Adaptive Neurofuzzy Controller to Regulate UTSG Water Level in Nuclear Power Plants

Sudath R. Munasinghe, *Member, IEEE*, Min-Soeng Kim, *Student Member, IEEE*, and Ju-Jang Lee, *Senior Member, IEEE*

***Abstract*—A data-driven adaptive neurofuzzy controller is presented for the water-level control of U-tube steam generators in nuclear power plants. This neurofuzzy controller is capable of learning the control action principles from the data obtained using other methods of automatic or manual control. There are four inputs in the neurofuzzy system, yet only eighty fuzzy rules involved. Therefore, the fuzzy system is versatile and moderately compact. The versatility is due to the higher input space dimension that helps to learn more control principles. The compactness is due to the number of rules being not too many. A 10-h evaluation trial of the trained fuzzy controller demonstrated its capability in regulating the water level under random disturbances and reference level changes.**

***Index Terms*—Adaptive neurofuzzy system, nonminimum phase dynamics, Takagi–Sugeno fuzzy model, U-tube steam generator.**

NOMENCLATURE

p	power level (percent);
u	feedwater flow (kilograms per second);
v	steam flow (kilograms per second);
y	water level (millimeters);
r	reference water level (millimeters);
$e^l = r - y$	level error (millimeters);
$e^f = v - u$	flow error (kilograms per second);
v_p	rated steam flow at power p (kilograms per second).

I. INTRODUCTION

NUCLEAR POWER plants generate electricity by driving the armature coupled to a steam turbine. The steam is generated by the u-tube steam generator (UTSG). The water level of the UTSG should be maintained within safe limits. A too high of a water level produces wet steam that could damage the turbine blades; therefore, the turbine trips off. On the other hand, too low of a water level causes poor cooling of the nuclear reactor; therefore, the reactor trips off. In both cases, the power plant shuts down unintentionally. The water-level regulation of UTSG is a very difficult control problem, and it is one of the major reasons for unintended shutdowns of nuclear power plants. The difficulty arises due to reasons such as nonlinearity of dynamics, nonminimum phase dynamics (also known as reverse dynamics), and unreliable sensor feedback (at low power) [1]. Therefore, UTSG plants are always looked after and manually controlled by expert practitioners, whereas automatic control is considered only for trivial operations.

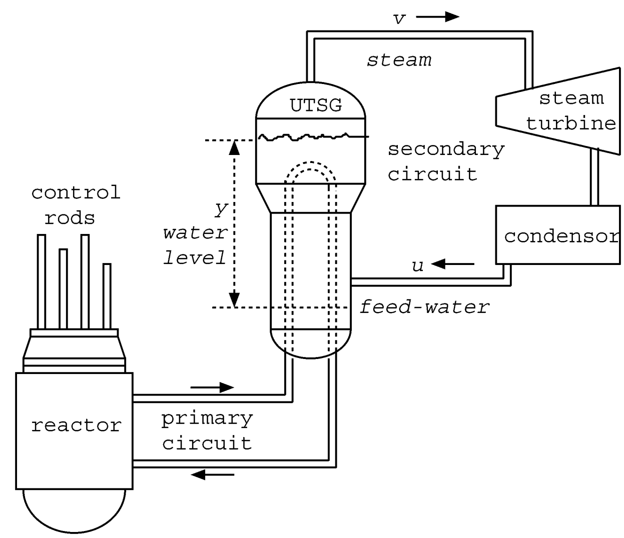

Fig. 1. Simplest schematic of a nuclear power plant with the u-tube steam generator.

Thermodynamics of UTSG makes it very difficult to model the water level theoretically. Even though it were accomplished, the theoretical models [2]–[5] are too complicated to be considered as candidates for control system design. Therefore, Irving [6] developed a simplified linear dynamic model, in which the model parameters change as the operating point changes. He also specified those parameters for five specific operating points. Irving's model is the most popular UTSG model in control research, and it assumes that the reverse dynamics of feed-water and steam to be identical, which has been extensively followed by many successive researchers later on [7]–[9]. Many attempts have been made to design controllers for the UTSG water level over the last two decades. Na [7] reported a PID control of UTSG water level, where he used a model predictive technique (based on standard Irving's model) to automatically tune the PID gains. Later on, he developed an adaptive predictive controller for UTSG [9]. Kothera [10] presented model predictive control of the UTSG water level using a further simplified UTSG model. Kim [11] argued that the nonminimum phase dynamics of feed-water and steam should not necessarily be identical and distinguished the two effects, introducing two more model parameters to the standard

Manuscript received August 24, 2003; revised October 10, 2004. This work was supported by the Ministry of Commerce, Industry, and Energy of Korea under Research Grant R-2003-0-035-0-00.

The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Dejeon 305-701, Korea (e-mail: rohan@odyssey.kaist.ac.kr).

Digital Object Identifier 10.1109/TNS.2004.842723

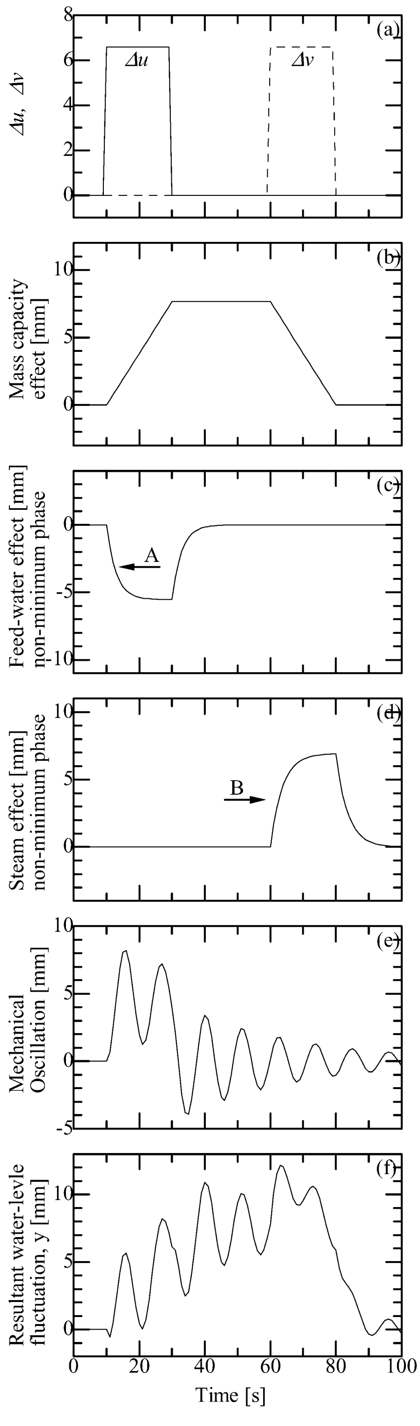

Fig. 2. UTSG dynamics at 50% of the rated power. All graphs show the deviation form its steady state value at the specified power level. Plant excitations in (a) $\Delta u = \Delta v = 6.6$ (kilograms per second) is 1% of the plant flow rates at 50% rated power.

