A Plasma-Etching Process Modeling Via a Polynomial Neural Network

Dongwon Kim, Byungwhan Kim, and Gwi-Tae Park

A *plasma* is a collection of charged particles and on average is electrically neutral. In fabricating integrated circuits, plasma etching is a key means to transfer a photoresist pattern into an underlayer material. To construct a predictive model of plasma-etching processes, a polynomial neural network (PNN) is applied. This process was characterized by a full factorial experiment, and two attributes modeled are its etch rate and DC bias. According to the number of input variables and type of polynomials to each node, the prediction performance of the PNN was optimized. The various performances of the PNN in diverse environments were compared to three types of statistical regression models and the adaptive network fuzzy inference system (ANFIS). As the demonstrated high-prediction ability in the simulation results shows, the PNN is efficient and much more accurate from the point of view of approximation and prediction abilities.

Keywords: Plasma-etching process, polynomial neural network, statistical regression model, adaptive network fuzzy inference system.

I. Introduction

A *plasma* is a collection of charged particles and on average is electrically neutral. In fabricating integrated circuits, plasma etching is a key means of forming fine patterns for manufacturing integrated circuits. Owing to complex chemical reactions between the plasma variables, ions and radicals, and the material surfaces, it has been extremely difficult to model plasma etching. Thus, there have been many reports on constructing predictive etch models using intelligent systems such as a backpropagation neural network [1]-[4] and fuzzy system [5]. In this paper, a polynomial neural network (PNN) [6], [7] is applied to model a plasma etching. The PNN employed here is a GMDH-type algorithm [8], which is one of the useful approximator techniques. The PNN has an architecture similar to feedforward neural networks whose neurons are replaced by polynomial nodes. The output of each node in the PNN is obtained using several types of polynomials such as a linear, quadratic, and modified quadratic of input variables. These polynomials are called partial descriptions (PDs). The PNN has fewer nodes than a backpropagation neural network, but the nodes are more flexible. The PNN model is compared to the three types of statistical regression models and the adaptive network fuzzy inference system (ANFIS). The experimental data examined here were obtained from the etching of silicon carbide (SiC) material. The process was characterized by a 2^5 factorial experiment [9]. The process input parameters that were varied in the design include source power, bias power, pressure, O_2 fraction, and a gap between the coil antenna and chuck holder. The process outputs modeled are etch rate and electrical DC bias.

Manuscript received Aug. 29, 2003; revised Apr. 29, 2004.

This research was partly supported by University IT Research Center Project.

Dongwon Kim (phone: +82 2 929 5185, email: dwkim@elec.korea.ac.kr) and Gwi-Tae Park (email: gtpark@korea.ac.kr) are with the Department of Electrical Engineering, Korea University, Seoul, Korea.

Byungwhan Kim (email: kbwhan@sejong.ac.kr) is with the Department of Electronic Engineering, Bio Engineering Research Center, Sejong University, Seoul, Korea.

II. Experimental Data

We collected the experimental data from SiC etching in a C_2F_6/O_2 plasma. We isolated the plasma generated inside a chamber from the planar-coupled coils using a dielectric window. A multipolar magnet was additionally equipped outside the chamber to achieve a high plasma density while maintaining high spatial uniformity. The cylindrical chamber has a radius of 80 mm and a height of 40 mm. Prior to the initiation of gas flows, the chamber was evacuated using a turbo (TUROVAC 3430MC) and rotary (Edward High Vacuum E2M40) pump, thereby maintaining a base pressure of about 10^{-6} Torr. We precisely controlled the gas flow rates through mass flow controllers and controlled the process pressure, measured by the baratron, pinary, and penning gauges, using the throttle valve. The coolant was fed to a chuck holder to minimize damage to the equipment from a surge in the temperature while the system was running. We fabricated test patterns on n-type, 2-inch, 4H-SiC wafers with an epi doping density of 1.0E15 N_D-N_A/cm^3. Using a mask with a line width of 3 μm, we patterned a negative photoresist on the epi-layered SiC. The patterns were then etched in C_2F_6/O_2 plasma. The DC bias was measured during the process run. After removing the photoresist through the lift-off process, another vertical etch rate was measured using scanning electron microscopy. Using a 2^5 factorial experiment [9], the etch process was characterized. The process parameters that were varied in the design include the source power, bias power, pressure, O_2 fraction, and gap. Their experimental ranges are shown in Table 1.

Table 1. Experimental parameters and ranges.

Parameters	Range	Unit
Source power	600–900	Watts
Bias power	50–150	Watts
Pressure	4–16	mTorr
O_2 fraction	0–80	%
Gap	6–12	cm

The total flow rates of both C_2F_6 and O_2 were set to 30 sccm. The data for the etch rate consisted of 32 and 15 experiments for training and testing networks, respectively. Other DC bias data were composed of 34 and 17 experiments for training and testing models, respectively.

III. Polynomial Neural Network

1. Polynomial Neural Network Architecture and Its Algorithm

Since the PNN is applied to plasma-etched data, its fundamentals are briefly explained. Each polynomial in the PNN algorithm represents a PD, and the best model is determined by selecting the most significant input variables and polynomial order. The design procedures are detailed in [6], [7]. Here, the architecture and algorithm of the PNN is briefly explained. The PNN is operated in the following steps.

Step 1. We define the input variables such as $x_{1i}, x_{2i}, \cdots x_{Ni}$ related to output variable y_i, where N and i are the number of the entire input variables and input-output data sets, respectively.

Step 2. The input-output data sets are separated into training (n_{tr}) data sets and testing (n_{te}) data sets. Obviously, we have $n = n_{tr} + n_{te}$. The training data set is used to construct a PNN model. And the testing data set is used to evaluate the constructed PNN model.

Step 3. The structure of the PNN is strongly dependent on the number of input variables and the order of PD in each layer. Two kinds of PNN structures, namely, the basic PNN structure and the modified PNN structure, can be available. Each of them comes with two cases.

(a) Basic PNN structure – The number of input variables of the PDs is the same in every layer.

(b) Modified PNN structure – The number of input variables of the PDs varies from layer to layer.

Case 1. The polynomial order of the PDs is the same in each layer of the network.

Case 2. The polynomial order of the PDs in the 2nd or higher layer is different from the one in the 1st layer.

