Journal of the Korean Nuclear Society
Volume 33, Number 5, pp.483~497, October, 2001

A Neuro-Fuzzy Inference System for Sensor Failure Detection Using Wavelet Denoising, PCA and SPRT

Man Gyun Na
Chosun University
375 Seosuk-dong, Dong-gu, Kwangju 501-759, Korea
E-mail: magyna@chosun.ac.kr

(Received April 26, 2001)

Abstract

In this work, a neuro-fuzzy inference system combined with the wavelet denoising, PCA (principal component analysis) and SPRT (sequential probability ratio test) methods is developed to detect the relevant sensor failure using other sensor signals. The wavelet denoising technique is applied to remove noise components in input signals into the neuro-fuzzy system. The PCA is used to reduce the dimension of an input space without losing a significant amount of information. The PCA makes easy the selection of the input signals into the neuro-fuzzy system. Also, a lower dimensional input space usually reduces the time necessary to train a neuro-fuzzy system. The parameters of the neuro-fuzzy inference system which estimates the relevant sensor signal are optimized by a genetic algorithm and a least-squares algorithm. The residuals between the estimated signals and the measured signals are used to detect whether the sensors are failed or not. The SPRT is used in this failure detection algorithm. The proposed sensor-monitoring algorithm was verified through applications to the pressurizer water level and the hot-leg flowrate sensors in pressurized water reactors.

Key Words : neuro-fuzzy inference system, principal component analysis, sequential probability ratio test, wavelet denoising, sensor failure detection

1. Introduction

In nuclear power plants, measurement outputs from many different channels are used in control and safety critical systems and for plant state identification. Therefore, these outputs must be validated to increase the reliability of operator decisions and automatic plant operations. Sensor validation can be done through accurate mathematical modeling and computer coding of a process which are usually very difficult. Also, traditional methods for sensor validation can involve periodic instrument calibrations. These calibrations are expensive in labor and process downtime. Many periodic sensor calibration methods require the process to be shut down, the instrument taken out of service, and the instrument loaded and calibrated. These methods

can induce equipment failure and incorrect calibrations due to adjustments made under nonservice conditions. Recently, the neural networks have been used for this sensor monitoring as a powerful tool [1-5]. Through training, the neural network is very good at phenomenal nonlinear function approximation and pattern recognition, especially when expert 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 neuro-fuzzy inference system can be difficult since the subtle differences may occur between different transients. Therefore, it is necessary to preprocess the transient signals.

A wavelet denoising technique is applied to remove noise components in the input signals on the neuro-fuzzy inference system. 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 can miss, aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity [6]. The dimension of the input signals to a neuro-fuzzy inference system had better be reduced to save the time necessary to train the neuro-fuzzy inference system. Principal component analysis (PCA) [7-8] 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. Also, the PCA method makes easy the selection of the input to the neuro-fuzzy inference system.

By using the input signals preprocessed by the wavelet denoising technique and the principal component analysis, a neuro-fuzzy inference system estimates the relevant signals. The neuro-fuzzy system parameters such as the membership functions and the connectives between layers in a neuro-fuzzy inference system will be optimized by a genetic algorithm and a least-squares algorithm.

An important problem in sensor monitoring is whether a sensor is decided to be failed or not after only one abnormal observation. It is sure that several measurements can give a reliable result. At every new sample, a new mean and a new variance may be computed and then, these quantities may be used to check if the sensor is failed or not. However, this procedure requires too many samples to obtain a meaningful mean and a meaningful variance and also, during the acquisition of the samples, a significant degradation of the process monitored may occur. Therefore, in this work the sequential probability ratio test (SPRT) [9] was used. The method can detect a failure using the degree of degradation and the continuous behavior of the sensor, without having to calculate a new mean and a new variance at each sample. The signal estimated by the neuro-fuzzy inference system is compared with the measured signal, and then the SPRT monitors the sensor using the residuals.

The proposed algorithm was applied to the sensor monitoring of the pressurizer water level and the hot-leg flowrate of pressurized water reactors [10].

2. Preprocessing of Sensor Signals

2.1. Wavelet Denoising

The denoising objective is to suppress the noise part of a noisy signal and to recover a denoised signal. 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 signal $f(t)$ be expressed as

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

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. The size of the subspace can generally be increased by changing the time scale of the scaling functions that is generated from the basic scaling function by scaling and translation expressed as $\varphi_{j,k}(t) = 2^{j/2}\,\varphi(2^j t-k)$[11].

By introducing 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 $\varphi_{j,k}(t)$, the important features of a signal can be better described. 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:

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

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

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

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

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 representations of the original signal [12]. Thus, 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 [11] which consists of low and high pass filters and by downsampling it. Downsampling means throwing away every second data point. This concept is described in Fig. 1.

