Integration of Connectionist Methods and Chaotic Time-Series Analysis for the Prediction of Process Data

R. Kozma,* N. K. Kasabov, and J. S. Kim
Department of Information Science, University of Otago, Dunedin, New Zealand

A. Cohen
Waste Solutions Ltd., Mosgiel, New Zealand

A connectionist-based time-series analysis method is described that includes chaotic characterization, fractal analysis together with statistical data processing in an adaptive fuzzy neural network environment. The applied fuzzy neural network (FuNN) can utilize as well as generate knowledge during an iterative learning and adaptation procedure. Two major aspects of the present work are (1) incorporating knowledge into the fuzzy neural network based on the nonlinear deterministic, chaotic analysis of the signals and (2) refining and updating the knowledge base by the FuNN using adaptive learning techniques. Examples include the standard gas-furnace benchmark data analysis and also an application to a case study of multivariate signal analysis as part of a project for establishing a plantwise monitoring and process control system. © 1998 John Wiley & Sons, Inc.

1. INTRODUCTION

Hybrid systems' development is one of the most intensively evolving area of connectionist science. Hybrid systems utilize various soft computing methods like artificial neural networks, fuzzy engines, evolutionary computation, and chaos. It is a challenging task to implement hybrid techniques for time-series analysis and prediction, which is the topic of the present paper. Modeling and predicting time series are difficult problems that often require advanced tools of dynamic characterization of the system under investigation.[1–3] Methods of chaos analysis are natural candidates in this field as they can grasp the multilevel, nonlinear dynamics of natural processes. Embedding chaos into neuro-fuzzy models enhances the advantages of the latter as nonlinear approximators; see, e.g., Refs. 4–6. In the present paper the methodological founda-

*Author to whom correspondence should be addressed. E-mail: rkozma@commerce.otago.ac.nz.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 13, 519–538 (1998)
 CCC 0884-8173/98/060519-20

tions of a connectionist-based time-series analysis method are introduced, including fractal analysis of the time signals in conjunction with a fuzzy neural network architecture.[5,7]

Fractal analysis has been applied to a variety of research fields to characterize highly irregular and nonstationary data. The fundamental concept of fractal theory was first proposed by Mandelbrot[8] to describe the complex shapes that appear in nature. When a time series is fractal, its self-similarity can be described by the fractal dimension value D. The evaluation of the fractal dimension usually requires a complex rectification procedure. In this paper a relatively simple algorithm is used following Higuchi.[9,10] Higuchi's method is based on the calculation of the fractional curve length in the case of self-affine geometry of graphs of time series.

Fractal analysis can supplement other signal processing methods aiming at reliable system characterization under changing external conditions.[9,11] It will be shown in this paper that results of statistical and chaotic time-series analyses can be incorporated into a connectionist-based, fuzzy neural network model aiming at time-series modeling and systems control. The results are illustrated on the example of gas-furnace benchmark test data and using actual process data measured at a wastewater treatment plant.

2. TIME-SERIES ANALYSIS BASED ON HIERARCHICAL MODELING BY FUZZY NEURAL NETWORKS

2.1. Strategy of a Recursive Connectionist Signal Characterization Method

The proposed time-series analysis method consists of three major steps:

(1) Characterization of the signal with various statistical and dynamical evaluation methods.
(2) Developing a model for the description of past and present behavior and, within a certain time frame, for predicting future states.
(3) Iterative updating/refinement of the characterization conducted in step 1 using model calculations.

The goal of the first step is to reveal some major features of the investigated signal, like stationarity/nonstationarity, the presence/absence of periodic components, and revealing whether the signal is random or contains nonlinear deterministic, chaotic components. Results of the characterization step have important implications on the choice of modeling methods in the further analysis.[11] For example, if the signal is stationary with possible periodic components, conventional methods of correlation analysis in time and frequency domains are usually sufficient. For nonstationary signals with nonlinear dynamics, however, methods of chaos analysis can be more appropriate.

Once the main properties of the signal have been identified, several important quantitative features are determined, like the dominant time scales in

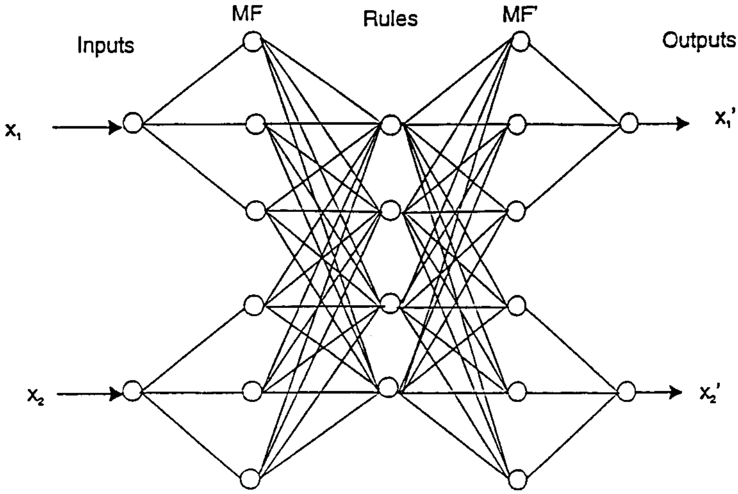

Figure 1. General FuNN architecture.

the signal, possible occurrence of self-affinity and fractal/multifractal features, etc. All these descriptors contain important information for the design of the signal model.

In the present work we concentrate on signals with nonlinear dynamics modeled by fuzzy neural networks (FuNNs).[7] FuNNs use a multilayer perceptron (MLP) network and a modified backpropagation training algorithm. The general FuNN architecture (see Fig. 1) consists of five layers of neurons with partial feedforward connections. It allows for the combination of both data and rules into one system, thus producing the synergistic benefits associated with the two sources. In addition, it allows for adaptive learning in a dynamically changing environment.

The properly trained and tuned FuNN is used for the extraction of knowledge on the system behavior. The extracted knowledge can be used for updating the identified characteristics of the signal, e.g., dominant time scales, extent of temporal memory, etc. In this way a closed feedback loop is generated which can be used for iteratively refining the description and also for adapting the inferred model to changing environment. In the following section, details of the applied dynamic characterization are given.

2.2. Fractal Analysis of Time Series

A significant part of the time-series analysis methods is based on the assumption that the investigated time series can be described as stochastic processes with Gaussian probability density functions. Moreover, by calculating the power spectral density functions by the commonly used FFT algorithm, it is assumed that the time series is stationary. These assumptions are quite restrictive and they are rarely met in real life. Methods of fractal analysis do not make such assumptions and they can provide a solution to the arising problems.

In the present work the system is investigated (1) by calculating the fractal dimension of the temporal signals and also (2) by determining the effective memory of the system using the structure of the FuNN trained by a structural learning algorithm. Here the major steps of the analysis are outlined.

