Proceedings of the Korean Nuclear Spring Meeting
Gyeongju, Korea, May 2003

A Software Sensor Using a Black Box Modeling Method

Man Gyun Na, Dong Won Jung, Sun Ho Shin, Sun Mi Lee and Young Rok Sim[*]

Department of Nuclear Engineering, Chosun University
375 Seosuk-dong, Dong-gu, Gwangju 501-759, Republic of Korea

Abstract

In this paper, a software sensor using a black box modeling method has been developed to monitor existing hardware sensors. The black box modeling is accomplished by a fuzzy inference system that 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 software sensor was applied to monitoring the feedwater flowrate. The feedwater flowrate is measured by Venturi meters in most current pressurized water reactors (PWRs). These 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. The proposed software sensor was verified by using the numerical simulation data of MARS code for Kori nuclear power plant unit 1 and also, the real plant data of Yonggwang nuclear power plant unit 3. In a result using the numerical simulation data, the relative two-sigma errors are 0.11% and the relative maximum error is 0.22%. In another result using the real plant data, the relative two-sigma errors are 0.65% and the relative maximum error is 2.73%. These errors are so small that the proposed method can be applied successfully to validate and monitor the existing feedwater flow meters.

1. Introduction

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; Régnier et al. 1996; Linko, Luopa, and Zhu, 1997; Chéruy, 1997; Masson et al., 1999). Software sensor design consists of building an estimate of some quantity of interest. An estimate of a physical variable can be accomplished through mechanistic mathematical modeling or black-box modeling. That is, there are tow kinds of software sensors: model-based and data-based. 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 data-based 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 proposed to design hardware sensors that can replace a physical measurement or validate an existing one. That is, the selection of the input signals to the fuzzy inference system and its rule generation are automated to optimally estimate relevant physical variables.

Thermal reactor power is typically evaluated by secondary system calorimetric calculations that strongly depend on the accurate measurement of feedwater flowrate, and also, is directly proportional to the feedwater flowrate. Therefore, it is very important to accurately measure the feedwater flowrate in order to monitor the thermal performance of a nuclear power plant and a lot of researchers have been interested in overcoming the inaccurate measurement problem of the feedwater flowrate (Kavaklioglu and Upadhyaya, 1994; Heo, Choi, and Chang, 2000). 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 fouling increases the measured pressure drop across the meter, which in turn results in an overestimation of the feedwater flowrate. Therefore, in this paper, a developed software sensor is applied to measuring the feedwater flowrate by combining an empirical data based model using a fuzzy inference system and other partial measurements of the reactor system.

[*] Current Address: SAMCHANG ENTERPRISE CO., LTD

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, Luopa, and Zhu, 1997; Masson et al., 1999).

In this work, a fuzzy inference system will be used to design a software sensor. The fuzzy inference system combines linguistic and numerical information (mainly input-output pairs). Since the fuzzy inference system is constructed from fuzzy *if-then* rules, linguistic information can be directly incorporated and on the other hand, numerical information is incorporated by training the fuzzy inference system to match the target input-output pairs. The main advantages of the fuzzy inference system are the possibility of implementing *rule of thumb* experience, intuition, heuristics and the fact that it does not need a mathematical model of a process.

The inputs and outputs of the fuzzy inference system to be used as software sensors are real-valued variables. Therefore, in this work, instead of considering the Mamdani (1975) type fuzzy *if-then* rules in the form which requires time-consuming defuzzification calculation, a Takagi- Sugeno (1985) type fuzzy inference system is used where the i -th rule can be described as follows:

$$\text{If } x_1 \text{ is } A_{i1} \ AND \, \Lambda \ \ AND \ x_m \text{ is } A_{im}\text{, then } \hat{y}^i \text{ is } f^i(x_1, \Lambda\ , x_m)\,, \tag{1}$$

where

x_j = the input linguistic variable to the fuzzy inference system (j = 1, 2, ..., m),

A_{ij} = the membership function of the j -th input variable for the antecedent of the i -th rule (i = 1, 2, ..., n),

$\hat{y}^i$ = the output of the i -th rule.

