A Fuzzy Neural Network Model for the Estimation of the Feeding Rate to an Anaerobic Waste Water Treatment Process

Jaesoo KIM[1], Robert KOZMA[1], Nikola KASABOV[1], Bert GOLS[2], Michel GEERINK[2], and Tico COHEN[2]

Department of Information Science, Knowledge Engineering Lab.,
University of Otago[1]
Waste Solutions Ltd.[2]

Abstract: *Biological processes are among the most challenging to predict and control. It has been recognised that the development of an intelligent system for the recognition, prediction and control of process states in a complex, nonlinear biological process control is difficult. Such unpredictable system behaviour requires an advanced, intelligent control system which learns from observations of the process dynamics and takes appropriate control action to avoid collapse of the biological culture. In the present study, a hybrid system called fuzzy neural network is considered, where the role of the fuzzy neural network is to estimate the correct feed demand as a function of the process responses. The feed material is an organic and/or inorganic mixture of chemical compounds for the bacteria to grow on. Small amounts of the feed sources must be added and the response of the bacteria must be measured. This is no easy task because the process sensors used are non-specific and their response would vary during the developmental stages of the process. This hybrid control strategy retains the advantages of both neural networks and fuzzy control. These strengths include fast and accurate learning, good generalisation capabilities, excellent explanation facilities in the form of semantically meaningful fuzzy rules, and the ability to accommodate both numerical data and existing expert knowledge about the problem under consideration. The application to the estimation and prediction of the correct feed demand shows the power of this strategy as compared with conventional fuzzy control.*

Keywords: *Fuzzy neural Networks, Hybrid learning, Knowledge extraction and insertion, Estimation, Biological process and control, Bacterial system, Total organic carbon (TOC).*

1 Introduction

Over the last decade, significant advances have been made in two distinct technological areas: fuzzy logic and computational neural networks. The theory of fuzzy logic [18] provides a mathematical framework to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. Also, it provides a mathematical morphology to emulate certain perceptual and linguistic attributes associated with human cognition. On the other hand, the computational neural network paradigms have evolved in the process of understanding the learning and adaptive features of neuronal mechanisms inherent in certain biological species. The integration of two fields has given birth to an emerging technological field– the fuzzy neural networks.

[1]Jaesoo KIM Department of Information Science, University of Otago, PO Box 56, Dunedin, New Zealand: Tel. : +64-3-479 5397; Fax. : +64-3-479 8311; Email: jskim@kaa.otago.ac.nz

The fuzzy neural networks have the potential to capture the benefits of the two fascinating fields, fuzzy logic and neural networks, into a single entity [4, 13, 1, 11, 6, 5, 17, 3, 8, 9, 10].

One of the applications of fuzzy neural networks is in advanced process control. The advanced process control application in this paper will focus on monitoring, description and control of a biological process. The biological process used in this paper is an experimental system for anaerobic waste water treatment. During this process bacteria decompose organic matter, which is gradually degraded to biogas, a mixture of methane and carbon dioxide. Up to 99 % of the organic matter can be converted, depending on the nature of the organic matter. The anaerobic process is used as an example, because the process elements and dynamics are typical for a number of biotechnological applications as they involve the growth of microorganisms growing under anaerobic conditions.

Biological processes are among the most challenging to predict and control [15, 7]. The behaviour of cells and microorganisms on different conditions is dependent on many factors, which are often not fully understood. Especially in mixed bacterial cultures of many different species interrelation between these bacteria adds a lot of complexity to the system. Such unpredictable system behaviour requires an advanced, intelligent control system, which learns from the observations of the process dynamics and takes appropriate control action.

Another problem with bacterial systems is that on-line sensors for direct measurement of the essential parameters are very expensive and not always very accurate. This means that especially in systems where low concentrations are used alternative methods need to be applied. A commonly used method is trying to find a relationship between easily measurable parameters and the desired parameters [2]. These relationships however are often very complex and not fully understood. This is where an intelligent system can be utilised.

At the start of the operation of a biological reactor, the right ingredients must be present. These are the bacteria that will convert feed material for their metabolism. The feed material is a mixture with mainly organic compounds for the bacteria to grow on. When a bacterial culture has not been in contact with a specific feed material for some time the bacteria need some time to adapt to this material. In this start up stage the bacteria are more sensitive to overloading with feed material. In a system that is not adapted, overloading will lead to an accumulation of the substrate or of one of the intermediate products, which are likely to be toxic and will lead to inhibition of growth and eventually to the dead of the bacterial culture. Adapted systems will have the ability to convert more substrate and prevent accumulation of intermediate products. Adapted cultures have the ability to provide more buffering capacity to the reactor mixture. On the other hand, if insufficient feed material is supplied the bacteria will fail to grow and without growth there is no acclimatisation and the process will fail as well. Process failure will be characterised by washout of bacteria. This occurs when the growth of bacteria is lower than the amount of bacteria that flow out. If the process succeeds in the initial critical stage, the bacteria will start to grow successfully. The growth will be self-accelerating and the demand for feed increases to a certain maximum. Further increase of the feed will lead to overloading of the process and again, process failure.