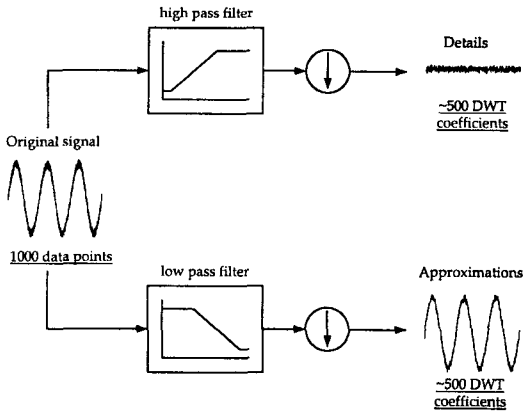

Fig. 1. Filtering and Downsampling

2.2. Principal Component Analysis (PCA)

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 and is used to reduce the dimension of an input space into the neuro-fuzzy inference system. A lower dimensional input space will reduce the time necessary to train a neuro-fuzzy inference system. The PCA method can be chosen as a method of preprocessing data to extract uncorrelated features from the data. The method also makes the transformed vectors orthogonal and uncorrelated.

Given a signal vector $\mathbf{x}$ of p dimensions, $\mathbf{x} = [x_1\ x_2\ \cdots\ x_p]^T$, its true mean and covariance matrix are replaced with the sample mean $\mathbf{m}$ and the sample covariance matrix $\mathbf{S}$ because they are seldom known. The eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_p$, 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:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \tag{5}$$

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:

$$\mathbf{z} = \mathbf{x}^T\mathbf{P} \tag{6}$$

where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_p]$.

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. Usually there is variability in the data with random noise, this variability is in most cases of no concern, and by transforming to a lower dimensional space this noise can sometimes be removed.

3. Neuro-fuzzy Inference System 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 neuro-fuzzy inference system or an adaptive fuzzy system. In a fuzzy inference system, the i-th rule can be described using the first-order Sugeno-Takagi type [13] as follows:

$$\text{If } x_1 \text{ is } A_{i1} \; AND \cdots AND \; x_m \text{ is } A_{im}, \text{ then } y_i \text{ is } A_{im}, \text{ then } y_i \text{ is } f_i(x_1, \cdots, x_m), \tag{7}$$

where

$x_1, \cdots, x_m$ = input variables to the neuro-fuzzy inference system (m = number of input variables),

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

y_i = output of the i-th rule,

$$f_i(x_1,\cdots,x_m) = \sum_{j=1}^{m} q_{ij}x_j + r_i, \tag{8}$$

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

r_i = bias of the j-th output,

n = number of rules.

In this work, the following Gaussian and sigmoid membership functions are used for each input variable:

$$A_{ij}(x_j) = \exp\left(-\frac{(x_j - c_{ij})^2}{2s_{ij}^2}\right), \tag{9}$$

$$A_{ij}(x_j) = \frac{1}{\exp\left(-\frac{x_j - c_{ij}}{s_{ij}}\right)+1}, \tag{10}$$

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 sigmoid membership function is used for the maximum and minimum center values in each input variable and the Gaussian membership function is used for other center values. The output of an arbitrary i-th rule, f_i, consists of the first-order polynomial of inputs as given in Eq. (8). The output of a fuzzy inference system with 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:

$$y = \sum_{i=1}^{n} \overline{w}_i f_i , \tag{11}$$

where

$$\overline{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i} , \tag{12}$$

$$w_i = \prod_{j=1}^{m} A_{ij}(x_j) . \tag{13}$$

3.2. Training of the Fuzzy Inference System

The neuro-fuzzy inference system is optimized 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 [14]. 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 [15].

To use a genetic algorithm, a solution to a given problem must be represented as a chromosome which can be thought of as a point in the search space of candidate solutions. 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. 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 [16-17]. In this work, a fitness 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, was suggested as follows:

$$F = \exp(-\mu_1 E_1 - \mu_2 E_2), \tag{14}$$

where μ_1 and μ_2 are the weighting coefficients, and E_1 and E_2 are overall sum of squared errors and maximum absolute error defined as

$$E_1 = \sum_{k=1}^{N} (y_d(k) - y(k))^2 , \tag{15}$$

$$E_2 = \max_k \{ |y_d(k) - y(k)| \} . \tag{16}$$

$y_d(k)$ and $y(k)$ denote the measured signal and the estimated signal, respectively.