Here, m is the number of input variables and n the number of fuzzy rules. There is generally no special restriction on the shape of membership functions. In this work, the symmetric Gaussian membership function is used to reduce the number of the parameters to be optimized. $f^i(x_1, \Lambda\ , x_m)$ is a polynomial in the input variables but it can be any function as long as it can appropriately describe the output of the fuzzy inference system within the fuzzy region specified by the antecedent of the rule. When the rule output is of the following form:

$$f^i(x_1,\Lambda\ ,x_m) = \sum_{j=1}^{m} q_{ij} x_j + r_i\,, \tag{2}$$

where

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

r_i = the bias of the i -th output,

the fuzzy inference system to be used in this work is called a first-order Takagi- Sugeno (1985) type fuzzy model since the output of an arbitrary i -th rule, f^i , is represented by the first-order polynomial of inputs as given in Eq. (2).

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 software sensor output estimated by the fuzzy inference system is given by:

$$\hat{y} = \sum_{i=1}^{n} \overline{w}^i f^i = \mathbf{w}^T \mathbf{q}\,, \tag{3}$$

where

$$\overline{w}^i = \frac{w^i}{\sum_{i=1}^{n} w^i}\,,$$

$$w^i = \prod_{j=1}^{m} A_{ij}(x_j),$$

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

$$\mathbf{w} = \left[\overline{w}^1 x_1 \Lambda \; \overline{w}^n x_1 \Lambda \; \Lambda \; \overline{w}^1 x_m \Lambda \; \overline{w}^n x_m \; \overline{w}^1 \Lambda \; \overline{w}^n\right]^T.$$

The superscript i in Eq. (3) indicates that the parameters are related to the i-th rule and the vector $\mathbf{q}$ is the consequent parameter vector that should be optimized by the training methods that will be described in next subsection.

3. Automatic Design of a Software Sensor

If the input signals to the fuzzy inference system to be used as a software sensor are selected and its fuzzy rules are generated, the design of a software sensor is completed. Therefore, a method that automates the input selection and the rule generation (rule number determination and its training method) will be described below.

3.1. Optimization of Input Signals and Fuzzy Rule Number

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 number of fuzzy inference rules depends on the number of selected inputs, it is required to select the optimum number of rules for selected inputs in order to prevent overfitting and underfitting problems (Na et al., 2003).

Genetic algorithms start from many points simultaneously climbing many peaks in parallel, and hence the probability of finding a false peak is reduced compared to the conventional methods that move from one point to another. Accordingly, genetic algorithms are less susceptible to being stuck at local minima than conventional search methods (Goldberg, 1989; Mitchell, 1996). Also, the genetic algorithm is the most useful method to solve optimization problems with multiple objectives. Therefore, it is proposed that a genetic algorithm will be applied to select proper input signals and to determine the optimum number of fuzzy rules. In genetic algorithm, the term *chromosome* refers to a candidate solution that minimizes a cost function, generally encoded as a bit string. As generation proceeds, populations of chromosomes are iteratively altered by biological mechanisms inspired by natural evolution such as selection, crossover and mutation.

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 such as small maximum error, small total squared error, the small number of input variables, and the small number of rules, is suggested as follows (Na et al., 2003):

$$F = \exp\left(-m_1 E_1 - m_2 E_2 - m_3 E_3 - m_4 E_4\right), \tag{4}$$

where m_1, m_2, m_3, and m_4 are the weighting coefficients, and E_1, E_2, E_3, and E_4 are the sum of squared errors, the maximum absolute error, the number of input variables, and the number of fuzzy inference rules, respectively, defined as

$$E_1 = \sum_{k=1}^{N} \left(y_k - \hat{y}_k\right)^2, \tag{5}$$

$$E_2 = \max_k \left\{\left|y_k - \hat{y}_k\right|\right\}, \tag{6}$$

$$E_3 = N_{input}, \tag{7}$$

$$E_4 = N_{rule}. \tag{8}$$

$y(k)$ and $\hat{y}(k)$ denote the actually measured signal and the signal estimated by a software sensor, respectively. Therefore, the optimization objectives of the genetic algorithm are to minimize the sum of squared errors, the maximum absolute error, the number of used input signals, and the number of fuzzy inference rules.