The fractal dimension of self-affine time series is estimated following the procedure.[9] Starting from a time series $X(j)$ ($j = 1, 2, \ldots, N$), the partial length of the graph of the time series is defined as

$$L_m(k) = \left\{ \left(\sum_{i=1}^{[(N-m)/k]} |X(m+ik) - X(m+(i-1)k)| \right) \frac{N-1}{[(N-m)/k] \cdot k} \right\} \Big/ k, \tag{1}$$

where k denotes the time interval and $m = 1, 2, \ldots, k$. Notation $[a]$ stands for the integer part of a. N is the number of data points used for calculating a set of $L(k)$ values. A new set of $L(k)$ is calculated sequentially at each time instant when new data have been acquired. This is followed by the estimation of the fractal dimension based on the $L(k)$ values.

For each time interval k, the average value of the length is given as

$$\langle L(k) \rangle = \frac{1}{k} \sum_{m=1}^{k} L_m(k). \tag{2}$$

By combining Eqs. 1 and 2, we obtain

$$\langle L(k) \rangle = \frac{1}{k} \sum_{m=1}^{k} \left\{ \frac{N-1}{k^2} \left(\frac{1}{[(N-m)/k]} \sum_{i=1}^{[(N-m)/k]} |X(m+ik) - X(m+(i-1)k)| \right) \right\}. \tag{3}$$

Here $\langle L(k) \rangle$ corresponds to the mean length of the curve when the wave number k is fixed. It is easy to see that Eq. 3 indeed gives a normalized curve length by summing the absolute values of the step-by-step changes along the graph of the time series.

If the relationship $L(k) \propto k^{-D}$ holds, the time series reveals fractal characteristics. In the case of a fractal time series, the log-log plot of the k versus $L(k)$ curve shows a linear behavior, and the negative slope of the curve gives the fractal dimension value D. In order to estimate the fractal dimension of time series based on the slope, a least squares method has been used. If the signal is fractal, the following relationship holds between k and $L(k)$,

$$\log(L(k)) = x_1 + x_2 \log k, \tag{4}$$

where x_1 and x_2 are parameters estimated by the least squares method. Finally, the fractal dimension D is given simply as

$$D = -x_2. \tag{5}$$

2.3. Relationship Between Fractal and Spectral Indices

In a lot of practically important cases, the power spectral density has a power-law behavior, i.e., low frequencies have higher energies compared to high frequencies following the unique relationship

$$P(f) \propto f^{-\alpha}. \tag{6}$$

Here $P(f)$ is the power spectral density, f is the frequency, and α is a power-law index. A curve with a single power-law index for all frequencies is self-similar. The following approximate relationship has been obtained[10] between power-law index α and the fractal dimension D: over the range of $1 \leq D \leq 2$,

$$D = (5 - \alpha)/2. \tag{7}$$

Note that the above relation is just an approximation. It holds for a class of processes called fractional Brownian motion with $1 < \alpha < 3$, but it breaks down for $\alpha < 1$ and $\alpha > 3$. In many practical situations, the signal often reveals bi- or multifractal features; i.e., the slope of the k versus $L(k)$ curve depends on the range of wave numbers k. In a multifractal signal, the estimation of a set of fractal dimension values is performed by dividing the whole k range into several segments and applying linear fitting over each subregion. The proper choice of subregions in the wave number domain is rather complicated. Intelligent data processing methods based on artificial neural networks can be applied to this aim.[12] Results obtained by statistical time-series analysis and fractal analysis are used in the design of the FuNN architecture as described in the next section.

3. THE ARCHITECTURE OF FuNNs

The FuNN model is designed to be used in a distributed environment. The architecture facilitates learning from data and approximate reasoning, as well as fuzzy rules extraction and insertion. The FuNN is an adaptable fuzzy neural network where the membership functions of the fuzzy predicates, as well as the fuzzy rules inserted before training or adaptation, may adapt and change according to new data. Below a brief description of the components of the FuNN architecture and the philosophy behind this architecture are given. For further details on FuNNs, see Refs. 5 and 7.

3.1. Input Layers

The input layer of neurons represents the input variables as crisp values. These values are fed to the condition element layer, which performs fuzzification. This is implemented using three-point triangular membership functions with centers represented as the weights into this condition element layer. The triangles are completed with the minimum and maximum points attached to adjacent centers or shouldered in the case of the first and last membership functions. The triangular membership functions are allowed to be nonsymmetrical, and any input value will belong to a maximum of two membership functions with degrees differing from zero. These membership degrees for any given input will always sum to 1, ensuring that there will be rules to fire for any point in the input space.

The applied triangular membership functions make the fuzzification and the defuzzification procedures fast without compromising the accuracy of the solution. Under the FuNN architecture, labels can be attached to weights when the network is constructed. When adaptation is taking place, the centers are spatially constrained according to some rules, e.g., the membership function weight representing "low" will always have a center less than "medium," which will always be less than "high," etc.

3.2. Rule Layer

In the rule layer each node represents a single fuzzy rule. The layer is potentially expandable, in that nodes can be added to represent more rules as the network adapts, or potentially shrinkable. The activation function is the sigmoidal logistic function. The semantic meaning of the activation of a node is that it represents the degree to which input data match the antecedent component of an associated fuzzy rule. However, the synergistic nature of rules in a fuzzy-neural architecture must be remembered when interpreting such rules. The connection weights from the condition element layer to the rule layer represent semantically the degrees of importance of the corresponding condition elements for the activation of this node. The values of the connection weights to and from the rule layer can be limited during training to be within a certain interval, say $[-1, 1]$, thus introducing nonlinearity into the synaptic weights.

3.3. Output Layers

In the action element layer, a node represents a fuzzy label from the fuzzy quantization space of an output variable. The activation of the node represents the degree to which this membership function is supported by the current data used for recall. The activation function for the nodes of this layer is the sigmoidal logistic function with a variable gain factor as in the previous layer. This gain factor should be adjusted appropriately given the size of the weight boundary. The output layer performs a modified center of gravity defuzzification. Singletons representing centers of gravity of membership functions are attached to the connections from the action to the output layer. Linear activation functions are used here. The shape of the output membership functions is not restricted and may be changed during training and adaptation.

