Proceedings of the Korean Nuclear Society Spring Meeting

Kori, Korea, May 2000

A Fuzzy Neural Network for Sensor Signal Estimation

Man Gyun Na

Chosun University

375 Seosuk-dong, Dong-gu, Kwangju 501-759, Korea

Abstract

In this work, a fuzzy neural network is used to estimate the relevant sensor signal using other sensor signals. Noise components in input signals into the fuzzy neural network are removed through the wavelet denoising technique. Principal component analysis (PCA) is used to reduce the dimension of an input space without losing a significant amount of information. A lower dimensional input space will also usually reduce the time necessary to train a fuzzy-neural network. Also, the principal component analysis makes easy the selection of the input signals into the fuzzy neural network. The fuzzy neural network parameters are optimized by two learning methods. A genetic algorithm is used to optimize the antecedent parameters of the fuzzy neural network and a least-squares algorithm is used to solve the consequent parameters. The proposed algorithm was verified through the application to the pressurizer water level and the hot-leg flowrate measurements in pressurized water reactors.

1. Introduction

Past research shows that the fuzzy neural network is a powerful tool for sensor monitoring. Through training, the fuzzy neural network is very good at phenomenal nonlinear function approximation and pattern recognition, especially when expert's diagnostic knowledge and the prior relation of fault symptom model are not clear. The direct use of transient signals in the time domain to the input of a fuzzy neural network can be difficult since the subtle differences may occur between different transients. Therefore, it is necessary to preprocess the transient signals.

In this work, noise components in the input signals are removed through the wavelet denoising technique. Wavelets have the ability to analyze a localized area of a larger signal. Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity [1]. The dimension of the input signals into the fuzzy neural network is reduced through the principal component analysis (PCA). Principal component analysis [2-3] is used to reduce the dimension of an input space without losing a significant amount of information. This method transforms the input space into an orthogonal space that can be chosen to be of lower dimension with minimal loss of information. A lower dimensional input space will also usually reduce the time necessary to train a fuzzy neural network. Also, the PCA method makes easy the selection of the input signals into the fuzzy neural network.

The signals preprocessed by the wavelet denoising technique and principal component analysis are applied to the inputs of the fuzzy neural network and the relevant signals are estimated. The fuzzy neural network parameters such as the membership functions and the connectives between layers in a fuzzy neural network must be optimized for good performance. The fuzzy neural network parameters are optimized by two learning methods. A genetic algorithm is used to optimize the antecedent parameters of the fuzzy neural network and a least-squares algorithm is used to solve its consequent parameters.

The proposed algorithm was applied to the pressurizer water level and the hot-leg flowrate measurements in pressurized water reactors.

2. Preprocessing of the Sensor Signals

2.1 Wavelet Denoising

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis consists of breaking up a signal into shifted and scaled versions of the original wavelet called mother wavelet.

Let a function or a signal $f(t)$ be expressed as

$$f(t) = \sum_k c_k \varphi_k(t) \quad \text{for any } f(t) \in V_0 , \tag{1}$$

where V_0 is the subspace of $L^2(R)$ (the space of all functions with a well defined integral of the square of the modulus of the function) spanned by the scaling functions $\varphi_k(t)$ with all integers k from minus infinity to infinity. One can generally increase the size of the subspace spanned by changing the time scale of the scaling functions that is generated from the basic scaling function by scaling and translation by $\varphi_{j,k}(t) = 2^{j/2}\varphi(2^j t - k)$. For $j > 0$, the span is larger since $\varphi_{j,k}(t)$ is narrower and is translated in smaller steps. It represents finer detail. For $j < 0$, the span is smaller since $\varphi_{j,k}(t)$ is wider and is translated in larger steps, and it represents only coarse information [4].

The important features of a signal can be better described, not by using $\varphi_{j,k}(t)$ and increasing j to increase the size of the subspace spanned by the scaling functions, but by defining a slightly different set of the wavelet functions $\psi_{j,k}(t)$ that span the differences between the spaces spanned by the various scales of the scaling function. If $f(t) \in V_{j+1}$ can be expressed at a scale of $j+1$, when it is expressed at one scale lower resolution, wavelets are necessary for the detail not available at a scale of j as follows:

$$f(t) = \sum_k a_{j+1}(k)\varphi_{j+1,k}(t) = \sum_k a_j(k)\varphi_{j,k}(t) + \sum_k d_j(k)\psi_{j,k}(t) . \tag{2}$$

Since $\varphi_{j,k}(t)$ and $\psi_{j,k}(t)$ are orthonormal,

$$a_j(k) = \sum_m l(m-2k) a_{j+1}(m) \,, \tag{3}$$

$$d_j(k) = \sum_m h(m-2k) a_{j+1}(m) \,. \tag{4}$$

In the digital signal processing, the filtering of the input signal is thought as a moving average with the coefficients being the weights like Eqs. (3) and (4). Wavelet decomposition is to obtain low pass approximations and high pass details. An approximation is a low resolution representation of the original signal, while a detail is the difference between two successive low resolution representation of the original signal [5]. An approximation contains the general trend of the original signal, while a detail contains the high frequency contents of the original signal. Approximation and details are obtained through a successive convolution process. The detail and approximation of the original signal are obtained by passing it through a filter bank [4], which consists of low and high pass filters. Wavelet decomposition also includes down-sampling operations. Down-sampling operations keep only the even samples of the signal.

