A New Architecture of Genetically Optimized Self-Organizing Fuzzy Polynomial Neural Networks by Means of Information Granulation

Ho-Sung Park*, Sung-Kwun Oh**, and Tae-Chon Ahn*

* School of Electrical Electronic and Information Engineering, Wonkwang University, Iksan, Korea
(Tel : +82-63-850-6344; E-mail: (neuron, tcahn)@wonkwang.ac.kr)
**Department of Electrical Engineering, Suwon University, Hwaseong, Korea
(Tel : +82-31-229-8162; E-mail: ohsk@suwon.ac.kr)

Abstract: This paper introduces a new architecture of genetically optimized self-organizing fuzzy polynomial neural networks by means of information granulation. The conventional SOFPNNs developed so far are based on mechanisms of self-organization and evolutionary optimization. The augmented genetically optimized SOFPNN using Information Granulation (namely IG_gSOFPNN) results in a structurally and parametrically optimized model and comes with a higher level of flexibility in comparison to the one we encounter in the conventional FPNN. With the aid of the information granulation, we determine the initial location (apexes) of membership functions and initial values of polynomial function being used in the premised and consequence part of the fuzzy rules respectively. The GA-based design procedure being applied at each layer of genetically optimized self-organizing fuzzy polynomial neural networks leads to the selection of preferred nodes with specific local characteristics (such as the number of input variables, the order of the polynomial, a collection of the specific subset of input variables, and the number of membership function) available within the network. To evaluate the performance of the IG_gSOFPNN, the model is experimented with using gas furnace process data. A comparative analysis shows that the proposed IG_gSOFPNN is model with higher accuracy as well as more superb predictive capability than intelligent models presented previously.

Keywords: Self-organizing fuzzy polynomial neural networks, Information granulation, Genetic algorithms, accuracy and predictive capability.

1. INTRODUCTION

When the complexity of the system to be modeled increases, both experimental data and some prior domain knowledge (conveyed by the model developer) are of importance to complete an efficient design procedure. It is also worth stressing that the nonlinear form of the model acts as a two-edge sword: while we gain flexibility to cope with experimental data, we are provided with an abundance of nonlinear dependencies that need to be exploited in a systematic manner. In particular, when dealing with high-order nonlinear and multivariable equations of the model, we require a vast amount of data necessary for estimating all its parameters [1-2].

To help alleviate the problems, one among the first approaches along systematic design of nonlinear relationships between system's inputs and outputs comes under the name of a Group Method of Data Handling (GMDH). GMDH was developed in the late 1960's by Ivakhnenko[3-4] as a vechicle for identifying nonlinear relations between input and output variables. GMDH-type algorithms have been extensively used since the mid-1970's for prediction and modeling complex nonlinear processes. While providing with a systematic design procedure, GMDH comes with some drawbacks. To alleviate the problems associated with the GMDH, Self-Organizing Neural Networks (SONN, called "SOFPNN") were introduced by Oh and Pedrycz [5-7] as a new category of neural networks or neuro-fuzzy networks. Although the SOFPNN has a flexible architecture whose potential can be fully utilized through a systematic design, it is difficult to obtain the structurally and parametrically optimized network because of the limited design of the nodes located in each layer of the SOFPNN.

In this study, in considering the above problems coming with the conventional SOFPNN, we introduce a new structure and organization of fuzzy rules as well as a new genetic design approach. The new meaning of fuzzy rules, information granules melt into the fuzzy rules. In a nutshell, each fuzzy rule describes the related information granule. The determination of the optimal values of the parameters available within an individual FPN (viz. the number of input variables, the order of the polynomial, a collection of preferred nodes, and the number of MF) leads to a structurally and parametrically optimized network through the genetic approach. Similarly, we end up with an architecture which is more flexible as well as simpler than the conventional SOFPNN. To evaluate the performance of the proposed model, we exploit a well-known time series data [8]. Furthermore, the network is directly contrasted with several existing neurofuzzy models.

2. THE ARCHITECTURE OF FUZZY POLYNOMIAL NERUON (FPN) AND ITS TOPOLOGY

2.1 Architecture of Fuzzy Polynomial Neuron(FPN)

The FPN encapsulates a family of nonlinear "IF-THEN" ruels. As visualized in Fig. 1, the FPN consists of two basic functional modules. The first one, labeled by **F**, is a collection of fuzzy sets (here $\{A_l\}$ and $\{B_k\}$) that form an interface between the input numeric variables and the processing part realized by the neuron. Here, x_q and x_p denote input variables.