The prediction of the feed rate and the estimation of the feed demand as a function of the process responses will be the objective of this work. The intent of this paper is to demonstrate how a novel fuzzy neural network, FuNN (Fuzzy Neural Network) [8, 9, 10], can be used for the estimation of the correct feed rate.

2 Data Acquisition and Analysis

As process variables, the following measured parameters are used:

- **temperature**: The temperature dependency of all parameters is related to the change of the growth rate depending on the temperature. The growth rate behaves as an optimum curve with an optimum value depending on the type of organism. Deviations from the optimum value will result in the decrease of the growth rate, which results in lower gas and CH_4 production. If the growth rate decreases too much the pH can decrease and the ORP can increase due to accumulation of intermediate products.

- **pH**: A low pH indicates an accumulation of intermediate products (in this case fatty acids). This means the bacteria are growing under stressed conditions. This will result in a decrease of the gas production and CH_4 concentration. During an accumulation of intermediate products the ORP will go up.

- **Oxidation Reduction Potential (ORP)**: The ORP measures the balance of all chemical oxidation-reduction couples in the reactor. Every chemical oxidant will be reduced to some extent if it has oxidised another chemical. This makes a pair for each oxidant or reductant. The overall balance between oxidants and reductants is measured by the ORP in mV. ORP is a difficult concept and is hard to interpret, but it may indicate the difference between different process states. Production of hydrogen gas and anaerobic conditions in general tend to lower the ORP, aerobic conditions will raise the ORP. A stressed system may have a higher ORP than a healthy system.

- **CH_4**: The methane concentration off the gas produced in the reactor is measured. Successful biodegradation is parred with production of methane gas. In a stressed system other bacteria, producing carbon dioxide and other gases will predominate. This leads to a lower methane content of the gas.

- **gas production**: Gas production is a direct measure for the metabolic activity of the bacteria. Successful biodegradation is parred with production of methane gas. In a stressed system other bacteria, producing carbon dioxide and other gases will predominate. This leads to a lower methane content of the gas.

- **feed rate supply**: The feed rate supply is measured to calculate the amount of feed supplied to the system. On basis of the gas, and especially the methane production the conversion rate and treatment efficiency can be calculated which will be used to estimate the right feeding rate. The feed material is an organic and/or inorganic mixture of chemicals compounds for the bacteria to grow on.

A problem with biological process control is that on-line sensors which detect the essential parameters do not exist, and if they exist they are prohibitely expensive. Several parameters such as *pH* and ORP (Oxidation Reduction Potential) can be measured, but their significance and relation to the process is often not fully understood. It is also common for sensors and process variables to behave differently during different stages in the process. For instance, the dynamics of gas production in an anaerobic process are much slower in a stressed system than in an unstressed, well adapted system.

Figures 1 shows the measured parameters over the time for 7 days. The measurement interval was every 1 minute. The figures present over 10,000 measurement intervals.

Although the significance and relation of the essential parameters to the process is often not fully understood, various parameters have a direct or indirect impact on the value of a parameter. The dependencies are not always symmetric, e.g., if the gas increases, the ORP decreases, but the reverse must not be true, etc. This means that changes in one parameter may cause changes in the other parameter and because more complex reactions are possible, relations between parameters are too complex to catch: the parameters are all related in one way or another.

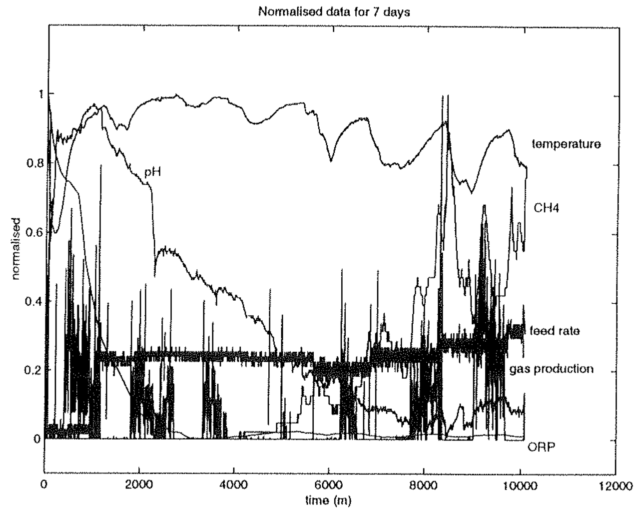

Figure 1: **Normalised data for 7 days from an experimental control system**

	temp.	pH	ORP	CH_4	gas
temp.	1.0	0.31	-0.26	-0.47	-0.07
pH	0.31	1.0	0.65	-0.70	0.13
ORP	-0.26	0.65	1.0	-0.32	0.22
CH_4	-0.47	-0.70	-0.32	1.0	0.15
gas	-0.07	0.13	0.22	0.15	1.0
TOC	0.11	-0.56	-0.84	0.44	-0.17

Table 1: **Correlation coefficients from measured data.**

Table 1 shows the correlation coefficients of the measured data from an experimental control system for anaerobic wastewater treatment. The correlation coefficients vary from -1 to +1 and indicate the correlation, or the anti-correlation of two parameters. A correlation coefficient of 0 means there is no correlation detectable.

The correlations of Table 1 appear also in Fig. 2 and Fig.3. The correlation of CH_4 to TOC (feed rate) and the negative correlation of pH and ORP to TOC is obvious.

