Pergamon

Computers chem. Engng, Vol. 21, Suppl., pp. S1149-S1154, 1997

Printed in Great Britain

PII:S0098-1354(97)00204-4

00098-1354/97 $17.00+0.00

The false nearest neighbors algorithm: An overview

Carl Rhodes*
Chemical Engineering, 210-41
California Institute of Technology
Pasadena, CA 91125 USA
car@aut.ee.ethz.ch

Manfred Morari
Institut für Automatik
ETH - Z/ETL
CH-8092 Zürich Switzerland
morari@aut.ee.ethz.ch

Abstract - The false nearest neighbors (FNN) algorithm is presented as a method for determining the proper regression vector for recreating the dynamics of nonlinear systems. This algorithm which was originally developed for the analysis of chaotic time-series, is used to determine the proper regression vector for input/output system identification and inferential prediction using time-series data. The FNN algorithm is presented, and the problem of analyzing noise corrupted time-series is discussed. The choice of adjustable parameters in the algorithm and the data requirements for the algorithm are also discussed. The application of the algorithm to the identification of an electrical leg stimulation experiment and an industrial pulp digester model is presented and the results are analyzed.

INTRODUCTION

Recently, there has been a great deal of work examining "black-box" identification of nonlinear systems from input/output data. The common focus of these works is determining and approximating a functional relationship between a given regression vector (which commonly consists of past outputs and inputs) and the future output of the system. In other words, a set of observed regressors $\psi(t)$ for $t = 1 \ldots N$ and observed outputs $y(t)$ related to that regression vector are given. All these methods attempt to find a functional relationship

$$y(t) = G[\psi(t)] \tag{1}$$

which minimizes some error function E which is often of the form

$$E = \sum_{i=1}^{N} \mathrm{Err}[y(i) - G[\psi(i)]] \tag{2}$$

where Err is typically some norm operator. Commonly, the regression vector for single-input/single-output systems consists of delayed versions of the input and output.

$$\psi_{l,m}(t) = [y(t-\tau), \ldots, y(t-l\tau), u(t-\tau), \ldots, u(t-m\tau)] \tag{3}$$

An overview of recent work in the field of nonlinear identification can be found in (Sjöberg *et al.*, 1995). While many works focus on ways of parameterizing and determining the optimal function G, there is relatively little work for nonlinear systems on determining the proper form of the regressor ψ. Specifically, in nearly all of these studies the number of delayed terms in the regression vector (l and m) is assumed to be known. The function G is then estimated using the observed regressors and outputs of the time-series. If the differences between the approximated function $G[\psi(t)]$ and the output $y(t)$ are large, there could be two sources of the error. The first is the estimated function G does not do a good job of approximating the relationship between the regressor and the output. The second possibility is the regressor $\psi(t)$ does not contain enough information to accurately predict the future output. When the prediction error is large, it is impossible to tell from the outputs of these algorithms whether the source of the error is from the functional approximation or a regression vector which does not contain enough terms.

In the identification of linear systems, determining the exact source of the error is not a problem. Due to the linearity of the system, choosing the regression vector is the only critical step since the functional relationship is defined by the linearity of the system. Since computation of the unknown parameters involves relatively little computation, determining the proper regression vector involves computing the optimal linear model for various regression vectors and using some function (such as AIC or F-Test) to determine when the error no longer significantly decreases with respect to the complexity resulting from additional terms (Söderström and Stoica, 1989). For nonlinear systems, this method is infeasible since nonlinear optimizations require large amounts of computational effort and it is difficult to guarantee that nonlinear models converge to an optimal solution.

For nonlinear systems, it would be preferable to break the identification process into two distinct parts. First, the proper regression vector should be determined. The proper regression vector should contain only those terms needed to accurately predict the output. Once the proper regression vector is determined, then the functional relationship between the regressor and the output should be estimated.

While there is extensive work in determining the proper regression vector for linear systems (AIC, F-test, etc), there is relatively little work in the field of nonlinear systems. A method based on geometric ideas for determining the proper dimension for the regression vector will be presented here. As will be shown in the following section, this method does not involve any specific model structure and for this reason this method can be used in conjunction with any nonlinear modeling scheme.

