LA-UR 94-3116

Conf-9409112--1

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

TITLE: FUZZY CONTROLLERS IN NUCLEAR MATERIAL ACCOUNTING

AUTHOR(S): Andrew Zardecki

SUBMITTED TO: First International Workshop on Fuzzy Logic and Intelligent Technologies in Nuclear Science, Mol, Belgium, September 19-21, 1994

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

DISCLAIMER

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

FUZZY CONTROLLERS IN NUCLEAR MATERIAL ACCOUNTING*

Andrew Zardecki
Los Alamos National Laboratory, MS E541
Los Alamos, NM 87545, USA

ABSTRACT

Fuzzy controllers are applied to predicting and modeling a time series, with particular emphasis on anomaly detection in nuclear material inventory differences. As compared to neural networks, the fuzzy controllers can operate in real time; their learning process does not require many iterations to converge. For this reason fuzzy controllers are potentially useful in time series forecasting, where we want to detect and identify trends in real time. We describe an object-oriented implementation of the algorithm advanced by Wang and Mendel. Numerical results are presented both for inventory data and time series corresponding to chaotic situations, such as encountered in the context of strange attractors. In the latter case, the effects of noise on the predictive power of the fuzzy controller are explored.

1. Introduction

The control of nuclear materials is usually viewed as consisting of two distinct problems. First, there is the problem of guarding against losses. Second, one should be able to demonstrate that the amount of material lost is below some specified amount; that is, the steps taken to guard against losses are effective.[1]

Materials accounting for safeguards is based on the continuity equation, which states that the net amount of nuclear material transferred into a close, bounded volume during some time interval must equal the increase in material stored within the volume during the same interval. In the analysis of inventory data, the basic quantity is the inventory difference (material unaccounted for, or MUF), defined in terms of the physical inventory measurement I_n and the net material transfer T_n as

$$M_n = I_n + T_n - I_{n+1}, \tag{1}$$

where n refers to the nth balance period.[2] MUF is the algebraic difference between a book inventory and a physical inventory. A negative M_n represents a gain of material, whereas positive M_n represents material loss.

The problem of nuclear material control has traditionally been formulated in terms of classical statistical methods or by using specialized techniques, such as Kalman filtering. With the advent of fast computers, several new techniques are beginning to be used in nuclear safeguards. Starting with the pioneering work of Lapedes and Farber,[3] the neural networks have been advantageously applied to predicting and modeling time series, in

*Work supported by the US Department of Energy, Office of Safeguards and Security.

domains as distinct as chaotic dynamics and predicting corporate bond ratings. A methodology for monitoring nuclear reactor systems has been formulated by Ikonomopoulos, et al.,[4] in terms of a combination of neural networks and fuzzy logic. A recent study by Burr and Wangen[5] uses nonlinear time series analysis for safeguards data from reprocessing plants.

For most real-world control and signal processing problems, the information concerning design and evaluation can be classified into two kinds: numerical information obtained from sensor measurements and linguistic information obtained from human experts. Generally, neural control is suited for using numerical data pairs (input-output pairs), whereas fuzzy control is an effective approach to utilizing linguistic rules. When fuzzy rules are generated from numerical data pairs, the two kinds of information are combined into a common framework.[6]

As compared to neural networks, the fuzzy controllers can operate in real time; their learning process does not require many iterations to converge. For this reason fuzzy controllers deserve their legitimacy in time series forecasting, where we want to detect and identify trends in real time. From the standpoint of mathematics, both neural networks and fuzzy controllers stand on a solid footing: they can be viewed as universal approximators.[7]

This paper describes an object-oriented implementation of the algorithm advanced by Wang and Mendel.[6] Numerical results are presented both for time series with randomness and time series corresponding to chaotic situations, such as encountered in the context of strange attractors. In the latter case, the effect of noise on the predictive power of the fuzzy controller is explored. In addition, by introducing a distance between observed and predicted data, one can apply the results of this study to a pattern recognition of temporal signatures.

