A Novel Soft Computing Technique for the Shortcoming of the Polynomial Neural Network

Dongwon Kim, Sung-Hoe Huh, Sam-Jun Seo, and Gwi-Tae Park

Abstract: In this paper, we introduce a new soft computing technique that dwells on the ideas of combining fuzzy rules in a fuzzy system with polynomial neural networks (PNN). The PNN is a flexible neural architecture whose structure is developed through the modeling process. Unfortunately, the PNN has a fatal drawback in that it cannot be constructed for nonlinear systems with only a small amount of input variables. To overcome this limitation in the conventional PNN, we employed one of three principal soft computing components such as a fuzzy system. As such, a space of input variables is partitioned into several subspaces by the fuzzy system and these subspaces are utilized as new input variables to the PNN architecture. The proposed soft computing technique is achieved by merging the fuzzy system and the PNN into one unified framework. As a result, we can find a workable synergistic environment and the main characteristics of the two modeling techniques are harmonized. Thus, the proposed method alleviates the problems of PNN while providing superb performance. Identification results of the three-input nonlinear static function and nonlinear system with two inputs will be demonstrated to demonstrate the performance of the proposed approach.

Keywords: Fuzzy system, nonlinear system modeling, soft computing technique, unified framework, polynomial neural network.

1. INTRODUCTION

System modeling is important for analysis, control, and automation as well as scientific research. Recently, a significant amount of attention has been directed to advanced techniques of system modeling. When dealing with high-order nonlinear and multivariable systems, a vast amount of data is required for estimating all its parameters, which causes computational intensity and requires supercomputer facilities. As a modeling technique, there is a group method of data handling (GMDH)-type algorithm [1-8]. The GMDH algorithm [1-4] introduced by Ivakhnenko in the early 1970's is an analysis technique for identifying nonlinear relationships between the inputs and outputs of a given system. One of the GMDH-type algorithms is the polynomial neural network (PNN) [7,8]. The PNN provides an automated selection of essential input variables and builds hierarchical polynomial regressions as well as a partial description (PD), of required complexity. In addition, high-order regression often leads to a severely ill-conditioned system of equations. However, the PNN avoids this problem by constantly eliminating variables at each layer. Therefore, complex systems can be modeled without specific knowledge of the system or a massive amount of data. It is revealed in [7,8] that the PNN shows a superb performance in comparison to the previous fuzzy modeling methods. However, the approximation capabilities of the PNN are limited because of its critical handicap. If two or three numbers of input variables are considered, the PNN cannot be constructed flexibly. The above-mentioned problem can be solved by using the *soft computing (SC) technique.* SC, an innovative approach to constructing computationally intelligent systems, has become the center of attention. In confronting real-world computing problems, it is frequently advantageous to use several computing techniques synergistically rather than exclusively, resulting in construction of complementary hybrid intelligent systems. As a result, we employed one of three principal *soft computing* components, fuzzy systems, to overcome the well-known shortcoming of the conventional PNN. The fuzzy modeling method is a highly advanced technique and has been studied to deal with complex, ill-defined and uncertain systems in which the conventional mathematical model fails to give satisfactory results. In the fuzzy model, there is a premise part and a consequent part in an if-then rule. The input space of the premise in the model is parti-

Manuscript received July 8, 2003; accepted March 23, 2004. Recommended by Editorial Board member Eun Tai Kim under the direction of Editor Jin Bae Park.

Dongwon Kim, Sung-Hoe Huh, and Gwi-Tae Park are with the Department of Electrical Engineering, Korea University, 1, 5-ka, Anam-dong, Seongbuk-gu, Seoul 136-701, Korea (e-mail: {upground, sungh, gtpark}@korea.ac.kr).

Sam-Jun Seo is with the Department of Electrical and Electronic Engineering, Anyang University, 708-113 Anyang 5-dong, Manan-gu, Anyang-shi, Kyunggi-do, 430-714, Korea (e-mail: ssj@aycc.anyang.ac.kr).

tioned into several fuzzy subspaces, which is equal to the number of fuzzy rules [9]. The number of rules corresponds to the number of subspaces. Hence, the input space can be partitioned according to the number of rules in the fuzzy inference system (FIS). Consequently, we can deal with the critical restriction of the PNN by using the FIS. A space of input variables is partitioned into several subspaces by the fuzzy system and these subspaces are utilized as new input variables to the PNN architecture. Recently, hybrid architectures have appeared, as shown in [23-25], as a result of the combination of fuzzy rules, neural networks, FNN and PNN. But unfortunately, various experiments were not conducted in [23,24]. In [25], too many trials must be done to obtain comparable results. Furthermore, the genetic optimization algorithm must be applied to adjust design parameters such as learning rate, momentum term, and membership parameters. As such, the structures in [25] are immense. As can be seen from the section below, we did not employ any special optimization technique and various experiments are examined.

In this paper, a new soft computing technique has been investigated and a comparative study of this technique for nonlinear system modeling is presented. The proposed SC technique combines the fuzzy system and PNN into one methodology. In other words, this method is achieved by merging fuzzy systems and polynomial neural networks in one unified framework. As a result, we can find a workable synergistic environment and the main characteristics of the two modeling techniques are harmonized. Thus, the proposed method provides superb performance in comparison to the previous fuzzy modeling methods and also alleviates the problems of PNN such that if small numbers of input variables are considered, the PNN cannot be constructed flexibly.