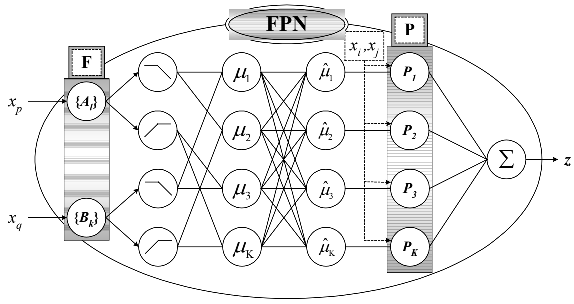


Fig. 1 A general topology of the generic FPN module.

The second module (denoted here by **P**) is about the function – based nonlinear (polynomial) processing. This nonlinear processing involves some input variables (x_i and x_j)

In other words, FPN realizes a family of multiple-input single-output rules. Each rule, refer again to Fig. 1, reads in the form

$$if\ x_p\ is\ \boldsymbol{A}_l\ and\ \boldsymbol{B}_k\ then\ z\ is\ P_{lk}(x_i, x_j, \boldsymbol{a}_{lk}) \qquad (1)$$

where $\boldsymbol{a}_{lk}$ is a vector of the parameters of the conclusion part of the rule while $P_{lk}(x_i, x_j, \boldsymbol{a}_{lk})$ denotes the regression polynomial forming the consequence part of the fuzzy rule which uses several types of high-order polynomials (linear, quadratic, and modified quadratic) besides the constant function forming the simplest version of the consequence; refer to Table 1.

Table 1 Different forms of regression polynomial building a FPN.

No. of inputs / Order of the polynomial	1	2	3
0(Type 1)	Constant	Constant	Constant
1(Type 2)	Linear	Bilinear	Trilinear
2(Type 3)	Quadratic	Biquadratic-1	Triquadratic-1
2(Type 4)		Biquadratic-2	Triquadratic-2

2.2 Topology of FPN

Proceeding with the overall SOFPNN architecture essential design decisions have to be made with regard to the number of input variables and the order of the polynomial forming the conclusion part of the rules as well as a collection of the specific subset of input variables. The consequence part can be expressed as a linear, quadratic, or modified quadratic polynomial as mentioned earlier. Especially for the consequence part, we consider two kinds of input vector formats in the conclusion part of the fuzzy rules of the 1^{st} layer, namely i) selected inputs and ii) entire system inputs, see Table 2.

Table 2 Polynomial type according to the number of input variables in the conclusion part of fuzzy rules.

Input vector / Type of the Consequence polynomial	Selected input variables in the premise part	Selected input variables in the consequence part
Type T	A	A
Type T*	A	B

Where notation **A**: Vector of the selected input variables (x_1, x_2,…, x_i), **B**: Vector of the entire system input variables(x_1, x_2, …x_i, x_j …), Type T: f(A)=f(x_1, x_2,…, x_i) - type of a polynomial function standing in the consequence part of the fuzzy rules, Type T*: f(B)=f(x_1, x_2, …x_i, x_j …) - type of a polynomial function occurring in the consequence part of the fuzzy rules.

Proceeding with each layer of SOFPNN, the design alternatives available within a single FPN can be carried out with regard to the entire collection of the input variables or its selected subset as they occur in the consequence part of fuzzy rules encountered at the 1^{st} layer. Following these criteria, we distinguish between two fundamental types (Type T, Type T*), namely

Type T- the input variables in the conditional part of fuzzy rules are the same as those occurring in the conclusion part of the fuzzy rules (Z_i of (A) FPN (x_p, x_q; x_i, x_j)).

Type T*- the entire collection of the input variables is kept as input variables in the conclusion part of the fuzzy rules (Z_i of (B) FPN (x_p, x_q; x_1, x_2, x_3, x_4)).

3. OPTIMIZATION OF IG_GSOFPNN

3.1 Information Granulation based on Hard C-Means clustering method

Information granulation is linked collections of objects (data points, in particular) drawn together by the criteria of indistinguishability, similarity or functionality [9]. It is a procedure to extract meaningful concepts from large collections of numerical data and an inherent activity of human being carried out with the purposed of better comprehension of the problem. We reveal the structure in data with the aid of Hard C-means clustering method[10], which leads to a collection of sets. We also construct the input interface of the fuzzy model, which plays a crucial role in transforming the collected data into a conceptual framework using the obtained information granules on the above phase. Through HCM, we determine the initial location (apexes) of membership functions and initial values of polynomial function being used in the premise and consequence part of the fuzzy rules respectively.

