A Software Sensor for Feedwater Flow Monitoring

* Man Gyun Na\textsuperscript{1)}, Yoon Joon Lee \textsuperscript{2)}

\textsuperscript{1)} Department of Nuclear Engineering, Chosun University
375 Seosuk-dong, Dong-gu, Gwangju 501-759, Republic of Korea
magyna@chosun.ac.kr

\textsuperscript{2)} Department of Nuclear and Energy Engineering, Cheju National University
1 Ara-Il-dong, Jeju-do 690-756, Republic of Korea
yjlee@cheju.ac.kr

ABSTRACT

Venturi meters can decrease the thermal performance of nuclear power plants because the feedwater flowrate can be over-measured because of their fouling phenomena that make corrosion products accumulate in the feedwater flow meters due to long-term operation. Therefore, in this paper, a software sensor using a fuzzy inference system is developed in order to increase the thermal efficiency by estimating online the feedwater flowrate accurately. The fuzzy inference system to be used for black box modeling of the feedwater system is equipped with an automatic design algorithm that automates the selection of the input signals to the fuzzy inference system and its fuzzy rule generation including parameter optimization. The proposed algorithm was verified by using the numerical simulation data of MARS code for Kori-1 and also, the real nuclear plant data (YG-3). In the simulations using numerical simulation data and real plant data, the RMS error and the relative maximum error are so small that the proposed method can be applied successfully to validate and monitor the existing feedwater flow meters.

KEYWORDS

feedwater measurement, fuzzy inference system, software sensor, genetic algorithm

1. INTRODUCTION

It is very important to accurately measure the feedwater flowrate in order to monitor the thermal performance of a nuclear power plant (NPP). Venturi meters are used to measure the feedwater flowrate in most current pressurized water reactors (PWRs). These meters can induce measurement drift due to corrosion product buildup near the meter orifice because of long-term operation. This venturi meter fouling is known to be the most significant contributor to derating in PWRs. The
amount of derating ranges from 0.5% to 3%. Therefore, a lot of researchers have been interested in overcoming the inaccurate measurement problem of the feedwater flowrate (Kavaklioglu and Upadhyaya, 1994, Heo, 2000).

Due to the fouling phenomena of the venturi meter, the accuracy of the existing hardware sensors is not sufficient. Therefore, in this paper, a software sensor is developed to measure the feedwater flowrate by combining an empirical data-based model using a fuzzy inference system and other partial measurements of the reactor system. Software sensor design consists of building an estimate of some quantity of interest. The software sensor can be used either to replace a physical measurement or to validate an existing one.

Recently, many researchers have paid much attention to software sensors or inferential sensing, which use other readily available on-line measurements because these software sensors can either replace the hardware sensors or be used in parallel with them to provide redundancy and verify whether the hardware sensors are drifting (Choi and Park, 2001, Linko et al, 2001, Masson, 1999). When the process model for evaluating the process variables is a priori unknown or difficult to model like the steam generator system at hand, the problem can be stated in terms of black-box modeling. The fuzzy inference system is widely used for this black-box modeling. Therefore, in this work, a fuzzy inference system equipped with an automatic design algorithm is developed in order to increase the thermal efficiency by estimating on line the feedwater flowrate accurately. Particularly, the selection of the input signals to the fuzzy inference system and its rule generation are automated to optimally estimate the feedwater flowrate.

2. A SOFTWARE SENSOR USING A FUZZY INFERENCE SYSTEM

There are two types of approaches in developing software sensors. One is a method that estimates required parameters on the basis of a deterministic model and the other is the black-box modeling method that depends only on the measured values. Black-box modeling approaches such as artificial intelligence are more favored because they can model complicated processes which are difficult to be described by analytical and mechanistic methods. Therefore, black-box model approaches for building software sensors have widely been attempted. Also, recently, artificial intelligence such as fuzzy inference systems and artificial neural networks has been paid much attention from many researchers because artificial intelligence can model complex nonlinear systems easily (Choi and Park, 2001, Linko et al, 2001, Masson, 1999).