2. POLYNOMIAL NEURAL NETWORK AND ITS PROBLEM DESCRIPTION

The PNN algorithm [7,8] is based on the group method of data handing (GMDH) [1] and utilizes a class of polynomials such as linear, quadratic, and modified quadratic types (see Table 1). These polynomials are referred to as partial descriptions (PDs). By choosing the most significant input variables and polynomial types among various types of forms available, we can obtain the PDs in each layer. The PNN is developed to identify the model of nonlinear complex systems by the use of the input-output data set. This data set is divided into two parts, that is, the training data set and the testing data set. The total number of nodes is given by the combination of a fixed number of inputs among entire input variables. The number of input variables and types of polynomials of each node are therefore determined in

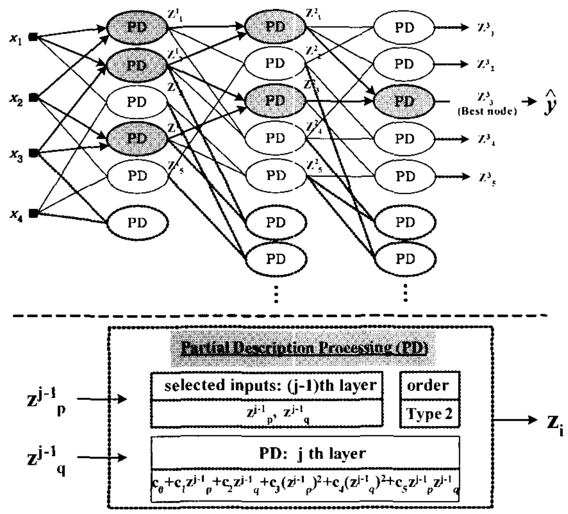


Fig. 1. Overall architecture of the PNN.

advance by the designer. By using the chosen input variables and type corresponding to each node, we construct a PD for each node. We determine the coefficients of the PD by the least square method using a given training data set, and finally we obtain the estimated output of each node. Furthermore, we evaluate each PD to check its predictive capability for the output variable using the testing data set. We then compare these values and choose the PDs that give the best predictive performance. In the sequel, we construct the second layer in the same way, considering the output variable of each chosen node in the first layer as the new input variables for the second layer. Afterwards, we repeat this procedure until the termination criterion has been satisfied. Once the final layer has been constructed, only the individual node characterized by the best performance is selected as the output node. The remaining nodes in that layer are discarded. Furthermore, all the nodes in the previous layers that have no influence over the selected output nodes are also removed by tracing the data flow path on each layer. Finally, the PNN model is obtained.

As an illustrative example, the overall architecture of the PNN through the design procedure stated above is provided in Fig. 1. In the figure, 4 input variables (x_1, ..., x_4), 3 layers, and a partial description (PD) processing example are considered. Where, Z^{j-1}_i means output of the ith node in the j-1th layer, which is employed as a new input of the jth layer. Black nodes have influence on the best node (output node) and these networks represent the ultimate PNN model. Meanwhile, solid line nodes have no influence over the output node. In addition, the dotted line nodes are excluded in choosing PDs with the best predictive performance in the corresponding layer owing to poor performance. Therefore, the solid line nodes and dotted line nodes should not be present in the final PNN model.

Table. 1. Different forms of the polynomials used in the proposed method.

No. of inputs / polynomial		1	2	3
FIS	PNN			
Type 1	·	c	c	c
Type 2	Type 1	l-1	l-2	l-3
Type 3	Type 2	q-1	q-2	q-3
Type 4	Type 3	mq-1	mq-2	mq-3

c: constants

l-2 (bilinear) = $c_0 + c_1x_1 + c_2x_2$

q-2 (biquadratic) = bilinear+ $c_3x_1^2 + c_4x_2^2 + c_5x_1x_2$

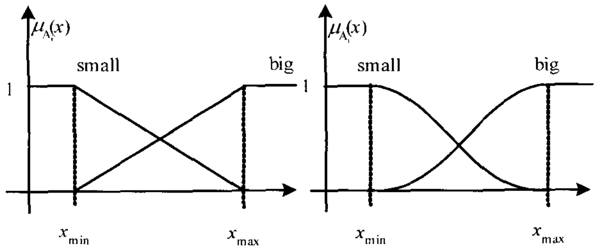
mq-2 (modified biquadratic) = bilinear $+c_3x_1x_2$

There are two types of PNN structures, namely the basic and the modified structure. The basic PNN structure consists of nodes for which the number of input variables is identical in every layer, but in the modified PNN structure, the number of input variables to each node in every layer can be changed. Each type comes with two cases, case 1 and case 2. In both structures the order of the polynomials in the PD may or may not vary from layer to layer. Detailed design procedures of the PNN algorithm can be found in [7, 8]. As shown in Fig. 1, if 1 to 3 input variables are considered, the PNN cannot construct a flexible neural architecture. In the following section, we propose a new soft computing technique to overcome this restriction, although a small number of input variables are considered.

3. FUZZY INFERENCE SYSTEM

The fuzzy inference system (FIS) [9-11] is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules, and fuzzy reasoning. In this paper, we use the Sugeno fuzzy model in which since each rule has a crisp output; the overall output is obtained via weighted average, thus avoiding the time-consuming process of defuzzification. When we consider fuzzy rules in the Sugeno fuzzy model, the consequent part can be expressed by constant or linear, quadratic, and modified quadratic polynomials as shown in Table 1 [10]. Depending on the types of consequent polynomials, the modeling performance will vary. Moreover, we can exploit various forms of membership functions (MFs) such as triangular and Gaussian types for fuzzy set in the premise part of the fuzzy rules. These are additional factors contributing to the flexibility of the proposed approach.

For simplicity, the nonlinear system to be identified is assumed to have two input variables and each input variable has two fuzzy sets, respectively. For the Sugeno-fuzzy model, a common rule set is as follows:



(a) Triangular MF. (b) Gaussian MF.

Fig. 2. Type of membership function typically used in the proposed method.

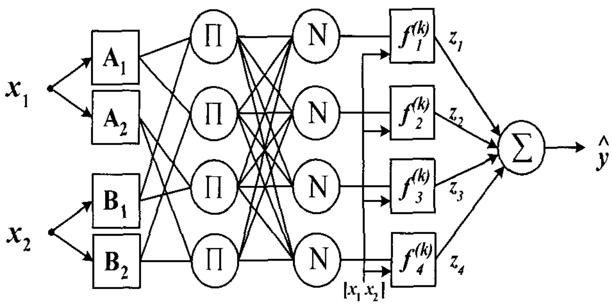


Fig. 3. ANFIS architecture that is equivalent to the Sugeno fuzzy model.