Step 4. We determine arbitrarily the number of input variables and type of polynomial in the PDs. The polynomials differ according to the number of input variables and the polynomial order. Several types of polynomials are shown in Table 2. Because the outputs of the nodes of the preceding layer become the input variables for the current layer, the total number of PDs located at the current layer is determined by the number of selected input variables (r) from the nodes of the preceding layer. The total number of PDs in the current layer is equal to the combination, ${}_N\mathrm{C}_r$, that is $\frac{N!}{r!(N-r)!}$, where N is the number of nodes in the preceding layer.

Table 2. Different types of the polynomial in PDs.

No. of inputs / Order of the polynomial	1	2	3
1 (Type 1)	Linear	Bilinear	Trilinear
2 (Type 2)	Quadratic	Biquadratic	Triquadratic
2 (Type 3)	Modified quadratic	Modified biquadratic	Modified triquadratic

For an example, the specific forms of a PD in the case of two inputs are given as

- Bilinear = $c_0 + c_1 x_1 + c_2 x_2$ (1)
- Biquadratic = $c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1^2 + c_4 x_2^2 + c_5 x_1 x_2$ (2)
- Modified biquadratic = $c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2$, (3)

where c_i is the regression coefficients.

Step 5. The vector of the coefficients of the PDs is determined using a standard mean squared error by minimizing the following index:

$$E_k = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} (y_i - z_{ki})^2, \qquad k = 1,2,\cdots, \frac{N!}{r!(N-r)!}, \tag{4}$$

where z_{ki} denotes the output of the k-th node with respect to the i-th data, and n_{tr} is the number of training data subsets.

This step is completed repeatedly for all the nodes in the current layer.

Step 6. The predictive capability of each PD is evaluated by a performance index using the testing data set. We then choose w PDs among ${}_N C_r$ PDs in due order from the best predictive capability (the lowest value of the performance index). Here, w (30) is the pre-defined number of PDs that must be preserved to the next layer. The outputs of the chosen PDs serve as inputs to the next layer.

Step 7. The PNN algorithm terminates when the number of layers predetermined by the designer is reached. Here, the number of total layers was limited to 5.

Step 8. If the stopping criterion is not satisfied, the next layer is constructed by repeating steps 4 through 8.

Figure 1 shows a PNN architecture. In the figure, four input variables $(x_1, \cdots, x_4)$, three layers, and a PD processing example are considered. $z^{j-1}{}_i$ indicates the output of the i-th node in the $(j-1)$th layer, which is employed as a new input of the j-th layer. The black nodes have influence on the best node (output node), and these networks represent the ultimate PNN model. Meanwhile, the solid line nodes have no influence over the output node. In addition, owing to poor performance, the dotted line nodes are excluded when choosing the PDs with the best predictive performance in the corresponding layer. Therefore, the solid line nodes and dotted line nodes should not be present in the final PNN model.

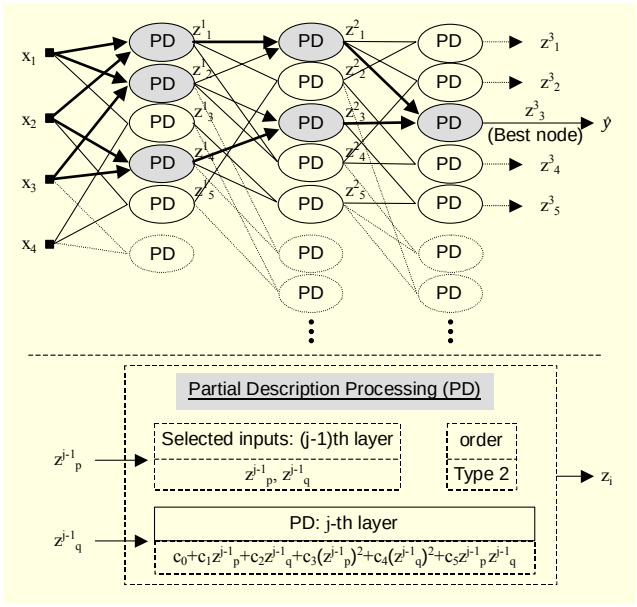

Fig. 1. Overall architecture of the PNN.

2. Comparison of Neural Network and PNN

Neural networks (NNs) have been widely used for modeling nonlinear systems. The approximation capability of NNs also has been investigated by many researchers. NNs provide an excellent flexibility in mapping complex 'input-output' dependencies. The use of NNs has, however, some disadvantages compared with the PNN. In particular, the equations built during NNs training are opaque, and NNs do not distinguish inputs by their significance, leaving the responsibility to select significant inputs to a user. Also, the number of nodes and layers of the NNs are fixed by the user, and while there are many factors contributing to the flexibility of the NNs such as training tolerance, hidden neurons, initial weight distribution, and two gradients of activation functions, the factors contributing to the flexibility of the PNN are developed through the modeling process. The training of NNs is a kind of statistical estimation often using algorithms that are slow. If noise is considerable in a data sample, the generated models tend to be overfitted in order to achieve good results, whereas the PNN model creates an optimal complex model systematically and autonomously. The optimal complex model is a model that optimally balances model quality on a given data set and its generalization power on new, not previously seen, data with respect to the data's noise level and the task of modeling (prediction). It thus solves the basic problems of experimental systems analysis, systematically avoiding "overfitted" models based on the data's information only. This makes the PNN method a most automated, fast, and very efficient supplement and alternative to the predictions of plasma etching methods.

IV. Simulation Results

Using the PNN, statistical regression technique, and ANFIS, predictive models are constructed and compared. The accuracy

measured by the training and testing data was quantified by the root mean-squared error.

1. Etch Rate

The etch rate is modeled by using the PNN. The data for the etch rate was divided into two sets of training and test data. The training data consisted of 32 patterns, 30 from the experimental design and two replications for the center point. Actually, two of the expected 32 patterns from the experimental design employed were unable to be collected due to unacceptable plasma conditions. The test data consisted of 15 patterns not pertaining to the training data. As stated earlier, the PNN architecture consists of five layers. Each PD in a layer is defined by two training factors, the number of inputs and the polynomial type. Depending on their combinations, the PNN prediction is expected to vary considerably. The input number for the first and other four layers varies from 1 to 4. Similarly, the polynomial type varies from 1 to 3. The results are summarized in the figures and tables. Figures 2 through 4 show the performances of the basic PNN model when the polynomial type varies from 1 to 3 and the number of inputs for the first and other four layers are either 2, 3, or 4. In the

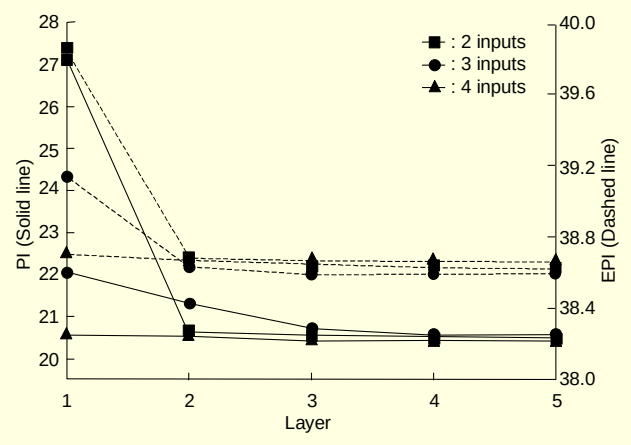

Fig. 2. PI and EPI behavior with respect to layers (Type 1).