*Current mailing address: Institut für Automatik, ETH - Z/ETL, CH-8092 Zürich Switzerland

FNN: DETERMINING THE PROPER DIMENSION

The false nearest neighbors (FNN) algorithm was originally developed for determining the smallest dimension regression vector needed to recreate the dynamics of autonomous chaotic systems (Kennel *et al.*, 1992; Abarbanel *et al.*, 1993). The idea behind the FNN algorithm is geometric in nature. If there is enough information in the regression vector to predict the future output, then two regression vectors which are close in the regressor space will also have future outputs which are close. If there are not enough terms present in the regressor vector to recreate the dynamics of the system, then there will be some neighborhoods in the regressor space which have a wide range of associated future outputs. Trajectories which are close in the regression space with vastly different outputs can be thought of as *false neighbors*, since they are close in the regression space only because of projection onto a space with a dimension too small to represent the dynamics of the system. For noise-free data, the number of false neighbors will drop to zero when the dimension of the regression vector is large enough to allow accurate prediction of future outputs.
In order to determine whether neighbors are true or false, a test must be defined to determine whether the neighbors have future outputs which are "far apart". For this purpose, a ratio test determines whether the distance between future outputs is significantly larger than the distance between time-delay regression vectors which are close in the regressor space. If the distance between future outputs is "large" when divided by the distance between two points which are "nearest neighbors" in the regressor space, then the neighbors are considered to be false.
Here is the FNN algorithm for input/output systems.

1. Form the set of regressors

$$\psi_{l,m}(k) = [y(k-\tau),\ldots,y(k-l\tau), u(k-\tau),\ldots,u(k-m\tau)] \quad (4)$$

and related outputs $y(k)$ from time-series data.

2. Identify the closest point (in the Euclidean sense) to a given point in the regression space. That is, for a given regressor $\psi_{l,m}(k)$ find another regressor $\psi_{l,m}(j)$ in the data set such that the following distance d is minimized.

$$d = \|\psi_{l,m}(k) - \psi_{l,m}(j)\|_2 \quad (5)$$

It should be noted that times k and j themselves are not necessarily close to one another. In fact, if k and j are always close to one another the sampling time may be too small and there may be problems in accurately estimating the dimension of the regression vector (Fredkin and Rice, 1995).

3. Determine if the following expression is true or false

$$\frac{|\, y(k) - y(j)\,|}{\|\psi_{l,m}(k) - \psi_{l,m}(j)\|_2} \le R \quad (6)$$

where R is a previously chosen threshold value. If expression 6 is true, then the neighbor are recorded as *true* neighbors. If the expression is false, then the neighbors are *false* neighbors.

4. Continue the algorithm for all times k in the data set. Calculate the percentage of points in the data set which have false nearest neighbors.

5. Continue the algorithm for increasing l and m until the percentage of false nearest neighbors drops to zero (or some acceptably small number). If the percentage of false neighbors is large, then the regressor vector must be extended to include more delayed input and/or output terms.

THRESHOLD SELECTION AND DATA REQUIREMENTS

The one adjustable parameter in the above algorithms is the threshold R used in the ratio test (Equation 6). The proper choice of the threshold is important. If R is too small, then the percentage of false nearest neighbors may not drop to zero in the proper dimension. On the other hand, if R is too large it is possible that the FNN algorithm will predict that future outputs are predictable with a regressor dimension that is too small. In order to properly select the threshold R and understand the results of the FNN algorithm, analysis of what passing or failing this threshold test refers to in a geometric sense will be given.
For reasons of analysis, assume that the number of delayed terms (l and m of Equation 3) and the function (G of Equation 1) relating these delayed terms to the future output are known. For all regression vectors embedded in the proper dimension, if the effects of noise are excluded, two regression vectors which are close in the regression space and their future outputs are related in the following way