Rule 1: If x_1 is A_1 and x_2 is B_1, then $y_1 = f_1^{(k)}(x_1, x_2)$

Rule 2: If x_1 is A_1 and x_2 is B_2, then $y_2 = f_2^{(k)}(x_1, x_2)$

Rule 3: If x_1 is A_2 and x_2 is B_1, then $y_3 = f_3^{(k)}(x_1, x_2)$

Rule 4: If x_1 is A_2 and x_2 is B_2, then $y_4 = f_4^{(k)}(x_1, x_2)$

where A_i and B_i in the premise part of the rules are linguistic values (such as "small" or "big") associated with input variables x_1 and x_2, respectively. $f_j^{(k)}(x_1, x_2)$, (k=1, ..., 4) is one of the consequent polynomial functions with Type K in Table 1 for the j-th rule.

As depicted in Fig. 2, two types of membership functions for the A_i and B_i were examined, which include triangular and Gaussian. Fig. 3 is an adaptive neuro-fuzzy inference system (ANFIS) [11,12] architecture that is equivalent to a two-input Sugeno fuzzy model with four rules, where each input is assumed to have two associated MFs. The output of each rule, z_j (j=1, ...4), is used as an input variable to the next PNN.

For the adaptation of parameters in ANFIS [11],

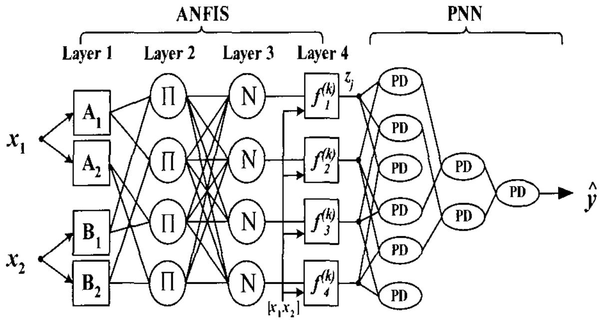


Fig. 4. Architecture of the proposed method.

gradient descent (GD) algorithms and the recursive least-squares estimation (RLSE) algorithm are employed for adjusting both premise and consequent parameters iteratively. However, we do not employ the complex hybrid learning algorithm but rather the general LSE for adjusting the coefficients in the consequent polynomial function with Type *K*.

4. NOVEL SOFT COMPUTING TECHNIQUE

The proposed soft computing technique for the shortcoming of the conventional PNN is presented in this section. This architecture is obtained by combining the ANFIS architecture with the cascade connection PNN. The architecture of the model can be constructed as shown in Fig. 4.

In what follows, we summarize the architecture of the network by considering the functionality of the individual layers.

[layer 1]: Each node *i* in this layer generates membership grades of a linguistic value. For instance, the node function of the *i*th node might be a Gaussian type such as

$$\mu_{A_i}(x) = \exp(-(\frac{x-c_i}{a_i})^2), \quad (1)$$

where *x* is the input to node *i*, and A_i is the linguistic value associated with this node function. *a* and *c* are the MFs width and center, respectively.

[layer 2]: Every node in this layer multiplies the incoming signals and sends the product out. For instance,

$$w_j = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2), \quad i=1,2, \quad j=1,2,\ldots4. \quad (2)$$

Each node output represents the firing strength of a rule.

[layer 3]: The *j*th node of this layer calculates the ratio of the *j*th rules' firing strength to the sum of the combined rules' firing strength,

$$\bar{w}_j = \frac{w_j}{w_1 + w_2 + w_3 + w_4}. \quad (3)$$

[layer 4]: Node *j* in this layer has the node function.

$$z_j = \bar{w}_j f_j^{(k)} = \bar{w}_j(c_{j0} + c_{j1}x_1 + c_{j2}x_2), \quad (4)$$

where the function of Type K $f_j^{(k)}$ is shown in Table 1.

[layer 5 or higher]: These layers consist of several steps. The steps are as follows;

Step 1- We define the input variables such as $x_1 = z_1, x_2 = z_2, x_3 = z_3, x_4 = z_4$ related to output variable y.

Step 2- The PNN structure is selected on the basis of the number of input variables and the order of PDs in each layer. Two kinds of PNN structures with two cases can be available. One is the basic PNN structure and the other is the modified PNN structure with Case 1 and Case 2, respectively.

Step 3- We determine the regression polynomial structure of a PD. When *r* input variables are selected from 4 input variables $x_1, x_2, \ldots, x_4$ in the preceding layer, the total number of nodes (PDs) in the current layer is determined by $k = 4!/((4-r)!r!$. Possible types of regression polynomials are contained in Table 1.

Step 4- The coefficients of the PD are determined using standard mean squared errors (MSE).

Step 5- Each PD in the current layer is estimated and evaluated with the training and testing data. Starting from the PD with the smallest performance index measured with the test data, PDs are selected by a pre-defined number *w*. The outputs of the chosen PDs serve as an input to the next layer. In this study, the *w* has been set to 30 in each layer.

Step 6- The PNN algorithm terminates when the number of layers predetermined by the designer is reached. Here, the number of total layers has been limited to 5.

Step 7- If the termination criterion is not satisfied, the next layer is constructed by repeating steps 3 through 7.

5. SIMULATION RESULTS

In this section, we show the performance of our new method for two well known types of nonlinear system modeling. One is a three-input nonlinear function which was studied previously in [13-17] and the other is a nonlinear static system already exploited in fuzzy modeling [9,18-22].

5.1. Three-input nonlinear function

We will demonstrate how the proposed approach can be employed to model a highly nonlinear function. The performance obtained in this example will also be compared with the individual fuzzy system and the PNN as well as earlier works. The function to be iden-

Table 2. Simulation results of fuzzy system.

MF		Fuzzy Inference System Type 1	Type 2	Type 3	Type 4
T	PI	8.83	0.11	373.92	402.54
	EPI	12.86	8.96	588.26	456.45
G	PI	10.30	2.e-12	3e-010	3e-12
	EPI	11.64	35.91	15316	32.63

T: Triangular MF, G: Gaussian MF

Table 3. Simulation results of PNN and its parameters.