TABLE I
UTSG Model Parameters

$p[\%]$	G_1	G_2	G_3	G_4	τ_1	τ_2	τ_3	T	v_p
5	0.058	7.704	9.63	0.181	41.9	38.72	48.4	119.6	57.4
15	0.058	3.568	4.45	0.226	26.3	17.20	21.5	60.5	180.8
30	0.058	1.464	1.83	0.310	3.6	43.40	4.5	17.7	381.7
50	0.058	0.840	1.05	0.215	34.8	2.88	3.6	14.2	660.0
100	0.058	0.376	0.47	0.105	28.6	2.72	3.4	11.7	1434.7

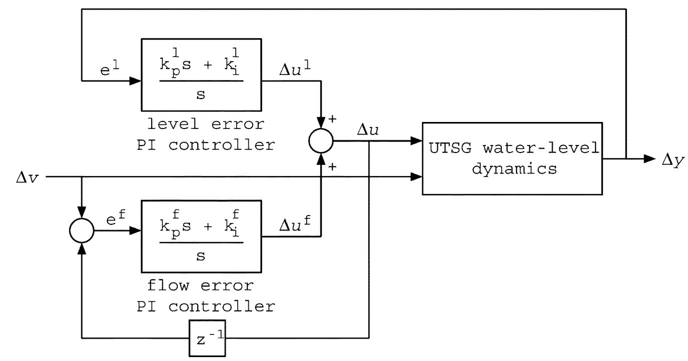

Fig. 3. Two PI controller system to regulate UTSG water level.

Irving's model. It appears that Kim's model is more general and could be reshaped to Irving's model merely by equating nonminimum phase parameters.

The difficulty of modeling and control of UTSG water level inspired researchers to investigate model-free (data-driven) techniques such as fuzzy and adaptive learning systems. Fuzzy reasoning [12]–[14] can be used to interpret uncertain incomplete data in order to make an intelligent guess of the desirable control action, which is exactly what is needed where there is no accurate model (therefore no proper controller either) of the process. The plant, the controller, or the combined closed-loop system can be modeled by a fuzzy system in terms of an *if-then* rule-set. For complex systems, the rules can be automatically generated by grid partition of the input product space or by identifying input space clusters [15], [16]. The *if-then* rules, in the form of Takagi–Sugeno (TS) fuzzy inference [17], actually model-specific local behaviors of the plant, and the weighted sum of the rule outputs, approximates the actual output of the plant. Therefore, a TS fuzzy systems is a linear multimodel representation of a complex, nonlinear plant [18], where plant dynamics can be revealed by inspecting the characteristics of the individual TS rules. The transparency and simplicity of the TS rule base can be improved by further refining the rule base by combining similar rules together and optimizing the important rules [19]–[22]. Cho [8] developed a fuzzy controller for UTSG, where he used two inputs with isosceles triangular membership functions in the premise (input) part and singletons in the consequent (output) part. He used Irving's model to derive the fuzzy control rules by looking at a desired phase-plane trajectory of water level against flow mismatch $(v - u)$. Na [25] proposed a fuzzy controller based on Irving's model, where he used a genetic algorithm to generate membership functions and rules.

A predominant amount of UTSG controller designs to date have considered identical reverse dynamics for feed-water and steam flows to simplify modeling. This is a good approximation, yet not necessarily true in general. We argue that this assumption not be made in UTSG modeling so that the controller is given more degrees of freedom in regulating a general UTSG plant, including those where this assumption is indeed true. The results so far reported on UTSG controller performance have been limited to a rejection of a single predetermined steam disturbance, or a single level tracking without the presence of disturbances. These scenarios are ideal, whereas in real practice the reference level changes, and disturbances occur independently