2.2 Principal Component Analysis

Principal component analysis is used to reduce the dimension of an input space without losing a significant amount of information. A lower dimensional input space will also reduce the time necessary to train a fuzzy neural network. The PCA method can be chosen as a method of preprocessing data to extract uncorrelated features from the data.

The PCA method involves linearly transforming the input space into an orthogonal space that can be chosen to be of lower dimension with minimal loss of information. The method also makes the transformed vectors orthogonal and uncorrelated.

Given a signal vector $\mathbf{x}$ of p dimensions, $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix}^T$, its mean vector $\boldsymbol{\mu}$ and covariance matrix Σ are described by

$$\boldsymbol{\mu} = E\{\mathbf{x}\} = \begin{bmatrix} m_1 & m_2 & \cdots & m_p \end{bmatrix}^T \,, \tag{5}$$

$$\Sigma = E\left\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T\right\}. \tag{6}$$

Since the true mean and the true covariance matrix are seldom known, the mean and covariance matrix are replaced with the sample mean $\mathbf{m}$ and the sample covariance matrix $\mathbf{S}$. The eigenvalues $l_1, l_2, \cdots, l_p$, and the corresponding orthonormal eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_p$ of the covariance matrix $\mathbf{S}$ are calculated, and then arranged according to their magnitude:

$$l_1 \geq l_2 \geq \cdots \geq l_p \,. \tag{7}$$

The eigenvectors $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_p$ are called the principal components. The eigenvalues are proportional to the amount of variance (information) represented by the corresponding principal component. The transformation to the principal component space can be written as:

$$\mathbf{z} = \mathbf{x}^T \mathbf{P} \,, \tag{8}$$

where $\mathbf{P} = \lfloor \mathbf{p}_1 \ \mathbf{p}_2 \ \cdots \mathbf{p}_p \rfloor$.

The feature vector $\mathbf{z}$ can be transformed back into the original data vector $\mathbf{x}$ without a loss of information as long as the number of features, m, is equal to the dimension of the original space, p. For $m < p$, some information is usually lost. The objective is to choose a small m that does not lose much information. Many times there is variability in the data from random noise sources, this variability is usually of no concern, and by transforming to a lower dimensional space this noise can sometimes be removed.

3. Fuzzy Neural Network for Sensor Signal Estimation

3.1 Fuzzy Inference System

A system that consists of a fuzzy inference system and its neuronal training system is usually called a fuzzy neural network or an adaptive fuzzy system. In a fuzzy inference system, the i-th rule can be described using the first-order Sugeno-Takagi type [6] as follows:

$$\text{If } x_1 \text{ is } A_{i1} \ AND \cdots AND \ x_m \text{ is } A_{im}, \text{ then } y_i \text{ is } f_i(x_1, \cdots, x_m), \tag{9}$$

where

$x_1, \cdots, x_m$ = input variables to the fuzzy neural network (m = number of input variables)

$A_{i1}, \cdots, A_{im}$ = antecedent membership function of each input variable for the i-th rule (i = 1, 2, ..., n)

y_i = output of the i-th rule

$$f_i(x_1,\cdots,x_m) = \sum_{j=1}^{m} q_{ij} x_j + r_i \tag{10}$$

q_{ij} = weighting value of the j-th input onto the i-th output

r_i = bias of the i-th output

n = number of rules.

In this work, the following Gaussian and sigmoid membership functions are used for each input variable as shown in Fig. 1:

$$A_{ij}(x_j) = \exp\left(-\frac{(x_j - c_{ij})^2}{2s_{ij}^2} \right) \tag{11}$$

and

$$A_{ij}(x_j) = \frac{1}{\exp\left(-\frac{x_j - c_{ij}}{s_{ij}} \right) + 1} \tag{12}$$

where

c_{ij} = center position of a membership function for the i-th rule and the j-th input

s_{ij} = sharpness of a membership function for the i -th rule and the j -th input.

The output of an arbitrary i -th rule, f_i, is composed of the first-order polynomial of inputs as given in Eq. (10). The output of a fuzzy neural network with n rules is obtained by weighting the real values of consequent part for all rules with the corresponding membership grade. The output is obtained as follows:

$$y = \sum_{i=1}^{n} \overline{w}_i f_i \, , \tag{13}$$

where

$$\overline{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{14}$$

$$w_i = \prod_{j=1}^{m} A_{ij}(x_j) \, . \tag{15}$$

The membership value for rule i , w_i , means a compatibility grade between antecedent parts of "if x_1 is A and x_2 is B". A and B are generic terms for the fuzzy linguistic sets like PB (positive big), ZO (zero) and NB (negative big). The multiplicative weight in Eq. (15) is preferred over the minimum weight because of its smoothness properties. The fuzzy neural network described above is shown in Fig. 2. In Figure 2, x_1, x_2 and x_m are the input values to the fuzzy neural network.

3.2 Training of the Fuzzy Inference System

Fuzzy inference system parameters such as the membership functions and the connectives between layers in a fuzzy neural network must be optimized for good performance. This is accomplished by adapting the antecedent parameters (membership function parameters) and consequent parameters (the polynomial coefficients of the consequent part) so that a specified objective function is minimized.