3.4. Functionality and Features of FuNNs

One of the advantages of the FuNN architecture is that it manages to provide a fuzzy logic system without having to unnecessarily extend the traditional MLP. Since standard transfer functions—linear and sigmoidal—are used along with a slightly modified backpropagation algorithm, the main departure being the constraining rules, much of the theory regarding such networks is still applicable. There are four versions of weight updating in the FuNN according to the mode of training and adaptation. [5,7,13,14] These are not mutually exclusive versions, but are all provided within the same environment and the versions can be switched as needed. The methods of adaptation are as follows:

(1) A partially adaptive training, where the membership functions (MF) of the input and the output variables do not change during training and a modified backpropagation algorithm is used for the purpose of rule adaptation. This adaptation mode can be suitable for systems where the membership functions to be used are known in advance or where the implementation is constrained by the problem in some way.
(2) A fully adaptive training with an extended backpropagation algorithm.[14] This version allows changes to be made to both rules and membership functions, subject to constraints necessary for retaining semantic meaning.
(3) A partially adaptive version as in method 1, but a forgetting factor is introduced.
(4) A partially adaptive version with the use of a genetic algorithm for adapting the membership functions. This mode does not alter the rules. The algorithm is described in Ref. 14.
(5) Adaptive training with the use of the "Method of training and zeroing".[7] This method employs a standard backpropagation algorithm, but small connection weights (below a certain threshold) are "zeroed" regularly using a variable threshold. These connections can be left in the structure for further change or can be pruned.

These modes can either be used as alternatives or they can be combined as is most appropriate for the given problem. It may be useful to use several different modes in an alternative manner, with each version of the adaptation algorithm best suited to some part of the adaptation task. A MS Windows version of FuNN, which is part of an integrated hybrid development tool[5] called FuzzyCOPE/2, is made available free from the WWW site: http://divcom.otago.ac.nz:800/COM/INFOSCI/KEL/fuzzycop.htm.

FuNN allows for different training and adaptation strategies to be tested before the most suitable is selected for a certain application. It has other interesting features in addition to the features discussed above. Some of them are (a) a FuNN has a symmetrical structure and it can be used to build a replicator NN—a FuNN can be trained with output data being the input ones and a compressed encoding can be picked up from the rule layer— and (b) a FuNN structure may be used to represent all stages in a cognitive task, i.e., perception, data filtering, knowledge acquisition, and knowledge modification, action.

4. INTEGRATING FuNN AND CHAOTIC TIME-SERIES ANALYSIS

A neural network can eventually be trained to approximate a chaotic function, but can be connectionist structure be trained with chaotic data in such a way that it captures structurally the main characteristics of the chaotic process? This problem is solved here by applying a method of structural learning with forgetting to a FuNN. After training with forgetting, the FuNN structurally represents the underlying rules of the chaotic process.

In the theoretical description of dynamic systems, usually a set of nonlinear differential equations is analyzed. By formulating these differential equations, certain assumptions are introduced on the system behavior. These assumptions are inherent in the dynamic system models. A neural network, as a distributed parameter system, is free of such assumptions. Therefore, it can be regarded also as an alternative description of nonlinear dynamics. Combining fuzzy systems with neural networks, the description of the nonlinear system can be expressed in the form of high-level, symbolic rules. Depending on the nature of the analyzed problem, these rules can approximate closely the differential equations or they can exhibit a more general behavior. This point will be illustrated in the example of estimation of the memory of the system using structural learning in FuNN.

4.1. Structural Learning Methods

In the past few years, significant efforts have been devoted to elaborate algorithms that find optimal neural network architectures. Two major approaches can be distinguished: either growing an increasingly elaborate network starting from a simple architecture or reducing the size and complexity of an initially very complex neural network.[15–17] The latter approach is called network pruning. In the present study, a special type of network pruning is applied, which is a modified backpropagation learning algorithm with forgetting the connection weights. Modified backpropagation with forgetting belongs to the class of structural learning algorithms. By applying learning with forgetting, the weights decrease continuously unless they are reinforced by the backpropagation rule. At the end of the training, only the essential weights deviate significantly from zero. By pruning the weights that are close to zero, a skeleton network is obtained.

Based on the skeleton structure, knowledge can be obtained about the analyzed patterns. A neural network that is able to create and process abstract knowledge is called an artificially intelligent network, as opposed to computational nets, which simply process numerical data. The structural learning method applied in this work has the potential to generate artificially intelligent neural networks in the above sense. NNs trained by the forgetting algorithm are expected to have better generalization properties than those trained by standard backpropagation, because the skeleton structure obtained after training with forgetting is usually more suitable for the given problem than a predefined architecture used in standard backpropagation. Moreover, by decreasing the effective number of weights during training with forgetting, overtraining can be avoided. A disadvantage of backpropagation with forgetting is the increased computational time. This problem, however, can be compensated for by removing the unnecessary connections.

4.2. Implementation of Structural Learning in the FuNN Environment

In this section, the main features of the backpropagation learning algorithm with forgetting are summarized. The basic idea is to update the connection weights as follows.[16]

$$\Delta w_{ij} = \Delta w'_{ij} - \gamma \operatorname{sgn}(w_{ij}) \tag{8}$$

Here $\Delta w'_{ij}$ is the change of the ijth weight using the standard backpropagation algorithm, γ is the so-called forgetting rate, and $\operatorname{sgn}(x)$ denotes the signum function. The second term on the right-hand side of Eq. 8 describes a decreasing tendency for the connection weights. Indeed, the weights decrease continuously unless they are reinforced by the backpropagation rule. The corresponding cost function is given by

$$J = \sum_i (y_i - y_i^*)^2 + \gamma' \sum_{i,j} |w_{i,j}| \tag{9}$$

Here y_i and y_i^* are the actual and the target values of the network outputs, respectively. $\gamma' = \lambda\gamma$, where λ is the learning rate and γ is the forgetting rate. The first term is the usual sum of squared errors (SSE) between the actual and target values of the output of the NN. The second term is the sum of the absolute values of weights (SW) with an appropriate proportionality constant.

Structural learning with forgetting can be viewed as a pruning algorithm in which a significantly reduced network is obtained after training. The second term in Eq. 9 represents the sum-of-weights penalty condition. A large variety of penalty terms are applied in the literature, including exponential weight decay, optimal brain damage, enthalpy pruning, and lateral inhibition.[15] The results of comparative studies show that structural learning with forgetting scores very well compared with other methods as far as the discovery of regularities and generalization capabilities are concerned.[66]

At the beginning of the training the quadratic error function plays a dominant role in the cost function in Eq. 9. SSE drops quickly while the sum of the weights changes only slightly. As the learning advances, SSE changes less intensively, because cost function J can be reduced via decreasing the absolute values of the weights as well. The decreased convergence is a disadvantage of backpropagation with forgetting. Nevertheless, this shortcoming can be compensated for by reducing the system size as the forgetting progresses and the number of active nodes decreases. Exactly this is done during zeroing in the FuNN system.

The proper choice of the forgetting rate is crucial to the success of the modified backpropagation with forgetting. If the forgetting rate is large, the weights decrease quickly and the network structure becomes rigid. In this case, the convergence of the training can be very poor, which causes unsatisfactory testing performance. If the forgetting rate is too small, the effect of forgetting appears slowly, and very long training is needed to achieve proper performance of the network. If the forgetting rate is properly selected, a better generalization can be achieved by the forgetting algorithm than by the standard backpropagation method and overtraining can be avoided. Various strategies exist for optimization of the forgetting.[16–18] In this work such a pruning level is selected that allows knowledge extraction from the FuNN without deteriorating the convergence of the training algorithm.