The fuzzy rules of IG_gSOFPNN is given as follows:

$$R^j : If\ x_1\ is\ A_{ji}\ and \cdots x_k\ is\ A_{jk}\ then\ y_j - M_j$$
$$= f_j\{(x_1 - v_{j1}), (x_2 - v_{j2}), \cdots, (x_k - v_{jk})\}$$

Where, A_{jk} mean the fuzzy set, the apex of which is defined as the center point of information granule (cluster). M_j and v_{jk} are the center points of new created input-output variables by information granule.

3.2 Genetic optimization of IG_gSOFPNN

GAs is a stochastic search technique based on the principles of evolution, natural selection, and genetic recombination by simulating "survival of the fittest" in a population of potential solutions (individuals) to the problem at hand. GAs are capable of globally exploring a solution space, pursuing potentially fruitful paths while also examining random points to reduce the likelihood of setting for a local optimum. The main features of genetic algorithms concern individuals viewed as strings, population-based optimization (carried out in the genotype space) and stochastic search mechanisms (such as selection and crossover). In order to enhance the learning of the FPNN and augment its performance, we use genetic algorithms to carry out the structural optimization of the network by optimally selecting such parameters as the number of input variables(nodes), the order of polynomial, input variables and the number of MF within a FPN.

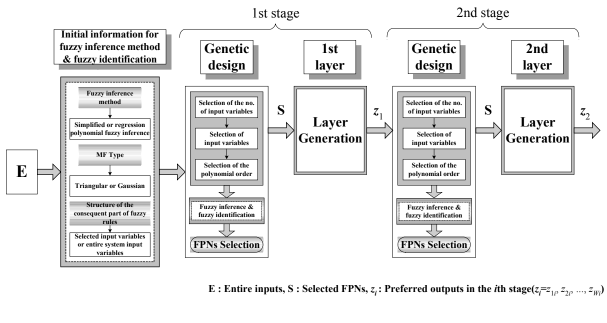


Fig. 2 Overall genetically-driven structural optimization process of IG_gSOFPNN.

In this study, we consider GA that uses serial method of binary type, roulette-wheel as the selection operator, one-point crossover, and an invert operation in the mutation operator [11]. To retain the best individual and carry it over to the next generation, we use elitist strategy [12]. The overall genetically-driven structural optimization process of IG_gSOFPNN is shown in Fig. 2.

4. THE ALGORITHM AND DESIGN PROCEDURE OF IG_GSOFPNN

The framework of the design procedure of the gSOFPNN with aid of the Information granulation (IG) comprises the following steps.

[Step 1] *Determine system's input variables.*
Define system's input variables $\boldsymbol{x}_i$(i=1, 2, …, n) related to the output variable y. If required, the normalization of input data is carried out as well.

[Step 2] *Form training and testing data.*
The input-output data set ($\boldsymbol{x}_i$, y_i)=(x_{1i}, x_{2i}, …, x_{ni}, y_i), i=1, 2, …, N (with N being the total number of data points) is divided into two parts, that is a training and testing dataset.

[Step 3] *Determine apexes of MF by HCM clustering method.*

[Step 4] *Decide initial design information for constructing the IG_gSOFPNN structure.*
Here we make decisions as to the number of essential design parameters.
(a) Fuzzy inference method and fuzzy identification scheme that is -Fuzzy inference method, -MF type: Triangular or Gaussian-like MF, -No. of MFs per each input of a node (or FPN), -Structure of the consequence part of fuzzy rules.
(b) Termination aspect is considered - The maximum number of layers with intent to achieve a sound balance between model accuracy and its complexity.
(c) The maximum number of input variables coming to each node in the corresponding layer.
(d) The total number (W) of nodes to be retained (selected) at the next generation of the FPNN algorithm.

[Step 5] *Decide a structure of the FPN structure using genetic design.*
This phase concerns the selection of the number of input variables, the polynomial order, the input variables, and the no. of MFs to be assigned in each node of the corresponding layer as shown in Fig. 3 . These important decisions are carried out through an extensive genetic optimization. The 1st sub-chromosome contains the number of input variables, the 2nd sub-chromosome involves the order of the polynomial of the node, the 3rd sub-chromosome contains input variables, and the 4th sub-chromosome (remaining bits) is contains the number of membership functions coming to the corresponding node (FPN). All these elements are optimized when running the GA.