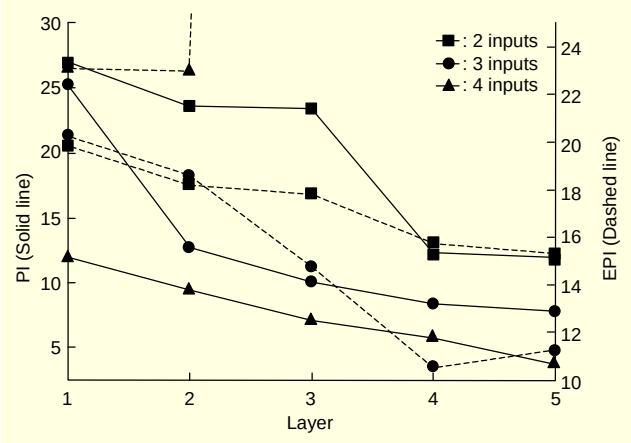

Fig. 3. PI and EPI behavior with respect to layers (Type 2).

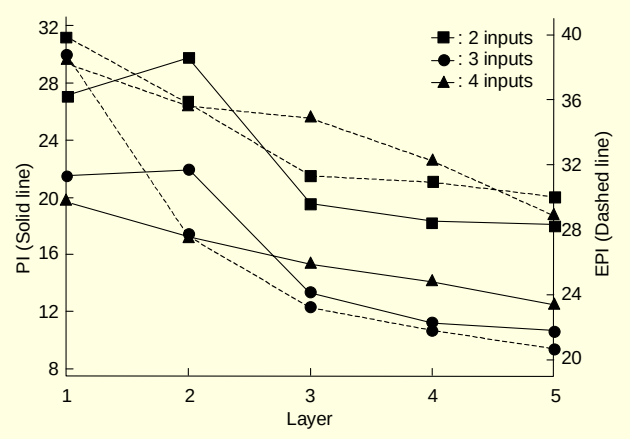

Fig. 4. PI and EPI behavior with respect to layers (Type 3).

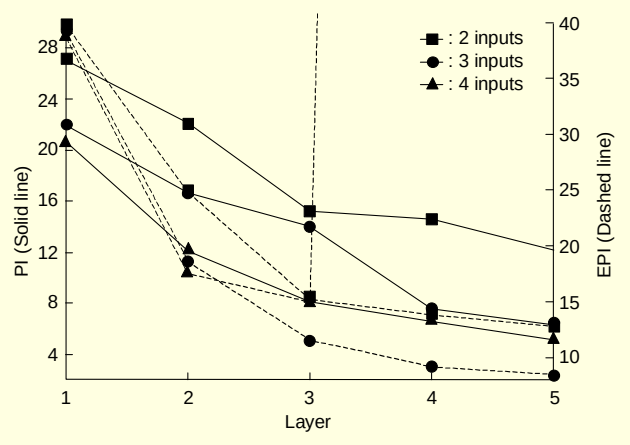

Fig. 5. PI and EPI behavior with respect to layers and polynomial type (Type 1→2).

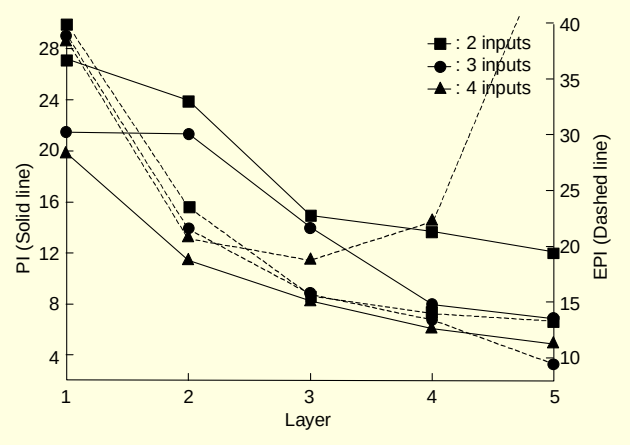

Fig. 6. PI and EPI behavior with respect to layers and polynomial type (Type 3→2).

figures, PI denotes the performance index of the model for the training data, while EPI describes the performance of the model for the testing data.

Figures 5 and 6 show the performances of the basic PNN model when the polynomial type for the first and other four layers varies from 1 to 2 and 3 to 2, respectively.

Figures 7, 8, and 9 show the performances of the modified PNN model when the number of inputs for the first and other four layers varies from 2 to 4, 3 to 2, and 3 to 4, respectively.

Figures 10 and 11 show the performances of the modified PNN when the number of inputs for the first and other four layers varies from 2 to 4, 3 to 2, and 3 to 4, and the polynomial types vary from 1 to 2, and 3 to 2.

In the case of the basic PNN, the values PI=6.3183 and EPI=12.4130 in Fig. 5 are obtained by the use of three inputs for every node in all the layers, with Type 1 in the first layer and Type 2 in the other four layers. On the other hand, the results of the modified PNN, PI=3.1607 and EPI=6.9059 in Fig. 11, are obtained by the use of three input variables with Type 3 for every node in the first layer and four input variables with Type 2 for every node from the second layer to the fifth

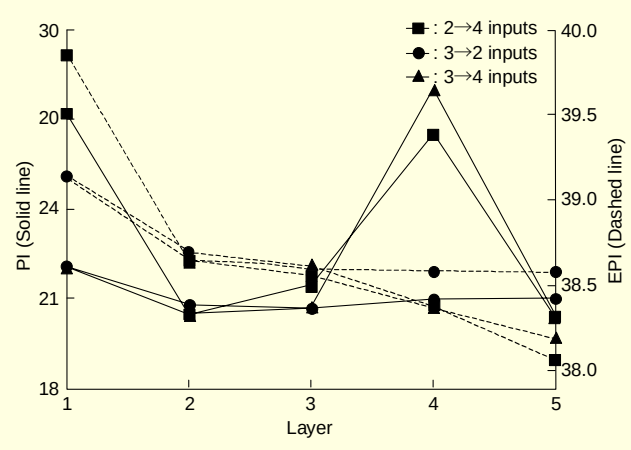

Fig. 7. PI and EPI behavior with respect to layers and inputs (Type 1).