2. Generating Fuzzy Rules

Following Ref. 6, we summarize the algorithm that allows us to generate fuzzy rules from numeric data. The algorithm consists of five steps, the first of which simply divides the input and output spaces into fuzzy regions and assigns to each region a fuzzy membership function. Each membership function is triangular; however, a different shape, e.g., trapezoidal, would not change the essence of the method.

The second step, in which fuzzy rules are generated from given data pairs, is crucial. Suppose we are given a set of desired input-output data pairs:

$$\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}; y^{(1)}\right), \quad \left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}; y^{(2)}\right), \ldots \qquad (2)$$

where x_1, x_2, and x_3 are inputs and y is the output. We generate a set of fuzzy rules from the input-output pairs of Eq. (1) and use these fuzzy rules to determine a mapping f: $(x_1, x_2, x_3) \rightarrow y$.

To resolve possible conflicts in rules definition, in the third step we assign a degree to each rule. This is a numerical value equal to the product of the degrees of an individual rule's members. When some *a priori* information about the data is available, we can modify the rule degree by an additional multiplicative factor reflecting this information.

In the fourth step we create a combined fuzzy rule base. If M_1, M_2, and M_3 are the numbers of quantized levels each rule if-member can take, then the number N_R of possible rules is given by $N_R = M_1 \times M_2 \times M_3$. The arising three-dimensional parameter space can be viewed as consisting of N_R cells. If a linguistic rule is an "and" rule, it fills only one cell; an "or" rule fills all the cells in the rows or columns corresponding to the regions of the IF part.

The final fifth step defines the centroid defuzzification scheme. For inputs x_1, x_2, and x_3, let $m^i_I(x_1)$, $m^i_I(x_2)$, and $m^i_I(x_3)$ denote the input membership values of the ith fuzzy rule. Using product operation, we determine the degree of the output, m^i_O, corresponding to x_1, x_2, and x_3 as

$$m^i_O = m^i_I(x_1)\, m^i_I(x_2)\, m^i_I(x_3) \quad . \tag{3}$$

If the total number of rules is K, then the following centroid defuzzification formula defines the output,

$$y = \frac{\sum_{i=1}^{K} m^i_O \langle y^i \rangle}{\sum_{i=1}^{K} m^i_O} \,, \tag{4}$$

where $\langle y^i \rangle$ denotes the center value of the output region corresponding to ith rule.

The implementation of the Wang-Mendel algorithm is based on the approach of Viot,[8] adapted to the dynamic rule generation. The centroid defuzzification scheme described earlier has been somewhat simplified in the actual numerical implementation by employing the center of gravity method instead.

3. Signal Extraction

To illustrate former considerations, we apply our algorithm to two sample problems. In both cases we deal with a discrete time series $z(k)$, $k = 1, 2, \ldots$, which is simply a mapping from a set of whole numbers to reals. The prediction (forecasting) problem consists in finding $z(k+1)$, given a window of past n measurements $z(k-n+1)$, $z(k-n+2), \ldots, z(k)$. The length n of the window affects the maximum number of rules and leads quickly to combinatorial explosion as n increases. For this reason, a reduction scheme of a fuzzy logic system has recently been proposed.[9]

The term single-step prediction or one-step-ahead prediction is used when the measurement window is given the actual values of the observed time series. In iterated single-step predictions, the predicted output is fed back as input for the next prediction and all other inputs are shifted back one unit. That the fuzzy controller is not well adapted to multi-step prediction follows from the effect of error amplification by a nonlinear system. At each time step, a small forecast error is magnified by a large amount producing, as a rule, uniform time series without any reference to the original one. This is reminiscent of the sensitive dependence to the initial conditions found in chaotic systems.[10]

In the following we investigate the significance of the length of measurement window and the robustness of the controller with respect to an additive noise. In both cases we attempt to extract a narrow Gaussian signal superimposed on the time series under study.