If we fix 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, we can use the least-squares method to determine the remaining parameters. When a total of N input-output pattern data for training are given, from Eq. (11) the consequent parameters are chosen such that the pattern data satisfy the following equation:

$$\mathbf{y} = \mathbf{W}\mathbf{q} , \tag{17}$$

where **y** is the output data, **q** is the parameter vector, and the matrix includes the input data defined as, respectively

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

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

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

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

The neuro-fuzzy inference outputs are represented by the $N \times (m+1)n$ -dimensional matrix **W** and $(m+1)n$-dimensional parameter vector **q**. In order to solve the parameter vector **q** in Eq. (17), the matrix **W** should be invertible but is not usually a square matrix. Therefore, we solve the vector using the pseudo-inverse as follows:

$$\mathbf{q} = \left(\mathbf{W}^T\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{y}. \qquad (18)$$

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. Failure Detection Using SPRT

In sensor monitoring, at every new sample, a new mean and a new variance may be computed to check if the sensor is degraded or not. However, this procedure requires too many samples to obtain a meaningful mean and a meaningful variance. During the acquisition of the samples, a significant degradation of the process monitored may occur. So a method is required to detect a failure using the degree of failure and the continuous behavior of the sensor, without having to calculate a new mean and a new variance at each sample. The SPRT (Sequential Probability Ratio Test) which is a statistical model developed by Wald in 1945 [9] satisfies these requirements.

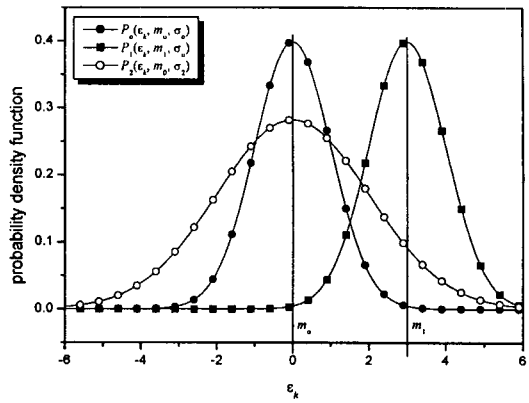

Fig. 2. Probability Density Functions of Residual Signals

The objective of sensor degradation detection is to detect the failure as soon as possible with a very small probability of making a wrong decision. In the application of sensor failure detection, the SPRT is dealing with the residual (difference between the sensor measurement and the sensor estimate). Normally the residual signals are randomly distributed, so they are nearly uncorrelated and have a Gaussian (normal) distribution $P_i(\varepsilon_k, m_i, \sigma_i)$, where ε_k is the residual signal at time k, and m_i and σ_i are the mean and the standard deviation under hypothesis i, respectively (refer to Fig. 2). The sensor failure can be stated in terms of a change in the mean m or a change in the variance σ^2. The basis for the SPRT lies in the likelihood ratio, which is given by

$$\gamma_k = \frac{P_1(\varepsilon_k \mid H_1)}{P_0(\varepsilon_k \mid H_0)}, \qquad (19)$$

where H_1 represents a hypothesis that the sensor is degraded and H_0 represents a hypothesis that the sensor is normal. The ratio is updated at every sampling step. If a set of samples x_i, i=1,2, ···, n,

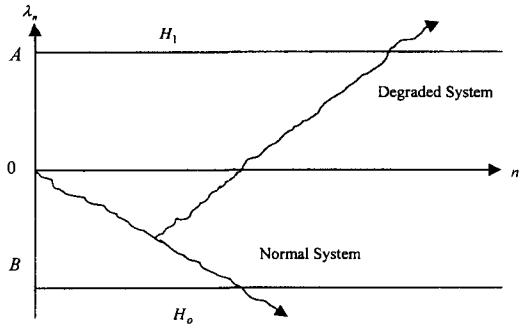

Fig. 3. Trajectories of LLR' s

is collected with a density function P describing each sample in the set, an overall likelihood ratio is given by

$$\gamma_n = \frac{P_1(\varepsilon_1 \mid H_1)\cdot P_1(\varepsilon_2 \mid H_1)\cdot P_1(\varepsilon_3 \mid H_1)\cdots P_1(\varepsilon_n \mid H_1)}{P_0(\varepsilon_1 \mid H_0)\cdot P_0(\varepsilon_2 \mid H_0)\cdot P_0(\varepsilon_3 \mid H_0)\cdots P_0(\varepsilon_n \mid H_0)} \quad (20)$$

By taking the logarithm of the foregoing equation and replacing the probability density functions in terms of residuals, means and variances, the log likelihood ratio (LLR, λ_n) can be written as following recurrent form:

$$\lambda_n = \lambda_{n-1} + \ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{(\varepsilon_n - m_0)^2}{2\sigma_0^2} - \frac{(\varepsilon_n - m_1)^2}{2\sigma_1^2}. \quad (21)$$

This is the form we use for deriving the sensor drift detection algorithm.