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 et al., 2003) will be used in this work. It is possible to reduce the computational time by lowering the probability of selecting the inputs that are almost not related to the output, though much related to some other inputs. Note that the correlation coefficient matrix of the original data set is equal to the covariance matrix of the data after the data have been standardized. This correlation matrix indicates how closely the variables (signals) linearly depend on each another. That is, the high specific (i, j) component of the correlation matrix means that the two corresponding (i-th and j-th) input variables are closely related to each other. The correlation coefficients between the input variables and the output variable are used to initialize the input signals selection bits of the chromosomes of the genetic algorithm (refer to Fig. 1). A chromosome is encoded as a bit string which consists of two parts of bits where one is related to the input signals selection and another is related to the fuzzy rule number. The input signals selection part is composed of the same bit number as the number of acquired usable input variables, and one '1' in this part represents that the corresponding input signal is selected and zero '0' represents that the corresponding input signal is not selected.

Explaining this process in detail as shown in Fig. 1, the correlation degree (dotted line) between the output (circle) bit and the selected input (triangle) bit has to be as large as possible and the correlation degrees (solid lines) between the selected input (triangle) bit and the possible inputs (cross) bits have to be as small as possible. To run a conventional genetic algorithm for input selection, each bit of the chromosomes is usually randomly assigned one or zero which represent that the corresponding input (bit) is selected or not, respectively. However, in this modified genetic algorithm, there is a high probability that the corresponding (triangle) bit is assigned one in case a correlation between the specific input (triangle) and the output is high and correlations between the specific input (triangle) and the possible inputs (cross) are low. On the contrary, there is a high probability that the corresponding (triangle) bit is assigned zero when the correlation between the specific input and the output is low and the correlations between the specific input (triangle) and the possible inputs (cross) are high. This helps to reduce the computational time by reducing the probability of selecting the inputs that are almost not related to the output, though much related to some other inputs. Although the correlation coefficient analysis is not able to determine the nonlinear relationship between the input variables and the output variables, since the correlation analysis is used once at first or intermittently, the genetic algorithm is able to sufficiently compensate for the nonlinear relationship during a few dozens of generations.

3.2. Fuzzy Rule Optimization

Conventional optimization algorithms including a back-propagation algorithm are susceptible to being stuck at local minima. Therefore, in this work the genetic algorithm that prevents the local minimum problem is used to optimize the fuzzy rules (membership parameters). However, 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. Thus, the genetic algorithm is used to learn the antecedent parameters (center position and sharpness of membership functions), and the least-squares algorithm is used to solve the consequent parameters q_{ij} and r_i (the polynomial coefficients of the consequent part).

Therefore, the genetic algorithm is used to select the input signals and the rule number of the fuzzy inference system and also, is used to optimize the fuzzy rules to be described in this subsection. The objective of the genetic algorithm as a problem of fuzzy parameters optimization is to minimize the overall sum of 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. Since this basis function expansion is linear in its adjustable parameters, the least-squares method can be used to determine the remaining parameters. From a total number of N input-output training data that are target values, the consequent parameters are chosen to minimize the difference between the target values and the estimated values:

$$J = \frac{1}{2}\sum_{k=1}^{N}\left(y_k - \hat{y}_k\right)^2 = \frac{1}{2}(\mathbf{y} - \hat{\mathbf{y}})^2 , \qquad (9)$$

where

$$\mathbf{y} = \left[y_1 \; y_2 \; \cdots \; y_N\right]^T ,$$

$$\hat{\mathbf{y}} = \left[\hat{y}_1 \; \hat{y}_2 \; \cdots \; \hat{y}_N\right]^T .$$

Using Eq. (3), the equation for minimizing the cost function is as follows:

$$\mathbf{y} = \mathbf{Wq}\,, \tag{10}$$

where

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

$$\mathbf{W} = \left[\mathbf{w}_1 \;\; \mathbf{w}_2 \;\Lambda \;\; \mathbf{w}_N\right]^T,$$

$$\mathbf{w}_k = \left[\overline{w}^1 x_1 \Lambda \;\; \overline{w}^n x_1 \Lambda \;\; \Lambda \;\; \overline{w}^1 x_m \Lambda \;\; \overline{w}^n x_m \;\; \overline{w}^1 \Lambda \;\; \overline{w}^n\right]^T, \quad k = 1, 2, \Lambda \;, N\,.$$

$\mathbf{y}$ is the output data vector, $\mathbf{q}$ is the parameter vector, and the matrix $\mathbf{W}$ includes the input data.