The adaptation methods of most fuzzy inference systems rely on the back-propagation algorithm [7]. The back-propagation algorithm is a general method for recursively solving for parameter optimization. Since this conventional optimization algorithm is susceptible to getting stuck at local optima, the genetic algorithm is used in this work. However, the genetic algorithm requires much time if there are many parameters to be optimized. Therefore, the least-squares method that is a one-pass optimization method is combined for a part of the parameters. The genetic algorithm is used to optimize the antecedent parameters c_{ij} and s_{ij}, and the least-squares algorithm is used to solve the consequent parameters q_{ij} and r_i .

In genetic algorithms, the term *chromosome* typically refers to a candidate solution to a problem, generally encoded as a bit string. Each chromosome can be thought of as a point in the search space of candidate solutions. The genetic algorithms process populations of chromosomes, successively replacing one such population with another. The genetic algorithms require a fitness function that assigns a score to each chromosome in the current population. The fitness of a chromosome (individual) depends on how well that chromosome solves the problem at hand [8-9]. The fitness of a chromosome is calculated by means of the energy (cost function) of the individual. The chromosome that has lower energy has higher fitness. Therefore, one uses a cost function that evaluates the extent to which each individual is suitable for the given objectives such as small maximum error together with small total squared error. To accomplish the aforementioned objectives, the two energy functions are expressed by

$$E_1 = \sum_{k=1}^{N} \left(y_d(k) - y(k)\right)^2 , \tag{16}$$

$$E_2 = \max\left\{ \left| y_d(k) - y(k) \right|_{all\ k} \right\}, \tag{17}$$

where $y_d(k)$ and $y(k)$ denote the measured signal and the estimated signal, respectively, and E_1 and E_2 are overall sum of squared errors, maximum absolute error, respectively. The fitness function is defined as follows:

$$F = \exp\left(\frac{d - E_t}{r}\right) , \tag{18}$$

where $E_t = aE_1 + bE_2$, and a and b are the weighting coefficients. The parameters d and r are introduced in Eq. (18) so as to prevent outreaching the calculation range of a computer.

Initially, after an initial population of chromosomes is randomly generated, then the typical genetic algorithm evolves the population through the following three operators.

1) Selection Operator: This operator selects individuals (chromosomes) in the population for reproduction. The goodness of each individual depends on its fitness. Fitness may be determined by an objective function. The fitter the chromosome, the more times it is likely to be selected to be reproduced.

2) Crossover Operator: After two individuals are chosen from the population using the selection operator, the crossover operator randomly chooses a crossover site along the bit strings and exchanges the subsequences before and after that crossover site between the two individuals to create two offspring. For example, the strings *000000* and *111111* could be crossed over after the second locus in each to produce the two offspring *110000* and *001111*. The two new offspring created from this mating are put into the next generation of the population. By recombining portions of good individuals, this process is likely to create even better individuals.

3) Mutation Operator: With some low probability, a portion of the new individuals will have some of their bits flipped. Mutation can occur at each bit position in a string with some probability, usually very small. Its purpose is to maintain diversity within the population and inhibit premature convergence.

To use a genetic algorithm, a solution to a given problem must be represented as a chromosome. Since the genetic algorithm, in this work, optimizes the antecedent parameters, each chromosome contains the antecedent parameters c_{ij} and s_{ij} which describe the fuzzy membership functions. The genetic algorithm then creates a population of solutions (chromosomes) and applies genetic operators such as selection, crossover and mutation to evolve the solutions in order to find the best one.

If one fixes some parameters of the fuzzy inference system by the genetic algorithm, the resulting fuzzy inference system is equivalent to a series expansion of some basis functions. This basis function expansion is linear in its adjustable parameters. Therefore, one can use the least-squares method to determine the remaining parameters. When some input-output pattern data for training are given, from Eq. (13) the consequent parameters are chosen such that the pattern data satisfy the following equation:

$$\mathbf{y} = \mathbf{Wq}\,, \tag{19}$$

where $\mathbf{y}$ is the output data, $\mathbf{q}$ is the parameter vector, and the matrix $\mathbf{W}$ includes the input data defined as, respectively

$$\mathbf{y} = \left[y^1 \; y^2 \; \cdots \; y^N \right]^T ,$$

$$\mathbf{q} = [q_{11} \cdots q_{n1} \cdots \cdots q_{1m} \cdots q_{nm} \; r_1 \cdots r_n]^T ,$$

$$\mathbf{W} = \left[\mathbf{w}^1 \; \mathbf{w}^2 \; \cdots \; \mathbf{w}^N \right]^T ,$$

$$\mathbf{w}^k = [\overline{w}_1 x_1 \cdots \overline{w}_n x_1 \cdots \cdots \overline{w}_1 x_m \cdots \overline{w}_n x_m \; \overline{w}_1 \cdots \overline{w}_n]^T , \quad k = 1, 2, \cdots, N.$$

N is the number of the input-output data pairs. The fuzzy logic outputs are represented by $N \times (m+1)n$-dimensional matrix with N rows equal to the number of data pairs and $(m+1)n$ columns equal to $(m+1)$ times the number of rules. In order to solve the parameter vector $\mathbf{q}$ in Eq. (19), the matrix $\mathbf{W}$ should be invertible but is not usually a square matrix. Therefore, one solves the vector using the pseudoinverse as follows:

$$\mathbf{q} = \left(\mathbf{W}^T \mathbf{W} \right)^{-1} \mathbf{W}^T \mathbf{y}\,. \tag{20}$$

The least-squares method is a one-pass regression procedure and is therefore much faster than the back-propagation algorithm and the genetic algorithm.