3.1. Length of the Measurement Window

To investigate the effect of the measurement window length n, we employ the diffusion plant MUF data of Ref. 1. Steps 1–4 of the algorithm described in the previous section are used to generate the fuzzy rule base. Typically, we use between 100 and 200 input-output pairs to generate the rules. We use seven levels to quantize the input and output parameter space. In order to apply this method to signal detection, a Gaussian signal with magnitude of 0.1 and variance of 1 was added at time $t_0 = 100$. Figure 1, in which $n = 3$, shows the predicted time series normalized to unity and the difference between the predicted and observed values in the lower part. The Gaussian signal is detectable by an unusual spike in the difference signal. This is accompanied, however, by false alarms around times $t_1 = 15$ and $t_2 = 45$. There is simply not enough data to generate a reliable library of rules. The training process covering 140 time periods leads to a set of about 30 rules. The same data set can be used to train a back-propagation neural network. With little increase in predicting power, the cost to pay is an increase in the computer time by two orders of magnitude.

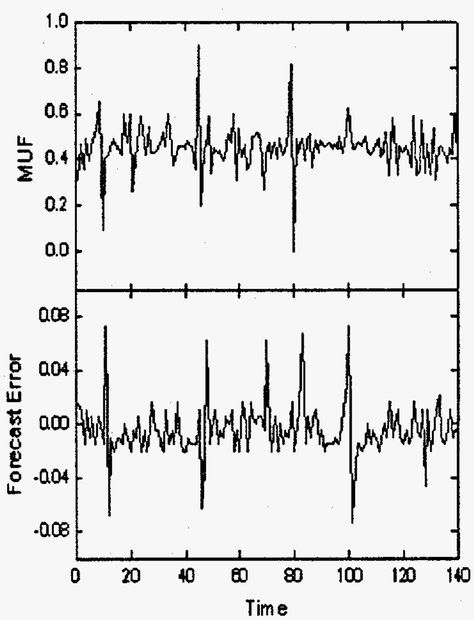

FIGURE 1. Diffusion plant MUF normalized data of Ref. 1 and the difference between the observed and predicted time series. Length of measurement window n = 3.

The forecasting results for $n = 4$ are shown in Fig. 2. It is evident that it is not the measurement window length, but the scarcity of data, that causes false alarms in the forecast.

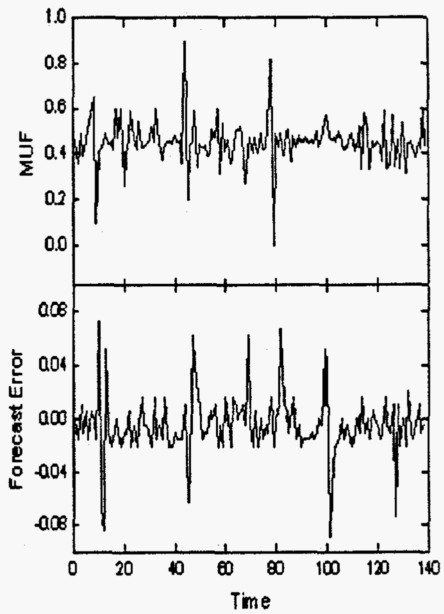

FIGURE 2. Diffusion plant MUF normalized data of Ref. 1 and the difference between the observed and predicted time series. Length of measurement window $n = 4$.

For $n = 2$, the forecasting power of the fuzzy controller breaks down, as illustrated in Fig. 3. This may be interpreted in terms of the slope and curvature of the time series graph. When $n = 2$, the only information available to the controller is the time series slope; the curvature information requires $n = 3$, or more. Surprisingly enough, we recover the correct trend of the MUF time series for $n = 1$ (see Fig. 3). In this case, which generates seven rules, only the preceding value of a forecasted point is known. If the model has too many free parameters relative to the number of cases in the training set, it can overfit the data. Rather than learning the basic structure of the data, enabling it to generalize well, the model learns irrelevant details of the individual cases.