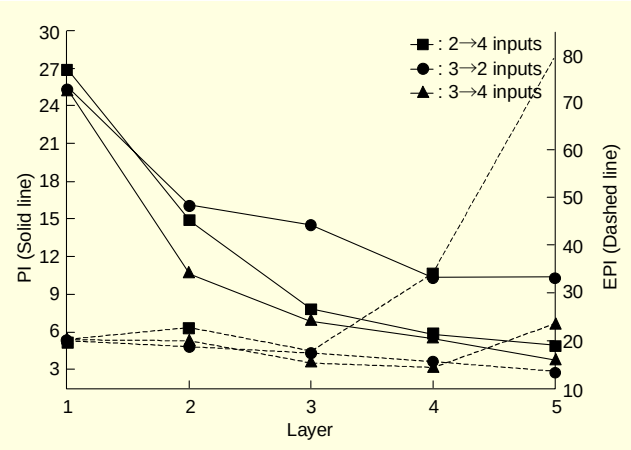

Fig. 8. PI and EPI behavior with respect to layers and inputs (Type 2).

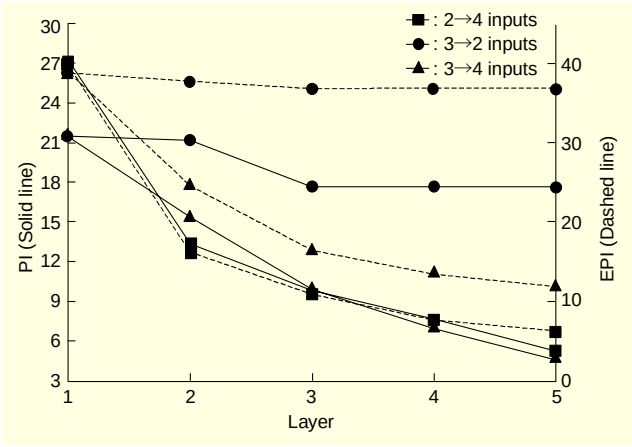

Fig. 9. PI and EPI behavior with respect to layers and inputs (Type 3).

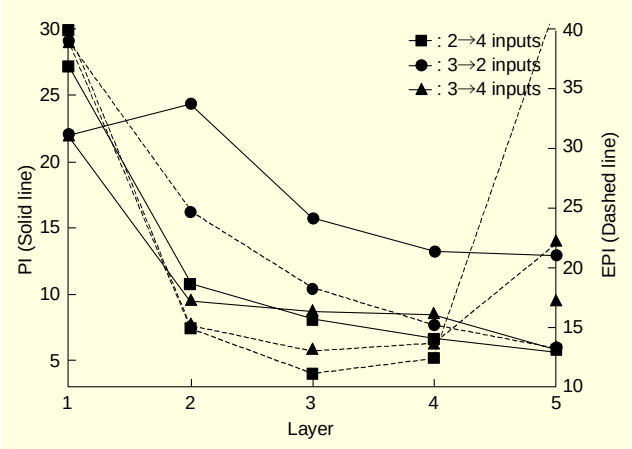

Fig. 10. PI and EPI behavior with respect to layers, inputs, and polynomial type (Type 1→2).

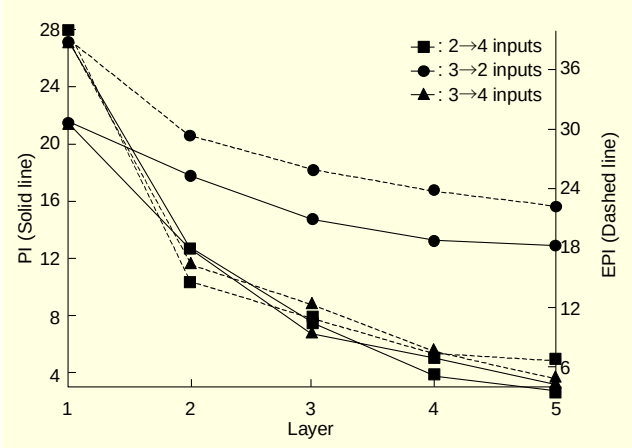

Fig. 11. PI and EPI behavior with respect to layers, inputs, and polynomial type (Type 3→2).

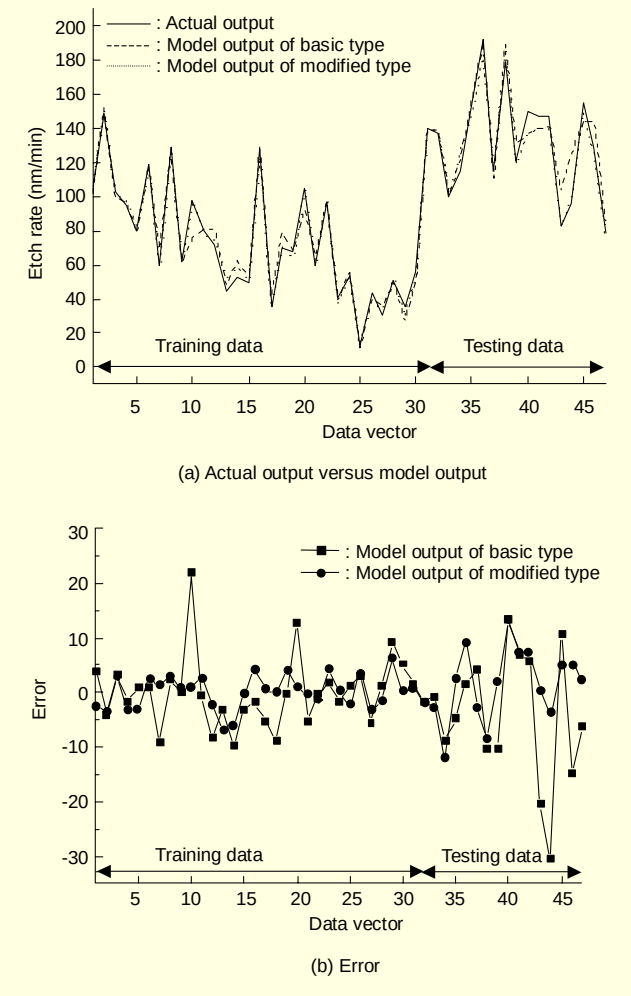

Fig. 12. Identification performance of PNN.

layer. To ascertain the model performance visually, model predictions are compared with the actual measurements for all training and test data as illustrated in Fig. 12(a). This is also supported by the residual errors shown in Fig. 12(b). As represented in Fig. 12, the PNN model was well trained and attained a high predictive ability.