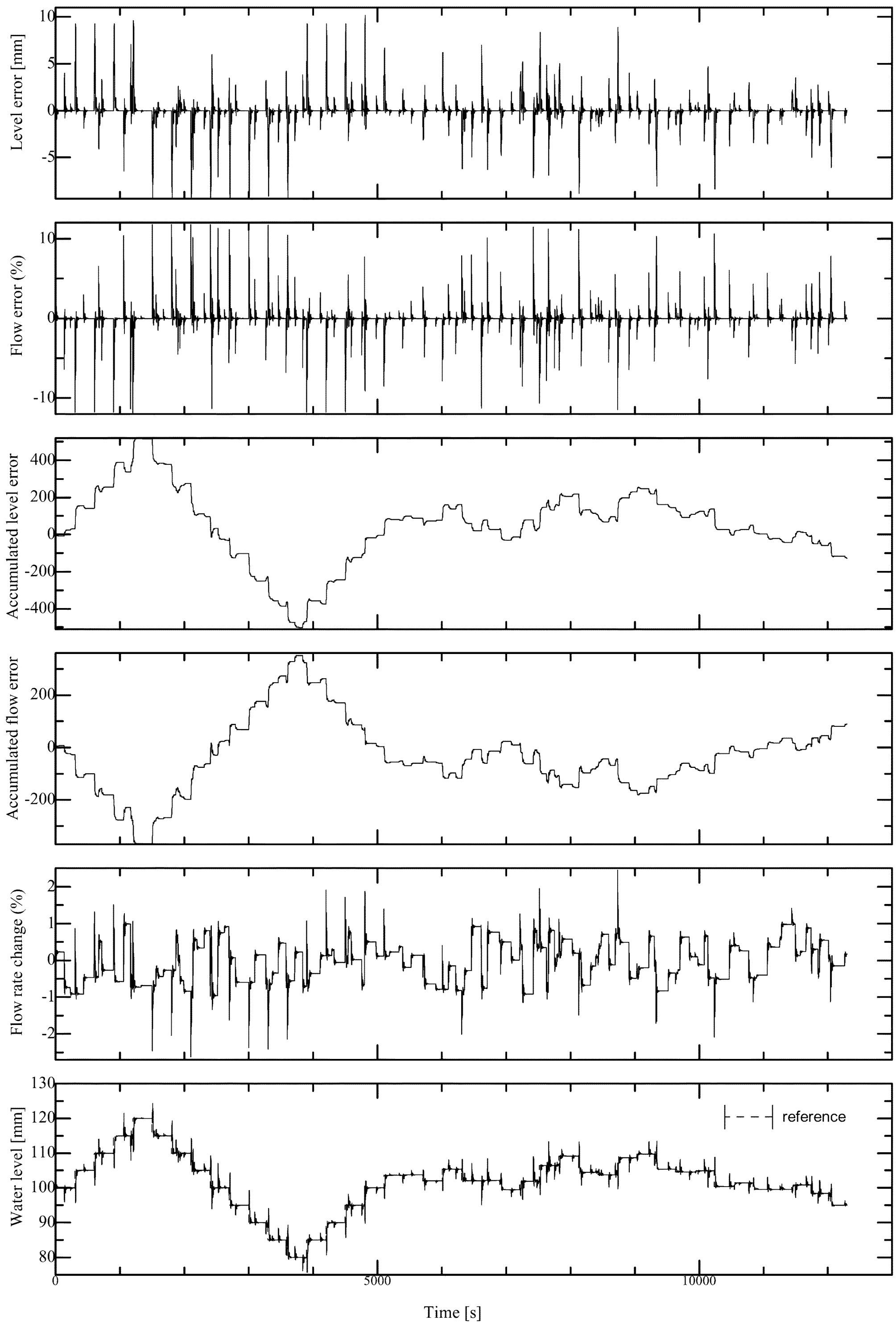

Fig. 4. Simulation under PI control at $p = 50\%$. During first 5100 s, the operator changes the reference level within 80–120 mm, and during the next 7200 s, reference level changes randomly. Random disturbances are introduced in the steam flow at all times. The time interval between consecutive disturbances is 200 s, and between consecutive reference changes is 300 s. The magnitude and sign of disturbances and reference changes are also randomly determined.

and frequently overlap each other so that the controller stability becomes a worried concern. However, it is extremely difficult to provide a hard stability proof for intelligent controllers; therefore, a prolonged simulation is necessary to demonstrate their stable operation.

In this work, we make efforts to fill some of the mentioned voids in UTSG literature. One motivation is to accommodate nonidentical reverse dynamics in UTSG modeling. The other motivation is to construct a data-driven intelligent controller (an adaptive neurofuzzy controller). The training data for this controller are taken form a prolonged UTSG simulation using few PI controllers that are specific to operating power. We uses Kim's UTSG model [11] in which reverse dynamics of feedwater and steam are not identical. The neurofuzzy controller has five inputs, 80 TS fuzzy rules, 532 parameters, and one overall output, and it was trained by the ANFIS hybrid algorithm (see the Appendix) [23]. The trained fuzzy system was used to regulate the UTSG water level under conditions similar to actual UTSG operations, where steam disturbances and reference level changes occur randomly, while overlapping each other. The 10-h simulation of the plant under these conditions provides convincing results about the stability of the fuzzy controller, and the low root mean square (RMS) error of the water level verifies its capability.

II. UTSG

In nuclear power industry, u-tube steam generator is a major component, where the steam is generated. The simplest schematic of the overall nuclear power plant is shown in Fig. 1. The heat generated at the nuclear reactor is taken away by forced-circulated water in the primary circuit. This water is contaminated by radioactive particles; therefore, the primary circuit is isolated from the rest of the system. The primary circuit has an inverted u-tube bundle submerged in the water column of the steam generator, where the heat transfer takes place from primary circuit to secondary circuit that makes secondary circuit water reach the state of bulk-boiling. The generated steam of the secondary circuit (with more than 99.9% dryness) is sent to the turbine, which is coupled to an armature to generate electricity.

As shown in Fig. 1, the water level y of the UTSG should be maintained within its lower and upper limits. Failure to maintain water level would lead to the following serious consequences including unintended plant shutdowns and system damage [1]:

1) If low water level exposes the u-tubes, the heat transfer from the primary circuit to the secondary circuit will not take place efficiently. Consequently, primary circuit builds up heat within itself, which causes the reactor to trip off.
2) If the water level rises too high, the steam will contain more moisture (dryness $< 99.9\%$). And, the wet steam may damage the turbine blades; therefore, turbine trips off.

Thus, it is extremely important that the water level of the UTSG be regulated within its limits. At present, a significant percentage of plant shutdowns and system unavailability are reportedly due to failures in UTSG water level control, which is a very difficult problem as UTSG dynamics shows high nonlinearity and nonminimum phase behavior that can be approximated by the following linearized model for a given power level [11]:

$$Y(s) = \frac{G_1}{s}\{U(s) - V(s)\} - \frac{G_2}{1+\tau_2}U(s) + \frac{G_3}{1+\tau_3}V(s) + \frac{G_4 s}{\tau_2^{-2} + 4\pi^2 T^{-2} + 2\tau_1^{-1}s + s^2}U(s) \quad (1)$$

where the four terms on the right-hand side (RHS) are mass capacity effect, nonminimum phase effect of feed-water, nonminimum phase effects of steam, and the effect of mechanical oscillation, in that order. The model parameters of (1), i.e., $\{G_1, G_2, G_3, G_4, \tau_1, \tau_2, \tau_3, T\}$ are given in Table I for a generic plant. These parameters were originally published by Irving [6] for an ideal plant. Fig. 2 graphically illustrates UTSG dynamics given in (1) when the plant operates at 50% of its rated power. The reverse dynamics due to feed-water change Δu and steam flow change Δv are shown in Fig. 2(c) and (d) by $\leftarrow A$, and $\rightarrow B$, respectively. The two reverse dynamics have been assumed identical (except for sign) in [7] and [9]. In [7] and [10], the mechanical oscillation effect [Fig. 2(e)] has been neglected. These assumptions are helpful to simplify UTSG modeling, however, at an expense of loosing credibility to represent actual plants. In this paper, we consider mechanical oscillation effect as well as nonidentical reverse dynamics for steam and feed-water, therefore, to make the model more accurate in representing actual UTSG plants.