For a normal sensor, the likelihood ratio would decrease and eventually reach a specified bound A, a smaller value than zero. When the ratio reaches this bound, the decision is made that the sensor is normal, and the ratio is initialized by setting it equal to zero. For a degraded sensor the ratio would increase and eventually reach a specified bound B, a larger value than zero. When the ratio is equal to B, the decision is made that the sensor is degraded. Figure 3 shows these processes. The decision boundaries A and B are chosen by a false alarm probability α and a missed alarm probability β; $A = \ln\left(\frac{\beta}{1-\alpha}\right)$ and $B = \ln\left(\frac{1-\beta}{\alpha}\right)$.

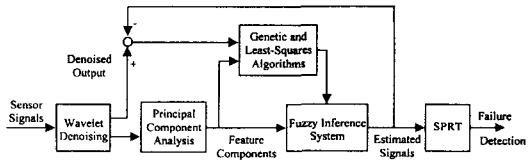

Fig. 4. Schematic Diagram of the Proposed Sensor-monitoring Algorithm

5. Applications

The proposed algorithm that the above-mentioned methods are combined is described in Fig. 4. The proposed algorithm was applied to the pressurizer water level and the hot-leg flowrate sensors. The input-output data were obtained for the load-decrease transients from the simulation of the MARS code [18] which is a unified version of COBRA/TF and RELAP5/MOD3. 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 input-output data consist of a total of 14 different signals. Noise is added to model the real data of the nuclear power plant. The noise is proportional to the maximum variation σ_{max} of each signal and is chosen from a uniform distribution on the interval $(-0.02\sigma_{max}, 0.02\sigma_{max})$. In all computer simulations, the wavelet denoising technique was applied to all measurement signals and the Daubechies wavelet function was used [19]. Figures 5 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

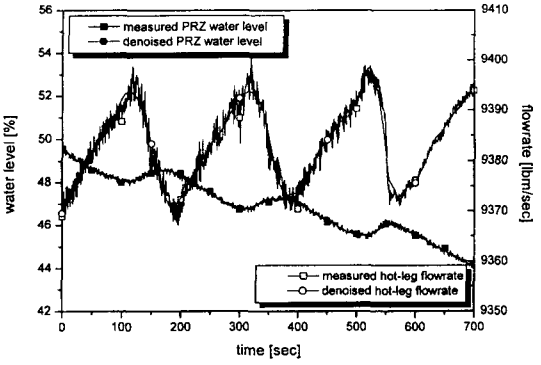

Fig. 5. Wavelet Denoising of the Pressurizer Water Level and Hot-leg Flowrate

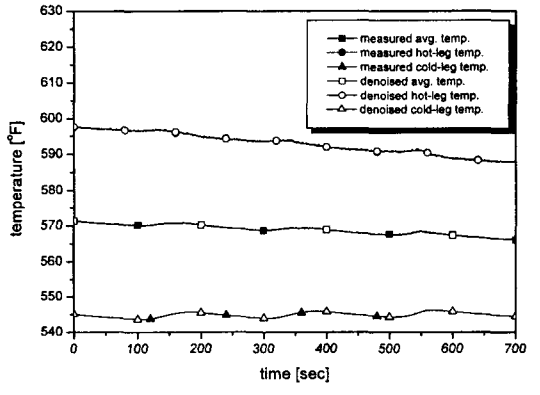

Fig. 6. Wavelet Denoising of the Coolant Average, Hot-leg, and Cold-leg Temperatures

used in this work and partially selected from a total of 14 different measured signals. Each signal consists of a total of 700 discrete time points where its sampling period is 1 sec. The neuro-fuzzy inference system was trained using one fifth of all the given data in the training stage and was verified using the remaining data in the verification stage. The false alarm probability α and the missed alarm probability β are chosen as 0.0001 and 0.1, respectively.

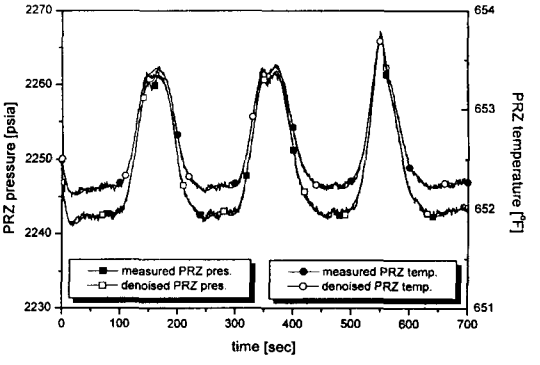

Fig. 7. Wavelet Denoising of the Pressurizer Pressure and Temperature

5.1. Pressurizer Water Level

In case that the principal component analysis is applied to the pressurizer water level monitoring, the input signals into the PCA are the coolant average temperature, hot-leg flowrate, and pressurizer temperature and pressure. We want to use a small number of inputs, if possible, because there is a smaller possibility that it is related with faulty and unreliable sensor signals. They are chosen through correlation analysis and several computer simulations among the signals that are considered to have a little close relationship with the pressurizer water level. The input signals into the neuro-fuzzy inference system 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). In case that the principal component analysis is not applied, the input signals into the neuro-fuzzy inference system 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