For the comparison with statistical regression models, three types of regression models were constructed. Their forms were obtained as the following.

Type 1:

$$y = -11.561 + 0.12108x_1 + 0.38199x_2 + 0.42708x_3 + 0.20156x_4 - 5.4916x_5 \quad (5)$$

Type 2:

$$\begin{aligned} y = {} & -501.19 + 0.62593x_1 - 0.060522x_2 - 2.6828x_3 + 1.2095x_4 + 78.053x_5 \\ & - 0.00040773x_1^2 + 0.0033633x_2^2 - 0.026861x_3^2 - 0.013099x_4^2 \\ & - 4.353x_5^2 + 0.00010938x_1x_2 + 0.008316x_1x_3 + 0.00002343x_1x_4 \\ & + 0.0012326x_1x_5 - 0.0052604x_2x_3 + 0.00022656x_2x_4 \\ & - 0.030052x_2x_5 + 0.0093424x_3x_4 - 0.25955x_3x_5 - 0.012109x_4x_5 \end{aligned} \quad (6)$$

Type 3:

$$\begin{aligned} y = {} & 5.4259 + 0.020361x_1 + 0.6031x_2 - 3.1448x_3 + 0.18047x_4 \\ & + 0.60282x_5 + 0.00016959x_1x_2 + 0.008316x_1x_3 + 0.00002343x_1x_4 \\ & + 0.000229126x_1x_5 - 0.0052604x_2x_3 + 0.00022656x_2x_4 \\ & - 0.030052x_2x_5 + 0.0074608x_3x_4 - 0.25955x_3x_5 - 0.012109x_4x_5 \end{aligned} \quad (7)$$

The corresponding PI and EPI are contained in Table 3. In addition, the ANFIS model was constructed with variations in the type of membership function. The corresponding PI and EPI are also contained in Table 3.

Table 3. Performance index values of some identification models.

Model		Performance index	
		PI	EPI
Regression model	Type 1	20.414	56.476
	Type 2	9.3755	19.028
	Type 3	17.647	56.164
ANFIS [10]	Triangular	0.3750	27.6330
	Gaussian	0.3751	27.2463
Our model	Basic	6.3183	12.4130
	Modified	3.1607	6.9059

2. DC-Bias

As in the case of the etch rate, the DC bias was modeled by using the PNN and its identification performance compared with three types of statistical regression models and the ANFIS. The results are contained in Table 4. For a comparison with statistical regression models, three types of regression models were also constructed. Their forms were obtained as the following.

Type 1:

$$y = 43.423 - 0.10187x_1 + 2.8906x_2 + 2.8906x_3 - 0.14453x_4 + 16.49x_5 \quad (8)$$

Type 2:

$$\begin{aligned} y = {} & 713.09 - 1.0885x_1 - 4.0459x_2 + 35.03x_3 + 4.1024x_4 - 16.829x_5 \\ & + 0.00064582x_1^2 + 0.029232x_2^2 - 1.8572x_3^2 - 0.051371x_4^2 \\ & + 1.129x_5^2 - 0.0000625x_1x_2 + 0.0060764x_1x_3 - 0.00052604x_1x_4 \\ & - 0.0017361x_1x_5 - 0.0015625x_2x_3 + 0.0013906x_2x_4 \\ & + 0.12188x_2x_5 - 0.0095052x_3x_4 + 0.11632x_3x_5 + 0.023698x_4x_5 \end{aligned} \quad (9)$$

Type 3:

$$\begin{aligned} y = {} & 185.69 - 0.11972x_1 + 1.8006x_2 - 2.1146x_3 - 0.0072917x_4 \\ & + 3.4931x_5 - 0.0000625x_1x_2 + 0.0060764x_1x_3 - 0.00052604x_1x_4 \\ & - 0.0017361x_1x_5 - 0.0015625x_2x_3 + 0.0013906x_2x_4 \\ & + 0.12188x_2x_5 - 0.0095052x_3x_4 + 0.11632x_3x_5 + 0.023698x_4x_5 \end{aligned} \quad (10)$$

Again, a series of comprehensive experiments of the PNN was conducted, and the results are summarized in the same way as before.

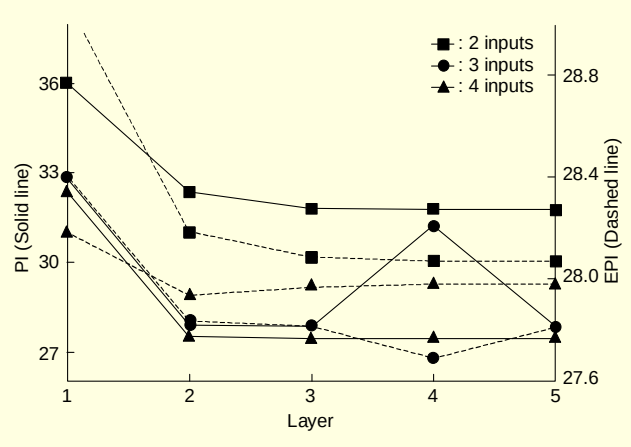

Fig. 13. PI and EPI behavior with respect to layers (Type 1).

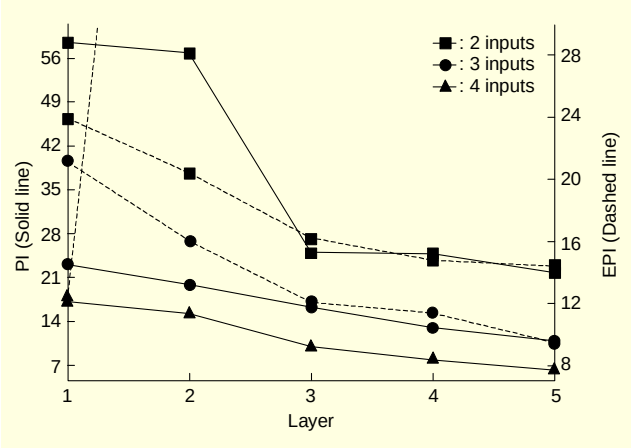

Fig. 14. PI and EPI behavior with respect to layers (Type 2).