TABLE II
SIMULATION CONDITIONS OF THE UTSG

range of reference water-level	[80,120]mm
Interval between reference water-level changes	300[s]
probability of reference water-level change	0.3
upper bound of reference water-level change	5[mm]
probability of occurrence of steam disturbance	0.9
Interval between consecutive steam disturbances	200[s]
minimum duration of steam disturbance	15[s]
upper bound of steam disturbance	1% of v_p
sampling interval	1[s]

TABLE III
NEAR-OPTIMUM PI GAINS FOR SPECIFIC POWER LEVELS

$p[\%]$	k_p^l	k_i^l	k_p^f	k_i^f
5	2.70	0.09	0.50	1.00
15	2.80	0.20	0.50	1.00
30	1.50	7.00	0.50	1.00
50	2.00	0.70	0.50	1.00
100	2.80	2.00	0.50	1.00

The UTSG water level control is inherently a very difficult problem due to the following two particular reasons [1]:

1) *Nonminimum phase dynamics (known as "swell" and "shrink" in UTSG literature):* "Swelling" behavior refers to a temporary increase in water level in response to a reduction of liquid water mass in the steam generator. "Swelling" is momentarily observed when steam flow rate undergoes a sudden increment ($v \rightarrow v + \delta v$) [Fig. 2(a) and (d)] or feed-water flow rate undergoes a sudden drop ($u \rightarrow u - \delta u$). "Shrinking" behavior is the exact opposite of "swelling," and it refers to a temporary decrease in the water level, against an increase of the liquid water mass in the steam generator [Fig. 2(a) and (c)]. These behaviors, though they last momentarily, are in exact opposition of the response one would expect upon the nature of steam or feed-water flow changes introduced to the system. These reverse behaviors make it very difficult to regulate the UTSG water level.
2) *Errors of flow rate measurements:* The most critical and widely used feedback signals are steam flow rate v and feed-water flow rate u. It is more often the case that these signals are not accurate enough during startup transients, and at low power operations. Under these conditions, flow rates are small in magnitude, and the process noise corrupts them beyond the limit of being useful feedback signals.

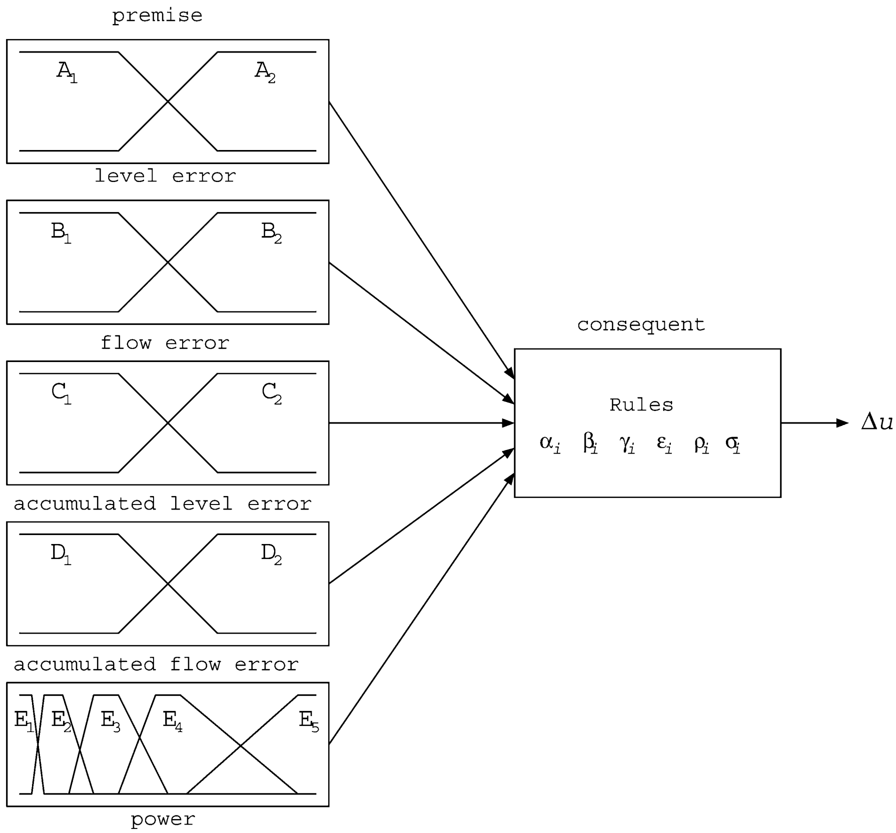

Fig. 5. Takagi–Sugeno-type adaptive neurofuzzy system.

III. Designing the Adaptive Neurofuzzy Controller

Using two PI controllers, one for level error control and another one for the flow error control, it is possible to regulate the water level at all specific power levels given in Table I. The structure of the two PI controllers is shown in Fig. 3, whereas the PI control law of the two controllers are given by

$$\Delta u^l(t) = k_p^l e^l(t) + k_i^l \int_0^t e^l(t)dt \tag{2}$$

$$\Delta u^f(t) = k_p^f e^f(t) + k_i^f \int_0^t e^f(t)dt. \tag{3}$$

And the water-level adjustment (increment or decrement) is determined by summing the outputs of the two controllers as follows:

$$\Delta u(t) = \Delta u^l(t) + \Delta u^f(t). \tag{4}$$

Using this PI control structure, a prolonged simulation of the UTSG plant was carried out under the conditions given in Table II, and the results are shown in Fig. 4. Starting from the beginning of the simulation, the reference water level was intentionally changed in 5-mm steps in every 300-s intervals, over the entire range of 100 mm→120 mm→80 mm→100 mm, which takes 5100 s in total. Then, another 2 h was given for random reference changes. At all times, random steam disturbance were introduced according to the specifications given in Table II.

The total duration of one simulation epoch is, therefore, 12 300 s. This simulation was iteratively carried out while intuitively tuning PI parameters of the level error controller, i.e., k_p^l, and k_i^l, and the nearly optimal settings shown in Table III were found. The PI gains for flow error controller, i.e., k_p^f and k_i^f, were set to acceptable values and kept unchanged for the sake of simplicity and reduced dimensionality in intuitive tuning of the control gains. There are six random number sequences in this simulation, i.e., three for water level change (decision, magnitude, and sign), and three for steam disturbances (occurrence, magnitude, and sign), which were not maintained constant in repetitive simulations during gain tuning process. Therefore, every iteration generates a different random number sequence. We argue that it does not affect the gain tuning process because the simulation duration is sufficiently long (12 300 s) that the RMS error of the water level would not be affected as a result of not using constant random number sequences in each iteration. On the other hand, using different random number sequences in successive simulation epochs is essential for a generalization of the tuning process. The following data have been obtained under the control of the tuned PI controllers:

1) level error, $e^l = r - y$;
2) percentage flow error, $e^f\% = e^f/v_p \times 100$;
3) accumulated level error, $\int e^l dt$;
4) accumulated flow error, $\int e^f dt$;
5) percentage change in flow rate, $\Delta u\% = \Delta u/v_p$.