In this work, a Takagi-Sugeno (1985) type fuzzy inference system to be used to design a software sensor is applied to verify and monitor an existing venturi meter which measures the feedwater flowrate. Its \( i \)-th rule can be described as follows:

\[
\text{If } x_i \text{ is } A_{i1} \text{ AND } \cdots \text{ AND } x_m \text{ is } A_{im}, \text{ then } \hat{y}_i = f^i(x_1, \cdots, x_m),
\]  

(1)
where \( x_j \) is the input linguistic variable to the fuzzy inference system \( (j = 1, 2, \ldots, m) \), \( A_j \) the membership function of the \( j \)-th input variable for the antecedent of the \( i \)-th rule \( (i = 1, 2, \ldots, n) \), and \( \hat{y}_i \) the output of the \( i \)-th rule. Also, the rule output is of the following form:

\[
f^i(x_1, \cdots, x_m) = \sum_{j=1}^{m} q_{ij} x_j + r_i,
\]

where \( q_{ij} \) is the weighting value of the \( j \)-th input on the \( i \)-th rule output and \( r_i \) the bias of the \( i \)-th rule output. The output of a fuzzy inference system with \( n \) fuzzy rules is a weighted sum of the consequent of all the fuzzy rules. Therefore, the output of the software sensor is given by:

\[
\hat{y} = \sum_{i=1}^{n} \bar{w}^i f^i = w^T q,
\]

where \( \bar{w}^i = \frac{w^i}{\sum_{i=1}^{n} w^i} \), \( w^i = \prod_{j=1}^{m} A_j(x_j) \), \( q = [q_{i1} \cdots q_{in} \cdots q_{im} \cdots q_{mn} r_1 \cdots r_n]^T \), and \( w = [\bar{w}^1 x_1 \cdots \bar{w}^m x_m \cdots \bar{w}^n x_m \bar{w}^1 \cdots \bar{w}^n]^T \).

### 3. AUTOMATIC DESIGN OF A SOFTWARE SENSOR

#### 3.1. Automatic Structuring

The number of variables to be input to the fuzzy inference system has to be optimized for several reasons. First, irrelevant inputs will result in an unstable model. Thus, it becomes important to use only high information predictors. Secondly, since the generalization may degrade if colinearity is present among the variables, it is necessary to remove highly correlated variables. Finally, when building a black-box model with many input variables, a large number of observations are required to span the complete input space. The number of required observations grows exponentially with the number of input variables, which makes a dimension reduction essential to obtain a good model. In addition, since the optimum number of fuzzy inference rules depends on selected inputs and its number, it is required to select the optimum number of rules for selected inputs in order to prevent overfitting and underfitting problems (Na, 2003).

The genetic algorithms require a fitness function that assigns a score to each chromosome (candidate solution) in the current population. In this paper, a fitness function that evaluates the extent to which each candidate solution is suitable for the multiple objectives that minimize a maximum error and a root mean squared error along with the small number of input variables and the small number of rules, is suggested as follows:
Since genetic algorithms are computationally expensive, it is necessary to reduce the computation time of genetic algorithms. A modified genetic algorithm proposed in the literature (Na, 2003) is used in this work.

### 3.2. Parameter Optimization

Since the genetic algorithm requires much computational time if there are many parameters being involved, the genetic algorithm is combined with a least-squares algorithm. The objective of the genetic algorithm for a problem of fuzzy parameters optimization is to minimize the root mean squared errors and the maximum absolute error (refer to Eqs. (4) through (6)), which results in achieving the membership function optimization. If some parameters of the fuzzy inference system are fixed by the genetic algorithm, the resulting fuzzy inference system can be described as a series of expansions of some basis functions. This basis function expansion is linear in its adjustable parameters as shown in Eq. (3), \( \hat{y} = w^T q \), since \( w^T \) has been known by the genetic algorithm. Therefore, the least-squares method can be used to determine the remaining parameters. From a total number of \( N \) input-output training data pairs that are target values, the consequent parameters \( q \) are chosen to minimize the square of the difference between the target values \( y \) and the estimated values \( \hat{y} \):