Inputs	layer		Polynomial Neural Networks Type 1	Type 2	Type 3
2	1	PI	15.04	11.92	13.64
		EPI	15.19	13.28	14.12
	2	PI	13.90	5.90	10.17
		EPI	10.42	6.44	11.64
	3	PI	11.93	6.54	9.70
		EPI	9.26	4.92	10.84
	4	PI	11.18	3.30	11.21
		EPI	9.48	4.38	12.61
	5	PI	12.31	2.59	9.37
		EPI	10.31	8.52	13.38
3	1	PI	12.65	5.78	9.48
		EPI	11.04	6.81	12.95

tified is a three-input nonlinear system given by

$$y = (1 + x_1^{0.5} + x_2^{-1} + x_3^{-1.5})^2, \qquad (5)$$

which is widely used by Takagi and Hayashi [13], Sugeno and Kang [14], and Kondo [15] to test their modeling approaches. Forty pairs of input-output data are obtained from (5) [17].

The data is divided into a training data set and a testing data set. The training data set is used for the identification of the model, while the testing data set is used solely for the verification of the identified model. To compare the performance, the same performance index, average percentage error (APE) adopted in [13-17] is used.

$$APE = \frac{1}{m}\sum_{i=1}^{m}\frac{|y_i - \hat{y}_i|}{y_i}\times 100 \quad (\%), \qquad (6)$$

where m is the number of data pairs and y_i and $\hat{y}_i$ are the ith actual output and model output, respectively.

A series of comprehensive experiments was conducted and the results have been summarized as figures and tables. First, the fuzzy system and the PNN are exploited for the comparison of the proposed model, which has high accuracy and superior generalization capabilities.

Table 2 provides simulation results of the fuzzy system only. Triangular and Gaussian MF, and consequent polynomials of fuzzy rules are considered as modeling options. Here the FIS contains eight rules, with two MFs assigned to each input variable.

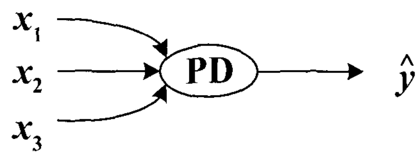
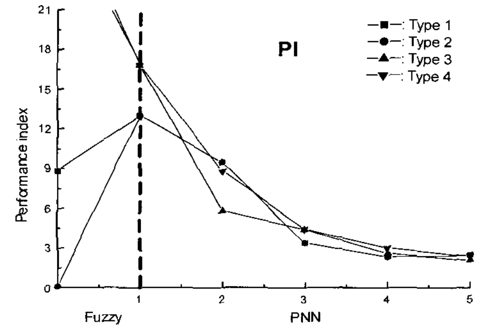
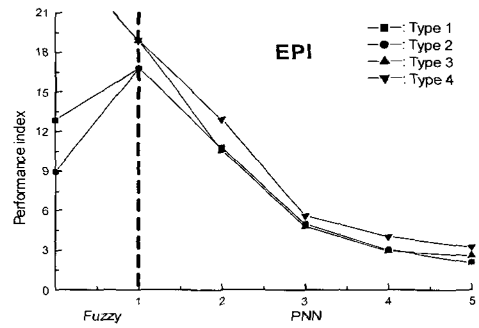


Fig. 5. Incomplete PNN architecture when 3 inputs are used.



(a) Performance index for the training data.



(b) Performance index for the testing data.

Fig. 6. Performance index of the proposed model (triangular MF and basic PNN).

In Table 3, the values of performance index via number of layers of the conventional PNN with Type 1 through Type 3 are shown. When three inputs are used, only 1 node is produced. Therefore, the PNN networks cannot be constructed completely for this nonlinear system. The incomplete network of the PNN in which the main characteristics of the PNN cannot be revealed is shown in Fig. 5.

Considering the design of the proposed model, 2 triangular and Gaussian MFs are assigned to each input variable just as in the case of the fuzzy system. Fig. 6 depicts the preferred results of the performance index, PI and EPI, produced in successive layers of the model according to the consequent type of the fuzzy rules. These values are produced when the triangular MF for the fuzzy rules and basic PNN are employed. The values of the performance index in the last layer and its modeling parameters are summarized in Table 4.

Table 4. Results of the 5th layer of the proposed model in Fig. 6 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Basic PNN				Results of 5th Layer	
Consequent Type	Membership Function	Inputs		Order			
		1st layer	2-5 layer	1st layer	2-5 layer	PI	EPI
Type 1	triangular	3	3	Type 3	Type 3	2.4403	2.0686
Type 2	triangular	3	3	Type 3	Type 3	2.4403	2.0686
Type 3	triangular	3	3	Type 3	Type 3	2.0687	2.5783
Type 4	triangular	3	3	Type 3	Type 3	2.3842	3.2501

Table 5. Results of the 5th layer of the proposed model in Fig. 7 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Basic PNN				Results of 5th Layer	
Consequent Type	Membership Function	Inputs		Order			
		1st layer	2-5 layer	1st layer	2-5 layer	PI	EPI
Type 1	Gaussian	3	3	Type 2	Type 2	0.34752	1.1011
Type 2	Gaussian	3	3	Type 2	Type 2	0.34752	1.1011
Type 3	Gaussian	3	3	Type 2	Type 2	0.34752	1.1011
Type 4	Gaussian	3	3	Type 2	Type 2	0.34752	1.1011

Table 6. Results of the 5th layer of the proposed model in Fig. 8 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Modified PNN				Results of 5th Layer	
Consequent Type	Membership Function	Inputs		Order			
		1st layer	2-5 layer	1st layer	2-5 layer	PI	EPI
Type 1	triangular	2	4	Type 3	Type 2	0.0004	0.72972
Type 2	triangular	2	4	Type 3	Type 2	0.0002	1.1311
Type 3	triangular	3	4	Type 1	Type 3	0.1609	0.3697
Type 4	triangular	3	4	Type 1	Type 3	0.1750	0.5269