The same data profiles are graphically shown in Fig. 4. Similar data were obtained for all specific power levels $\%p \in \{5, 15, 30, 50, 100\}$, and these data were used to train a Takagi–Sugeno-type adaptive neurofuzzy controller [23] with trapezoidal membership functions as shown in Fig. 5. This

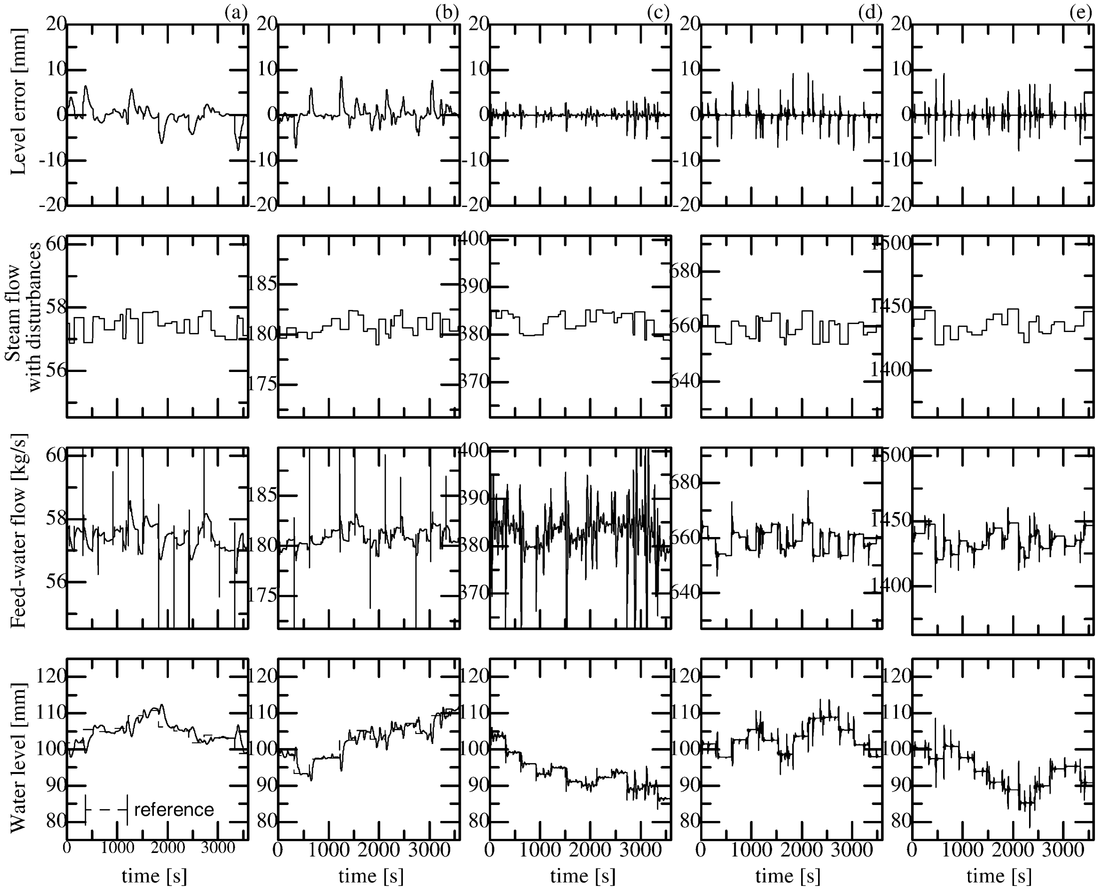

Fig. 6. Fuzzy controller performance at specific power levels. (a) 5%, (b) 15%, (c) 30%, (d) 50%, and (e) 100%.

neurofuzzy controller has five inputs—the four data sequences of the first four subframes in Fig. 4 and the power level $\%p$, whereas its desired output is the %feed-water change given in the fifth subframe in Fig. 4. The Takagi–Sugeno fuzzy inferences [17] used in the neurofuzzy system is described as follows: R_i: **if** e^l is A_j and e^f is B_k and $\int e^l dt$ is C_l and $\int e^f dt$ is D_m and p is E_n **then**

$$\Delta u_i = \alpha_i e^l + \beta_i e^f + \gamma_i \int e^l dt + \epsilon_i \int e^f dt + \rho_i p + \sigma_i$$
$$w_i = \mu_{e^l}(A_j) \times \mu_{e^f}(B_k) \times \mu_{\int e^l dt}(C_l) \times \mu_{\int e^f dt}(D_m) \times \mu_p(E_n). \quad (5)$$

The output of the TS fuzzy system would be the weighted sum of the individual rule outputs as given by

$$\Delta u = \frac{\sum_i^N w_i \Delta u_i}{\sum_i^N w_i} \quad (6)$$

where $N = j \times k \times l \times m \times n = 80$ is the number of rules as $j = k = l = m = 2$, and $n = 5$ are the number of fuzzy labels used for the corresponding inputs. The fuzzy system output for Δu is compared with the corresponding PI controller output for Δu, which is given in the training data in Fig. 4, and the mismatch is used to adapt the consequent parameters $\alpha_i, \beta_i, \gamma_i, \epsilon_i, \rho_i, \sigma_i$ and premise parameters $a_{mf}, b_{mf}, c_{mf}, d_{mf}$; $mf \in \{A_j, B_k, C_l, D_m\}$; $j, k, l, m \in \{1, 2\}$, that specify the trapezoidal membership functions of A_j, B_k, C_l, D_m. This adaptation algorithm was proposed by Jang [23], which uses least squares estimates of the consequent parameters, and gradient-based error backpropagation [26] for adapting premise parameters (see the Appendix). We have used two membership functions, each for the first four inputs, which were initialized by grid-partitioning of the training data, whereas the fifth input (i.e., power) was assigned five membership functions, which were handcrafted so that they have unity membership at the respective singleton values of power. Therefore, the number of rules is limited to 80, which is a manageable size for the neurofuzzy controller. In the premise part, the 13 $(2+2+2+2+5)$ trapezoidal membership functions require 52 $(= 13 \times 4)$ parameters, whereas in the consequent part the 80 $(= 2 \times 2 \times 2 \times 2 \times 5)$ rules need 480 $(= 80 \times 6)$ coefficients. The adaptation of the neurofuzzy system takes about 30 min/epoch on a 1.5-GHz 256-MB RAM Pentium IV system for a dataset of 12 300 entries. Training showed a very little error starting from the initialization (epoch = 1); thus, we stopped training after 15 epochs, and the trained fuzzy controller was used to regulate the UTSG plant for a 10-h duration.