Table 1. Relative Information of Each Feature Component

Feature component	Pressurizer water level	Hot-leg flowrate
1st	65.9654	64.3666
2nd	31.3311	35.1428
3rd	2.7035	0.4894
4th	0.0000	0.0012

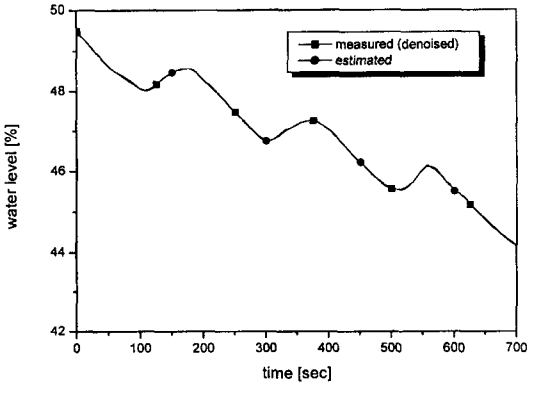

Fig. 8. Estimation of the Pressurizer Water Level (using the verification data that were not used in the training stage)

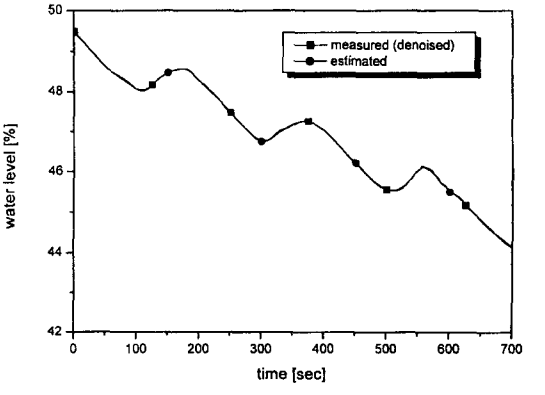

Fig. 9. Failure Detection of the Pressurizer Water Level Sensor (with uniform noise)

very closely related with the pressurizer water level (refer to Table 2). Irrespective of the PCA application, note that the same four input signals into the neuro-fuzzy inference system are used to compare their error levels between cases with PCA application and without PCA application. There are not clear differences in maximum relative errors and standard deviations of the training data and the verification data but the large difference of the total squared error between the training data and the verification data in Table 3 is due to the fact that the number of the verification data is four times more than that of the training data. Consequently, the difference of the results between the verification data and the training data is very small.

Table 3.a shows total squared relative percent error, maximum relative percent error, and the standard deviation of residuals in case the PCA method is applied or not. The result with PCA application is about three times better than that without PCA application. As shown in Fig. 8, through the verification simulation using the verification data that were not used in the training stage, it is known that the proposed algorithm actually estimates the pressurizer water level using other signals. Also, Table 3.b shows the results in case that Gaussian noise with mean zero and standard deviation $0.02\sigma_{max}$ is added. In this case, also, the result with PCA application is better than that without PCA application.

The pressurizer water level signal was purposely degraded in a degree of 3.0×10^{-5} of the measured values each time step from to verify the failure