\[
F = \exp(-\mu_1 E_1 - \mu_2 E_2 - \mu_3 E_3 - \mu_4 E_4),
\]

\[
E_1 = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (y_k - \hat{y}_k)^2},
\]

\[
E_2 = \max_k \left\{ |y_k - \hat{y}_k| \right\},
\]

\[
E_3 = N_{\text{input}},
\]

\[
E_4 = N_{\text{rule}}.
\]

\[
\hat{y} = w^T q,
\]

where \( y = [y_1 \ y_2 \ \cdots \ y_N]^T \).

The parameter vector \( q \) can be solved easily by using the pseudo-inverse of the matrix \( W \).

The process for automatically constructing the structure of the fuzzy inference system is described in Fig. 1. First, the input signals selection bits of the initial chromosomes are generated by using the correlation coefficient matrix to reduce the computational burden of the genetic algorithm and its rule number bits are allocated with more priority that their decoded value becomes a high number if the number of selected inputs is large. An outer loop for the selection stage of input sig-
nals and the number of fuzzy rules goes round until specific conditions are met as described by the fitness function. Also, in every selection stage of input signals and the number of fuzzy rules (outer loop), an inner loop for parameter optimization goes round repeatedly until specific conditions are met, too. In addition, in every selection stage of input signals and the number of fuzzy rules, a part of chromosomes with very low fitness is replaced by using the correlation analysis.

Fig. 1. Automatic design of a software sensor.

4. SENSOR FAULT DETECTION

The objective of sensor monitoring is to detect the failure as soon as possible with a very small probability of making a wrong decision. In this work, SPRT (Wald, 1945) that uses the residual are applied. Normally the residual signals are randomly distributed, so they are nearly uncorrelated and have a Gaussian (normal) distribution $P_i(\epsilon_k, m_i, \sigma_i)$, where $\epsilon_k$ is the residual signal at time $k$, and $m_i$ and $\sigma_i$ are the mean and the standard deviation under hypothesis $i$, respectively. The sensor failure can be stated in terms of a change in the mean $m$ or a change in the variance $\sigma^2$. If a set of samples $x_i$, $i=1,2,\ldots,n$, is collected with a density function $P$ describing each sample in the set, an overall likelihood ratio is given by

$$
\gamma_n = \frac{P_1(\epsilon_1 \mid H_1) \cdot P_2(\epsilon_2 \mid H_1) \cdot P_3(\epsilon_3 \mid H_1) \cdots P_n(\epsilon_n \mid H_1)}{P_0(\epsilon_1 \mid H_0) \cdot P_0(\epsilon_2 \mid H_0) \cdot P_0(\epsilon_3 \mid H_0) \cdots P_0(\epsilon_n \mid H_0)}, \quad (10)
$$
where $H_1$ represents a hypothesis that the sensor is degraded and $H_0$ represents a hypothesis that
the sensor is normal. By taking the logarithm of the above equation and replacing the probability
density functions in terms of residuals, means and variances, the log likelihood ratio (LLR, $\lambda_n$) can
be written as the following recurrent form:

$$\lambda_n = \lambda_{n-1} + \ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{(\varepsilon_n - m_0)^2}{2\sigma_0^2} - \frac{(\varepsilon_n - m_1)^2}{2\sigma_1^2}.$$  (11)

For a normal sensor, the log 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 sen-
sor is normal, and then the ratio is reinitialized 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. The decision boundaries $A$ and
$B$ are chosen by a false alarm probability $\alpha$ and a missed alarm probability $\beta$; $A = \ln\left(\frac{\beta}{1-\alpha}\right)$ and
$B = \ln\left(\frac{1-\beta}{\alpha}\right)$.