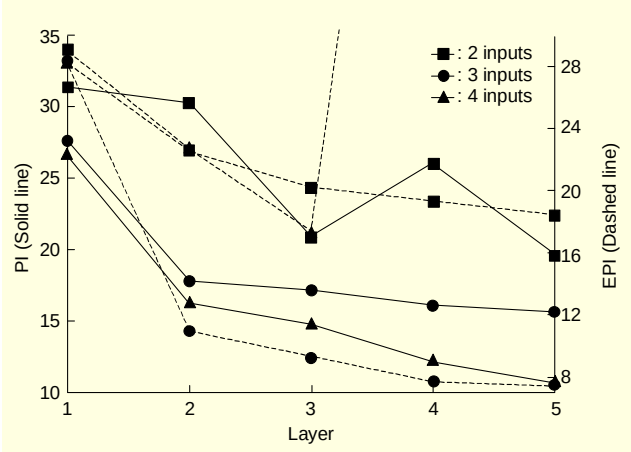

Fig. 15. PI and EPI behavior with respect to layers (Type 3).

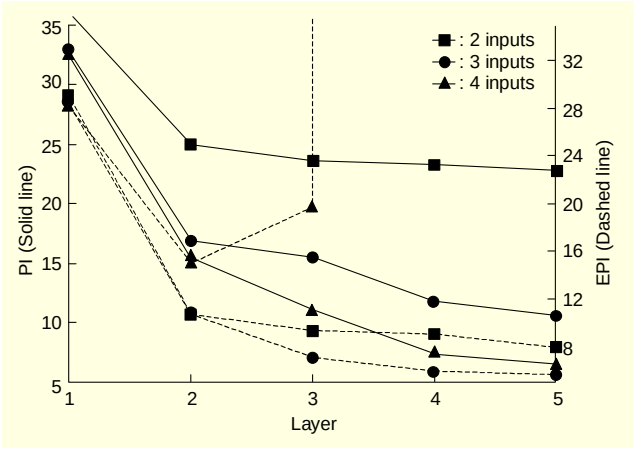

Fig. 16. PI and EPI behavior with respect to layers and polynomial type (Type 1→2).

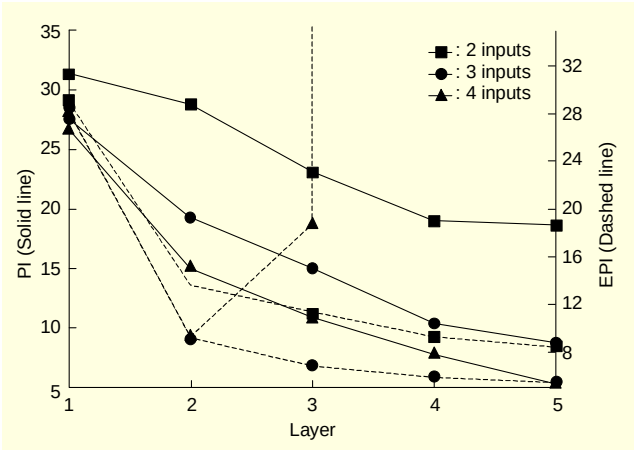

Fig. 17. PI and EPI behavior with respect to layers and polynomial type (Type 3→2).

In the case of the basic PNN, PI=8.7797 and EPI=7.6843 in Fig. 17 are obtained by the use of three inputs to every node in each layer with Type 3 in the first layer and Type 2 in the other four layers. On the other hand, the results of the modified PNN, PI=6.6168 and EPI=5.5893 in Fig. 22, are obtained by the use of two input variables with Type 3 to every node in the first layer and four input variables with Type 2 to every node from the second layer to the fifth layer. Figure 23 shows the comparisons with actual data and the lowest value of the basic and modified PNNs. Figure 23 also shows residual errors.

Table 4 provides a comparison of the PNN with the regression model and ANFIS. It is obvious that the PNN model outperforms the other models both in terms of accuracy and higher generalization capabilities. Owing to the complex and nonlinear chemical reactions between the experimental parameters and DC-bias, it has been difficult to model plasma etching. These complex characteristics of the DC-bias can be

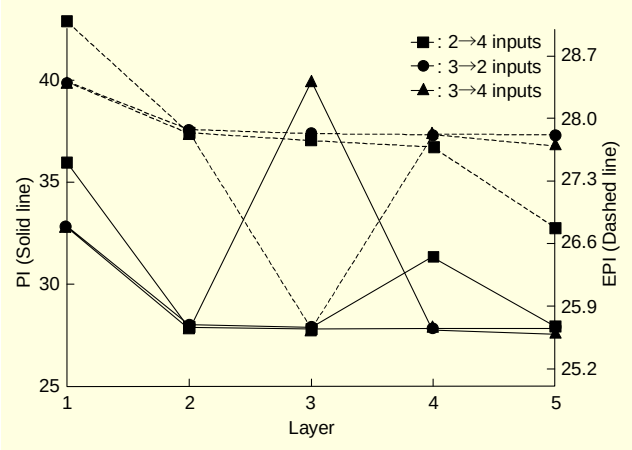

Fig. 18. PI and EPI behavior with respect to layers and inputs (Type 1).

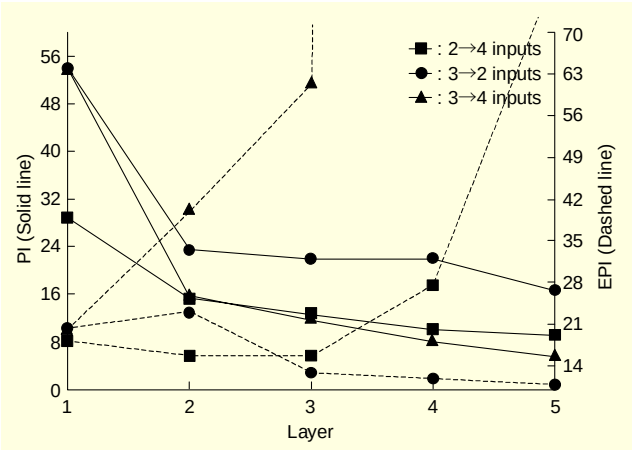

Fig. 19. PI and EPI behavior with respect to layers and inputs (Type 2).