[Step 6] *Carry out fuzzy inference and coefficient parameters estimation for fuzzy identification in the selected node(FPNs).*
New construction of consequence part of the fuzzy rule by HCM clustering method.

[Step 7] *Select nodes (FPNs) with the best predictive capability and construct their corresponding layer.*
To evaluate the performance of FPNs (nodes) constructed using the training dataset, the testing dataset is used. Based on this performance index, we calculate the fitness function to every node in each layer. The fitness function reads as

$$F(fitness \quad function) = \frac{1}{1+EPI} \tag{2}$$

where EPI denotes the performance index for the testing data (or validation data). In this case, the model is obtained by the training data and EPI is obtained from the testing data (or validation data) of the FPNN model constructed by the training data.

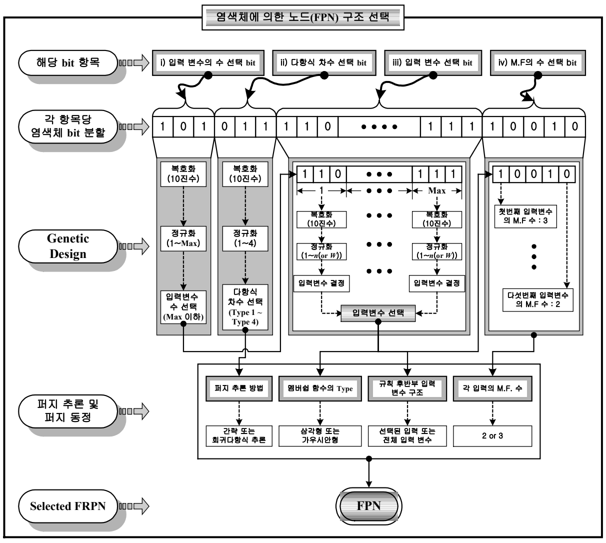


Fig. 3 The FPN design available in IG_gSOFPNN architecture by using a chromosome of GAs.

[Step 8] *Check the termination criterion.*
As far as the performance index is concerned, a termination is straightforward and comes in the form,

$$F_1 \le F_* \tag{3}$$

Where, F_1 denotes a maximal fitness value occurring at the current layer whereas F_* stands for a maximal fitness value that occurred at the previous layer.

[Step 9] *Determine new input variables for the next layer.*
If (3) has not been met, the model is expanded. The outputs of the preserved nodes (z_{1i}, z_{2i}, …, z_{Wi}) serves as new inputs to the next layer (x_{1j}, x_{2j}, …, x_{Wj})(j=i+1).

The IG_gSOFPNN algorithm is carried out by repeating steps 3-9 of the algorithm.

5. EXPERIMENTAL STUDY

We illustrate the performance of the network and elaborate on its development by experimenting with data coming from the gas furnace process. The time series data (296 input-output pairs) resulting from the gas furnace process has been intensively studied in the previous literature [8,]. The delayed terms of methane gas flow rate, $u(t)$ and carbon dioxide density, $y(t)$ are used as system input variables such as $u(t$-3), $u(t$-2), $u(t$-1), $y(t$-3), $y(t$-2), and $y(t$-1). The output variable is $y(t)$. The first part of the dataset (consisting of 148 pairs) was used for training. The remaining part of the series serves as a testing set. To come up with a quantitative evaluation of network, we use the standard MSE performance index. We choose the input variables of nodes in the 1st layer of SONN architecture from these input variables. Table 3 summarizes the list of parameters used in the genetic optimization of the IG_gSOFPNN.

In the optimization of each layer, we use 150 generations, 100 populations, a string of 41 bits, crossover rate equal to 0.65, and the probability of mutation set up to 0.1. A chromosome used in the genetic optimization consists of a string including 4 sub-chromosomes. The 1st chromosome contains the number of input variables, the 2nd chromosome contains the order of the polynomial, the 3rd chromosome contains input variables, and finally the 4th chromosome contains the number of membership function. The numbers of bits allocated to each sub- chromosome are equal to 3, 3, 30, and 5, respectively.