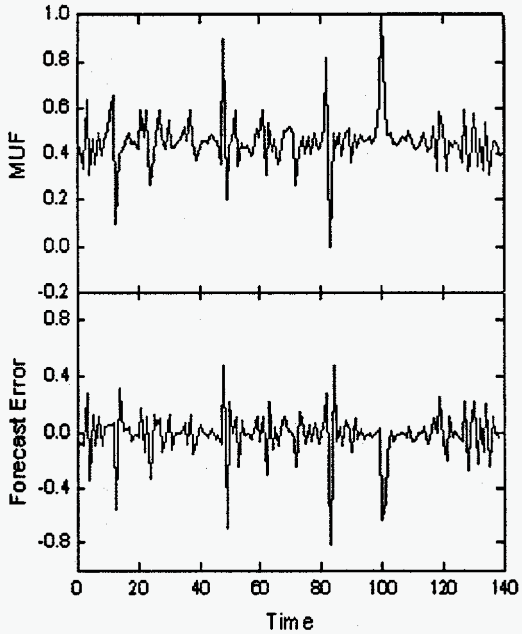

FIGURE 3. Diffusion plant MUF normalized data of Ref. 1 and the difference between the observed and predicted time series. Length of measurement window *n* = 2.

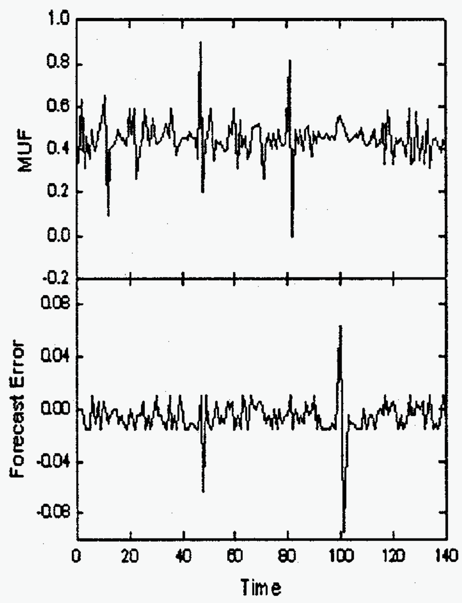

FIGURE 4. Diffusion plant MUF normalized data of Ref. 1 and the difference between the observed and predicted time series. Length of measurement window *n* = 1.

3.2. Impact of Noise

The second example, depicted in Fig. 5, deals with a chaotic dynamic system whose bounded subset has a non-integer fractal dimension. Such subsets are termed strange attractors.[10] The two-dimensional mapping studied by Henon has the form

$$x_{n+1} = 1 - cx_n^2 + y_n \,, \tag{5}$$

$$y_{n+1} = \beta x_n \,. \tag{6}$$

With the initial conditions given by $x_0 = 0.631$ and $y_0 = 0.189$, a strange attractor structure develops if we set $c = 1.4$ and $\beta = 0.3$. We train the fuzzy controller by considering in the x-y plane the Euclidean distance of the attractor points from the origin. The points shown in the lower part of Fig. 5 are the results of the recall phase.

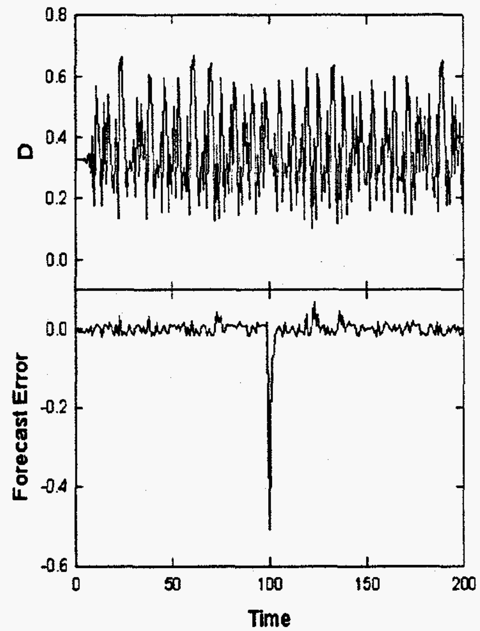

FIGURE 5. The Euclidean distance *D* from the origin $x = 0, y = 0$ of the vector time series corresponding to Henon's attractor. The lower figure showing the difference between the observed and predicted time series allows one to detect a superimposed signal at $t = 100$.