The output of the fuzzy inference system is represented by the $N \times (m+1)n$ -dimensional matrix $\mathbf{W}$ and the $(m+1)n$ -dimensional parameter vector $\mathbf{q}$. The parameter vector $\mathbf{q}$ in Eq. (10) is solved by using the pseudo-inverse of the matrix $\mathbf{W}$ as follows:

$$\mathbf{q} = \left(\mathbf{W}^T \mathbf{W}\right)^{-1} \mathbf{W}^T \mathbf{y}\,. \tag{11}$$

The process for automatically constructing the structure of the fuzzy inference system is described in Fig. 2. 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 of input signals and rule number goes round until specific conditions are met. Also, in every input signals and rule number selection step (outer loop), an inner loop for fuzzy rules optimization goes round repeatedly until specific conditions are met. In addition, in every input signals and rule number selection step a part of chromosomes with very low fitness is replaced by the correlation analysis.

4. Application to Feedwater Flowrate Estimation

The developed software sensor was applied to monitoring an existing venturi meter which measures the feedwater flowrate. 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 nuclear power plant unit 1 using a MARS code (Lee et al., 1999) that is a unified version of COBRA/TF and RELAP5/MOD3. Second, the proposed method was applied to the real plant starting data of Yonggwang nuclear power plant unit 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 remaining data (verification data) in the verification stage.

Figures 3 and 4 show the performance results of the proposed method for a data set acquired through numerical simulations of the computer code. Figure 3 shows the performance results for the training data from the numerical simulation. Fig. 3(a) shows the measured feed flowrate and its estimation error. Fig. 3(b) shows the histogram of its estimation error. The histogram resembles the Gaussian distribution. In this figure, it is shown that the relative maximum error compared with the rated value (469.01 kg/sec) is 0.15% and the relative two-sigma error 0.11%. Figure 4 shows the performance results for the verification data. Fig. 4(a) shows the measured feedwater flowrate and its estimation error which is small enough. Fig. 4(b) shows the histogram of its estimation error. In this figure, it is shown that the relative maximum error is 0.22% and the relative two-sigma error 0.11%. The results for the verification data are the same as that for the training data. Figure 5 shows simulation results in case feedwater flowrate is degraded on purpose. The estimated feedwater flowrate is almost the same as the accurate feedwater flowrate.

Table 1 summarizes the simulation results using the numerical simulation data. Twelve signals were acquired: S/G feedwater flowrate, S/G steam flowrate, S/G pressure, S/G temperature, S/G wide range level, S/G narrow range level, hot-leg temperature, cold-leg temperature, PZR pressure, PZR temperature, PZR water level and ex-core neutron detector signal. From the automatic design algorithm of a software sensor, three signals among these acquired 11 possible inputs except the S/G feedwater flowrate were selected as appropriate input signals to the fuzzy inference system for estimating the feedwater flowrate: steam generator steam flowrate, steam generator pressure, and steam generator narrow range water level. Also, the optimized number of fuzzy rules is 4. If we select three input signals heuristically, the selected input signals will be the steam generator steam flowrate, hot-leg temperature, and ex-core neutron detector signal through the correlation analysis. In the heuristic input selection method, these three inputs were selected through our intuition using the correlation analysis is and the number of fuzzy rules was selected to be 4 from the good performance results of many simulations. The proposed method has a maximum fitness value 0.8283 and the same relative two-sigma error 0.11% for both the training data and the verification data. On the other hand, the heuristic method has a maximum fitness value 0.7530 and the same relative two-sigma error 0.27% for both the training data and the verification data. The relative two-sigma error of the proposed method is about 145% better than that of the heuristic method. Also, the heuristic method has the same relative maximum error 0.38% for both the training data the verification data.

Figures 6 and 7 show the performance results of the proposed method for a data set acquired from a real nuclear power plant. Figure 6 shows the performance results for the training data. Fig. 6(a) shows the measured feed flowrate and its estimation error which is small enough. Fig. 6(b) shows the histogram of its estimation error. The histogram resembles the Gaussian distribution. In this figure, it is shown that the relative maximum error compared with the rated value (801.34 kg/sec) is 1.76% and the relative two-sigma error 0.65%. Figure 7 shows the performance results for the verification data. Fig. 7(a) shows the measured feed flowrate and its estimation error. Fig. 7(b) shows the histogram of its estimation error. In this figure, it is shown that the relative maximum error is 2.73% and the relative two-sigma error 0.65%. The results for the verification data are almost the same as that for the training data. Figure 8 shows simulation results in case feedwater flowrate is purposely degraded. The estimated feedwater flowrate is almost the same as the accurate feedwater flowrate.