4. Application to the Pressurizer Water Level and Hot-Leg Flowrate Measurements

The proposed algorithm that the above-mentioned methods are combined is described in Fig. 3. The proposed algorithm was applied to the pressurizer water level and the hot-leg flowrate measurements. The input-output data was obtained from the simulation of the MARS code [10] for the load decrease transients. The four important control algorithms were written into the input of the MARS code; the steam generator level, control rod, steam dump and pressurizer pressure (heater and spray) controls. The noise is added to model the real data of the nuclear power plant. The noise is proportional to the maximum variation $s_{\max}$ of each signal and is chosen from a uniform distribution on the interval ($-0.05 s_{\max}$, $0.05 s_{\max}$). In all computer simulations, the wavelet denoising technique was applied to all measurement signals and the Daubechies wavelet function was used [4]. Figures 4 through 7 show the measured and denoised signals of the pressurizer water level, hot-leg flowrate, coolant average temperature, hot-leg temperature, cold-leg temperature, pressurizer pressure, and pressurizer temperature that were used in this work. The fuzzy neural network was trained using one tenth of the given data in the training stage and was verified using the remaining data in the verification stage.

4.1 Pressurizer Water Level Estimation

When the principal component analysis is applied, the input signals into the PCA are the coolant average temperature, hot-leg flowrate, and pressurizer temperature and pressure. They are heuristically chosen among the data that are considered to have some relationship with the pressurizer water level. The input signals into the fuzzy neural network are a total of four signals: the first, the delayed first, the second and the third feature components. The first and second feature components have almost all information for the input signals into the PCA (refer to Table 1). When the principal component analysis is not applied, the input signals into the fuzzy neural network are a total of four signals: the coolant average temperature, the delayed coolant average temperature, the pressurizer temperature and pressure. The coolant average temperature is almost proportional to the pressurizer water level (refer to Figs. 3 and 5). Irrespective of the PCA application, four input signals into the fuzzy neural network are used.

Table 2 shows almost the same result irrespective of the PCA application, which is because the pressurizer level has the direct relationship with the coolant average temperature. The proposed algorithm actually estimates the pressurizer water level using other signals as shown in Fig. 8. Also, as shown in Fig. 9, this algorithm shows good performance for the verification data that were not used in the training stage.

4.2 Hot-Leg Flowrate Estimation

When the principal component analysis is applied, the input signals into the PCA are the hot-leg and cold-leg temperatures, and pressurizer water level and pressure. They are heuristically chosen among the data that are considered to have some relationship with the hot-leg flowrate. The input signals into the fuzzy neural network are a total of four signals: the first, the delayed first, the second and the third feature components that are the same ones used in the pressurizer water level application. When the principal component analysis is not applied, the input signals into the fuzzy neural network are the hot-leg temperature, the delayed hot-leg temperature, the pressurizer water level and the cold-leg temperature. The first two feature components have almost all information for the input signals into the PCA (refer to Table 1).

As shown in Table 2, better results are obtained when the PCA is applied. Different from the pressurizer water level case, in case there does not exist almost one-to-one mapping between one of the inputs and the output, the application of the PCA provides better performance as well as the easy selection of the input signals. The proposed algorithm estimates well the hot-leg flowrate using other signals as shown in Figs. 10 and 11.

5. Conclusions

In this work, a fuzzy neural network algorithm was proposed to estimate the relevant signals using other signals that are simply and heuristically selected. The input signals into the fuzzy neural network are preprocessed by the wavelet denoising method and principal component analysis. The input signals into the fuzzy neural network can be easily selected by the PCA and the first three feature components are used as its input signals. Also, the delayed first feature component is used to describe the sequential signal. In case there does not exist almost one-to-one mapping between one of the inputs and the output, the application of the PCA provides better performance as well as the easy selection of the input signals.

The proposed algorithm actually estimates the relevant signals using other signals. This algorithm can be used to monitor the sensor failure.

References

1. Michel Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi, Wavelet Toolbox User's Guide, MathWorks, Natick, MA (1996).
2. X. Z. Wang and R. F. Li, "Combining Conceptual Clustering and Principal Component Analysis for State Space Based Process Montoring," Ind. Eng. Chem. Res., Vol. 38, No. 11, pp. 4345-4358 (1999).
3. Junghui Chen and Jialin Liu, "Mixture Principal Component Analysis Models for Process Monitoring," Ind. Eng. Chem. Res., Vol. 38, No. 4, pp. 1478-1488 (1999).
4. C. Sidney Burrus, Ramesh A. Gopinath, and Haitai Guo, Introduction to Wavelets and Wavelet Transforms, Prentice-Hall, London (1998).
5. Pengju Kang, David Birtwhistle, and Kame Khouzam, "Transient Signal Analysis and Classification for Condition Monitoring of Power Switching Equipment Using Wavelet Transform and Artificial Neutral Network, 1998 2nd Intl. Conf. Knowledge-Based Intelligent Electronic Systems, 21-23 April 1998, Adelaide, Australia.
6. T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and Its Applications to Modeling and Control,” IEEE Trans. System, Man, Cybern., Vol. 1, pp. 116-132 (1985).
7. Man Gyun Na, Neuro-Fuzzy Control Applications in Pressurized Water Reactors: in Da Ruan (ed) Fuzzy Systems and Soft Computing in Nuclear Engineering, Springer-Verlag, Berlin Heidelberg New York, pp.172-207 (1999).
8. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, Massachusetts (1989).
9. M. Mitchell, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, Massachusetts (1996).
10. Won-Jae Lee, Bub-Don Chung, Jae-Jun Jeong, Kwi-Seok Ha, Moon-Kyu Hwang, Improved Features of MARS 1.4 and Verification, Korea Atomic Energy Research Institute, KAERI/TR-1386-99 (1999).