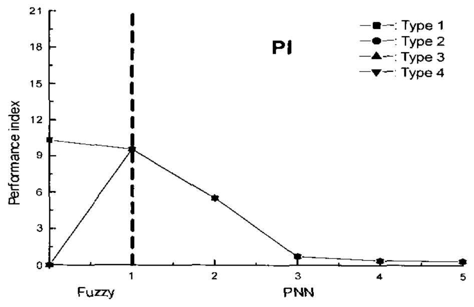
Table 7. Results of the 5th layer of the proposed model in Fig. 9 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Modified PNN				Results of 5th Layer	
Consequent Type	Membership Function	Inputs		Order			
		1st layer	2-5 layer	1st layer	2-5 layer	PI	EPI
Type 1	Gaussian	3	4	Type 3	Type 2	0.0006	1.1197
Type 2	Gaussian	2	4	Type 2	Type 2	0.0003	1.1259
Type 3	Gaussian	3	4	Type 3	Type 2	0.0006	1.3246
Type 4	Gaussian	3	4	Type 3	Type 2	0.0006	1.4057

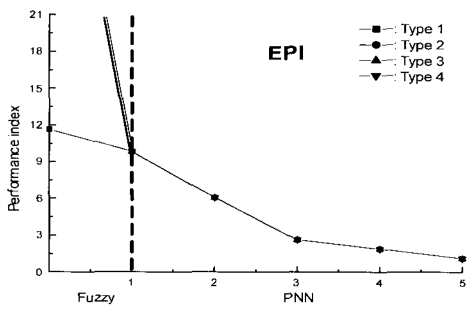
The preferred results of the proposed model with respect to the consequent type when the Gaussian MF for the fuzzy rules and basic PNN are employed are shown in Fig. 7 and its parameters are in Table 5. As shown in Fig. 7 and Table 5, if Gaussian MF and basic PNN are used for the proposed model, the results of the model are influenced very little by the consequent type of fuzzy rules. Meanwhile, the preferred results and their parameters with respect to the consequent type of fuzzy rules are shown in Fig. 8 and Table 6, respectively when the triangular MF for the fuzzy rules and modified PNN are employed. Here, we can see that the consequent types have a large effect on the performance index of the proposed model.

The preferred results and their parameters with respect to the consequent type of fuzzy rules are shown in Fig. 9 and Table 7, respectively when the Gaussian MF for the fuzzy rules and modified PNN are utilized. Here, the performance index of the proposed model is affected by the consequent type of fuzzy rules in a highly sensitive manner.

Fig. 10 shows the final structure of the proposed model and its identification performances when triangular MF for the fuzzy rule and modified PNN are utilized. For the consequent type, Type 1 is used and for the PNN, the 2 inputs-modified quadratic type in the 1st layer and 4 inputs-quadratic type in the 2nd layer or higher are used. The model output follows the

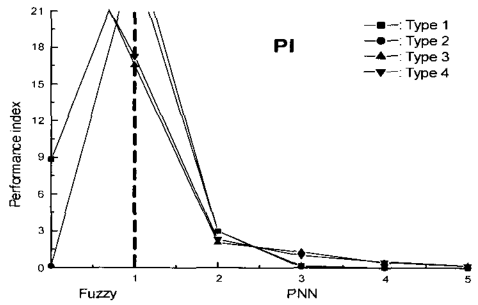


(a) Performance index for the training data.

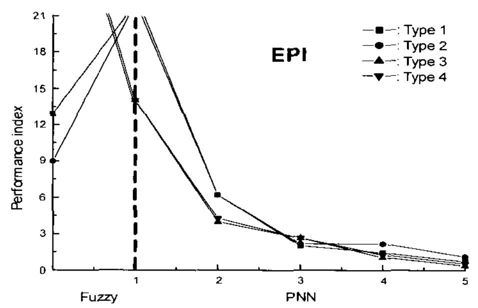


(b) Performance index for the testing data.

Fig. 7. Performance index of the proposed model (Gaussian MF and basic PNN).



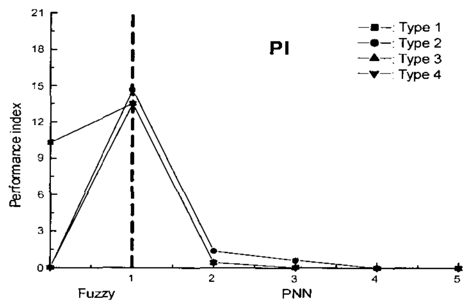
(a) Performance index for the training data.



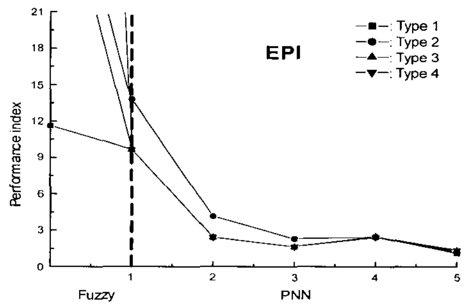
(b) Performance index for the testing data.

Fig. 8. Performance index of the proposed model (triangular MF and modified PNN).

actual output very well. Here, the values of the performance index of the model are equal to PI=0.0004, EPI=0.72972, respectively.



(a) Performance index for the training data.



(b) Performance index for the testing data.

Fig. 9. Performance index of the proposed model (Gaussian MF and modified PNN).

Fig. 10 shows the final structure of the proposed model and its identification performances when triangular MF for the fuzzy rule and modified PNN are utilized. For the consequent type, Type 1 is used and for the PNN, the 2 inputs-modified quadratic type in the 1st layer and 4 inputs-quadratic type in the 2nd layer or higher are used. The model output follows the actual output very well. Here, the values of the performance index of the model are equal to PI=0.0004, EPI=0.72972, respectively.

Table 8 provides a comparison of the proposed model with other techniques already proposed in the literature. The comparison is realized on the basis of the same performance index for the training and testing data set. The experimental results clearly reveal that the model outperforms the existing models both in terms of better approximation capabilities (PI) as well as superb generalization abilities (EPI).

