic Neural Network Architecture," the 9th Power Plant Dynamics, Control & Testing Symposium, Knoxville, Tennessee, May 24-26 1995.
8. Magdi A. Essawy, Mohammad Bodruzzaman, "Using A Recurrent Neural Network for Chaotic Behavior Control," to appear in the World Congress on Neural Networks '96, Sep. 15-20 1996.
9. Magdi A. Essawy, Mohammad Bodruzzaman, "Recurrent Neural Network-Based Monitoring and Control of Chaotic Fluidized Bed Combustion Systems," the 4th Annual HBCU Private Sector Energy Research and Development Technology Transfer Symposium, Greensboro, NC, April 2-3 1996.
10. Fuller, "Interpretation of Pilot Scale, Fluidized Bed Behavior Using Chaotic Time Series Analysis," 12th international FBC Conference, LA Jolla, CA, May, 1993.
11. S. Daw, "Characterization of Voidage and Pressure Signals from Fluidized Beds, Using Deterministic Chaos Theory," Fluidized Bed Combustion, ASME Conference 1991, pp 777-785.
12. S. Halow, "Characterizing Fluidized Bed Behavior, By Decomposition of Chaotic Phase Space Trajectories," 1993 Annual AIChE Meeting, St. Louis, MO, November, 1993.
13. Ding, C. Grebogi, E. Ott, T. Sauer, J. A. Yorke, "Plateau Onset for Correlation Dimension: When Does It Occur ?", Physics Review Letters 70, 3872, 1993.
14. Peter Grassberger, "Estimation of the Kolmogorov Entropy from a Chaotic Signal", Physical Review A, Vol. 28, No. 4, pp 2591-2593, October, 1983.