Table 1. Relative Information of Each Feature Component.

Feature Component	Pressurizer Water Level	Hot-Leg Flowrate
1st	63.0238	59.2284
2nd	30.4499	40.2783
3rd	6.5260	0.4854
4th	0.0002	0.0079

Table 2. Total Squared Error and Maximum Error (after 100 Generations Training).

		Pressurizer Water Level		Hot-Leg Flowrate	
PCA Application		Yes	No	Yes	No
Training data	Total squared error	0.4551	0.5695	98.6790	400.3078
	Maximum error	0.1205	0.1592	2.4351	5.9469
Verification Data	Total squared errors	4.2135	5.1379	907.1308	3630.3665
	Maximum error	0.1547	0.1742	2.4492	6.2430

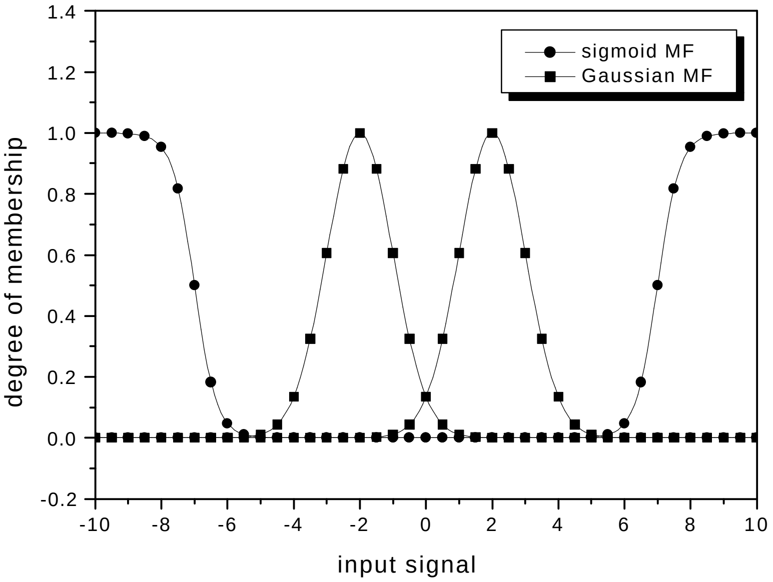

Fig. 1. Membership Functions.