2.1 Data Pre-processing

The quality of the results depends on the quality of the measured data which must be representative. To improve the quality of the data, obvious measuring errors were eliminated. If, for example, in quality process control some measured signals have to be investigated, it becomes necessary to filter these data in order to overcome the problems of noisy input. In addition to these methods some transformations of the measured data such as FFT, could improve the respective results. Both, filter methods as well as FFT, belong to the class of signal processing techniques.

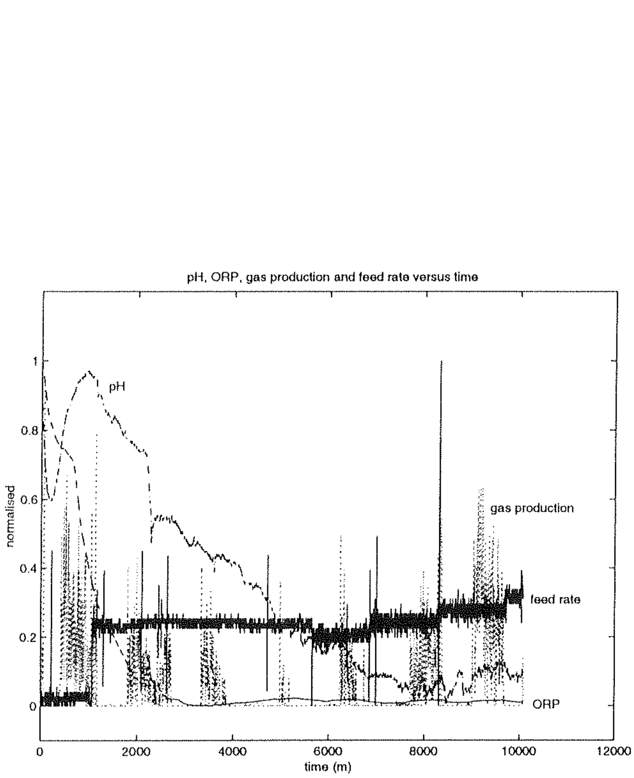

Figure 2: **pH, ORP, gas production and feed rate *versus* time**

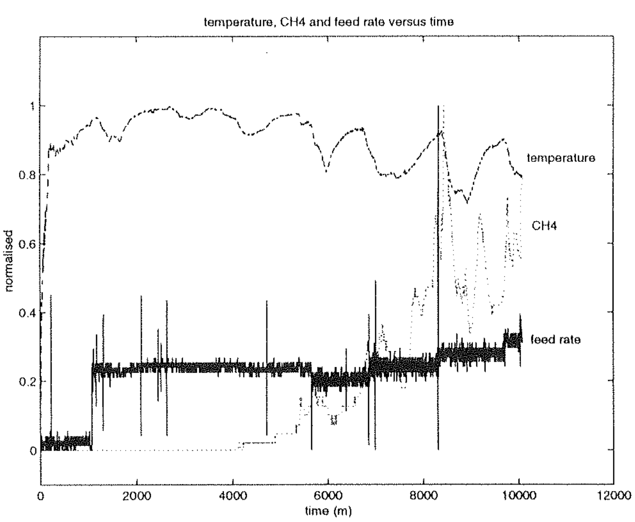

Figure 3: **temperature, CH_4 and feed rate *versus* time**

Statistical approaches could be used to detect relationships within a data set describing a special kind of application. Here correlation analysis, regression analysis, and discrimination analysis can be applied adequately. These methods could be used for example to facilitate the process of *feature extraction.* In the following study, we, however, employed an FFT algorithm and considered the giving of pattern information which is transformed using an FFT to the input layer of the fuzzy neural network, as will be explained in the next section. Data analysis have been performed with a commercially available software tool. For further information about FFT and filter methods see, e.g. [16, 14].

3 The Experimental System

The research will be target to the development of an intelligent system for the recognition, prediction and control of process states in a complex, non-linear biological process. Challenges will be put to the system to test its intelligent capabilities. These are (a) use of non-specific bacterial starter cultures; (b) use of toxic waste materials as feed for the process. At the start of operation of a biological reactor, the right ingredients must be present. These are bacteria that will catalyse the biochemical conversion processes and a feed material.

Initially the reactor is filled with a bacterial culture grown under aerobic conditions so that the bacteria are not adapted to the new environment. Little amounts of feed are supplied to the reactor in order to adapt the bacteria to the feed and the response will be measured. After the initial critical stage the bacteria will start to grow and the demand for feed increases. The feeding rate is increased till the maximum feed consumption for the reactor is reached. Further increase of the feed at that point will lead to overloading of the system. This will result in the accumulation of acids, the pH goes down and the system finally dies. This means that they can easily be inhibited in their growth if to much of the feed is supplied. On the other hand, if insufficient feed material is supplied, the bacteria would fail to grow and without growth there is no acclimatisation and the process will fail. Process failure will be characterised by washout of the bacteria and no process response. So the control system must be able to recognise the difference between a *healthy* and an *unhealthy* response of the bacteria and take appropriate action to avoid collapse of the biological culture.