IV. Results

The trained fuzzy controller demonstrated comparable performance to PI controller under the plant simulation conditions described in Table II, for a prolonged duration of 10 h. The first 1 h of the plant behavior at each power level is shown in Fig. 6.

The trained fuzzy controller is able to track random changes in reference water level satisfactorily, while rejecting random steam disturbances. The initial trapezoidal membership functions of the neurofuzzy network are as follows:

$$
\begin{aligned}
&A_1[-26.76, -17.58, -3.809, 5.37],\\
&A_2[-3.809, 5.37, 19.14, 28.32],\\
&B_1[-57.73, -38.5, -9.657, 9.574],\\
&B_2[-9.657, 9.574, 38.42, 57.65],\\
&C_1[-8428, -5596, -1348, 1484],\\
&C_2[-1348, 1484, 5731, 8563],\\
&D_1[-1324, -889.2, -237.3, 197.2],\\
&D_2[-237.3, 197.2, 849, 1284],\\
&E_1[-11.63, -2.125, 12.13, 21.63],\\
&E_2[12.13, 21.63, 35.88, 45.38],\\
&E_3[35.88, 45.38, 59.63, 69.13],\\
&E_4[59.63, 69.13, 83.38, 92.88],\\
&E_5[83.38, 92.88, 107.1, 116.6]
\end{aligned}
$$

and all consequent parameters were zeros. The initial membership functions were generated by grid-partitioning of training data, and it has zero membership value on either side of the training data distribution. This causes problems if an input variable swings beyond the range that grid-partitioning has already specified by looking at the training data. Under such a situation, control action may fail, and the plant may be destabilized. To eliminate this problem, the initial membership functions were stretched on either side as follows:

$$
\begin{aligned}
&A_1[\mathbf{-95.61}, \mathbf{-86.44}, -3.809, 5.37],\\
&A_2[-3.809, 5.37, \mathbf{87.99}, \mathbf{97.17}],\\
&B_1[\mathbf{-201.9}, \mathbf{-182.7}, -9.657, 9.574],\\
&B_2[-9.657, 9.574, \mathbf{182.7}, \mathbf{201.9}],\\
&C_1[\mathbf{-29670}, \mathbf{-26\,840}, -1348, 1484],\\
&C_2[-1348, 1484, \mathbf{26970}, \mathbf{29\,800}],\\
&D_1[\mathbf{-4584}, \mathbf{-4149}, -237.3, 197.2],\\
&D_2[-237.3, 197.2, \mathbf{4108}, \mathbf{4543}]
\end{aligned}
$$

The membership functions for power level were handcrafted as $E_1[\mathbf{-1}, \mathbf{0}, \mathbf{8}, \mathbf{12}]$, $E_2[\mathbf{8}, \mathbf{12}, \mathbf{18}, \mathbf{27}]$, $E_3[\mathbf{18}, \mathbf{27}, \mathbf{33}, \mathbf{47}]$, $E_4[\mathbf{33}, \mathbf{47}, \mathbf{53}, \mathbf{97}]$, and $E_5[\mathbf{53}, \mathbf{97}, \mathbf{100}, \mathbf{101}]$ using the known power levels. The boldface represents the modified premise parameters. Then, the neurofuzzy controller was trained for 15 epochs that showed a negligible error right from the beginning. The premise parameters of the trained fuzzy controller were rule1$[4.704, 0.5, 0.1742, 1.742, -3.787 \times 10^{-5} -7.573\,10^{-6}]$, rule2$[1.549\quad 0.1452\quad 0.1106\quad 0.5531\quad -0.2274\quad -0.015\,16]$, rule3$[0.3962\ 0.048\,59\ 1.832\ 0.2618\ -0.1893\ -0.006\,311] \cdots$ rule80$[0.1951\ 0.004\,271\ 0.1389\ 0.0697\ 0.0553\ 0.000\,553]$. The RMS errors of UTSG water-level control under PI and fuzzy controllers are listed in Table IV.

The neurofuzzy controller is trained to learn the principle of PI control. The training data are produced by five specific PI controllers at each power level. Plant dynamics and PI gains undergo significant variations on the operating power. Therefore, neurofuzzy controller learns the PI control, as it sees through the actions of all five PI controllers. Therefore, it may perform differently (better or worse) compared to PI controller at a specific power level. However, according to the results, the trained fuzzy controller performance is significantly comparable to the PI controller at all power levels.

TABLE IV
RMS ERROR WITH PI CONTROL AND NEUROFUZZY CONTROL

p[%]	PI control[mm]	Fuzzy control[mm]
5	2.47	2.27
15	1.96	1.77
30	0.82	0.78
50	1.15	1.19
100	1.35	1.29

In comparison with other reported results, steam disturbances have not been included in [10], whereas [7] and [9] have simulated only a single change in reference water level and a single steam disturbance that are perfectly isolated on the time line. It is, however, important to test how a UTSG controller performs when these events overlap, which is more likely the realistic situation. In our work, we have dealt with random disturbances and random changes in the reference water level, in that the randomness appears in the instance of occurrence, magnitude, and sign of these events. It makes the simulation in this paper more realistic, and that the results more trustworthy.

V. CONCLUSION AND FURTHER WORK

An adaptive neurofuzzy controller has been presented for the UTSG water-level control. This controller learned PI control principle and delivered satisfactory performance at all specific power levels. No simplifications were made in the UTSG model so that to accurately represent the realistic plant. System stability was demonstrated by carrying out simulations for prolonged durations under the conditions similar to real plant operations. Least number of membership functions (two) were used for the sake of compactness; however, sufficiently many input variables (five) were used for the versatility of the neurofuzzy controller. This proposed data-driven neurofuzzy controller can be trained off-line for any UTSG plant, given the actual data.

For further work, the fuzzy rules that are less important and/or redundant will be identified and removed to make the fuzzy controller more compact and transparent. We also plan to acquire actual UTSG data to train the neurofuzzy controller. How to outperform the original controller is another problem that we are working on in that we intend to modifying the PI data before they are used for training. This way, the controller is expected to outperform the original PI controller that is used to generate training data. In particular, we try to remove some high-frequency components in the PI output, i.e., %flow rate change, and use the filtered data as the target for fuzzy controller training. This way, we expect to find out an optimum damping level for the flow-rate change by trading off the speed of level tracking with nonminimum phase effect of feed-water flow. We hope that this will produce a new training method for data-driven UTSG controllers and other nonminimum phase systems.