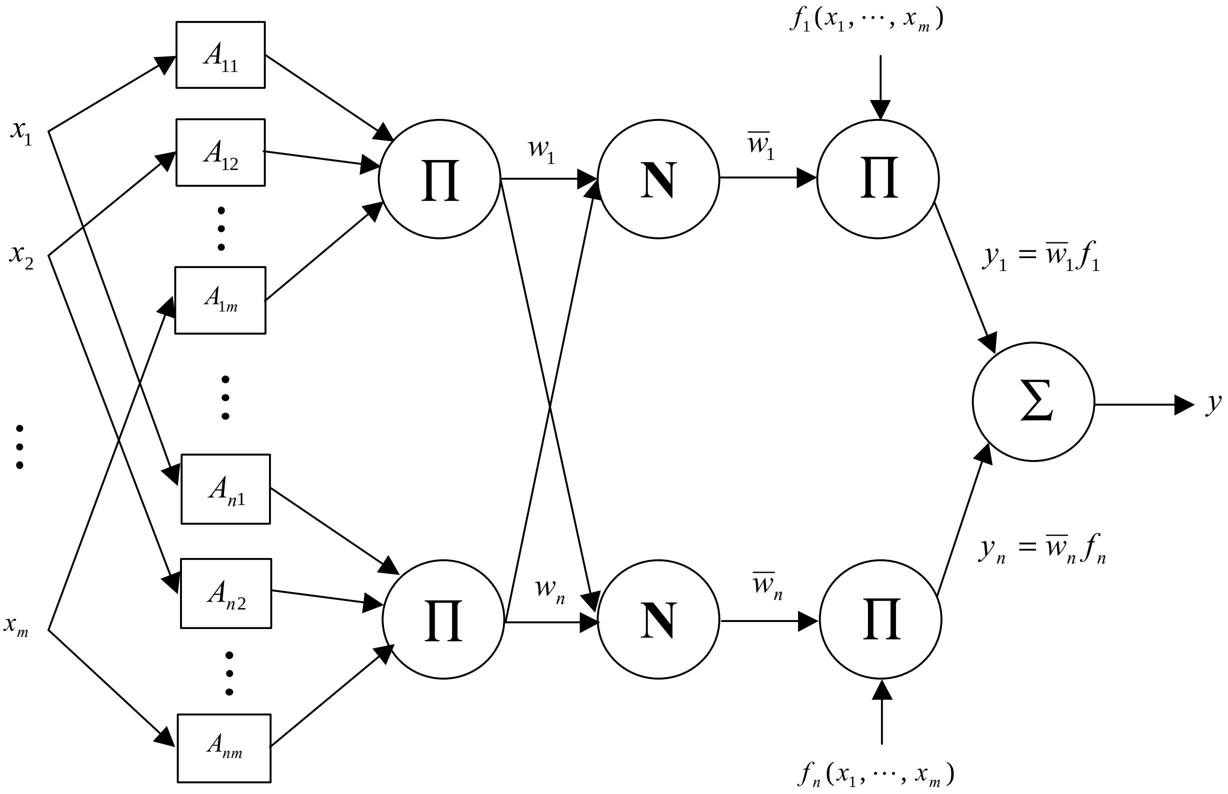

Fig. 2. Fuzzy Neural Network.

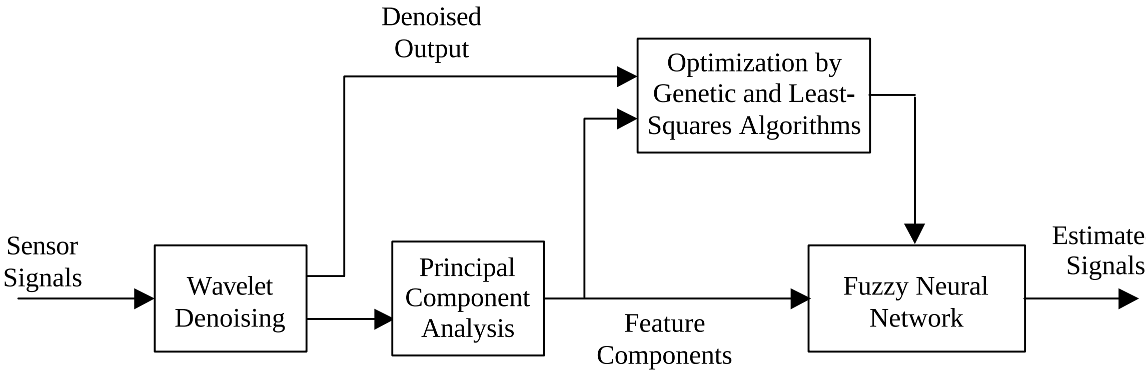

Fig. 3. Diagram of the Proposed Sensor Signal Estimation Algorithm.

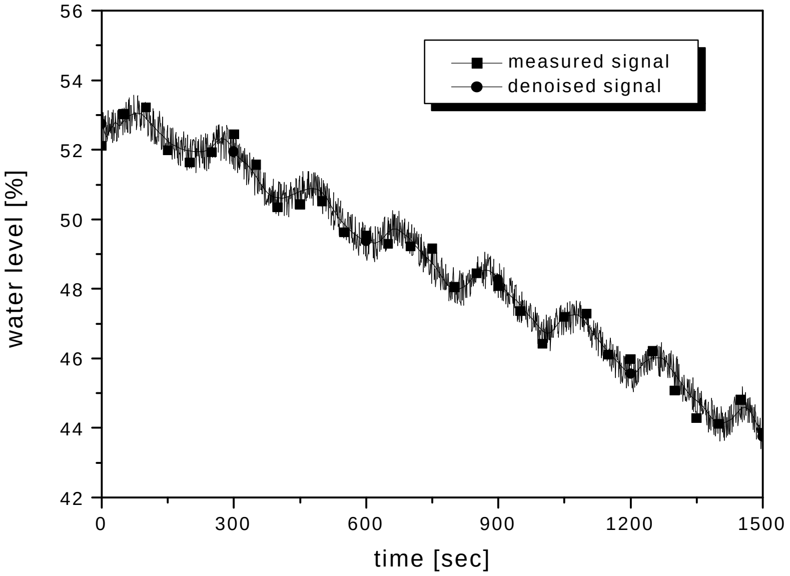

Fig. 4. Wavelet Denoising of the Pressurizer Water Level.

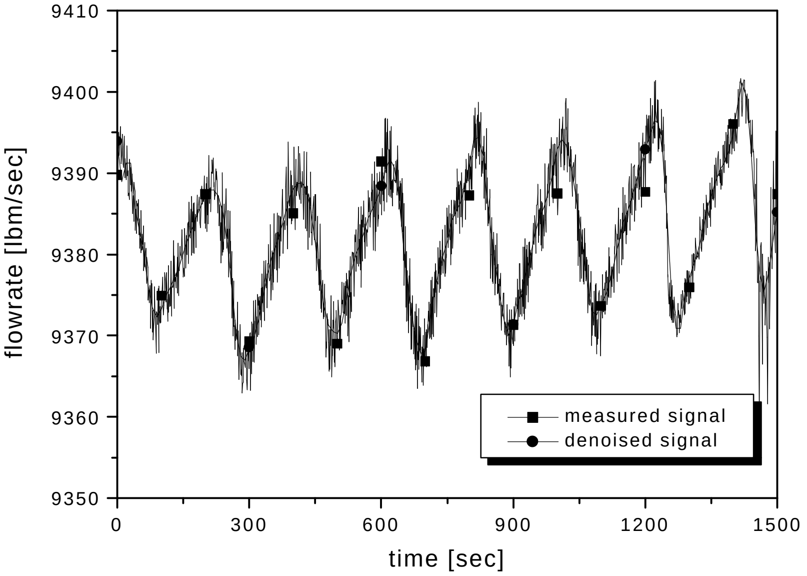

Fig. 5. Wavelet Denoising of the Hot-Leg Flowrate.