5. CASE STUDY OF GAS-FURNACE DATA

5.1. The Gas-Furnace Time Series

In order to demonstrate the potential of the proposed FuNN to time-series processing, a well-examined data set has been used—the gas-furnace data from Box and Jenkins.[19] This data set provides a nice example of time-series prediction and, having been used in many data analysis and signal processing studies, it is well benchmarked, allowing for comparisons between the current technique and alternatives. The data set consists of 292 consecutive values of methane at a time moment $(t-4)$, and the carbon dioxide (CO_2) produced in a furnace at a time moment $(t-1)$ as input variables, with the produced CO_2 at the moment (t) as an output variable.

Figure 2 shows the CO_2 time-series data, together with their various characteristic functions. There is a positive correlation up to four to five time lags, indicating the finite memory effect. The probability density function of the CO_2 signal shows significant deviation from Gaussian behavior. As the time series is rather short (comprising just 292 data points for both variables), the power spectrum has large statistical fluctuations. Due to the shortness of the time series, no fractal analysis has been conducted. Nevertheless, the rules extracted from the structure of the trained FuNN will provide useful information about the dynamics of the data set.

5.2. Analysis of Gas-Furnace Time Series by FuNNs

The task is to use different options built in a FuNN structure (1) to find the effective memory of the gas-furnace data through the use of the structural learning-with-forgetting method; (2) to explore different adaptive training tech-

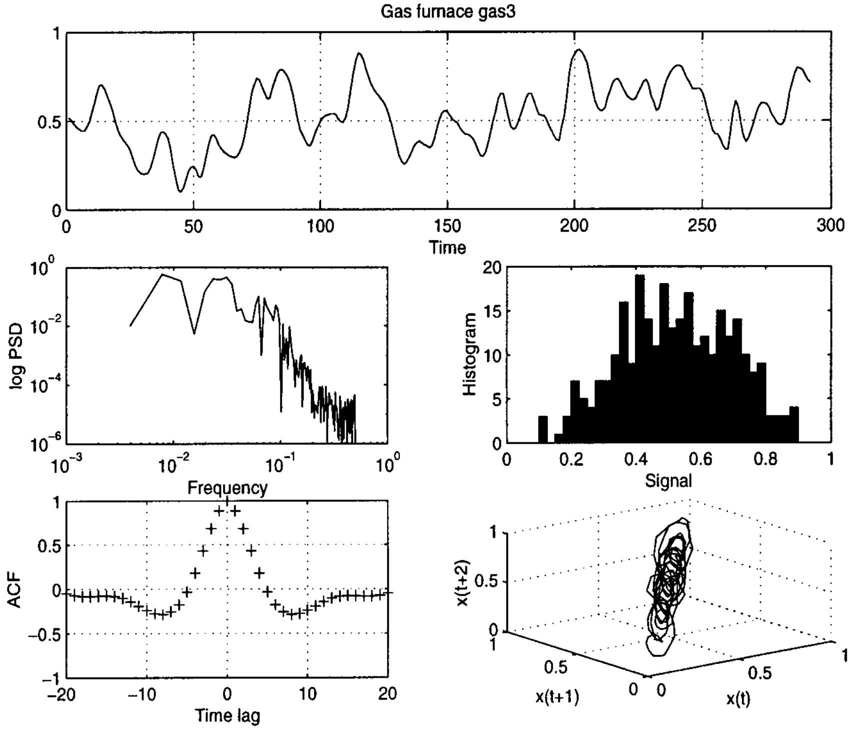

Figure 2. Characterization of CO_2 time series: (a) original time series; (b) power spectral density; (c) histogram; (d) autocorrelation function; (e) three-dimensional phase diagram.

niques in terms of learning error and generalization; (3) to extract rules that explain the behavior of the time series. For these purposes, the following experiment was performed: a 2-10-7-5-1 FuNN was trained for 1000 epochs with the modified backpropagation algorithm and fixed MFs. Further on, the training was extended up to 10,000 iterations with recording of the interim results of training at every 1000 iterations. Note that the applied structural learning assures that no overtraining occurs during the extended iterations. Forgetting was introduced with a forgetting factor of $\gamma = 10^{-4}$. The learning rate was variable during the training and individually set for each of the layers in the FuNN, while the momentum and the gain factor were 0.9 and 1, respectively, for layers 2–5. Figure 3(a) shows the actual and the estimated values for the CO_2. The one-step prediction error produced by the FuNN is given in Figure 3(b).

In order to investigate the memory effect in the methane and CO_2 data, further experiments were conducted with input nodes corresponding to the previous values of the signals up to nine time lags each. Three membership functions and 15 rules were used initially. Accordingly, the FuNN had a nodal

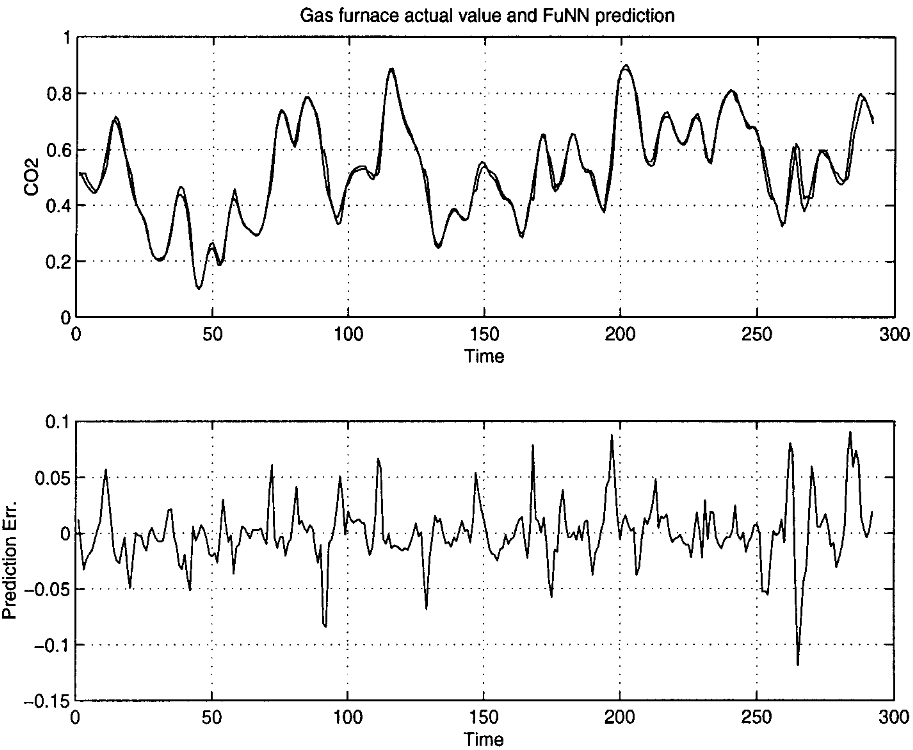

Figure 3. Results of FuNN modeling of CO_2 data: (a) actual and predicted CO_2 time series; (b) prediction error of FuNN.

structure of 18-54-15-3-1. A forgetting rate value of $\gamma = 10^{-3}$ resulted in significant pruning over the inputs and rule nodes as well. Only the nodes corresponding to the most recent time lags remain active for the methane, and the intensity of the connection weights exhibits a decay as time passes for the CO_2 as well. This is shown in Figure 4. It is also seen that the final number of rule nodes is four. The importance of the time lags is calculated as the average squared magnitude of the weights corresponding to the particular time lag. Data in Figure 5 clearly show that the extent of memory is about three and four time lags for CO_2 and methane signals.