5.2. Nonlinear static system

In this section, a double-input and single-output static function is chosen to be a target system for the new approach to the synergism of the fuzzy system and the PNN network. This function is represented as

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \qquad 1 \le x_1 , x_2 \le 5. \qquad (7)$$

This nonlinear static function has been widely used to evaluate modeling performance and has been reported by researchers, such as Sugeno [9], Kim [18,19],

Table 8. Comparison of identification error with some previous models.

Model		APE	
		PI	EPI
GMDH model[15]		4.7	5.7
Fuzzy[14]	Model 1	1.5	2.1
	Model 2	0.59	3.4
FNN [17]	Type 1	0.84	1.22
	Type 2	0.73	1.28
	Type 3	0.63	1.25
GD-FNN [16]		2.11	1.54
ANFIS [11]		0.043	1.066
FPNNs [25]		0.116	0.360
Fuzzy system (T^*+ Type 2)		0.115	8.967
PNN (2 inputs-Type 2)		2.597	8.525
Our model	*T+ basic PNN*	2.068	2.578
	G+ basic PNN	0.347	1.101
	T+ modified PNN	0.0004	0.729
	G+ modified PNN	0.0006	1.119

Table 9. Simulation results of fuzzy system.

	Fuzzy Inference System			
MF	Type 1	Type 2	Type 3	Type 4
T	0.268	0.089	0.037	0.087
G	0.321	0.055	0.021	0.046

Table 10. Simulation results of PNN and its parameters.

	Polynomial Neural Networks			
Inputs	layer	Type 1	Type 2	Type 3
2	1	0.299	0.096	0.268

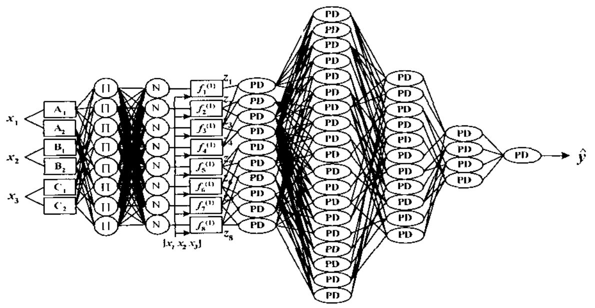
and Lin [22]. This system represents the nonlinear characteristic as shown in Fig. 11.

From the evenly distributed grid point of the input range of the above expression (7), 50 input-output data are obtained. The inputs are generated randomly and the corresponding output is then computed through the above relationship. The performance index is defined as the mean squared error

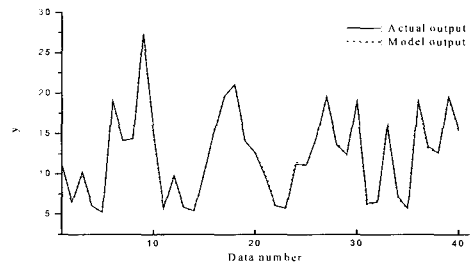
$$PI = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \quad (8)$$

with y_i being the actual output, $\hat{y}_i$ forming the output (estimate) of each node, and m as the number of data. Once again, a series of comprehensive experiments was conducted and the results are summarized in the same way as before with figures and tables.

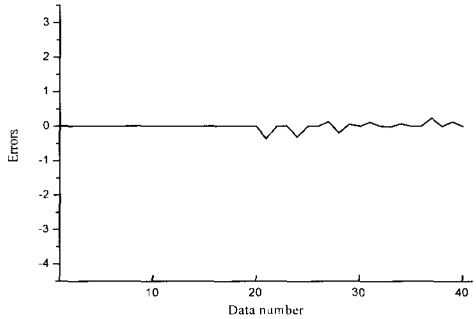
A detailed topology of the final network of the proposed model and its identification error are depicted in Fig. 15. When triangular MF for fuzzy rules and modified PNN are utilized, the value of the performance index of the network is equal to PI=0.00134. For the consequent type, Type 1 is used and for the PNN, the 2 inputs-modified quadratic type in the 1st layer and the 4 inputs-quadratic type in the 2-



(a) Final structure of the proposed model.



(b) Actual output versus model output.



(c) Error.

Fig. 10. Final structure of the proposed model and its identification performances.

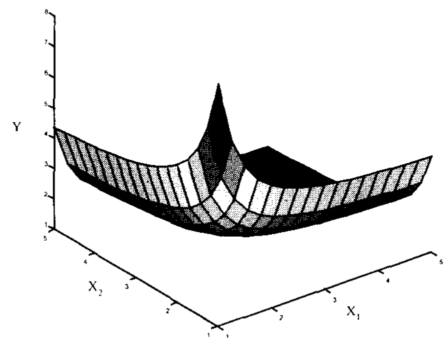


Fig. 11. Input-output relation of the two-input nonlinear system.

nd layer or higher are employed.

Table 13 contrasts the performance of the proposed method with other models studied in the previous literature. The experiment results show that our model

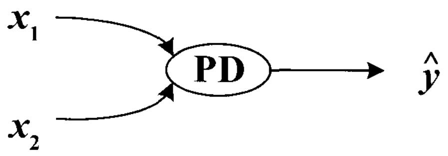


Fig. 12. Incomplete PNN architecture when 2 inputs are used.

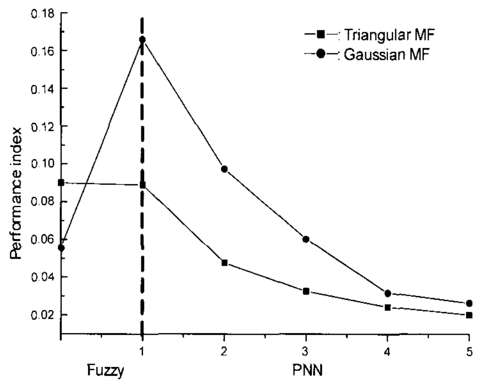


Fig. 13. Performance index of the proposed model (basic PNN used).