Table 2 summarizes the simulation results using the real plant data. Thirteen signals were acquired: S/G feedwater flowrate, S/G steam flowrate, S/G pressure, S/G temperature, S/G wide range level, S/G narrow range level, hot-leg temperature, cold-leg temperature, PZR pressure, PZR temperature, PZR water level, feedwater temperature, ex-core neutron detector signal. From the automatic design algorithm of a software sensor, four signals among these acquired 12 possible inputs except the S/G feedwater flowrate were automatically selected as appropriate input signals to the fuzzy inference system for estimating the feedwater flowrate: hot-leg temperature, cold-leg temperature, PZR temperature, and steam generator temperature. Also, the optimized number of fuzzy rules is 3. The heuristic input selection method has the number of fuzzy rules 4, and uses three inputs through our intuition using the correlation analysis: hot-leg temperature, feedwater temperature, and reactor power. The proposed method has a maximum fitness value 0.7137 and the same relative two-sigma error 0.65% for both the training data and the verification data. On the other hand, the heuristic method has a maximum fitness value 0.6809 and the relative two-sigma error 0.82 for the training data and 0.81 for the verification data. Also, the heuristic method has the relative maximum error 2.37% for the training data and 2.79% for the verification data.

5. Conclusions

A software sensor using a fuzzy inference system that has an automatic design algorithm has been developed to validate and monitor the existing hardware sensors. The developed software sensor has been applied to actually estimate the feedwater flowrate signal that is very important to evaluate the reactor thermal power. The proposed method was verified by using the numerical simulation output of MARS code for Kori nuclear power plant unit 1 and also, the real plant data of Yonggwang nuclear power plant unit 3. In a simulation using the numerical simulation data, the relative two-sigma errors are equally 0.11% for both the training data and the verification data and the relative maximum error is 0.15% for the training data and 0.22% for the verification data. In another simulation using the real plant data, the relative two-sigma errors are equally 0.65% for both the training data and its verification data, and the relative maximum error is 1.76% for the training data and 2.73% for the verification data. These errors are small enough and also, the results for the verification data are almost the same as that for the training data. Therefore, the developed software sensor can be applied successfully to validate and monitor the existing feedwater flowmeters.

References

Chéruy, A., 1997. Software sensors in bioprocess engineering. Journal of Biotechnology 52(3), 193-199.

Choi, D.-J., Park, H. 2001. A hybrid artificial neural network as a software sensor for optimal control of a waste water treatment process. Water Research 35(16), 3959-3967.

Goldberg, D.E. 1989. Genetic algorithms in search, optimization, and machine learning. Addison Wesley, Reading, Massachusetts.

Heo, G., Choi, S.S., Chang, S.H. 2000. Thermal power estimation by fouling phenomena compensation using wavelet and principal component analysis. Nuclear Engineering and Design 199(1-2), 31-40.

Kavaklioglu, K., Upadhyaya, B.R. 1994. Monitoring feedwater flow rate and component thermal performance of pressurized water reactors by means of artificial neural networks. Nuclear Technology 107, 112-123.

Lee, W.-J., Chung, B.-D., Jeong, J.-J., Ha, K.-S., Hwang, M.-K. 1999. Improved features of MARS 1.4 and verification. Korea Atomic Energy Research Institute, KAERI/TR-1386-99.

Linko, S., Luopa, J., Zhu, Y.-H. 1997. Neural networks as ‘software sensors’ in enzyme production. Journal of Biotechnology 52(3), 257-266.

Mamdani, E. H., Assilian, S. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Machine Studies 7, 1-13.

Masson, M.H., Canu, S., Grandvalet, Y., Lynggaard-Jensen, A. 1999. Software sensor design based on empirical data. Ecological Modelling 120(2-3), 131-139.

Mitchell, M. 1996. An introduction to genetic algorithms. The MIT Press, Cambridge, Massachusetts.

Na, M.G., Sim, Y.R., Park, K.H., Lee, S.M., Jung, D.W., Shin, S.H., Upadhyaya, B.R., Zhao, K., Lu, B. 2003. Sensor monitoring using a fuzzy neural network with an automatic structure constructor. IEEE Trans. Nucl. Sci. 50(2), 241-250.

Régnier, N., Defaye, G., Caralp, L., Vidal, C. 1996. Software sensor based control of exothermic batch reactors. Chemical Engineering Science 51(23), 5125-5136.