$$\frac{|\, y(k) - y(j)\,|}{\|\psi_{l,m}(k) - \psi_{l,m}(j)\|_2} \le \max_t \|DG(\psi_{l,m}(t))\|_2 \quad (7)$$

for all k and nearest neighbor j, where $DG(\psi)$ is the Jacobian of the function G at ψ. For the FNN algorithm to correctly find that there are no false nearest neighbors in the proper embedding dimension, the threshold R must be chosen such that it is larger than $\max_t \|DG(\psi_{l,m}(t))\|_2$.
Now suppose that the dimension of the regression vector is too small to accurately predict the output. Since the regression vector is too small for accurate prediction, there will be certain regions of the regression space where the output cannot be accurately predicted. This means that two points from the data set with the same location in the regression space can have future outputs which are very different. In the case of infinite data, where all the regressors in the data set have neighbors which are arbitrarily close, the threshold can be made arbitrarily large (Rhodes and Morari, 1996). For finite data, it has been observed that false neighbors tend to have outputs which are "very far" apart so a large threshold will still lead to false neighbors for regressors which don't contain enough delayed terms.
When analyzing a time-series with the FNN algorithm, the amount of data needed to carry out an accurate analysis also needs to be considered. For the ratio test in the algorithm to correctly interpret the idea of closeness in the output space, the nearest neighbors must be relatively close in the regression space. If distance between nearest neighbors is large, then the difference between their outputs can be very

large and the ratio test could still call the two data points true neighbors when the outputs are totally uncorrelated. For all neighbors to be close, the time-series data should "fill out" the regression space to be analyzed. To fill an ∞-norm unit ball of n-dimension with data such that all individual data points are within an δ-sized ball of one another, approximately δ^{-n} points are needed. As the number of dimensions is increased, the amount of data needed to cover the space increases exponentially. This is a well-known problem in nonlinear identification, and is known as the *curse of dimensionality* (Weigend and Gershenfeld, 1994). In order to accurately assess whether regression vectors of up to dimension n are able to accurately predict future outputs, a time-series of length approximately 10^n is needed.

Choosing a single threshold which will work well for all data sets is an impossible task. Fortunately, it has been observed that false neighbors tend to have future outputs which are very different when the embedding dimension is too small. Therefore, the threshold can be safely made fairly large and the algorithm will give the result that the embedding dimension is too small for accurate prediction. The authors have observed that a good choice of the threshold is in the range of 10-15. When the time-series is relatively short, the choice of threshold is more critical and it might be necessary to compare the results from a number of different thresholds to come to a reasonable conclusion on the regressor size.

NOISE CORRUPTION

The FNN algorithm is not robust to the presence of noise in the time-series. The problem can be illustrated in the following way. When noise is present in a time-series, two identical regression vectors which would have identical outputs if no noise were present in the time-series can have outputs which differ by some finite amount. Therefore even when the noise corrupted time-series is embedded in a regression space with the proper dimension, the original FNN ratio test can fail (Rhodes and Morari, 1996).

Assume (as before), that the proper embedding dimension and the function relating the delayed inputs and outputs to the future output are known. Also assume that the observed inputs ($u^{obs} = u + \delta^u$) and outputs ($y^{obs} = y + \delta^y$) contain magnitude bounded noise ($\mid \delta^u \mid \leq \delta^u_\infty$, $\mid \delta^y \mid \leq \delta^y_\infty$). The observed regression vectors will also contain noise $\psi^{obs}_{l,n}(t) = [y^{obs}(t-\tau), \ldots, u^{obs}(t-\tau), \ldots] = [y(t-\tau) + \delta^y_1, \ldots, u(t-\tau) + \delta^u_1, \ldots]$ By assuming all neighboring regression vectors are "near" in the regressor space, it is possible to find the following result which is in a form similar to the FNN threshold test.