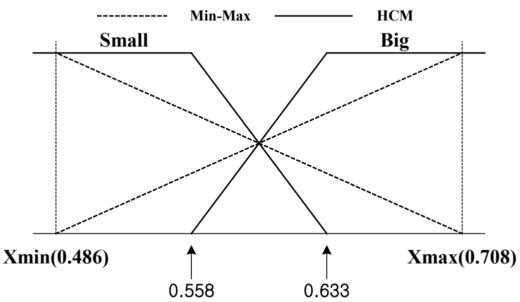
Table 3 Computational aspects of the genetic optimization of IG_gSOFPNN.

Parameters		1st layer	2nd to 3rd layer
GA	Maximum generation	150	150
	Total population size	100	100
	Selected population size (W)	30	30
	Crossover rate	0.65	0.65
	Mutation rate	0.1	0.1
	String length	3+3+30+5	3+3+30+5
IG_gSOFPNN	Maximal no.(Max) of inputs to be selected	$1 \le l \le$ Max(2~3)	$1 \le l \le$ Max(2~3)
	Polynomial type (Type T) of the consequent part of fuzzy rules(#)	$1 \le T \le 4$	$1 \le T \le 4$
	Consequent input type to be used for Type T (##)	Type T*	Type T
	Membership Function (MF) type	Triangular	Triangular
		Gaussian-like	Gaussian-like
	No. of MFs per input	2 or 3	2 or 3

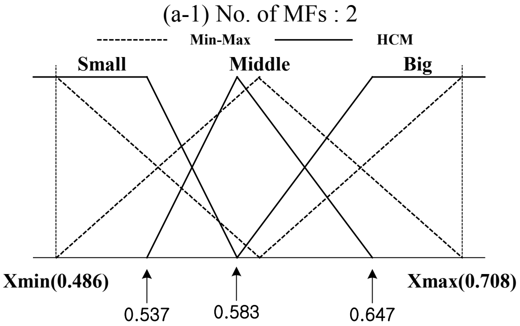
l, T, Max: integers, # and ## : refer to Tables 1-2 respectively.

The population size being selected from the total population size (100) is equal to 30. The process is realized as follows.100 nodes are generated in each layer of the network. The parameters of all nodes generated in each layer are estimated and the network is evaluated using both the training and testing data sets. Then we compare these values and choose 30 nodes that produce the best (lowest) value of the performance index. The maximal number (Max) of inputs to be selected is confined to two to three (2-3).

Fig. 4 shows the membership functions apexes of two input variable ($u(t\text{-}3)$, $y(t\text{-}1)$) according to the partition of fuzzy input spaces by a Min-Max method and the HCM clustering method.