5. APPLICATION TO THE FEEDWATER FLOWRATE MEASUREMENT

The proposed method was verified through two application cases. First, the proposed method
was applied to the numerical simulation data of the load-decrease transients in Kori-1 using a
MARS code (Lee et al, 1999). Second, the proposed method was applied to the real plant starting
data of YG-3. The software sensor using a fuzzy inference system was automatically structured using
a half of all the acquired data (training data) in the training stage and was verified using the re-
mainning data (test data) in the test stage. The proposed input selection method is compared with the
existing principal component analysis (PCA) method (Wang and Li, 1999) and a heuristic method.
In the heuristic method, inputs are selected through a correlation analysis among possible input sig-
nals. PCA 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.

Table 1 summarizes the simulation results using the numerical simulation data. Figure 2 shows
simulation results in case the feedwater flowrate starts to be gradually degraded on purpose from
200 sec. The estimated feedwater flowrate is almost the same as the accurate feedwater flowrate.
This is a natural result because the estimated feedwater flowrate is not affected at all by using unaf-
feected signals. The gradual degradation is detected for the first time by the proposed method.

Table 2 summarizes the simulation results using the real plant data. Figure 3 shows simulation
results in case feedwater flowrate start to purposely be degraded from 20 hr. The estimated feedwa-
ter flowrate is almost the same as the accurate feedwater flowrate. The gradual degradation is de-
tected early by the proposed method.
The Seminar of JSPS-KOSEF Core University Program on Energy Science & Technology
Subtask CR04-2-2 “Advancement on Operation & Maintenance for Nuclear Power Plant”
November 1-2, 2004, Tohoku University, Sendai, Japan

Table 1. Results for the Numerical Simulation Data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Relative maximum error (%)</th>
<th>Root mean square error (%)</th>
<th>Maximum Fitness</th>
<th>Selected Inputs</th>
<th>Number of rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>0.13</td>
<td>0.05</td>
<td>0.9647</td>
<td>S/G steam flowrate, S/G pressure, S/G wide-range level, hot-leg temperature</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>1.70</td>
<td>0.10</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>0.39</td>
<td>0.13</td>
<td>0.9177</td>
<td>4 principal components</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>0.37</td>
<td>0.13</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heuristic Input Selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>0.18</td>
<td>0.06</td>
<td>0.9613</td>
<td>S/G steam flowrate, S/G narrow-range level, S/G temperature, reactor power</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>0.50</td>
<td>0.06</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. Estimation of feedwater flowrate signal in case it is gradually degraded (numerical simulation data).

Fig. 3. Estimation of feedwater flowrate signal in case that it is gradually degraded (real plant data).

Table 2. Results for the Real Nuclear Plant Data.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Relative maximum error (%)</th>
<th>Root mean square error (%)</th>
<th>Maximum Fitness</th>
<th>Selected Inputs</th>
<th>Number of rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>2.10</td>
<td>0.24</td>
<td>0.8921</td>
<td>hot-leg temperature, cold-leg temperature, PZR temperature, S/G temperature</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>1.88</td>
<td>0.24</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>6.72</td>
<td>1.43</td>
<td>0.6205</td>
<td>6 principal components</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>17.97</td>
<td>1.53</td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>Heuristic Input Selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Data</td>
<td>2.48</td>
<td>0.27</td>
<td>0.8807</td>
<td>S/G wide-range level, S/G narrow-range level, feedwater temperature, reactor power</td>
<td>4</td>
</tr>
<tr>
<td>Test Data</td>
<td>3.81</td>
<td>0.29</td>
<td>-</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. CONCLUSIONS

A software sensor using a fuzzy inference system that has an automatic design algorithm has been developed to validate and monitor the existing feedwater flowrate. The developed software sensor actually estimates the feedwater flowrate signal using other signals, which removes the effect of the fouling degradation of the venture meters. The proposed algorithm was verified by using the numerical simulation output of MARS code for Kori-1 and also, the real plant data of YG-3. Although the application to the real plant has larger error than that to the numerical simulation data, these errors are small enough and also, the results for the test data are almost the same as that for the training data. So the developed software sensor can be applied successfully to validate and monitor the existing feedwater flow meters.

REFERENCES