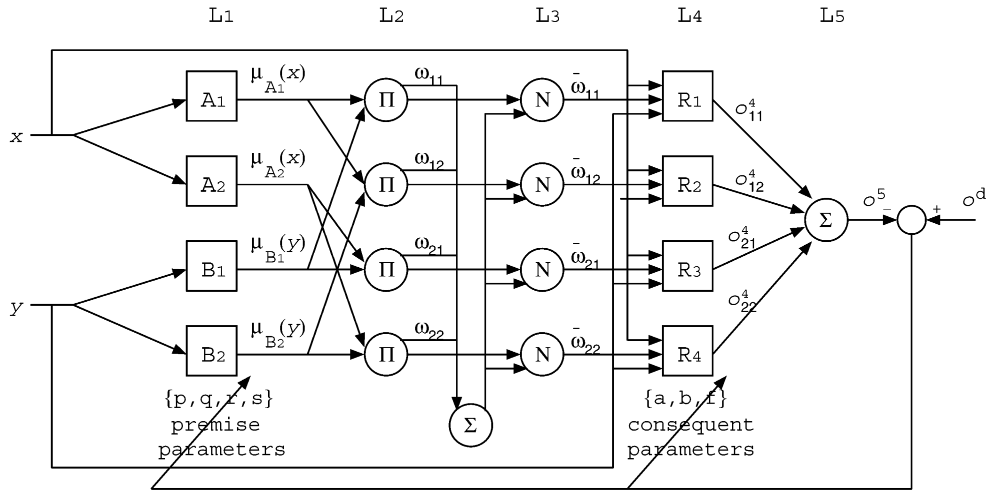

Fig. 7. Takagi–Sugeno-type adaptive neurofuzzy system with two inputs and one output.

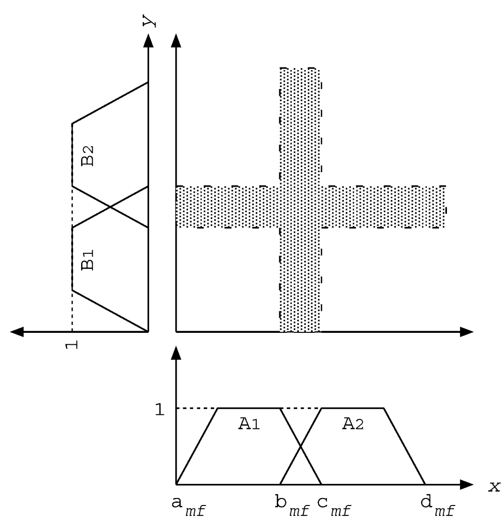

Fig. 8. Fuzzy grid partitioning of the input space.

APPENDIX
HYBRID LEARNING FOR ADAPTIVE NEUROFUZZY INFERENCE SYSTEMS (ANFIS)

An adaptive network (neurofuzzy) of two inputs and one output is shown in Fig. 7. The network has five layers, $L_1 \sim L_5$. The input vector is denoted by $[x, y]^T$, and o^5 is the output. The desired output is o^d. Fig. 8 shows the four subspaces of the fuzzy partitioning of input space by use of trapezoidal membership functions. These four subspaces are modeled by four TS rules that produce outputs $o_{ij}^4; i, j = 1 \sim 2$ as shown in Fig. 7, and the summation of these outputs determine the overall output of the network. The adaptive parameters of the network are of two categories–premise (input) parameters and consequent (output) parameters. The parameters that specify the membership functions are the premise parameters, i.e., a_{mf}, $b_{mf}, c_{mf}, d_{mf}; mf \in \{A_1, A_2, B_1, B_2\}$. The coefficients α_{ij}, β_{ij}, γ_{ij} of Takagi–Sugeno fuzzy rules, $o_{ij}^4 = \alpha_{ij}x + \beta_{ij}y + \gamma_{ij}$; $i, j = 1 \sim 2$ are the consequent parameters. Each layer of the network functions as follows:

1) Given the input, vector $[x, y]^T$ first layer calculates the membership functions $\mu_{A_1}(x)$, $\mu_{A_2}(x)$, $\mu_{B_1}(y)$, $\mu_{B_2}(y)$. For trapezoidal membership functions, $\mu_{A_1}(x)$ is calculated as follows:

$$\mu_{A_1}(x) = \begin{cases} \frac{x - a_{A_1}}{b_{A_1} - a_{A_1}}, & \text{if } a_{A_1} < x < b_{A_1} \\ 1, & \text{if } b_{A_1} \le x \le c_{A_1} \\ \frac{d_{A_1} - x}{d_{A_1} - c_{A_1}}, & \text{if } c_{A_1} < x < d_{A_1} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The other three memberships can be calculated in the same way.

2) The second layer calculates the firing strengths of rules

$$\omega_{ij} = \mu_{A_i} \times \mu_{B_j} \qquad i, j = 1, 2. \tag{8}$$

3) The third layer calculates the normalized firing strengths

$$\bar{\omega}_{ij} = \frac{\omega_{ij}}{\sum_{i=1}^{2} \sum_{j=1}^{2} \omega_{ij}}. \tag{9}$$

4) The fourth layer calculated the outputs of the TS rules

$$o_{ij}^4 = \bar{\omega}_{ij}(\alpha_{ij}x + \beta_{ij}y + \gamma_{ij}) \qquad i, j = 1, 2. \tag{10}$$

5) The fifth layer determined the overall output of the network

$$o^5 = \sum_{i=1}^{2} \sum_{j=1}^{2} \bar{\omega}_{ij}(\alpha_{ij}x + \beta_{ij}y + \gamma_{ij}). \tag{11}$$

A. Adapting Consequent Parameters by Least Squares Estimate

The forward pass of the network adapts the consequent parameters $\alpha_{ij}, \beta_{ij}, \gamma_{ij} \;\; i, j = 1, 2$. For this, (11) is rewritten as follows:

$$o^5 = \left[\{\bar{\omega}_{ij}\}_{1\times4}^T x \{\bar{\omega}_{ij}\}_{1\times4}^T y \{\bar{\omega}_{ij}\}_{1\times4}^T \right] \begin{bmatrix} \{\alpha_{ij}\}_{4\times1} \\ \{\beta_{ij}\}_{4\times1} \\ \{\gamma_{ij}\}_{4\times1} \end{bmatrix}. \tag{12}$$

For a batch of training data, $\{x_k, y_k, o_k^d\} \;\; k = 1 \sim N$

$$\{o_k^5\}_{N\times 1} = \begin{bmatrix} \{\bar{\omega}_{kj}\}_{4\times N} x_k \\ \{\bar{\omega}_{kj}\}_{4\times N} y_k \\ \{\bar{\omega}_{kj}\}_{4\times N} \end{bmatrix}^T \begin{bmatrix} \{\alpha_{ij}\}_{4\times 1} \\ \{\beta_{ij}\}_{4\times 1} \\ \{\gamma_{ij}\}_{4\times 1} \end{bmatrix} \tag{13}$$

where $k = 1 \sim N, i, j = 1, 2$. The consequent parameters α_{ij}, β_{ij}, γ_{ij} can be estimated in the best possible way by minimizing the square error of (13).