$$\frac{\mid y^{obs}(k) - y^{obs}(j) \mid}{\|\psi^{obs}_{l,m}(k) - \psi^{obs}_{l,m}(j)\|_2} \leq \max_t \|DG(\psi_{l,m}(t))\|_2 + \frac{2\sqrt{l{\delta^y_\infty}^2 + m{\delta^u_\infty}^2}\max_t \|DG(\psi_{l,m}(t))\|_2 + 2\delta^y_\infty}{\|\psi^{obs}_{l,m}(k) - \psi^{obs}_{l,m}(j)\|_2} \quad (8)$$

There are two distinct terms on the right hand side. The first term is due to the gain of the noise-free system, and the second term is due to noise contamination. Notice that the second term is inversely proportional to the separation between the nearest neighbors. In fact, if the original FNN ratio test is used to analyze noise corrupted time-series then the algorithm will report false neighbors in the proper embedding dimension which result only from noise. Even worse, the problem of false nearest neighbors occurring due to noise is compounded when more data are used for analysis (see (Rhodes and Morari, 1996) for more details).

In order to solve this problem, a new threshold test can be used for time-series where significant noise corruption is expected. A logical form of this threshold test based on the previous analysis is

$$\frac{\mid y(k) - y(j) \mid}{\|\psi_{l,m}(k) - \psi_{l,m}(j)\|_2} \leq R + \frac{2R\sqrt{l{\epsilon^y}^2 + m{\epsilon^u}^2} + 2\epsilon^y}{\|\psi_{l,m}(k) - \psi_{l,m}(j)\|_2} \quad (9)$$

Notice that this threshold test has two distinct limits. When the data are "dense", the second term on the right hand side will dominate. In this limit, the new threshold test simply determines whether the two neighbors outputs are within some specified noise-related distance ($2R\sqrt{l{\epsilon^y}^2 + m{\epsilon^u}^2} + 2\epsilon^y$) of one another. In the limit where neighbors are relatively far apart (where problems associated with noise corruption are not expected), this threshold test reverts back to the original FNN threshold test.

While this new threshold test does a better job in dealing with noise corrupted data, there are more parameters to be determined. The main parameter R should be chosen as in the original FNN algorithm, and the terms ϵ^y and ϵ^u should be no larger than the magnitude of the expected noise in the output and input respectively. Keep in mind that this threshold test accounts for the worst-case, and is somewhat conservative. For this reason, determining the exact magnitude of the noise is not always necessary. In fact it has been observed that by simply including these extra terms in the threshold test, better results are obtained even when the values of ϵ^y and ϵ^u are smaller than the magnitude of the noise (Rhodes and Morari, 1996).

CASE STUDIES

ELECTRICAL LEG STIMULATION

The first example is the identification of the dynamics of a human leg stimulated by an electrical signal. Jan Schultheiss carried out research to help paraplegics walk with the aid of a controlled electrical signal at the Swiss Paraplegic Center at Balgrist Hospital in Zürich Switzerland (del Re *et al.*, 1994). The data are from an experiment which measures the effect of stimulating the quadriceps muscle with an electrical signal. The input is the pulse-width of an electrical stimulation signal to the muscle, and the output is the angle of leg extension measured at the knee. The final goal of the project is to design a controller for helping paraplegics to walk by utilizing controlled electronic stimulation.

The time-series data consists of 654 input/output samples where the pulse-width of the electronic input which stimulates the leg is scaled to lie between 0 and 1. The leg angle output is scaled such that the full leg extension (or full quadriceps contraction) is 1, the knee being at a resting position is 0, and a leg which swings past its normal resting position on relaxation of the quadriceps causes a negative output. The sampling time for the system is 100 ms.

The original FNN algorithm with a threshold $R = 10$ was applied to the entire time-series. Since the time-series is relatively short, problems associated with noise corruption are not expected. The results of the FNN algorithm for this data set are given in Table 1. By examining the results, it can be seen that the percentage of false nearest neighbors drops to a low

percentage of 1.1 when the regression function takes the form

$$y(t) = G[y(t-1), y(t-2), u(t-1), u(t-2)] \quad (10)$$

While it is difficult to determine the definitive "proper" form of the regression vector since there are so little data, it appears that utilizing the model above should lead to accurate prediction.