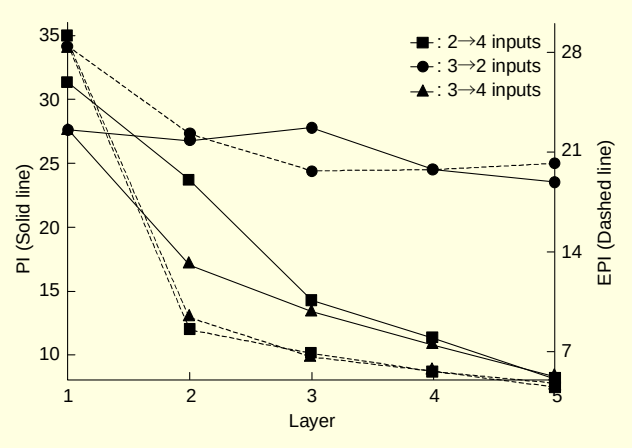

Fig. 20. PI and EPI behavior with respect to layers and inputs (Type 3).

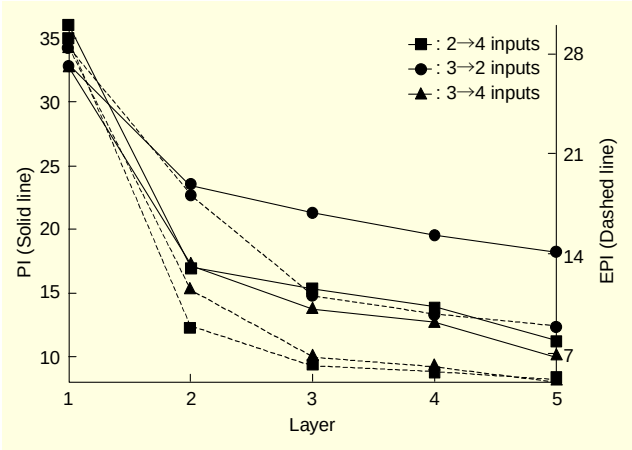

Fig. 21. PI and EPI behavior with respect to layers, inputs, and polynomial type (Type 1→2).

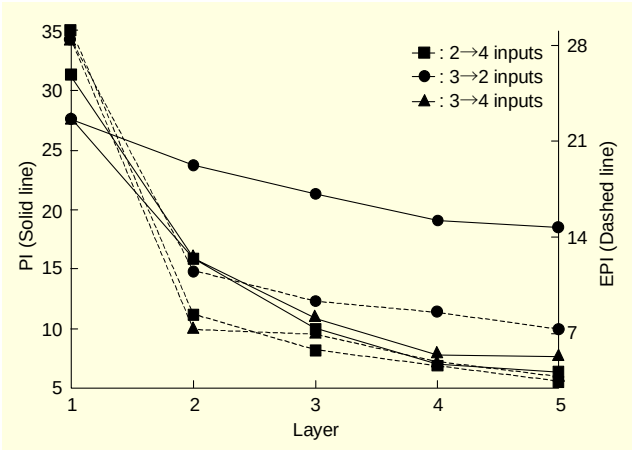

Fig. 22. PI and EPI behavior with respect to layers, inputs, and polynomial type (Type 3→2).

seen from Fig. 23. Because of these characteristics and a lesser number of testing data pairs, the EPIs are better than PIs.

V. Conclusion and Future Work

In this paper, a PNN was constructed and applied to model a plasma etching. The performance of the PNN was compared to those for the three statistical regression models and ANFIS. The etching process was characterized with a 2^5 factorial experiment. The PNN was evaluated with variations in the number of inputs and in the polynomial type. Compared to the optimized regression models and ANFIS, the PNN model demonstrated a considerably improved prediction in modeling either the etch rate or DC bias. This indicates that the PNN is an effective means to construct a predictive model for poorly defined complex systems characterized by the limited data set.

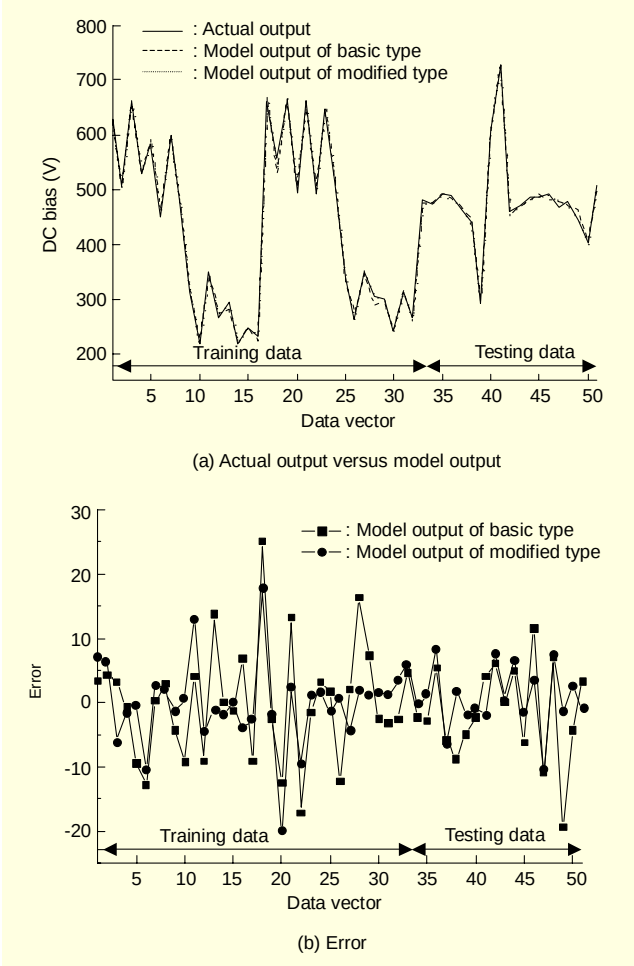

Fig. 23. Identification performance of PNN.

Table 4. Performance index values of some identification models.

Model		Performance index	
		PI	EPI
Regression model	Type 1	27.427	39.57
	Type 2	15.154	94.927
	Type 3	19.371	39.57
ANFIS [10]	Triangular	0.8495	33.3252
	Gaussian	0.8500	32.5949
Our model	Basic	8.7797	7.6843
	Modified	6.6168	5.5893

In this paper, we used testing data for the selection of the most predictive PDs. This means, somewhat, that the testing data is used in the course of PNN architecture building. Therefore a genuine prediction ability of the PNN model is not a guarantee. To get a more valid generalization ability, new selection criteria and data separating such as training, testing, and validation, from a methodological viewpoint, are needed.