Considering the chaotic nature of the Henon attractor, it is remarkable we can detect the signal at $t_0 = 100$ using 200 data points as our training set. In the presence of noise, at fine resolution below the noise magnitude, the self-similar features of the attractors are truncated. The fractal dimension can then be shown to measure the properties of noise and not of the dynamical system.[11]

The distance D was modified by adding a white noise term according to the formula

$$D = \sqrt{x^2 + y^2} + \sigma N \ , \tag{7}$$

where σ describes the strength of fluctuation and N is a white noise process with unit variance. In Figs. 6–8, we show the significance of the noise term for $\sigma = 0.1$, 1.0, and 10, respectively.

Another way to apply the developed theory to pattern recognition is to introduce a distance of two time series. Thus, if N refers to the number of time steps, the Euclidean distance per time step would be based on the sum of differences squared at the same value of time moment:

$$d(x, x^{(0)}) = \frac{1}{N}\sqrt{\sum_{n=1}^{N}\left(x_n - x_n^{(0)}\right)^2} \ . \tag{8}$$

A given temporal signature x is accepted, if its distance from a fiducial shape $x^{(0)}$ does not exceed a chosen threshold value. This simple principle has been applied to pattern recognition of time series having some simple shapes. For example, a sinusoidal pattern is easily distinguished from a pattern of rectangular pulses.

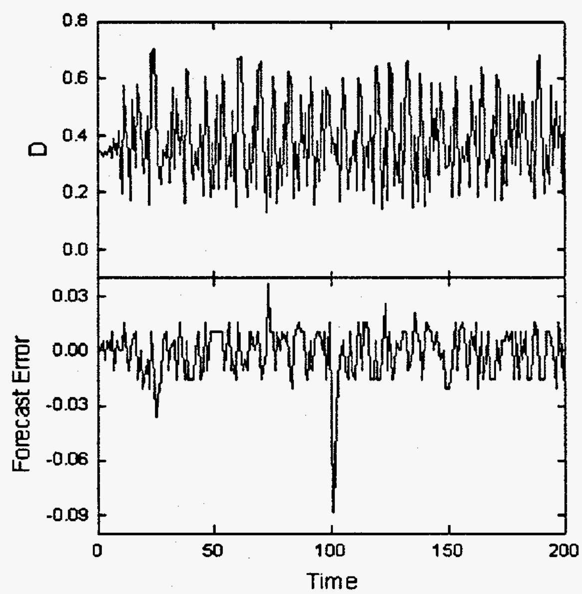

FIGURE 6. The Euclidean distance D from the origin $x = 0$, $y = 0$ of the time series corresponding to Henon's attractor. Noise strength $\sigma = 0.1$.

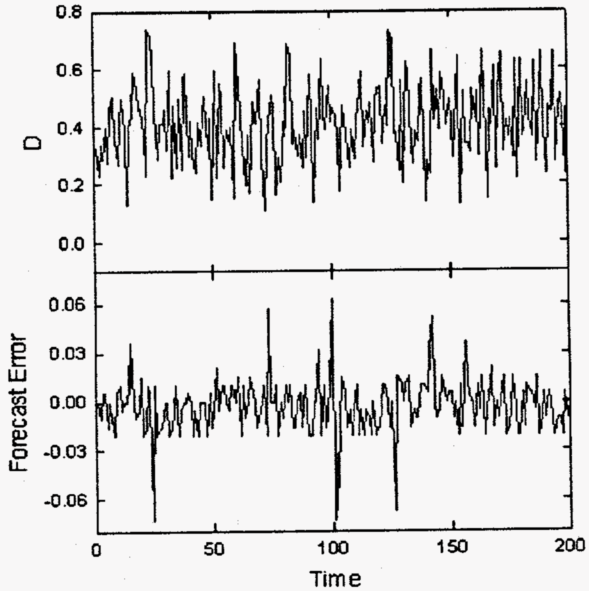

FIGURE 7. The Euclidean distance D from the origin $x = 0$, $y = 0$ of the time series corresponding to Henon's attractor. Noise strength $\sigma = 1$.