6. CASE STUDY OF SEWAGE TREATMENT PLANT MONITORING

6.1. Design of Plant Control System

A project has been initiated to update the control system of Morrinsville Sewage Treatment Plant, New Zealand.[20] The plant is located in the heart of New Zealand. It has to operate as an industrial treatment plant and be able to

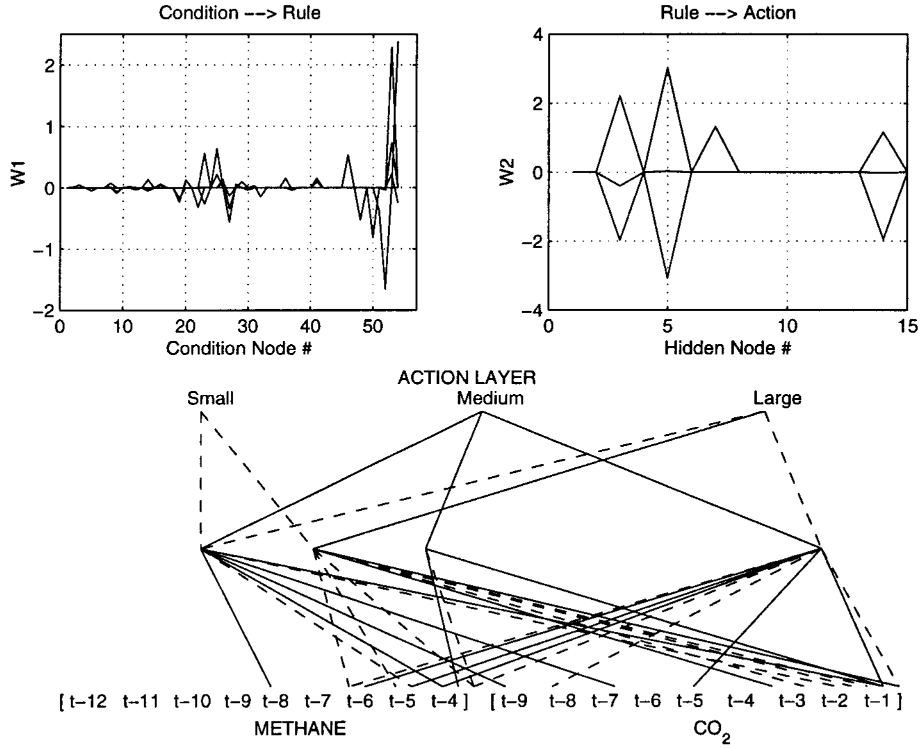

Figure 4. Memory effect in the gas-furnace data set: (a) distribution of weight matrix W1 (condition-to-rule layer) and (b) W2 (rule-to-action layer); (c) the skeleton of the hidden layers of the trained FuNN.

cope with variations of influent load and composition. Plant operating conditions vary greatly throughout the seasons, with winter months being characterized by low organic load, high storm water infiltration rates, and low-temperature operating conditions. During the summer, organic and nutrient loads are high, with low storm water infiltration and higher water temperatures. Peak storm water infiltration rates during wet weather conditions can be high because of the large catchment area. Peak organic load during the summer is up to twice the nominal load. In the framework of the project, an adaptive control system is designed in an attempt to cope with the variety of operational conditions. The schematic of the flow control system is shown in Figure 6.

The estimation of the outlet flow is based on process data measured on-line at various locations across the plant. There are 16 data channels available for further processing which include flow, temperature, turbidity, and pH. Data are extracted from the factory's Supervisory Control and Data Acquisition System at 1-min intervals and transmitted by FM radio link to the plant control system. The characteristics of the effluents reflect batch discharges such as cleaning-in-place cycles of spray dryers and other equipment. The resulting effluent charac-

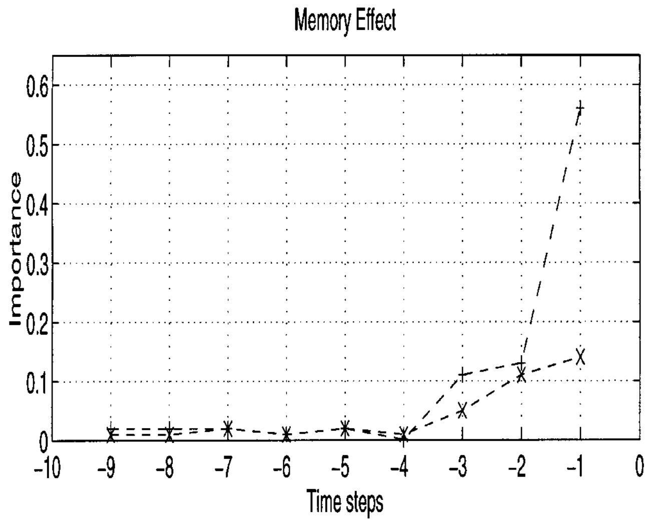

Figure 5. Importance of input nodes: + denotes CO_2 history; × denotes methane history.

teristics display a complex and rapidly changing pattern. Data recorded over a 3-week interval were available for the analysis.

6.2. Characterization and Prediction of the Flow Signal

The case study introduced in this paper concerns modeling and predicting outlet flow, which is a subtask of the comprehensive adaptive control system. Two signals have been included in our study: the outlet flow value and the temperature of the effluent. In Figure 7, characteristics of the outlet flow are illustrated using both statistical and chaotic features. The time-series plot

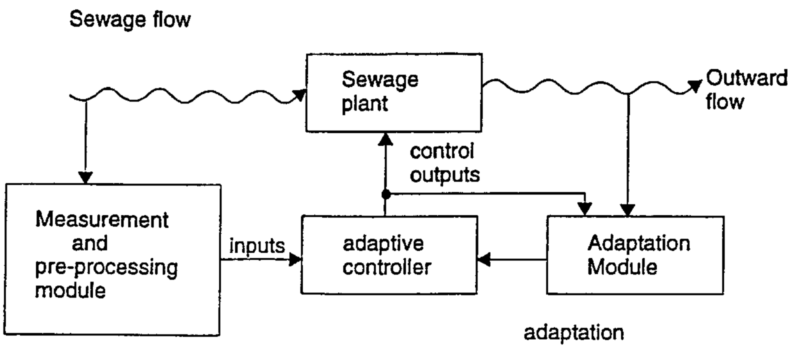

Figure 6. Schematic of adaptive control of the sewage treatment plant.