Table 2. Correlation Coefficient Matrix for Gathered Signals

	SF	FF	SP	ST	NL	WL	HT	CT	HF	AT	PP	PL	RP	PT
steam flowrate(SF)	1.0000	0.9897	-0.9566	-0.9573	-0.0031	-0.0039	0.9851	-0.4858	0.3202	0.9541	-0.1417	0.9544	0.9946	-0.1441
feed flowrate(FF)	0.9897	1.0000	-0.9353	-0.9362	-0.0919	-0.0916	0.9843	-0.4388	-0.3475	0.9605	-0.0896	0.9599	0.9877	-0.0920
steam pres.(SP)	-0.9566	-0.9353	1.0000	0.9999	-0.0260	-0.0252	-0.9078	0.7168	0.0436	-0.8368	0.2918	-0.8317	-0.9564	0.2942
steam temp.(ST)	-0.9573	-0.9362	0.9999	1.0000	-0.0263	-0.0254	-0.9085	0.7152	0.0476	-0.8377	0.2932	-0.8328	-0.9568	0.2956
S/G water level(NL)	-0.0031	-0.0919	-0.0260	-0.0263	1.0000	0.9985	-0.0368	-0.1170	0.0570	-0.0560	-0.2076	-0.0495	-0.0207	-0.2076
S/G wide-range level(WL)	-0.0039	-0.0916	-0.0252	-0.0254	0.9985	1.0000	-0.0372	0.1153	0.0565	-0.0562	-0.2065	-0.0496	-0.0212	-0.2065
hot-leg temp.(HT)	0.9851	0.9843	-0.9078	-0.9085	-0.0368	-0.0372	1.0000	-0.3655	0.4000	0.9891	0.0180	0.9852	0.9904	0.0156
cold-leg temp.(CT)	-0.4858	-0.4388	0.7168	0.7152	-0.1170	-0.1153	-0.3655	1.0000	0.6178	-0.2247	0.5781	-0.2110	-0.4899	0.5796
hot-leg flowrate(HF)	-0.3202	-0.3475	0.0436	0.0476	0.0570	0.0565	-0.4000	-0.6178	1.0000	-0.5152	-0.2655	-0.5446	-0.2837	-0.2644
RCS average temp.(AT)	0.9541	0.9605	-0.8368	-0.8377	-0.0560	-0.0562	0.9891	0.2247	-0.5152	1.0000	0.1131	0.9978	0.9593	0.1109
PRZR pressure(PP)	-0.1417	-0.0896	0.2918	0.2932	-0.2076	-0.2065	0.0180	0.5781	0.2655	0.1131	1.0000	0.0727	-0.0727	0.9999
PRZR water level(PL)	0.9544	0.9599	-0.8317	-0.8328	-0.0495	-0.0496	0.9852	-0.2110	0.5446	0.9978	0.0727	1.0000	0.9540	0.0705
reactor power (RP)	0.9946	0.9877	-0.9564	-0.9568	-0.0207	-0.0212	0.9904	-0.4899	-0.2837	0.9593	-0.0727	0.9540	1.0000	-0.0751
PRZR temp. (PT)	-0.1441	-0.0920	0.2942	0.2956	-0.2076	-0.2065	0.0156	0.5796	-0.2644	0.1109	0.9999	0.0705	-0.0751	1.0000

detection algorithm. The failure detection algorithm detected its gradual degradation 84 sec after the beginning of the gradual degradation in case the PCA is applied (refer to Fig. 9). The trip flag '1' in Fig. 9 represents that the sensor is determined to be failed. In case the PCA was not applied, the sensor was determined to be failed 206 sec after the beginning of the gradual degradation (refer to Fig. 9). Therefore, the failure detection algorithm with PCA application detected the gradual degradation 122 sec faster than the failure detection algorithm without PCA application. This is because that the standard deviation of the estimation errors is used to determine whether it is

Table 3. Total Squared Relative Error, Maximum Relative Error, and Standard Deviation of Residuals (after 100 generations training)

(a) uniform noise

PCA application		Pressurizer water level		Hot-leg flowrate	
		Yes	No	Yes	No
Training data	Total squared relative error [%]	2.0578e-001	1.4529e+000	4.0909e-003	7.3373e-003
	Maximum relative error [%]	8.4232e-002	2.4787e-001	1.1987e-002	1.5024e-002
	standard deviation of residuals	1.7915e-002	4.7736e-002	5.0725e-001	6.7942e-001
Verification data	Total squared relative error [%]	8.4174e-001	5.7896e+000	1.6224e-002	2.8884e-002
	Maximum relative error [%]	8.6148e-002	2.4990e-001	1.2007e-002	1.4981e-002
	standard deviation of residuals	1.8126e-002	4.7697e-002	5.0553e-001	6.7457e-001

(b) Gaussian noise

PCA application		Pressurizer water level		Hot-leg flowrate	
		Yes	No	Yes	No
Training data	Total squared relative error [%]	4.0053e-001	1.2081e+000	3.7736e-003	1.4514e-002
	Maximum relative error [%]	1.0789e-001	2.1306e-001	1.2475e-002	2.4139e-002
	standard deviation of residuals	2.4929e-002	4.3333e-002	4.8734e-001	9.5576e-001
Verification data	Total squared relative error [%]	1.5661e+000	4.7116e+000	1.4982e-002	5.7579e-002
	Maximum relative error [%]	1.1317e-001	2.1298e-001	1.3436e-002	2.5134e-002
	standard deviation of residuals	2.4686e-002	4.2868e-002	4.8596e-001	9.5267-001

failed or not. In order to check the bias (mean value) degradation in Eq. (21), the equation can be converted into the following equation by substituting $\sigma_1 = \sigma_0$ and $\mu_0 = 0$ since normal residual signals have zero mean values:

$$\lambda_n = \lambda_{n-1} + \frac{m_1}{\sigma_0^2}\left(\varepsilon_n - \frac{m_1}{2}\right). \tag{22}$$