The experimental system is shown in Figure 6. The process is an anaerobic digestion performed in anaerobic reactor. The bioreactor is maintained at 35° C by a heated water jacket. The feed is supplied to the bioreactor from a bottle by a feed pump P1 controlled by an output from the supervisory PC. Different pump rates are obtained by turning the pump on for a required part of 10 minutes. A weighing scale S1 measures the quantity of feed used. After filling the reactor, excess liquid will exit the reactor via an outlet at the top. This will contain a mixture of liquid and produced gas. The mixture is separated into three streams in a separator. The first stream is recirculated back into the reactor for mixing by pump P2, the recirculation. In the recirculation loop several sensors are inserted for the process variables including pH (S3), Oxidation Reduction Potential (ORP, S4) and the temperature (S5). The second stream is the excess liquid collected in a container. The third stream is the produced gas. An infrared methane analyser analyses the methane content of the gas. The gas is then collected in a Mariotti flask, which is a bottle filled with acidified water to prevent the produced CO_2 from dissolving. The gas displaces the water from the bottle, which is pressure balanced so that no pressure is needed to force the water out of the bottle. The displaced liquid is collected on a second weighing scale S2. Data from the sensors and the scales is collected and stored on minute intervals on the supervisory computer.

4 The Modelling Technique - the Fuzzy Neural Network FuNN

The fuzzy neural network FuNN [8, 9, 10] uses a multi-layered perceptron (MLP) network and a extended BP training algorithm. In this connectionist structure, the input and output nodes represent the input states and output control/decision signals respectively, and in the hidden layers, there are nodes functioning as membership functions (MFs) and rules. This eliminates the disadvantage of a normal feedforward multi-layer net which is difficult for an outside observer to understand or to modify.

The architecture facilitates learning from data and approximate reasoning, as well as fuzzy rule extraction and insertion. It allows for the combination of both numerical and fuzzy data and fuzzy rules to be used in one system, thus producing the synergistic benefits associated with the two sources. In addition, it allows for adaptive learning in a dynamically changing environment.

Below a brief description of the components of the FuNN structure and functionalities, and the philosophy behind this architecture, are given.

4.1 The architecture of FuNN

The general FuNN architecture consists of 5 layers with partial feedforward connections as shown in

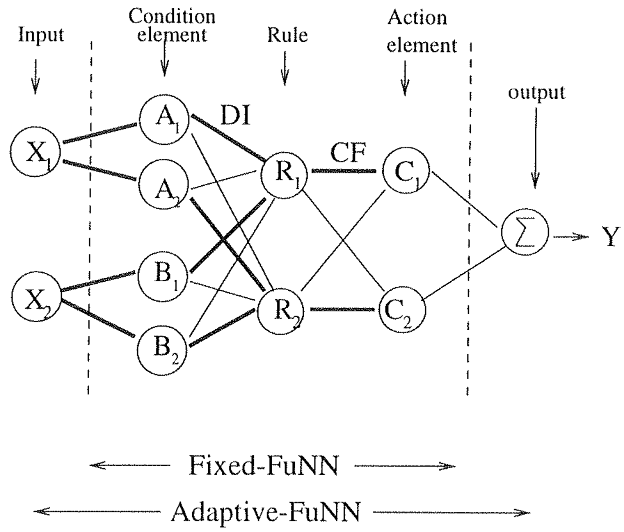

Figure 4: **A FuNN structure for two fuzzy rules:** *if* X_1 *is* A_1 *(*DI_{11} *and* X_2 *is* B_1 *(*DI_{12}*) then* Y *is (not)* C_2 *(*CF_{12}*)*

Fig. 4. In this connectionist structure a modified BP training algorithm was developed. The first and last layer act as the fuzzifier and the defuzzifier, respectively. In the condition layer, uniformly distributed triangular membership functions are used. Singletons are applied in between the action and the output layer, as connection weights, which represent the centre of a membership functions. FuNN is also adaptable 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 training data.

- **Input Layer:** Nodes in layer one are input nodes which represent input linguistic variables [19]. The nodes in this layer only transmit input values to the next layer, *condition element layer*.
- **Condition Layer:** Nodes in this layer act as fuzzification processors. The input values are fed to the condition element layer which performs fuzzification. This is implemented using three-point triangular membership functions with centres represented as connection weights. The triangles are completed with the minimum and maximum points attached to adjacent centres, or shouldered in the case of the first and last membership functions.

  The triangular membership functions are allowed to be non-symmetrical and any input value will belong to maximum of two membership functions with degrees differing from zero. It will always involve two membership functions unless the input value falls exactly on a membership function *centre* in which case the single membership will be activated. These membership degrees for any given input will always sum up to one, ensuring that some rules will be given the opportunity to fire for all points in the input space. This centre-based membership approach taken by FuNN avoids the problems of uncovered regions in the input space. These do not always limit centres and widths in such a way as to ensure complete coverage. While algorithms could be formulated and used in such cases to force the memberships to cover the input space, the simple centre-based approach taken by FuNN seems both more efficient and more natural, with fewer arbitrary restrictions. It should be noted that there are no *bias* connections necessary for this representation in FuNN.