In order to verify the results of the FNN algorithm, the nonlinear functional fitting algorithm MARS (Friedman, 1991) was used to estimate the function G for the same number of delay terms analyzed in Table 1. The first 500 points of the time-series are used to build a model and the MSPE (mean squared prediction error) is determined by comparing the results of the MARS modeling to the last 154 points of the time-series. After a model was found for a given regression vector, two types of error analysis were performed. The first determines the one-step ahead prediction error, where the actual past outputs and inputs are used to predict the next output of the system. The results of the one-step ahead analysis are given in Table 2. The second analysis involves a simulated model which utilizes the inputs and initial conditions of the physical system. Obviously, the second method typically has a larger error and the results of this method are given in Table 3.

In examining these results, it appears that a large percentage of FNN corresponds to larger values of the MSPE for both one-step prediction and simulation. For the one-step ahead prediction, the MSPE should always decrease when more terms are used. However, the simulation error is not guaranteed to decrease since the error resulting from a single step will compound when previous predicted outputs are used to determine future outputs. For these results, it appears that the MSPE does not significantly decrease when including more than two delayed version of both the input and output.

Without the FNN method, it would be necessary to build many models with different numbers of delayed terms. After these models were built the results would have to be analyzed and compared to determine a suitable number of delayed terms. With the FNN method, a suitable number of delayed terms can be determined in a single step. By utilizing the FNN algorithm, time can be saved when performing nonlinear identification.

SIMULATED PULP DIGESTER

The second example consists of simulation data which come from a fundamental model of a pulp digester. The data were provided to us in a dimensionless form by Ferhan Kayihan and Marc Gelormino of Weyerhaeuser Corporate Research and Development in order to test our identification methods. The data consist of 2 inputs and 2 outputs. For the generated time-series, the manipulated inputs changed randomly at the discrete sampling times of the system. The sampling time was chosen by the researchers at Weyerhaeuser to correspond roughly to the dominant time constant of the system. The time-series consists of 417 observed sampling periods.

While the FNN algorithm and analysis presented previously only consider single input/single output systems (SISO), it is straight-forward to extend the FNN algorithm to account for systems with multiple inputs and a single output (MISO). For a system with two inputs the regression vector must be extended, such that both inputs are included. The new prediction equation takes the following form

$$\begin{aligned} y(t) \;=\; & G[y(t-\tau), \ldots, y(t-l\tau), \\ & u_1(t-\tau), \ldots, u_1(t-m_1\tau), \\ & u_2(t-\tau), \ldots, u_2(t-m_2\tau)] \end{aligned} \quad (11)$$

where l, m_1, and m_2 are the number of delays which need to be determined by the FNN algorithm. By simply extending the regressor vector ψ of the original FNN algorithm to include the additional input term, the FNN algorithm can be used to search for the smallest number of terms needed to recreate the dynamics of the system. For systems with multiple outputs, each individual output should be analyzed separately by the FNN algorithm. It is possible that different outputs of the same system may need a different numbers of delayed terms to represent the dynamics.

For the first output, it was determined (using the average mutual information (Abarbanel *et al.*, 1993)) that there is a time delay of approximately three sampling times before either of the two inputs affect the first output. After accounting for this fact, the FNN algorithm is used to analyze the data from the system. A smaller threshold of $R = 5$ was used for this study because a larger number of terms in the regressor will be needed as a result of the two inputs. From the FNN algorithm, it appears that a representation in the form of

$$y_1(t) = G[y_1(t-1), u_1(t-3), u_1(t-4), u_2(t-3)] \quad (12)$$

should be able to recreate the observed dynamics.

To verify the results of the FNN algorithm a MARS model was built using the first 372 points in the time series. The results of the modeling were then verified using the last 43 points of the time series data. The results of the simulation from initial condition can be found in Figure 1. The model appear to be quite accurate.