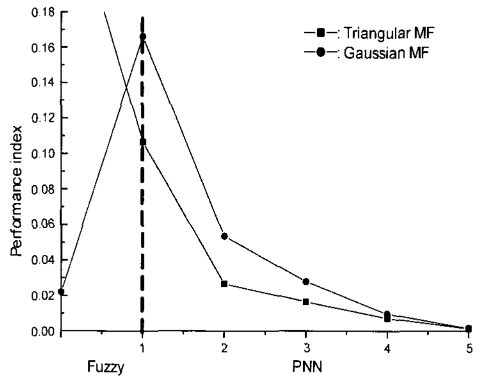
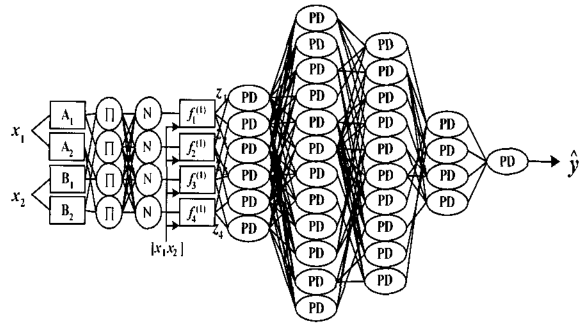


Fig. 14. Performance index of the proposed model (modified PNN used).

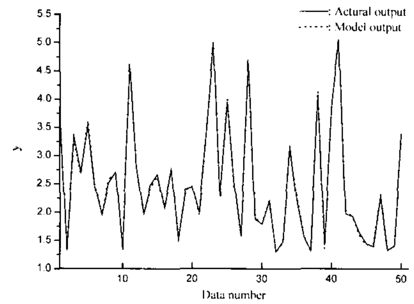
offers encouraging advantages and has superior performance. However, not every individual fuzzy system and conventional PNN can produce high-quality performance.

6. CONCLUSIONS AND FURTHER RESEARCH

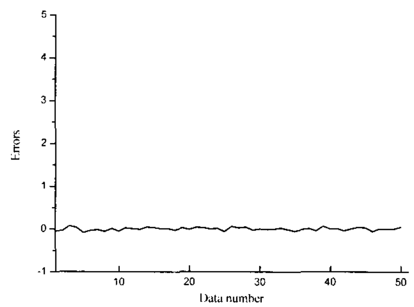
In this paper, we introduced a novel soft computing technique for the shortcoming of the polynomial neural network based on the fuzzy system and PNN. We discussed a diversity of topologies, and applied the model to nonlinear system modeling. As such, the proposed modeling technique is a sophisticated and versatile architecture capable of constructing models out of a limited data set and producing superb results. According to the types of membership functions, con-



(a) Final structure of the proposed model.



(b) Actual output versus model output.



(c) Error.

Fig. 15. Final structure of the proposed model and its identification performances.

sequent types of fuzzy rules, and structure of the PNN, we can see that the results of the model are varied. From the simulation results, we know that the proposed method is efficient and much more accurate than each individual fuzzy system and PNN, as well as other modeling methods. The proposed architecture is satisfactory from a performance point of view, but various problems still remain to be solved. There are many factors contributing to the flexibility of the model such as type of MF, number of MFs, consequent types, PNN structures and number of input and order (Type) for the PD. In this paper, these have been chosen by a trial-and-error procedure with a priori knowledge. Consequently, numerous results can

Table 11. Results of the proposed model in Fig. 13 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Basic PNN				Layer	PI
		Inputs		Order			
Consequent Type	MF	1st layer	2-5 layer	1st layer	2-5 layer		
						1	0.08893
						2	0.04781
Type 2	triangular	2	2	Type 2	Type 2	3	0.03259
						4	0.02439
						5	0.02013
						1	0.1661
						2	0.09740
Type 2	Gaussian	2	2	Type 2	Type 2	3	0.06030
						4	0.03183
						5	0.02650

Table 12. Results of the proposed model in Fig. 14 and its parameters.

Proposed Soft Computing Technique							
Fuzzy Rules		Modified PNN				Layer	PI
		Inputs		Order			
Consequent Type	MF	1st layer	2-5 layer	1st layer	2-5 layer		
						1	0.10661
						2	0.02662
Type 1	triangular	2	4	Type 3	Type 2	3	0.01638
						4	0.00711
						5	0.00134
						1	0.1661
						2	0.05343
Type 3	Gaussian	2	4	Type 2	Type 2	3	0.02785
						4	0.00963
						5	0.00161

Table 13. Comparison of identification error with some previous models.

Model		MSE PI
Sugeno [9]		0.079
Kim [18]		0.0197
Kim [19]		0.0089
Gomez-Skarmeta [20]		0.070
Hwang [21]		0.073
Lin [22]		0.0035
FPNNs [25]		0.0023
Fuzzy system (G+ Type 3)		0.021
Incomplete PNN (Type 2)		0.096
Our model	*T+ basic PNN*	0.020
	G+ basic PNN	0.026
	T+ modified PNN	0.0013
	G+ modified PNN	0.0016

be produced. As a result, there is no guarantee that the obtained model is the best one.

For performance improvement, the application of a simple network structure, optimal network topology, and a global optimization technique such as a genetic algorithm are needed.

REFERENCES

[1] A. G. Ivakhnenko, "Polynomial theory of complex systems," *IEEE Trans. on Syst. Man Cybern.*, vol. SMC-1, no. 1, pp. 364-378, 1971.

[2] A. G. Ivakhnenko, G. I. Krotov, and N. A. Ivakhnenko, "Identification of the mathematical model of a complex system by the self-organization method," *Theoretical Systems Ecology: Advances and Case Studies*, E. Halfon, Ed. New York: Academic, ch. 13, 1970.

[3] A. G. Ivakhnenko and N. A. Ivakhnenko, "Long-term prediction by GMDH algorithms using the unbiased criterion and the balance-of-variables criterion," *Sov. Automat. Contr.*, vol. 7, pp. 40-45, 1974.