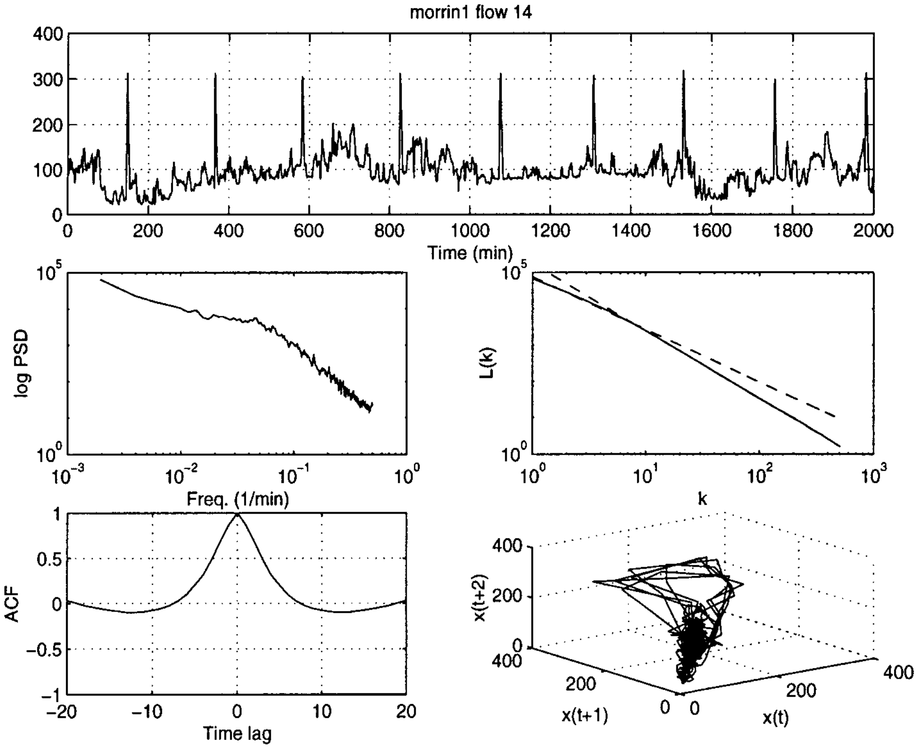

Figure 7. Characterization of flow outlet signal: (a) flow time series; (b) power spectral density; (c) autocorrelation; (d) fractional length of the time-series graph; (e) three-dimensional phase plot.

displays large fluctuations. The power spectrum has a bilinear shape in log-log coordinates, with a breaking point at frequency around 0.05 min^{-1}. The autocorrelation function diminishes beyond about six time lags.

The fractional length $L(k)$ versus wave number k shows a bifractal behavior. Least squares fits of the linear segments at low and high wave number values were conducted. The slope of the k versus $L(k)$ curve over k ranges [1−10] and [10−256] are denoted by D_{low} and D_{high}, respectively. The calculations give fractal dimension values $D_{\text{low}} = 1.85 \pm 0.02$ and $D_{\text{high}} = 1.41 \pm 0.05$. The phase plot of the flow signal indicates the presence of a three-dimensional loop. At higher k values ($k > 10$), the fractal curve is steeper, with D value close to 2, which is an indication of more random-like behavior. Similar conclusions can be drawn from the temperature signal in Figure 8, except for the absence of the low-dimensional loop in this case.

In modeling the flow signal by FuNN, six inputs were used which represent the flow and the temperature at times t, $t-1$, and $t-2$, respectively. We used 5 MFs and 20 rules and the nodal structure of the FuNN was 6-30-20-5-1. The

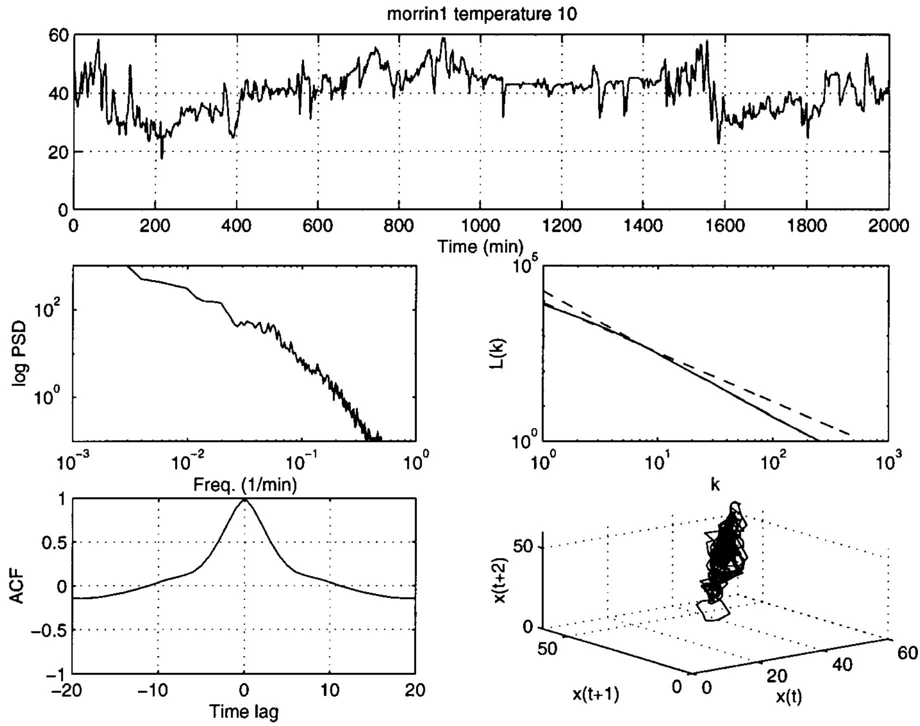

Figure 8. Characterization of temperature signal: (a) temperature time series: (b) power spectrum; (c) autocorrelation; (d) fractional length of the time-series graph; (e) three-dimensional phase plot.

output is the predicted flow value at time $t + 1$. The training of the FuNN was completed in a similar way as in the case of the gas-furnace data introduced previously in Section 5. Results of the prediction are shown in Figure 9. In Figure 9(a), the actual and predicted flow signals are given. Note that the two curves significantly overlap due to the accuracy of the FuNN model. Details of the prediction error are shown in Figure 9(b). It is seen that FuNN gives very good estimation with an absolute error of a few percent except for the "jumps" that occur at regular intervals. Detailed evaluation shows that the network gives a significant error at the first step of the "jump," but it quickly recovers and supplies accurate predictions in the consecutive time steps.