Takagi, T., Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Systems, Man, Cybern. 1, 116-132.

Table 1. Results for the numerical simulation data.

		Relative maximum error(%)	Relative $2s$ error(%)	Maximum Fitness	Selected Inputs	Number of rules
Proposed Input Selection Method	Training Data	0.15	0.11	0.8283	S/G steam flowrate, S/G pressure, S/G NR water level	4
	Verification Data	0.22	0.11	-		
Heuristic Input Selection Method	Training Data	0.38	0.27	0.7530	S/G steam flowrate, hot-leg temperature, ex-core neutron detector signal	4
	Verification Data	0.38	0.27	-		

Table 2. Results for the real nuclear plant data.

		Relative maximum error(%)	Relative $2s$ error(%)	Maximum Fitness	Selected Inputs	Number of rules
Proposed Input Selection Method	Training Data	1.76	0.65	0.7137	hot-leg temperature, cold-leg temperature, PZR temperature, S/G temperature	3
	Verification Data	2.73	0.65	-		
Heuristic Input Selection Method	Training Data	2.37	0.82	0.6809	hot-leg temperature, feedwater temperature, ex-core neutron detector signal	4
	Verification Data	2.79	0.81	-		

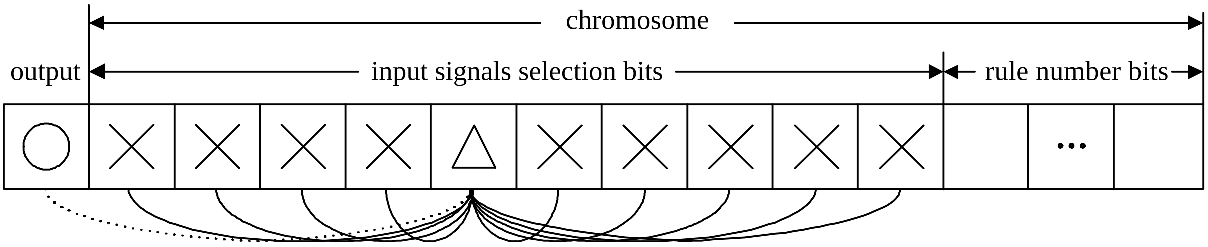


Fig. 1. The selection process of input signals and rule number.