Initially the membership functions are spaced equally over the weight space, although if any expert knowledge is available this can be used for initialisation. In order to maintain the semantic meaning of the membership functions some restrictions are introduced. When adaptation takes place the centres are limited to remain within equally sized partitions of the weight space. This avoids problems with violating the semantic ordering of membership functions. Therefore, under the FuNN architecture labels can be attached to weights when the network is constructed and these will remain valid for the *lifetime* of the network. For example, a membership function weight representing *low* always have a centre less than *medium*, which will always be less than *high*. Simple activation functions are used in the condition element nodes to perform fuzzification.

- **Rule Layer:** Each node in this layer is a rule node which represents a single fuzzy rule. Thus, all the nodes in this layer form a fuzzy rule base.

  The activation function is the sigmoid (logistic) function with a variable gain coefficient (a default value of 1 is used). The connection weights from the Condition Layer are initialised randomly with small values and fully connected.

  The semantic meaning of the activation of a node is that it represents the degree to which input data matches the antecedent component of the 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* (also called the *membership functions layer*) to the *Rule Layer* represent semantically the **degrees of importance** (DI) of the corresponding condition elements for the activation of this node.

- **Action Layer:** In this layer links define the consequences of the rules and a node represents a fuzzy label from the fuzzy quantisation space of an output variable. The activation of the node represents the degree to which this membership function is supported by all fuzzy rules together. So this is the level to which the membership function for this fuzzy linguistic label is *cut* according to the rules and current facts. The connections from the *Rule Layer* to the *Action Element Layer* represent conceptually the **confidence factors** (CF) or certainties of the corresponding rules when inferring fuzzy output values. They are subject to constraints that require them to remain in specified intervals as for the condition element layer with the same advantages of semantic interpretability. The activation function for the nodes of this layer is the sigmoid (logistic) function with the same or variable gain factor, and connection weights are initialised as in the previous layer. This gain factor should be adjusted appropriately given the size of the weight boundary.

- **Output Layer:** It represents the output variables of the system. This node and links attached to them act as the defuzzifier. This layer performs the centre of gravity (COG) defuzzification.

  Singletons are used as membership functions for the output labels, which is equivalent to having the centres only of triangular membership functions, as it was the case of the input variables, and are attached as connection weights to the corresponding links. Linear activation functions are used here.

  Adapting the output membership functions means moving the centres. The requirement that the membership degrees to which a particular output value belongs to the various fuzzy labels must always sum to one, is always satisfied. For each centre, there is a constraining band (partition) where this value can move to. This principle applies in the same way as the input membership function centres restrictions are.

Details of the supervised learning algorithms of FuNN are given below.

4.2 The FuNN Basic Learning Algorithm - a modified backpropagation algorithm

This section explains the algorithm used for the FuNN system. That includes a forward and a backward phase.

4.2.1 Forward Pass

This phase computes the activation values of all the nodes in the network from the first to fifth layers. In this a superscript indicates the layer and a subscript describes connection weights between layers.

- **Input Layer:** The nodes in this layer only transmit input values (crisp values) to the next layer directly without modification.
- **Condition Layer:** The output function of this node is the degree that the input belongs to the given membership function. The input weight represents the centre for that particular membership function, with the minimum and maximum determined using the adjacent membership's centres.

  In the case of the first and last membership function for a particular variable a shoulder is used instead. Hence, this layer acts as the fuzzifier. Each membership function is triangular and an input signal(x) activates only two neighbouring membership functions simultaneously, the sum of the grades of these two adjacent membership functions for any given input is always equal to 1.

  For a triangle-shaped membership function as in FuNN, the activation functions for a node (i) are:

$$\begin{aligned} Act_i^c &= 1 - \frac{x - a_i}{a_{i+1} - a_i},\ a_i < x < a_{i+1}, \\ Act_i^c &= 1 - \frac{a_i - x}{a_i - a_{i-1}},\ a_{i-1} < x < a_i, \\ Act_i^c &= 1,\ x = a_i, \end{aligned} \tag{1}$$

  where a is the centre of the triangular membership function.

- **Rule Layer:** The connections from the condition to this layer are used to perform precondition matching of fuzzy rules. The connection weights may be set either randomly and then trained or according to a set of rules, namely rules insertion. The net inputs and activations are respectively,

$$\begin{aligned} Net^r &= \sum_c w_{rc} Act^c, \\ Act^r &= \frac{1}{1 + \epsilon^{-g*Net^r}}, \end{aligned} \tag{2}$$

  where g is a gain factor.

- **Action Layer:** The nodes and connection weights in this layer function as those in the **Rule Layer** for Net input and activation:

$$\begin{aligned} Net^a &= \sum_r w_{ar} Act^r, \\ Act^a &= \frac{1}{1 + \epsilon^{-g*Net^a}}. \end{aligned} \tag{3}$$

- **Output Layer:** This layer performs defuzzification to produce a crisp output value. Among the commonly used defuzzification strategies, the *Centre of Gravity* (COG) method was used:

$$\begin{aligned} Net^o &= \sum_a w_{oa} Act^a, \\ Act^o &= \frac{Net^o}{\sum Act^a}. \end{aligned} \tag{4}$$