Figure 10(a) shows the hidden layers (layers 2–4) of the initial FuNN with a fully connected architecture. Layers 1 and 5 are not shown on Figure 10(a) as they correspond to the membership functions that are fixed (not tuned) in the present experiments. Rules extracted from the structure are illustrated graphically in Figure 10(b). It is seen that not only the number of rules decreased significantly (from 20 to 8), but also the number of inputs was reduced drastically. It is clear that the input temperature nodes have relatively weak connec-

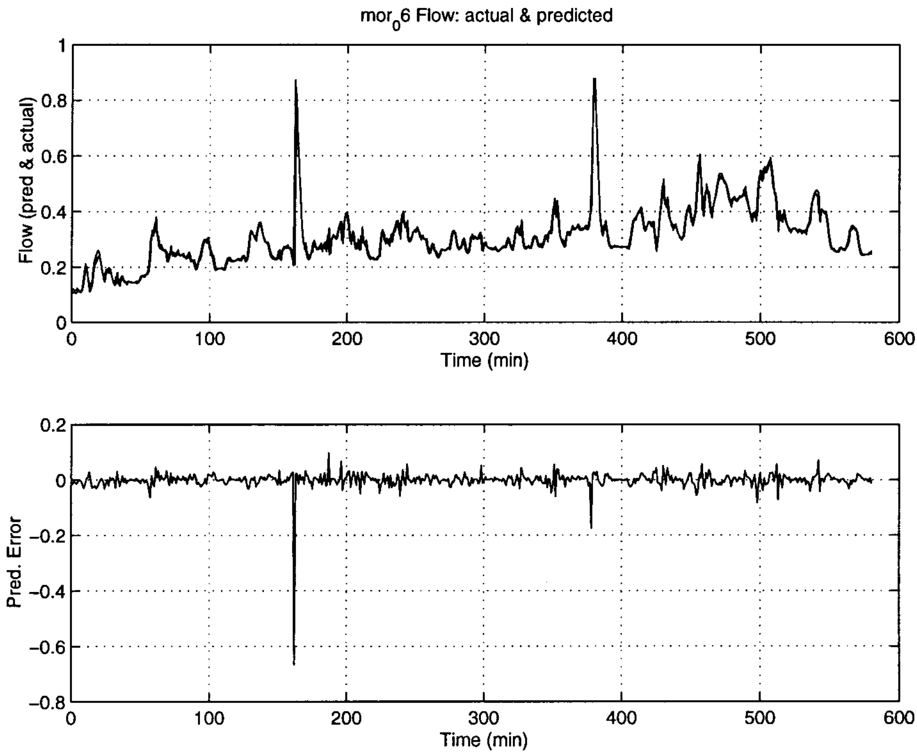

Figure 9. Prediction of flow by a 6-30-20-5-1 FuNN: (a) actual and predicted flow values; (b) error of the one-step prediction.

tions to the final rule nodes. Obviously, it is the flow signal, especially its most recent value at time t, that dominates the final node structure.

The effective memory of the flow signal is estimated in an experiment with 10 input nodes; see Figure 11. The inputs are the flow signal values at time instances $t, t-1, \ldots, t-9$. We used three MFs, which gives 30 nodes in the condition layer. Figure 12 clearly shows that the connections are the strongest within a finite time lag of about 5 min, where the importance of the weights is depicted as a function of the time lag. The importance fluctuates beyond the 5-min threshold as well, which is attributed to statistical effects.

7. CONCLUSION

In this work, a comprehensive signal processing method is outlined which consists of detailed characterization and consequent modeling/prediction of the signal. In the characterization phase, both statistical and chaotic methods are included. The modeling is based on a fuzzy neural network architecture. The

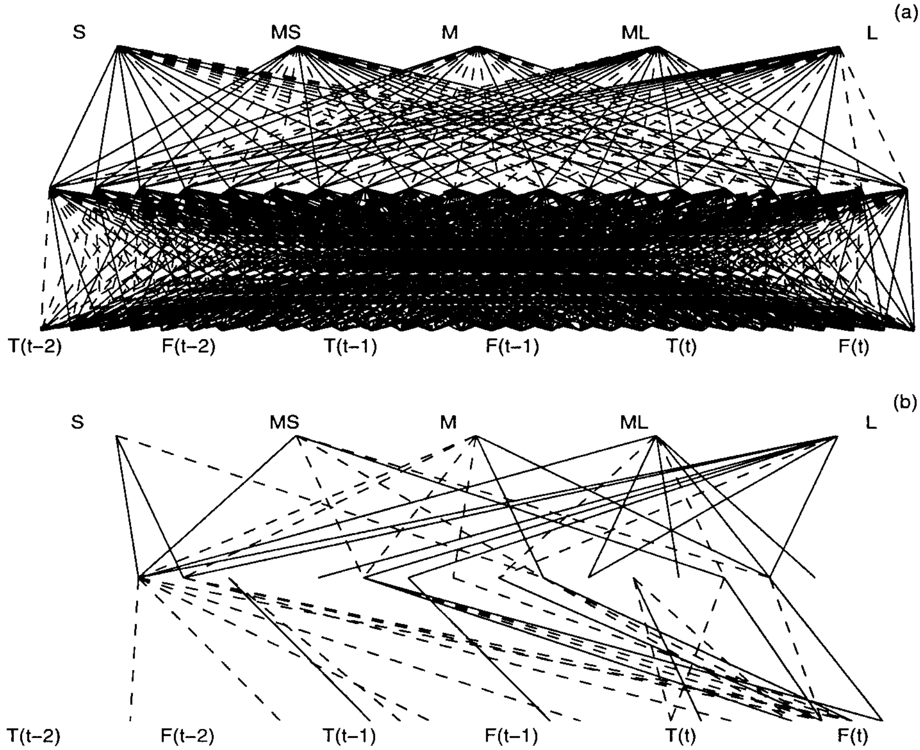

Figure 10. Structure of the hidden layers of FuNN with nodes 30-20-3: (a) initial fully connected structure with 20 rules; (b) skeleton of the trained FuNN structure with eight rules. Selective forgetting of the inputs is clearly visible.

proposed methodology has the following main features:

- The first step of the signal characterization is aimed at identifying characteristic features and dynamic ranges in the time series. The identification is based on the evaluation of the fractal dimension and determining its subbands of multifractalness.
- Next, rules have been extracted from the FuNN. The obtained rules are useful for understanding the dynamics of the system and also for identifying the essential subspace of input variables. In addition, the extracted rules support the characterization of the signals.
- A novel method has been established for identifying the effective memory of chaotic processes using structural learning with forgetting.