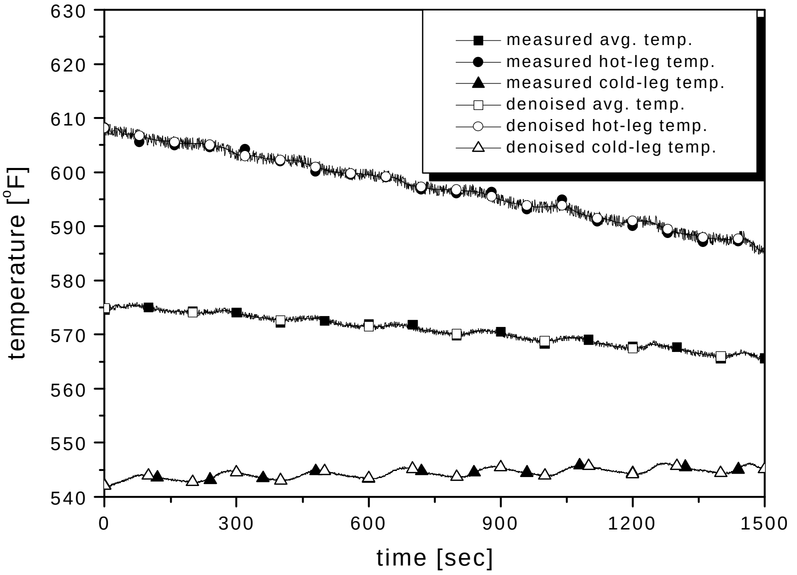

Fig. 6. Wavelet Denoising of the Coolant Average, Hot-Leg, and Cold-Leg Temperatures.

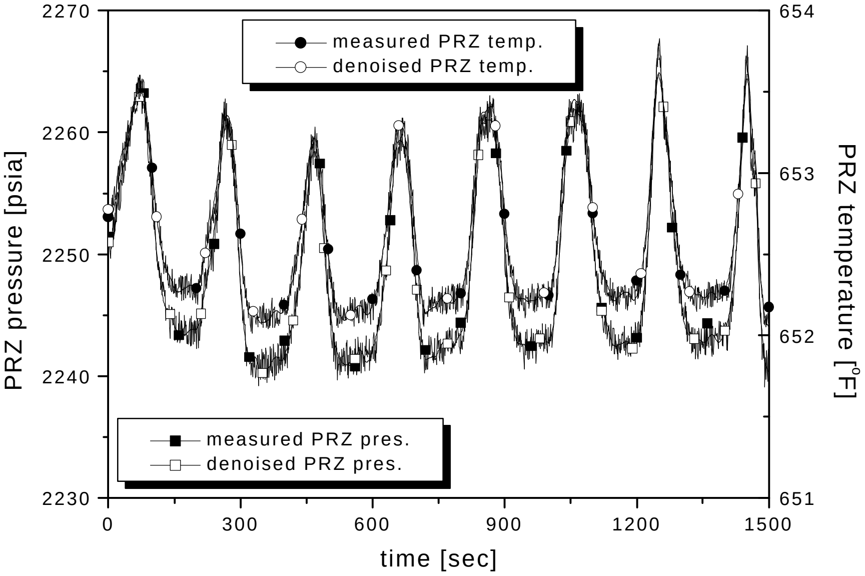

Fig. 7. Wavelet Denoising of the Pressurizer Pressure and Temperature.

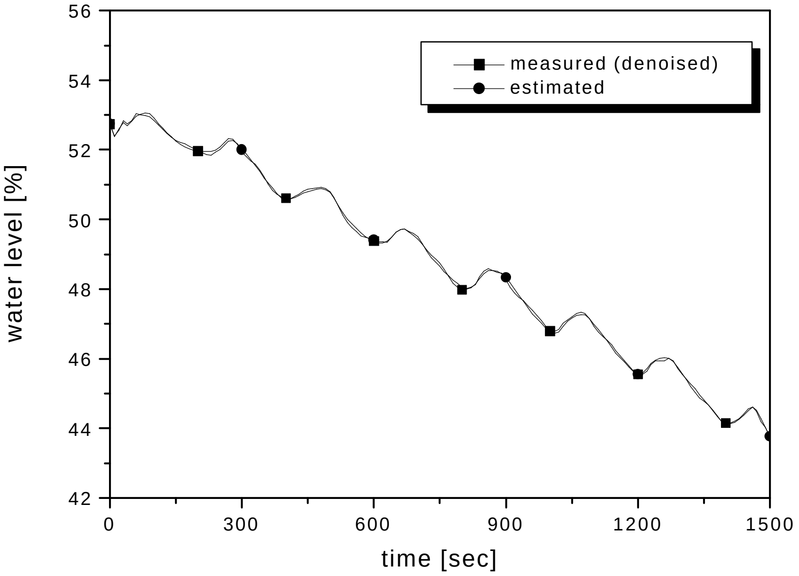

Fig. 8. Estimation of the Pressurizer Water Level (using the training data).