4.2.2 Backward Pass

The goal of the system is to minimise the following function. Here the standard BP learning algorithm is used:

$$E = \frac{1}{2} \sum (y^d - y^o)^2 \tag{5}$$

where y^d is the desired output and y^o is the current output. Hence the general learning rule (gradient descent) used is

$$\begin{aligned} \Delta w &\approx -\frac{\partial E}{\partial w}, \\ \Delta w_{t+1} &= \eta(-\frac{\partial E}{\partial w}) + \alpha \Delta w_t, \end{aligned} \tag{6}$$

where η is the learning rate and α is the momentum coefficient, and using *chain rule* we have

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial Net} \frac{\partial Net}{\partial w} = -\delta * Act. \tag{7}$$

Hence the weight update rule is:

$$\Delta w_{t+1} = \eta \delta Act + \alpha \Delta w_t. \tag{8}$$

- **Output Layer:** The error signal δ^o is derived as in the following:

$$\delta^o = -\frac{\partial E}{\partial Net^o} = -\frac{\partial E}{\partial Act^o} \frac{\partial Act^o}{\partial Net^o} = y^d - y^o \tag{9}$$

- **Action Layer:** The error for nodes in this layer is calculated based on fuzzification of desired outputs and activation of each node. The fuzzification of desired output for this layer is same as 1. Hence we have

$$\delta^a = f^{'}(Net^a) * E^a = Act^a(1 - Act^a) \sum (d^a - Act^a) \tag{10}$$

- **Rule Layer:** As in the Action layer, the error signals need to be computed and this error signal can be derived as

$$\delta^r = Act^r(1 - Act^r) \sum (w_{ar} \delta^a) \tag{11}$$

- **Condition Layer:** If inputs lies in the fuzzy segment, then the corresponding weight should be increased directly proportional to the propagated error from the previous layer, because the error is caused by the weight. This proposition can be represented by the following equation:

$$\delta^c = \frac{\partial Act_i^c}{\partial a_i} \sum (w_{rc}\delta^r). \qquad (12)$$

Using Eq. 1, the adaptive rule of the centre a_i, is derived as

$$\frac{\partial Act_i^c}{\partial a_i} = \begin{cases} \frac{\partial a_i - x}{\partial a_i - a_{i+1}{}^2}, & \text{if } a_i \leq x \leq a_{i+1}, \\ 0, & \text{otherwise} \end{cases} \qquad (13)$$

Hence the adaptive rule of connection weights becomes

$$\Delta w_{t+1} = \eta \delta^c x + \alpha \Delta w_t. \qquad (14)$$

5 Experimental Results and Discussions

In the present study, the role of the fuzzy neural network is to estimate patterns of changes in the TOC concentration of feed solution. In order to demonstrate the potential of the proposed FuNN to modelling of the biological process dynamics, the 5 measured parameters temperature, pH, ORP, CH_4, and gas production from an experimental control system, are used where the number of patterns was 10,080, as it was mentioned in Section 2. For learning the TOC concentration, the first 5,040 (t=1 to 5,040) patterns pairs (training data set) was used for training the FuNN while the remaining 5,040 pairs (t=5,041 to 10,080) were prepared for validating the identified model.

For this purpose, three triangular-type (small, medium, large) membership functions of input-output variables are attached and the following experiment was performed: a 5-15-10-3-1 FuNN was trained with the modified backpropagation algorithm and fixed MFs. A small learning rate of 0.01 is given and to stimulate learning capability the momentum is employed. It is observed that if the learning rate (α) is small, the gradient method will closely approximate the gradient path, but convergence will be slow since the gradient must be calculated many times. On the other hand, if α is large, convergence will initially be very fast, but the algorithm will oscillate about the optimum. Based on these observations, α was variable during training and individually set for each of the layers in the FuNN, while the momentum and the gain factor in the logistic activation function were 0.9 and 1, respectively, for layers 2 to 5.

Fig. 5 shows the measured and the estimated values of the feed rate (TOC) after 2,000 iterations with *fixed learning mode*, i.e., the MFs of input and output are frozen, and demonstrate how FuNN can effectively model a biological dynamics. After 2,000 epochs, we had root mean squared error (RMSE) = 0.0019. The fuzzy neural network estimates the TOC by using the actual input of the sensors. The results are good and are suitable for the estimation of the TOC by sensors.

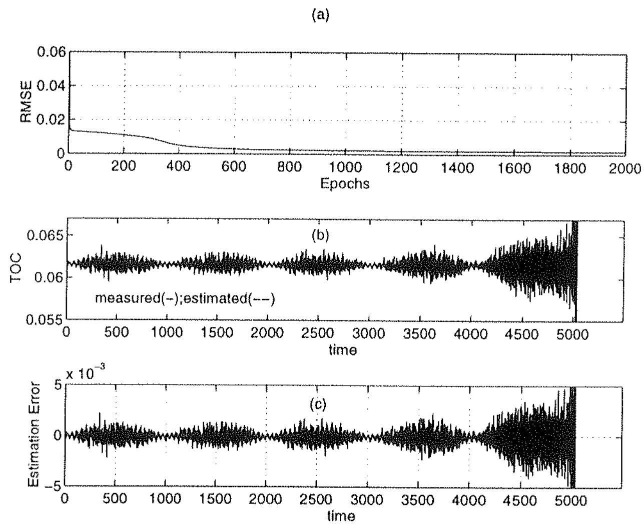

Figure 5: **Model performance of FuNN with three linguistic labels: (a) RMSE curves for the FuNN model; (b) TOC measured and estimated: Actual data is shown by the solid line, estimated data by the dotted line; (c) estimation error between measured and estimated data for test data.**