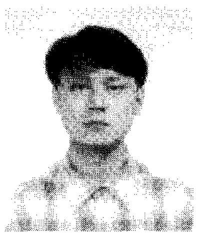
[4] A. G. Ivakhnenko and N. A. Ivakhnenko, "Long-term prediction by GMDH algorithms using the unbiased criterion and the balance-of-variables criterion, part 2," *Sov. Automat. Contr.*, vol. 8, pp. 24-38, 1975.

[5] S. J. Farlow, *Self-Organizing Methods in Modeling, GMDH Type-Algorithms*, New York: Marcel Dekker, 1984.

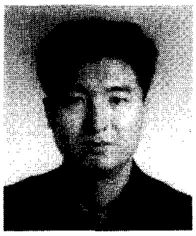
[6] D. W. Kim, "*Evolutionary Design of Self-Organizing Polynomial Neural Networks* (in Korean)," Master's thesis, Dept. Control Instrum.,

Wonkwang Univ., 2002.
[7] S. K. Oh, D. W. Kim, and B. J. Park, "A study on the optimal design of polynomial neural networks structure (in Korean)," *Trans. KIEE*, vol. 49D, no. 3, 2000.
[8] S. K. Oh and W. Pedrycz, "The design of self-organizing polynomial neural networks," *Inf. Sci.*, vol. 141, pp. 237-258, 2002.
[9] M. Sugeno, and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", *IEEE Trans. on Fuzzy Syst.* vol. 1, no. 1, pp. 7-31, 1993.
[10] D. W. Kim, S. K. Oh, and H. K. Kim, "A study on the self-organizing fuzzy polynomial neural networks," *Journal of KIEE*, vol. 11, no. 2, pp. 79-89, 2001.
[11] J. S. Jang, "ANFIS: Adaptive-networks-based fuzzy inference system," *IEEE Trans. on Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665-685, 1993.
[12] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy AND Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, 1997.
[13] H. Takagi and I. Hayashi, "NN-driven fuzzy reasoning," *Int. J. Approx. Reasoning*, vol. 5, no. 3, pp. 191-212, 1991.
[14] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, pp. 15-33, 1988.
[15] T. Kondo, "Revised GMDH algorithm estimating degree of the complete polynomial (in Japanese)," *Tran. Soc. Instrum. Control Eng.*, vol. 22, no. 9, pp. 928-934, 1986.
[16] S. Wu, M. J. Er, and Y. Gao, "A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks," *IEEE Trans. on Fuzzy Syst.*, vol. 9, no. 4, pp. 578-594, 2001.
[17] S. I. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE Trans. on Neural Netw.*, vol. 3, no. 5, pp. 801-806, 1992.
[18] E. T. Kim, M. K. Park, S. H. Ji and M. Park, "A new approach to fuzzy modeling", *IEEE Trans. on Fuzzy Syst.* vol. 5, no. 3, pp. 328-337, 1997.
[19] E. Kim, H. Lee, M. Park, and M. Park, "A simply identified Sugeno-type fuzzy model via double clustering," *Inf. Sci.*, vol. 110, pp. 25-39, 1998.
[20] A. F. Gomez-Skarmeta, M. Delgado, and M. A. Vila, "About the use of fuzzy clustering techniques for fuzzy model identification," *Fuzzy Sets Syst.*, vol. 106, pp. 179-188, 1999.
[21] H. S. Hwang, and K. B. Woo, "Linguistic fuzzy model identification," *IEE Proc.-Control Theory Appl.*, vol. 142, no. 6, pp. 537-544, 1995.
[22] Y. Lin, and G. A. Cunningham III, "A new approach to fuzzy-neural system modeling," *IEEE Trans. on Fuzzy Syst.* vol. 3, no. 2, pp. 190-198, 1995.
[23] D. W. Kim, J. H. Park, and G. T. Park, "Combination of fuzzy rule based model and self-organizing approximator technique: a new approach to nonlinear system modeling," *Proc. of Fuzz-IEEE*, pp. 1363-1367, 2003.
[24] D. W. Kim and G. T. Park, "Hybrid architecture of the neural networks and self-organizing approximator technique: a new approach to nonlinear system modeling," *Proc. of IEEE Int. Conf. Syst., Man, Cybern*, pp. 774-779, 2003.
[25] B. J. Park, W. Pedrycz, and S. K. Oh, "Fuzzy polynomial neural networks: hybrid architectures of fuzzy modeling," *IEEE Trans. on Fuzzy Syst.*, vol. 10, no. 5, pp. 607-621, 2002.

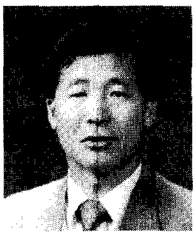
Dongwon Kim is currently a Ph.D. candidate in Electrical Engineering at Korea University, Seoul, Korea. His research interests are soft computing such as fuzzy systems, neural networks, genetic algorithms, GMDH-type algorithms, and their applications to complex systems.



Sung-Hoe Huh received his B.S. and M.S. degrees from Kookmin University, Seoul, Korea in 1994 and 1996. He is currently pursuing a Ph.D degree in the Department of Electrical Engineering at Korea University, Seoul, Korea. From 1998 to 2002, he was with the Information & Communication Research Center as a Researcher. Since 1996, he has been with the Intelligent System Research Center at the Korea Institute of Science and Technology, Seoul, Korea as a Student Researcher. His research interests are in intelligent control, digital control and motor drives.



Sam-Jun Seo received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Korea University in 1989, 1991 and 1996, respectively. He joined Anyang University in 1997 and is currently an Assistant Professor in Electrical and Electronic Engineering. His research interests include intelligent control, fuzzy systems, neural networks, and tele-operating systems.



Gwi-Tae Park received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from Korea University in 1975, 1977 and 1981, respectively. 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, respectively. 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, respectively.

Dr. Park is presently serving as the President of the Intelligent Building System (IBS)-Korea. His research interests include soft computing techniques, adaptive signal processing, computer & control networks and their applications to robots, home automation, security systems, smart car, and IBS.