In the future, the above results will be used to iteratively refine the characterization and also for reliable prediction. The ultimate goal of these efforts is to adapt the system model in a dynamically changing environment.

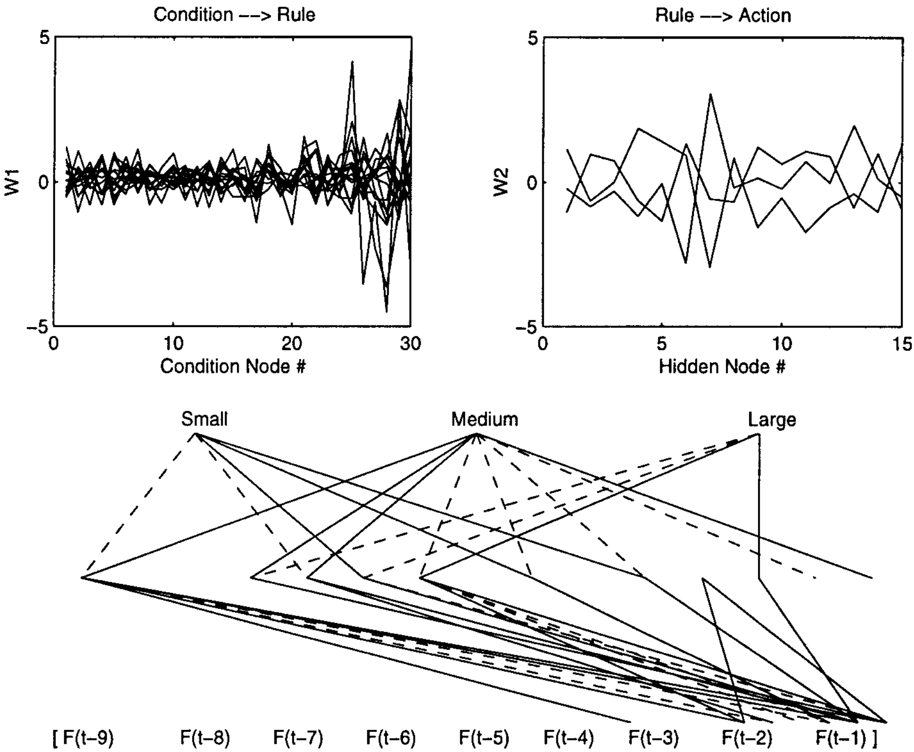

Figure 11. Structural learning in a FuNN with a structure 10-30-15-3-1.

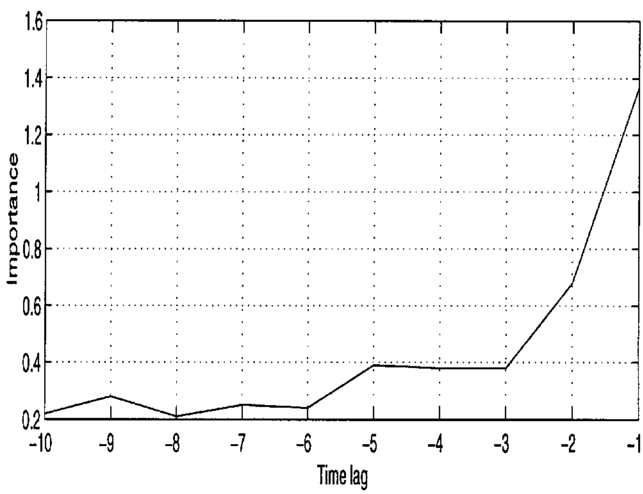

Figure 12. Memory effect in the outlet flow. The effective memory extends to about five steps backward.

This work was completed in the framework of the Foundation for Research, Science and Technology project FoRST-UO-606 on connectionist-based information systems (CBIS).

References

1. G. Sugihara and R.M. May, "Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series," *Nature*, **344**, 734–741 (1990).
2. M. Casdagli, "Nonlinear prediction of chaotic time series," *Physica D*, **35**, 335–356 (1989).
3. J.C. Principle, A. Rathie, and J.M. Kuo, "Prediction of chaotic time series with neural networks and the issue of dynamic modeling," *Int. J. Bifurcation Chaos*, **2**, 989–996 (1992).
4. J.S.R. Jang, "ANFIS: Adaptive-network-based fuzzy interference system," *IEEE Trans. Syst. Man Cybernetics*, **23**, 665–684 (1993).
5. N.K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, Cambridge, MA, 1996.
6. H. Szu and C. Szu, "Multiresolution analyses of fuzzy membership functions by chaotic neural networks," *Methods Concept Design Appl. Intell. Syst.*, **1**, 38–42 (1996).
7. N.K. Kasabov, "Adaptive connectionist production systems," *Neurocomputing*, **13**, 95–117 (1996).
8. B.B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, New York, 1982.
9. T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Physica D*, **31**, 277 (1988).
10. T. Higuchi, "Relationship between the fractal dimension and the power law index for a time series: A numerical investigation," *Physica D*, **46**, 254 (1990).
11. M. Sakuma, R. Kozma, and M. Kitamura, "Characterization of anomalies by applying methods of fractal analysis," *Nucl. Technol.*, **113**, 86–99 (1996).
12. M.J. Embrechts and Y. Danon, "Determining the fractal dimension of a time series with a neural net," in *Intelligent Engineering Systems through Artificial Neural Networks*, C.H. Dagli, L.I. Burke, B.R. Fernandez, and J. Ghosh, Eds., ASME Press, New York, 1993, Vol. 3, pp. 897–902.
13. N.K. Kasabov and R. Kozma, *Adaptive Fuzzy Neural Networks and Applications for Chaotic Time Series Analysis and Phoneme-Based Speech Recognition*, unpublished.
14. N.K. Kasabov, R. Kozma, and M.J. Watts, "Optimization and adaptation of fuzzy neural networks through genetic algorithms and structural learning," *Info. Sci.* **8** (1998). To appear.
15. R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Networks*, **4**, 740–747 (1993).
16. M. Ishikawa, "Structural learning with forgetting," *Neural Networks*, **9**, 509–521 (1996).
17. R. Kozma, "On the accuracy of mapping by neural networks trained by backpropagation with forgetting," *Neurocomputing*, **13**, 295–311 (1996).
18. D.A. Miller and J.M. Zurada, "Dynamics of structural learning with an adaptive forgetting rate," *Proc. IEEE ICNN97*, 1997, Vol. 4, pp. 1827–1832.
19. G. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden Day, San Francisco, 1970.
20. A. Cohen et al., "Application of computational intelligence for on-line control of a sequencing batch reactor at Morrinsville Sewage Treatment Plant," *Proc. IAWQ Conf. Advance Wastewater Treatment*, Sept. 1996.