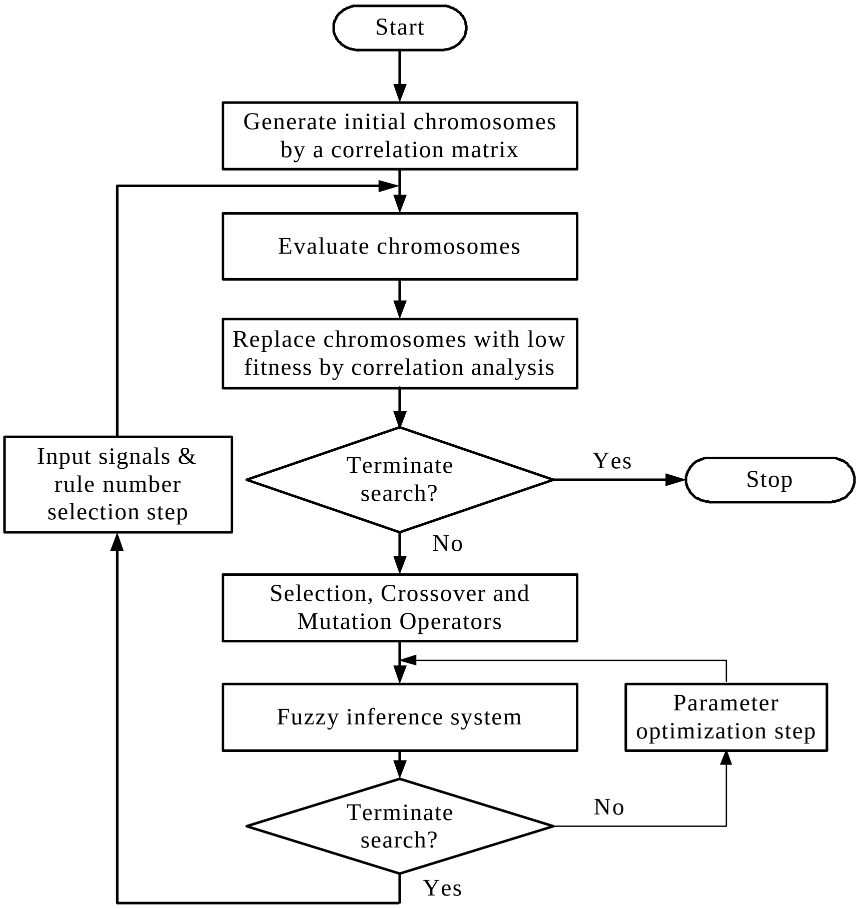
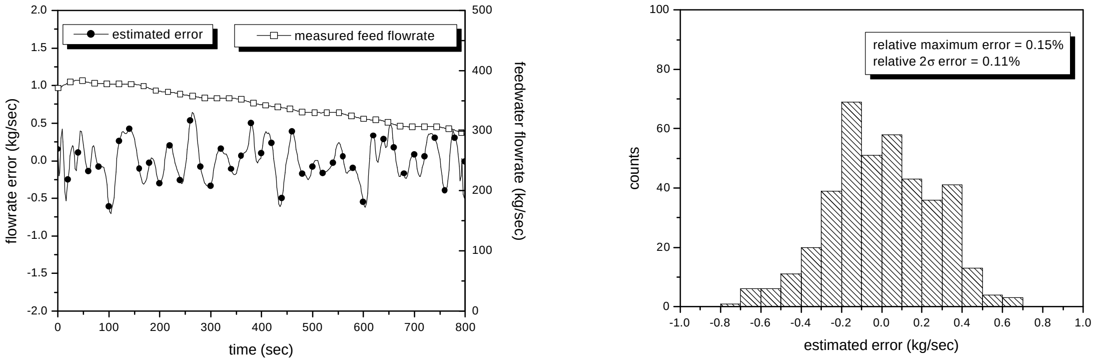
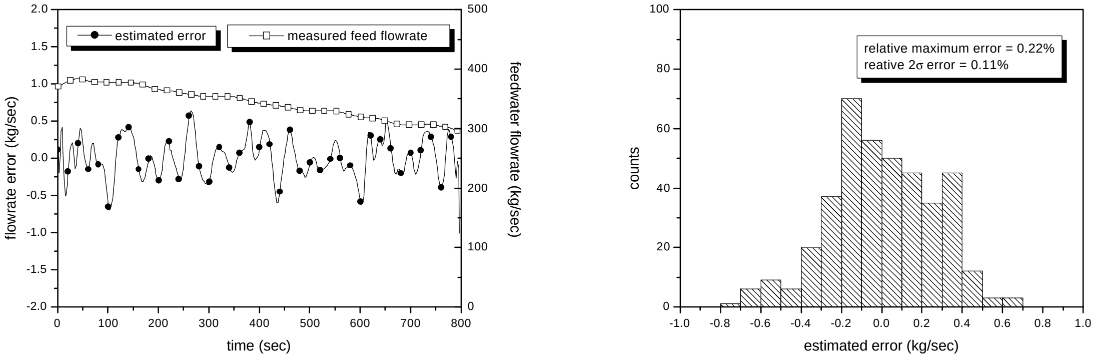

Start
Generate initial chromosomes by a correlation matrix
Evaluate chromosomes
Replace chromosomes with low fitness by correlation analysis
Terminate search?
Yes
Stop
No
Input signals & rule number selection step
Selection, Crossover and Mutation Operators
Fuzzy inference system
Parameter optimization step
Terminate search?
No
Yes

Fig. 2. A procedure for automatically constructing the structure of the fuzzy inference system.



(a) Feedwater flowrate error (b) Histogram of the feedwater flow estimation error

Fig. 3. Feedwater flowrate error and its histogram for the training data of a numerical simulation.



(a) Feedwater flowrate error (b) Histogram of the feedwater flow estimation error

Fig. 4. Feedwater flowrate error and its histogram for the verification data of a numerical simulation.