For the second output, there is no significant time delay between the input and output and from additional FNN analysis it appears a model of the form

$$y_2(t) = G[y_2(t-1), u_1(t-1), u_1(t-2), u_2(t-1)] \quad (13)$$

can be used to model the dynamics of the system. The results of MARS modeling for a simulation from initial conditions for output 2 can be found in Figure 2.

INFERENTIAL MEASUREMENT SELECTION

In many processes in the chemical industry, it is infeasible (for economic or physical reasons) to physically measure the output to be controlled. However in some of these cases, this output may be able to be determined or "inferred" from other outputs of the system. One example is composition control of a distillation column. Composition analysis introduces a significant measurement delay into the system and the equipment is expensive and difficult to maintain (Mejdell and Skogestad, 1991). Since temperature measurements can be made easily, the output compositions can be estimated using a number of temperature measurements along the column.

Since the temperature can be measured anywhere along the length of the column, the question of where

the temperature should be measured for accurate estimation of the composition remains. For linear systems, this problem has been examined using a number of different methods which are all based on utilization of a linear estimator. To the authors' knowledge, the problem of determining suitable measurement locations for nonlinear inferential prediction has not been studied.

The inferential measurement selection problem can be stated as follows. Say we wish to predict the primary output y^p as a function of a number of secondary outputs $y^s_1, \ldots, y^s_n$. The goal is to find the smallest set of secondary outputs which accurately predict the primary output. In mathematical terms, the smallest number l and the proper set of secondary measurements $i_1, \ldots, i_l$ which allow for accurate estimation of the primary variable should be found for the equation

$$y^p(t) = F[y^s_{i_1}(t), y^s_{i_2}(t), \ldots, y^s_{i_l}(t)] \qquad (14)$$

To solve this problem, the FNN algorithm can be modified in a minor way.

The FNN algorithm for inferential measurement selection is similar to the original algorithm. However, in this case the regression vector consists of secondary outputs rather than delayed versions of the input and output. In addition, there is no easy way to determine a method for adding terms to the regressor as in the input/output case. In fact, to be sure the optimal measurement set is chosen all combinations of measurements in each dimension must be evaluated. Therefore we substitute the regressor vector given here

$$\Omega_{i_1,\ldots,i_l}(k) = [y^s_{i_1}(k), \ldots, y^s_{i_l}(k)] \qquad (15)$$

in the place of the regressor vector which consists of delayed versions of the input and output $\psi_{l,m}(k)$ in the original FNN algorithm. The only other change which needs to be made is that the percentage of false neighbors needs to be computed for all combinations of secondary measurements in the dimension l. If the percentage of false nearest neighbors is large for all combinations of measurements in the given regression dimension l, the number of secondary measurements in the regressor vector should be increased.

CONCLUSIONS

The FNN algorithm is a useful tool for determining the proper embedding dimension for both input/output dynamics and inferential prediction. The examples presented here illustrate the success of the FNN in determining the proper regression vector for accurate prediction. While the FNN algorithm is successful in determining the embedding dimension, it does not give any clues about the proper functional relationship between the regression vector and the output. While this might appear to be a drawback, the main advantage of FNN is that the process of determining the proper regressor is not dependent on a single model structure. This is especially important in nonlinear systems since a single model structure may not do a good job representing all possible relationships between regressors and outputs. In addition, since FNN is not dependent on any single model structure it can work in conjunction with any nonlinear functional modeling scheme.

By determining the smallest regression vector dimension which allows accurate prediction of the output, the FNN algorithm should reduce the computational effort needed to perform nonlinear identification. Instead of repeating computations for a large number of identification experiments with different sized regression vectors, FNN can determine the proper embedding dimension using relatively little computation time. After the proper dimension is determined, only a single nonlinear function need to be estimated. For these reasons, an FNN assisted identification scheme should save time when solving difficult nonlinear identification problems since the proper number of delayed terms can be determined without fitting specific models to the system.