Several different methods for fuzzy rules extraction are applicable on the FuNN system [8, 9, 10]. For interpreting a FuNN structure in terms of aggregated fuzzy rules an algorithm is also implemented [10]. Each rule node is represented as one fuzzy rule. The strongest connection from a condition element node for an input variable to the rule node, along with the neighbouring condition element nodes, are represented in the associated rule. The connection weights of these connections are interpreted as degrees of importance (DI) attached to the corresponding condition elements. The extracted rules from the FuNN can be inserted in the other FuNN modules through the *rule insertion module* [10]. Extracted fuzzy if-then rules from FuNN are described in Table 2.

6 Concluding Remarks

Combined hybrid systems between neural networks and fuzzy logic are rapidly gaining popularity in the design of many complex systems. Experience shows that this type of combined system yields results sometimes superior to those obtained by the fuzzy control systems. Moreover, fuzzy neural networks are a promising paradigm in the area of *Soft Computing*.

They have strengths in both learning from data and monitoring knowledge.

In this paper, a fuzzy neural network called FuNN is presented as a demonstration control system and showed that a hybrid system is capable of modelling complex chemical and biological processes to estimate the quality parameter TOC (Total Organic Carbon). It is expected to be used further as a tool for development of adaptive systems responding to changing parameters and characteristics of the process control.

The main advantages of FuNN is to combine both the benefits of neural networks and fuzzy logic systems into an integrated system, with results being a FuNN system, which

- has faster learning speed than normal neural network learning algorithms,
- provides a good explanation on what has been learned by the network,
- provides a means for rules extraction and rules refinement, and item facilitates adaptive learning in a dynamically changing environment.

In particular, the adaptive learning algorithms developed here show that this is a promising approach to building adaptive intelligent information processing systems which suits many applications such as signal processing, speech recognition, time-series modelling and prediction, adaptive control, data mining and knowledge acquisition, and image processing.

Future research is anticipated in applying on-line adaptive control systems. Of course, the model we approached to process control systems is by no means exhaustive; for more approaches we need to further consider several other possibilities such as fuzzy-genetic, neuro-genetic, and neuro-fuzzy-genetic systems. Such systems are likely to dominated the area of hybrid intelligent information systems in the near future. On-line adaptation of fuzzy neural networks is still to be investigated. That may well be the most important criterion to compare different fuzzy neural network structures.

Acknowledgements

This work is partially supported by the PGSF UOO 606 research grant funded by the New Zealand Foundation for Research, Science and Technology. The final results of this project are available from: http://kel.otago.ac.nz/CBIS/.

References

[1] H.R. Berenji and P. Khedkar, "Fuzzy Rules for Guiding Reinforcement Learning," In *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (IPMU'92), pp. 511-514, Mallorca, 1992.

[2] D. Dochain, and G. Bastin, "On-line estimation and adaptive control of bioreactors," *Elsevier Science Publishers B.V.*, Amsterdam, 1990.

[3] W. Hauptmann, and K. Heesche, "A Neural Net Topology for Bidirectional Fuzzy-Neuro Transformation," *Proc. of the Fuzz-IEEE/IFES*, Yokohama, Japan, pp. 1511-1518, 1995.

[4] S. Horikawa, T. Fruhashi, S. Okuma, and Y. Uchikawa, "Composition Methods of Fuzzy Neural Networks," *Proc. of IEEE Industrial Electronics*, pp. 1253-1258, 1990.

[5] H. Ishibuchi, H. Tanaka, H. Okada, "Interpolation of Fuzzy If-Then Rules by Neural Networks," *Int. J. of Aprrox. Reasoning*, Vol. 10, No. 1, pp. 3-27, 1994.

[6] J.-S. Roger Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. on System, Man and Cybernetics*, Vol. 23, No. 3, pp. 665-685, May/June, 1993.

[7] M.N. Karim, T. Yoshida, S.L. Rivera, V.M. Saucedo, B. Eikens, and G.S. Oh, "Global and local neural network models in biotechnology-applications to different cultivation processes," *Journal of Fermentation & Bioengineering*, Vol. 83, No. 1, pp. 1-11, 1997.

[8] N. Kasabov, "Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering," *The MIT Press*, CA, MA, 550 pages, 1996.

[9] N. Kasabov, "Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems," *Fuzzy Sets and Systems*, Vol. 82, pp. 135-149, 1996.

[10] N. Kasabov, J. Kim, M. Watts, and A. Gray, "FuNN/2-A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition," *Information Sciences*, Vol. 101, No. 3-4, pp. 155-175, 1997.

[11] B. Kosko, "Neural Networks and Fuzzy Systems: A Dynamic Systems Approach to Machine Intelligence," *Prentice-Hall, Englewood Cliffs*, N.J., 1992.