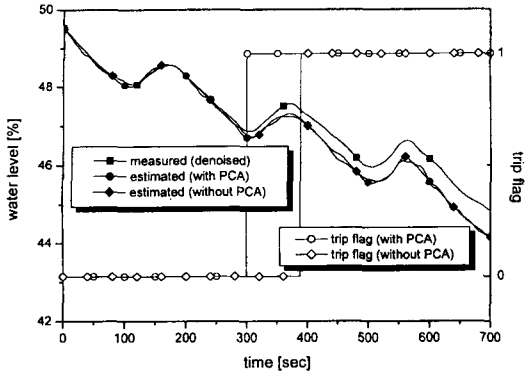

Fig. 10. Failure Detection of the Pressurizer Water Level Sensor (with Gaussian noise)

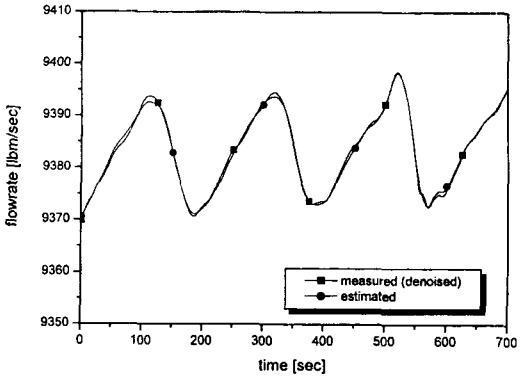

Fig. 11. Estimation of the Hot-leg Flowrate (using the verification data that were not used in the training stage)

As shown in Table 3, the standard deviation with PCA application is smaller than that without PCA application. Therefore, as known in the above equation, its log likelihood ratio (LLR) increases faster and reaches the specific bound B faster (refer to Fig. 3), which makes the detection time short.

Also, in case that Gaussian noise is added, the failure detection algorithm with PCA application detected the gradual degradation 89 sec faster than the failure detection algorithm without PCA application (refer to Fig. 10).

5.2. Hot-Leg Flowrate

In case that the principal component analysis is applied to this hot-leg flowrate monitoring, the input signals into the PCA are the hot-leg and cold-leg temperatures, and pressurizer water level and pressure. They are chosen through several computer simulations among the signals that are considered to have a little close relationship with the hot-leg flowrate (correlation analysis). The input signals into the neuro-fuzzy inference system 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. In case that the principal component analysis is not applied, the input signals into the neuro-fuzzy inference system 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 3.a, better results are obtained when the PCA is applied. 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 Fig. 11. Comparing with the relative errors of the pressurizer water level and the hot-leg flowrate cases, it is shown that the relative errors of the pressurizer water level is much larger that those of the hot-leg flowrate. However, this is because the magnitude of the hot-leg flowrate is much greater that that of the pressurizer water level and also the hot-leg flowrate change is relatively much smaller than the pressurizer water level change. Also, Table 3.b shows the results in case that Gaussian noise with mean zero and

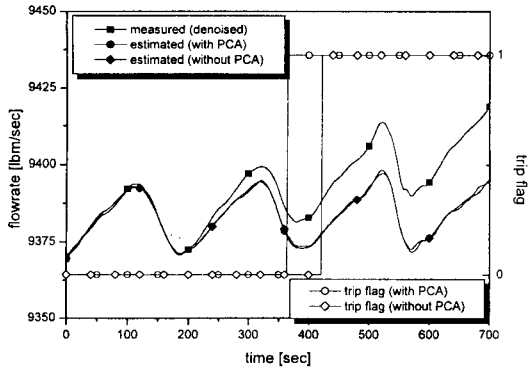

Fig. 12. Failure Detection of the Hot-leg Flowrate Sensor (with uniform noise)

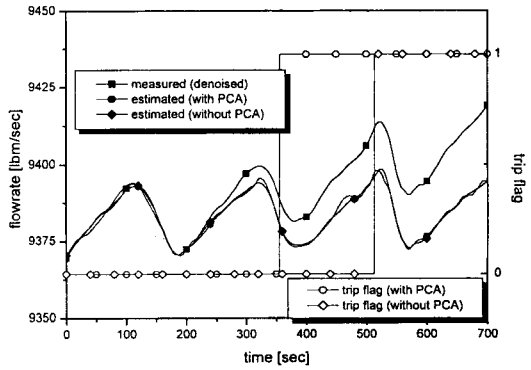

Fig. 13. Failure Detection of the Hot-leg Flowrate Sensor (with Gaussian noise)

standard deviation $0.02\sigma_{max}$ is added. In this case, also, the result with PCA application is better than that without PCA application.