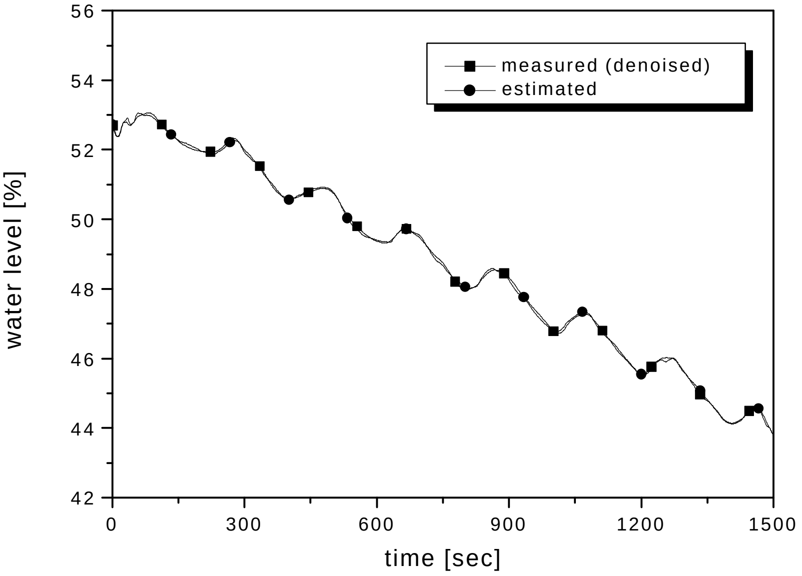

Fig. 9. Estimation of the Pressurizer Water Level
(using the verification data that was not used in the training stage).

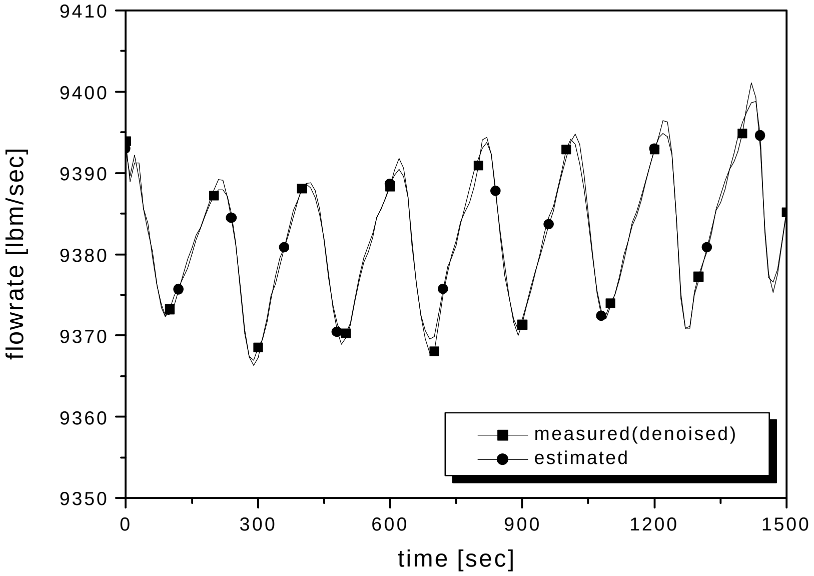

Fig. 10. Estimation of the Hot-Leg Flowrate (using the training data).

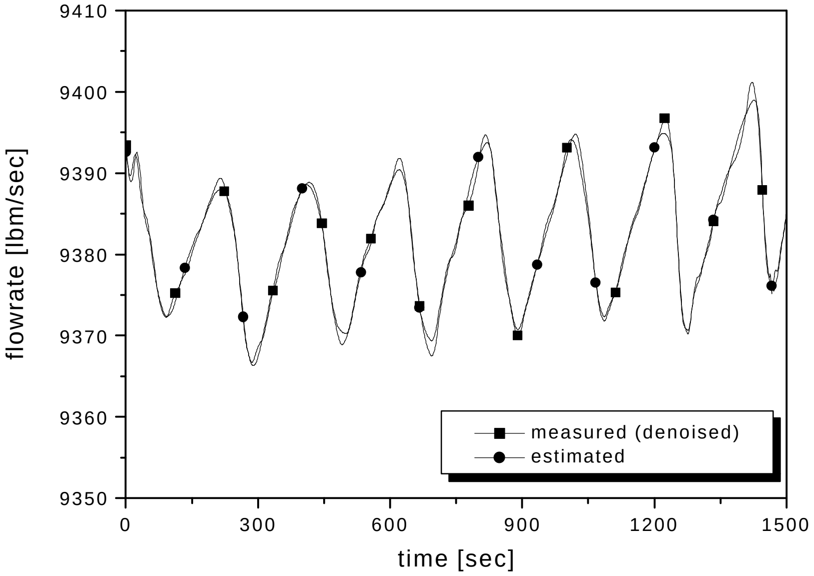

Fig. 11. Estimation of the Hot-Leg Flowrate
(using the verification data that was not used in the training stage).