B. Adapting Premise Parameters by Error Backpropagation

The membership functions (premise parameters) are adapted in such a way that the gradient of the square error with respect to firing strengths of each rule descends iteratively. For the same input output training dataset mentioned above, the output square error for the kth element is

$$E_k = \left(o_k^d - o_k^5\right)^2 \qquad k = 1 \sim N \tag{14}$$

and its gradient with respect to the normalized firing strength of each rule is

$$\frac{\partial E_k}{\partial \omega_{ijk}} = -2\left(o_k^d - o_k^5\right)\frac{\partial o_k^5}{\partial \omega_{ijk}}. \tag{15}$$

By substitution from (11) we have

$$\frac{\partial E_k}{\partial \omega_{ijk}} = -2\left(o_k^d - o_k^5\right)\left(\alpha_{ij}x_k + \beta_{ij}y_k + \gamma_{ijk}\right). \tag{16}$$

The gradient considering the entire training dataset is

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\sum_{k=1}^{N} \partial E_k}{\partial \omega_{ijk}}. \tag{17}$$

Then, the required incremental change of the firing strength is

$$\Delta \omega_{ij} = \frac{-\eta \partial E}{\partial \omega_{ij}} \tag{18}$$

where η is the learning rate.

ACKNOWLEDGMENT

The authors would like to thank Dr. B.-S. Moon (Korea Atomic Energy Research Institute, Daejeon, Korea) for his advise and guidance for this work.

REFERENCES

[1] A. G. Parlos and O. T. Rais, "Nonlinear control of U-tube steam generators via H_∞ control," *Contr. Eng. Pract.*, vol. 8, pp. 921–936, 2000.
[2] A. Hold, "UTSG-2 a theoretical model describing the transient behavior of a pressurized water reactor natural-circulation U-tube steam generator," *Nucl. Technol.*, vol. 90, Apr. 1990.
[3] M. R. Yeung and P. L. Chan, "Development and validation of a steam generator simulation model," *Nucl. Technol.*, vol. 92, Dec. 1992.
[4] W. H. Strohmayor, "Dynamic Modeling of Vertical U-Tube Steam Generators for Operational Safety Systems," Ph.D. thesis, MIT, Cambridge, MA, 1982.
[5] J. I. Choi, "Nonlinear Digital Computer Control for the Steam Generator System in a Pressurized Water Reactor Plant," Ph.D. thesis, MIT, Cambridge, MA, 1987.
[6] E. Irving, C. Miossec, and J. Tassart, "Toward efficient full automatic operation of the PWR steam generator with water level adaptive control," in *Proc. Int. Conf. Boiler Dynamics and Control in Nuclear Power Stations, British Nuclear Energy Society*, London, U.K., 1980, pp. 309–329.
[7] M. G. Na, "Auto-tuned PID controller using a model predictive control method for the steam generator water level," *IEEE Trans. Nucl. Sci.*, vol. 48, no. 5, pp. 1664–1671, Oct. 2001.
[8] B. H. Cho and H. C. No, "Desin of stability-guaranteed fuzzy logic controller for nuclear steam generators," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 2, pp. 716–730, Apr. 1996.
[9] M. G. Na, "Design of an adaptive predictive controller for steam generators," *IEEE Trans. Nucl. Sci.*, vol. 50, no. 1, pp. 186–193, Feb. 2003.
[10] M. V. Kothara, B. Mettler, M. Morari, P. Bendotti, and C.-M. Falinower, "Level control in the steam generator of a nuclear power plant," *IEEE Trans. Contr. Syst. Technol.*, vol. 8, no. 1, pp. 55–69, Jan. 2000.
[11] D. W. Kim, "A study of fuzzy logic controller for water level control of PWR steam generators," M.S. thesis, Dept. Nuclear Eng., Korea Advanced Inst. Sci. Technol., Dejeon, Korea, 1990.
[12] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404–418, Mar./Apr. 1990.
[13] ——, "Fuzzy logic in control systems: Fuzzy logic controller, Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 419–435, Mar./Apr. 1990.
[14] M. Jamshidi, "Fuzzy control of complex systems," in *Soft Computing*. Berlin, Germany: Springer-Verlag, 1997, vol. 1, pp. 42–56.
[15] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[16] J. C. Bezdek and J. Dunn, "Optimal fuzzy partition: A heuristic for estimating the parameters in a mixture of normal distributions," *IEEE Trans. Comput.*, vol. C-24, pp. 835–838, 1975.
[17] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," in *Proc. IFAC Symp. Fuzzy Information Knowledge Representation and Decision Analysis*, Jul. 1983, pp. 55–60.
[18] M. Setnes, R. Babuska, and H. B. Verbruggen, "Rule-based modeling: Precision and transparency," *IEEE Trans. Syst., Man, Cybern.*, pt. C, vol. 28, no. 1, pp. 165–169, Feb. 1998.
[19] M. Setnes, R. Babuska, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern., B*, vol. 28, no. 3, pp. 376–386, Jun. 1998.
[20] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 530–537, Nov. 1998.
[21] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: Complexity and performance," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 509–521, Oct. 2000.
[22] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 516–524, Aug. 2001.
[23] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May 1993.
[24] R. Babuska, *Fuzzy Modeling for Control*. Dordrecht, The Netherlands: Kluwer, 1998.
[25] M. G. Na, "Design of genetic-fuzzy controller for the nuclear steam generator water level control," *IEEE Trans. Nucl. Sci.*, vol. 45, no. 4, Aug. 1998.
[26] S. Haykin, *Neural Networks, a Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.