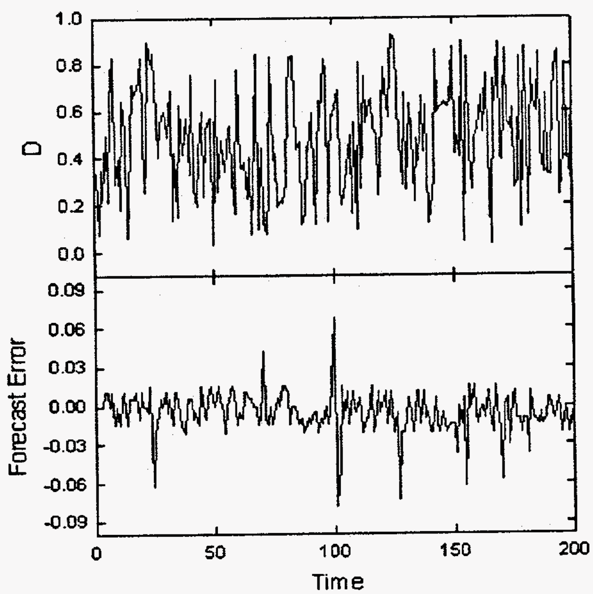

FIGURE 8. The Euclidean distance D from the origin $x = 0$, $y = 0$ of the time series corresponding to Henon's attractor. Noise strength $\sigma = 10$.

4. Conclusions

We have exploited the algorithm of Wang and Mendel to predict the time series corresponding both to random and chaotic systems. By considering the difference between the predicted and observed time series, we can detect unusual events superimposed on the time series. This technique can be exploited in nuclear safeguards. A distance function allows us to apply the theory to temporal pattern recognition.

References

1. L. Jaech, *Statistical Methods in Nuclear Material Control* (U. S. Atomic Energy Commission, Washington, DC, 1973).
2. P. Shipley, "Decision analysis for nuclear safeguards," in *Nuclear Safeguards Analysis*, edited by E. A. Hakkila (American Chemical Society, Washington, DC, 1978), pp. 35-64.
3. A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: prediction and system modeling," Los Alamos National Laboratory report LA-UR-87-2662, July 1987.
4. A. Ikonomopoulos, L. H. Tsoukalas, J. A. Mullens and R. E. Uhrig, "Monitoring nuclear reactor systems using neural networks and fuzzy logic," Proc. 1992 Topical Meeting on Advances in Reactor Physics, Charleston, SC, March 1992.
5. T. L. Burr and L. E. Wangen, "Authentication of reprocessing plant safeguards data through correlation analysis," Los Alamos National Laboratory draft report for POTAS Task D.89, June 1993.
6. L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Systems, Man and Cybernetics* **22** (1992) 1414–1427.
7. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks* **2** (1989) 359–366.
8. G. Viot "Fuzzy logic: Concepts to constructs," *AI Expert* **8** (1993) 26–33.
9. J. Hohensohn and J. M. Mendel, "Two-pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems," in *Proceedings of Third IEEE International Conference on Fuzzy Systems* (IEEE Service Center, Piscataway, NJ, 1994), pp. 696–700.
10. E. Ott, "Strange attractors and chaotic motions of a dynamical system," *Rev. Mod. Phys.* **43** (1981) 655–671.
11. A. Zardecki, "Noisy Ikeda Attractor," *Phys. Lett.* **90A** (1982) 274–277.