[12] R. Kozma, N. Kasabov, J. Kim, and T. Cohen, "Integration of Connectionist Methods and Chaotic Times Series Analysis for the Prediction of Process Data," *Int. J. of Intelligent Systems*, Vol. 13, No. 6, pp. 519-538, 1998.

[13] C.T. Lin and C.S.G. Lee, "Neural-Networks-Based Fuzzy Logic Control and Decision System," *IEEE Trans. on Computers*, Vol. 40, No. 12, pp. 1320-1366, December, 1991.

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, "Numerical Recipes," *Cambridge University Press*, 1986.

[15] Y.P. Tsai, C.F. Ouyang, W.L Chiang, and M.Y. Wu, "Construction of an on-line fuzzy controller for the dynamic activated sludge process," *Water Res.*, Vol. 28, pp. 913-921, 1994.

[16] User's Guide, *MATLAB Signal Processing Toolbox*, V5.1.

[17] R.R. Yager, "Modelling and formulating fuzzy knowledge bases using neural network," *Neural Networks*, Vol. 7, No. 8, pp. 1273-1283, 1994.

[18] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.

[19] ——, "Outline of a new approach to the analysis of complex systems and decision process," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 3, No. 1, pp. 28-44, January, 1973.

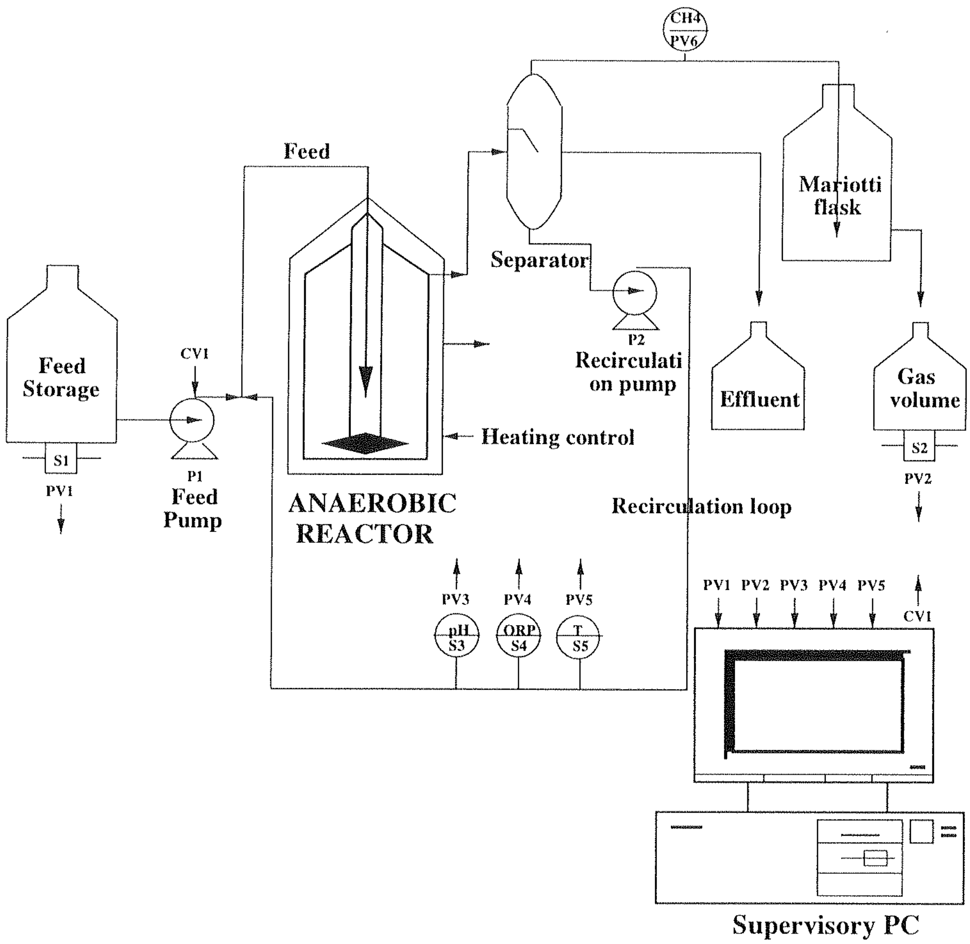

Figure 6: **Experimental setup for Biological Process Control: ORP (Oxidation Reduction Potential; PV (Process Variable); CV (Control Variable).**

Fuzzy rules	IF					THEN
	x_1	x_2	x_3	x_4	x_5	y
1	S(0.33)	S(0.22)	S(0.23)	M(0.38)	M(0.18)	S(0.37)
2	S(0.55)	M(0.23)	S(0.38)	S(0.39)	M(0.71)	S(0.94)
3	L(0.59)	L(1.04)	L(0.31)	L(1.03)	L(0.36)	L(2.74)
4	S(0.15)	S(0.69)	S(0.27)	M(0.11)	M(0.14)	S(0.53)
5	L(0.50)	L(0.21)	M(0.23)	L(0.07)	L(0.29)	L(0.52)
6	S(0.80)	S(0.79)	S(0.53)	M(0.41)	L(0.19)	S(1.25)
7	S(0.81)	S(0.64)	S(0.56)	S(0.35)	M(0.80)	S(1.12)

Table 2: **The fuzzy rules generated and their corresponding degrees.**