References

[1] B. Kim and G. Park, "Modeling Plasma Equipment Using Neural Networks," *IEEE Plasma Sci.*, vol. 29, no. 2, Feb. 2001, pp. 8-12.

[2] B. Kim and S. Park, "Characterization of Inductively Coupled Plasma Using Neural Networks," *IEEE Trans. Plasma Sci.*, vol. 30, no. 2, Apr. 2002, pp. 698-705.

[3] E. Rietman and E. Lory, "Use of Neural Networks in Semiconductor Manufacturing Processes: An Example for Plasma Etch Modeling," *IEEE Trans. Semicond. Manufact.*, vol. 6. no. 4, Nov. 1993, pp. 343-347.

[4] B. Kim and K.H. Kwon, "Qualitative Modeling of Silica Plasma Etching Using Neural Network," *J. Applied Physics*, vol. 93, no. 1, 2003, pp. 76-82.

[5] B. Kim and J. Park, "Qualitative Fuzzy Logic Model of Plasma Etch Process," *IEEE Trans. Plasma Sci.*, vol. 30, no. 2, Apr. 2002, pp. 673-678.

[6] D.W. Kim and G.T. Park, "Optimization of Polynomial Neural Networks: An Evolutionary Approach," *Trans. KIEE.* vol. 52D, no. 7, 2003, pp. 424-433.

[7] S.K. Oh, D.W. Kim, and B.J. Park, "A Study on the Optimal Design of Polynomial Neural Networks Structure," *Trans. KIEE.* 49D, 2000, pp, 145-156.

[8] A.G. Ivahnenko, "Polynomial Theory of Complex Systems," *IEEE Trans. Syst. Man Cybern.*, 1971, pp. 364-378.

[9] D.C. Mongomery, *Design and Analysis of Experiments*, Wiley, New York, 1991.

[10] J.S. Jang, "ANFIS: Adaptive-Networks-Based Fuzzy Inference System," *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 3, 1993, pp. 665-685.

Appendix

This appendix summarizes the architecture of the ANFIS. For simplicity, we assume the fuzzy inference system under consideration has two inputs, x_1 and x_2, and one output, z. Suppose that the rule base contains two fuzzy if-then rules.

Rule 1: If x_1 is A_1 and x_2 is B_1, then $f_1=p_1x_1+q_1x_2+r_1$
Rule 2: If x_1 is A_2 and x_2 is B_2, then $f_2=p_2x_1+q_2x_2+r_2$

The ANFIS architecture is shown in Fig. A1, illustrating the fuzzy reasoning mechanism. We describe the node function in each layer of the ANFIS as follows: (Note that O_i^j denotes the output of the i-th node in layer j)

Layer 1: Each node in this layer generates membership grades of a linguistic label. For instance, the node function of the i-th node might be

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}},$$

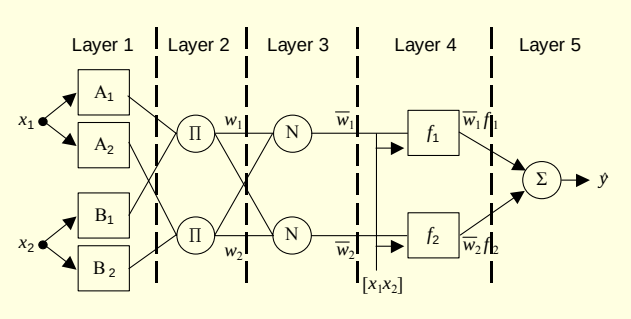

Fig. A1. ANFIS architecture.

where x is the input to node i; A_i is the linguistic label (small, large, etc) associated with this node; and {a, b, c} is the parameter set that changes the shape of the bell-shaped membership function. The parameters in this layer are referred to as the premise parameters.

Layer 2: Each node in this layer calculates the firing strength of each rule via multiplication.

$$O_i^2 = w_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2), \quad i = 1, 2$$

Layer 3: The i-th node of this layer calculates the ratio of the i-th rule's strength to the sum of all of the rules' firing strength.

$$O_i^3 = \overline{w_i} = \frac{w_i}{w_1 + w_2}$$

Layer 4: Node i in this layer has the following node function,

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x_1 + q_i x_2 + r_i),$$

where w_i is the output of layer 3 and $\{p, q, r\}$ is the parameter set. The parameters in this layer will be referred to as the consequent parameters.

Layer 5: The single node in this layer computes the overall output as the summation of all incoming signals

$$O_i^5 = \sum \overline{w_i} f_i = \frac{\sum w_i f_i}{\sum w_i}.$$

Dongwon Kim is currently a PhD candidate in electrical engineering at Korea University, Seoul, Korea. His research interests are in soft computing such as fuzzy systems, neural networks, genetic algorithms, GMDH-type algorithms, and their applications to semiconductor processes and complex systems. He received the Best Student Paper Competition Finalist Award at IEEE international conference on SMC 2003, the Excellent Presentation Award at SCIS & ISIS in 2002, the Prize from Seoam Scholarship Foundation in 2002, the Best Paper Award from KIEE in 2001, and he was the prize winner of the Student Paper Contest from IEEE Korea Council in 2000.

Byungwhan Kim received the BS and MS degrees in electrical engineering from Korea University in 1985 and 1987, and the PhD degree in the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, in 1995. He was a Principal Technical Staff at Hyundai Electronics in 1996 and a faculty member at Chonnam National University until 2000. He is now an Associate Professor in Electronic Engineering at Sejong University, Korea. His research interests include neurofuzzy modeling, diagnosis, and control of semiconductor processes. Dr. Kim was a recipient of the Electronic Packaging Fellow Award and Motorola Graduate Fellowship from Motorola and IEEE CPMT Society. He was awarded Academic Achievement Award from Sejong University. He is a member of KIEE and ICASE.

Gwi-Tae Park received the BS, MS and PhD degrees in electrical engineering from Korea University in 1975, 1977 and 1981. He was a Technical Staff Member in the Korea Nuclear Power Laboratory and an Electrical Engineering Faculty Member at Kwangwoon University in 1975 and 1978. He joined Korea University in 1981 where he is currently a Professor in Electrical Engineering. He was a Visiting Professor at the University of Illinois, UC and the University of Maryland, in 1984 and 1996. Dr. Park is presently serving as the President of the Intelligent Building System (IBS)-Korea. His research interests include soft computing technique and its application to semiconductor processes, adaptive signal processing, computer & control networks and their applications to robots, home automation, security systems, smart car, and IBS. He was a recipient of the KIEE Paper Award and Academic Achievement Award. He is a member of KIEE, ICASE, and KFIS.