NOTATION

y - System output
u - System input
ψ - Regressor of time-delayed terms
F, G - Functions relating regressors to outputs
R - FNN ratio threshold
l - Number of output terms
m - Number of input terms
δ - Magnitude bounded noise
ϵ - FNN noise threshold
y^p - Primary output
y^s - Secondary output
Ω - Regressor of secondary variables

ACKNOWLEDGMENTS

Partial support from the Department of Energy is gratefully acknowledged. The authors would also like to thank Jan Schultheiss for providing the leg stimulation data and Ferhan Kayihan and Marc Gelormino of Weyerhaeuser for providing the data from their fundamental pulp digester model simulation.

REFERENCES

Abarbanel, H. D. I., R. Brown, J. J. Sidorowich and L. S. Tsimring (1993). *Rev. Mod. Phys.* **65**, 1331–1392.

del Re, L., F. Kraus, J. Schultheiss and H. Gerber (1994). In: *American Control Conf.*. Vol. 2. pp. 2015–2019.

Fredkin, D. R. and J. A. Rice (1995). *Phys. Rev. E* **51**, 2950–2954.

Friedman, J. H. (1991). *The Annals of Statistics* **19**(1), 1–141.

Kennel, M. B., R. Brown and H. D. I. Abarbanel (1992). *Phys. Rev. A.* **45**, 3403–3411.

Mejdell, T. and S. Skogestad (1991). *Ind. Eng. Chem. Res.* **30**, 2543–2555.

Rhodes, C. and M. Morari (1996). IfA - ETH Technical Report AUT 96-03.

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson and A. Juditsky (1995). *Automatica* **31**(12), 1691–1724.

Söderström, T. and P. Stoica (1989). *System Identification.* Prentice Hall.

Weigend, A. S. and Gershenfeld, N. A., Eds.) (1994). *Time Series Prediction: Forcasting the Future and Understanding the Past.* Addison-Wesley Publishing Company.

% FNN Inputs m	Outputs l 0	1	2	3
0	100.0	72.2	13.5	4.2
1	87.4	36.5	5.5	2.6
2	82.2	27.8	1.1	0.3
3	74.5	23.2	0.5	0.3

Table 1: FNN results for leg stimulation data

MSPE Inputs m	Outputs l 0	1	2	3
0	4.7×10^{-2}	5.3×10^{-3}	4.1×10^{-4}	1.9×10^{-4}
1	7.7×10^{-2}	4.1×10^{-3}	2.7×10^{-4}	1.4×10^{-4}
2	2.2×10^{-2}	4.7×10^{-3}	1.9×10^{-4}	4.8×10^{-5}
3	1.8×10^{-2}	4.0×10^{-3}	1.6×10^{-4}	6.5×10^{-5}

Table 2: MSPE error for leg stimulation one-step ahead prediction

MSPE Inputs m	Outputs l 0	1	2	3
0	4.7×10^{-2}	4.6×10^{-2}	4.5×10^{-2}	4.3×10^{-2}
1	3.9×10^{-2}	1.4×10^{-2}	5.5×10^{-3}	8.1×10^{-3}
2	2.8×10^{-2}	2.1×10^{-2}	7.5×10^{-3}	2.9×10^{-3}
3	2.8×10^{-2}	2.3×10^{-2}	1.9×10^{-3}	2.9×10^{-3}

Table 3: MSPE error for leg stimulation simulation

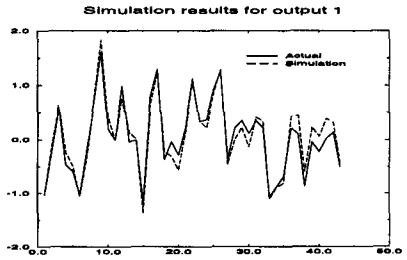

Figure 1: Results of MARS simulation for pulp digester output 1

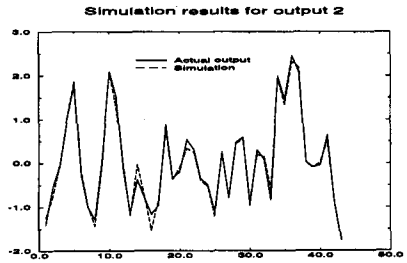

Figure 2: Results of MARS simulation for pulp digester output 2