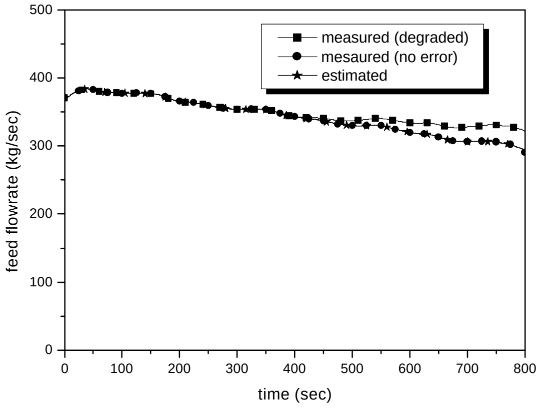
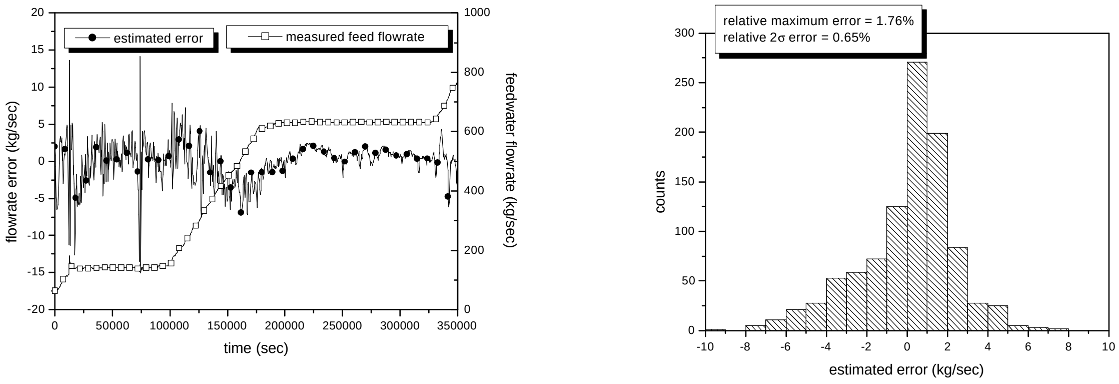
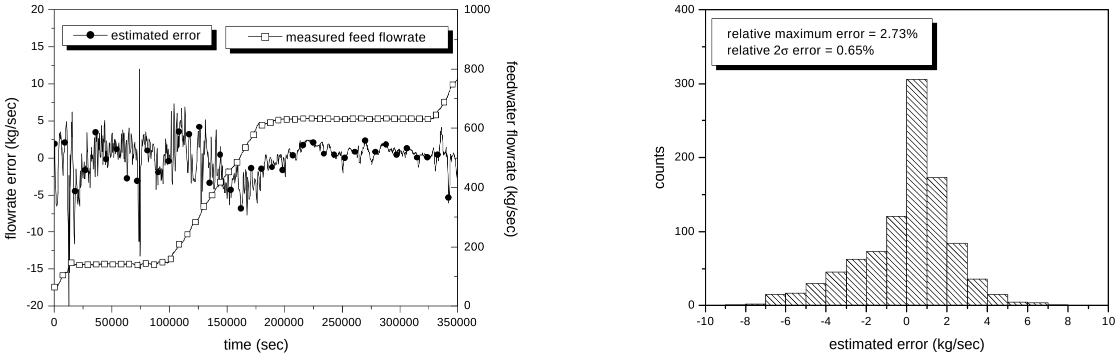


Fig. 5. Estimation of feedwater flowrate signal in case it is assumed that it is gradually degraded.



(a) Feedwater flowrate error (b) Histogram of the feedwater flow estimation error

Fig. 6. Feedwater flowrate error and its histogram for the training data of a real plant.



(a) Feedwater flowrate error (b) Histogram of the feedwater flow estimation error

Fig. 7. Feedwater flowrate error and its histogram for the verification data of a real plant.

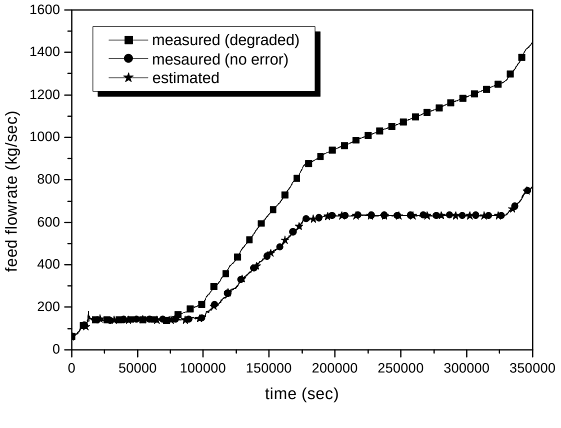


Fig. 8. Estimation of feedwater flowrate signal in case it is assumed that it is gradually degraded.