The hot-leg flowrate signal was on purpose degraded in a degree of 5.0×10^{-6} of the measured values each time step from 200 sec to verify the failure detection algorithm. The failure detection algorithm detected its gradual degradation 166 sec after the beginning of the gradual degradation in case the PCA is applied (refer to Fig. 12). In case the PCA was not applied, the sensor was determined to be failed 223 sec after the beginning of the gradual degradation (refer to Fig. 12). Therefore, the failure detection algorithm with PCA application detected the gradual degradation 57 sec faster than the failure detection algorithm without PCA application. Also, in case that Gaussian noise is added, the failure detection algorithm with PCA application detected the gradual degradation 158 sec faster than the failure detection algorithm without PCA application (refer to Fig. 13).

6. Conclusions

In this work, a neuro-fuzzy inference system was proposed to estimate the relevant signals using other signals that are selected through several computer simulations among the collected signals considered to have a little close relationship with the output signal by simple correlation analysis. The input signals into the neuro-fuzzy inference system are preprocessed by the wavelet denoising method and principal component analysis. The input signals into the neuro-fuzzy inference system can easily be selected by the PCA. The first three feature components are used as its input signals and also the delayed first feature component is used to describe the sequential signal. The application of the PCA provides better performance (smaller estimation error) as well as the easy selection of the input signals. The effect of the PCA was larger according as the noise level increased, although it is needless to say the wavelet denoising effect. Also, it is easier to select the input signals to the neuro-fuzzy inference system by the PCA and the failure detection algorithm with PCA application detects the gradual degradation failure faster than the failure detection algorithm without PCA application since the standard deviation of estimation errors with PCA application is smaller than that without PCA application.

Acknowledgment

This work has been conducted under a Korean Nuclear Energy Research Initiative (K-NERI) project supported by the Korea Institute of Science & Technology Evaluation and Planning (KISTEP) which is funded by Korea Ministry of Science and Technology (Korea MOST).

References

1. A. Erbay and B. R. Upadhyaya, "A Personal Computer-Based On-Line Signal Validation System for Nuclear Power Plants," *Nucl. Technology*, vol. 119, no. 1, pp. 63-75, July (1997).
2. J. W. Hines, D. J. Wrest, R. E. Uhrig, "Signal Validation Using an Adaptive Neural Fuzzy Inference System," *Nucl. Technology*, vol. 119, no. 2, pp. 181-193, (1997).
3. R. M. Singer, K. C. Gross, J. P. Herzog, R. W. King, and S. W. Wegerich, "Model-Based Nuclear Plant Monitoring and Fault Detection: Theoretical Foundations," Proc. 9th Intl. Conf. on Intelligence Systems Applications to Power Systems, Seoul, Korea, (1997).
4. P. Fantoni, S. Figedy, and A. Racz, "A Neuro-Fuzzy Model Applied Full Range Signal Validation of PWR Nuclear Power Plant Data," FLINS-98, Antwerpen, Belgium, (1998).
5. J. W. Heins, Andrei V. Gribok, Ibrahim Attieh, and Robert E. Uhrig, "Regularization Method for Inferential Sensing in Nuclear Power Plants," in Da Ruan (ed) *Fuzzy Systems and Soft Computing in Nuclear Engineering*, Springer-Verlag, Berlin Heidelberg New York, pp.285-314, (1999).
6. Michel Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi, *Wavelet Toolbox User's Guide*, MathWorks, Natick, MA, (1996).
7. 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).
8. 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).
9. *A. Wald, Sequential Analysis*, John Wiley & Sons, New York, (1947).
10. Man Gyun Na, "Neuro-Fuzzy Inference System Combining Wavelet Denoising, PCA And SPRT for Sensor Monitoring," NPIC & HMIT 2000, Washington, D.C., Nov. 13-17, (2000).
11. C. Sidney Burrus, Ramesh A. Gopinath, and Haitai Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice-Hall, London, (1998).
12. 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.
13. 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).
14. 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).
15. Man Gyun Na, "A Fuzzy Neural Network Combining Wavelet Denoising and PCA for

Sensor Signal Estimation," *J. Korean Nucl.* Soc., vol. 32, no. 5, pp. 485-494, Oct. (2000).

16. D. E. Goldberg, *Genetic Algorithms in Search, Optimization*, and Machine Learning, Addison Wesley, Reading, Massachusetts, (1989).
17. M. Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, (1996).
18. Won-Jae Lee, Bub-Dong 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).
19. I. Daubechies, Ten Lectures on Wavelets, SIAM, (1992).