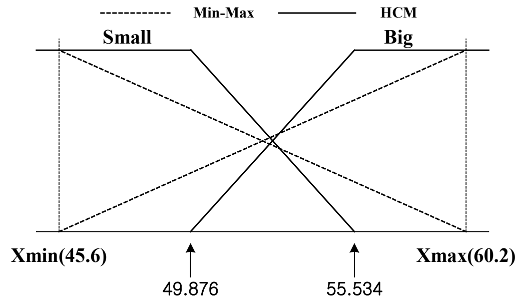


(a-1) No. of MFs : 2

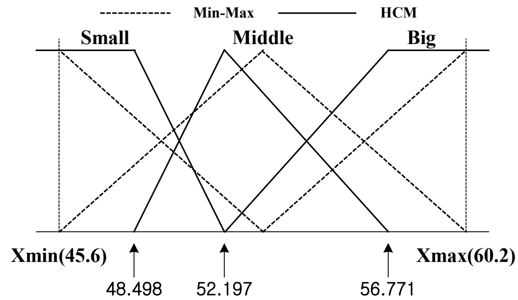


(a-2) No. of MFs : 3

(a) $u(t\text{-}3)$



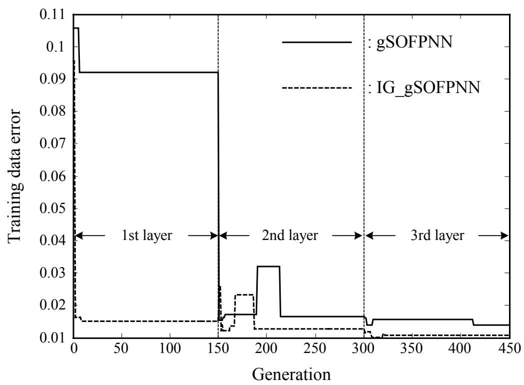
(b-1) No. of MFs : 2



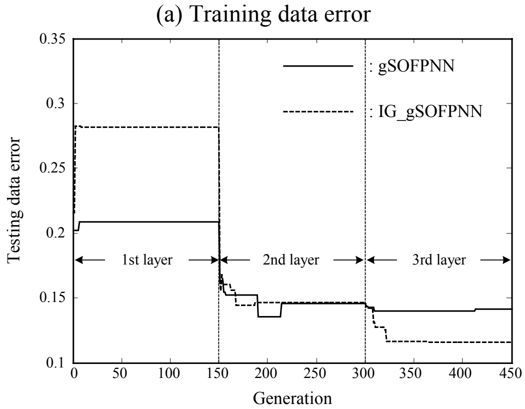
(b-2) No. of MFs : 3

(b) $y(t\text{-}3)$

Fig. 4 Definition of initial membership functions of gas furnace process.



(a) Training data error



(b) Testing data error

Fig. 5 The optimization process quantified by the values of the performance index.

Fig. 5 illustrates the different optimization process between gSOFPNN and the proposed IG_gSOFPNN by visualizing the values of the performance index obtained in successive generations of GA when using Type T. It also shows the optimized network architecture when using the Triangular MF(the Max of inputs to be selected is set to 2 with the structure composed of 3 layers).

Table 4 contains a comparative analysis including several previous fuzzy and neuro-fuzzy models. Compared with these models, the IG_gSOFPNN emerges as the one with high accuracy and improved prediction capability.

Table 4 Comparative analysis of the performance of the network; considered are models reported in the literature.

<table>
<tr><th colspan="4">Model</th><th>PI</th><th>PIs</th><th>EPIs</th></tr>
<tr><td colspan="4">Box and Jenkin's model [8]</td><td>0.710</td><td></td><td></td></tr>
<tr><td colspan="4">Oh and Pedrycz's model [13]</td><td></td><td>0.020</td><td>0.271</td></tr>
<tr><td colspan="4">Kim et al.'s model [14]</td><td></td><td>0.034</td><td>0.244</td></tr>
<tr><td colspan="4">Lin and Cunningham's model [15]</td><td></td><td>0.071</td><td>0.261</td></tr>
<tr><td rowspan="2">FPNN [16]</td><td>CASE I</td><td>Gaussian</td><td>5th layer</td><td></td><td>0.016</td><td>0.116</td></tr>
<tr><td>CASE II</td><td>Gaussian</td><td>5th layer</td><td></td><td>0.012</td><td>0.125</td></tr>
<tr><td colspan="2" rowspan="2">HFPNN [17]</td><td>Triangular</td><td>3rd layer</td><td></td><td>0.020</td><td>0.106</td></tr>
<tr><td>Gaussian</td><td>3rd layer</td><td></td><td>0.012</td><td>0.127</td></tr>
<tr><td rowspan="2">Proposed IG_gSO FPNN</td><td rowspan="2">Max=2 (Type T*)</td><td>Triangular</td><td>3rd layer</td><td></td><td>0.006</td><td>0.116</td></tr>
<tr><td>Gaussian</td><td>3rd layer</td><td></td><td>0.006</td><td>0.121</td></tr>
</table>

PI - performance index over the entire data set.
PIs - performance index on the training data set.
EPIs - performance index on the testing data set.

6. CONCLUDING REMARKS

In this study, we have developed the new architecture and formed the semantics of fuzzy rules and investigated the genetically optimized Self-Organizing Fuzzy Polynomial Neural Networks by means of Information Granulation (IG_gSOFPNN), and discussed their topologies.

The GA-based design procedure applied at each stage (layer) of the SOFPNN driven to information granulation leads to the selection of the preferred nodes (FPNs) with optimal local characteristics (such as the number of input variables, the order of the consequent polynomial of fuzzy rules, the number of membership functions, and a collection of specific subset of input variables) available within the FPN. These options contribute to the flexibility of the resulting architecture of the network. The comprehensive experimental studies involving well-known datasets quantify a superb performance of the network in comparison to the existing fuzzy and neuro-fuzzy models. More importantly, through the proposed framework of genetic optimization we can efficiently search for the optimal network architecture (being both structurally and parametrically optimized) and this design facet becomes crucial in improving the performance of the resulting model.

ACKNOWLEDGMENTS

This work has been supported by KESRI(R-2004-B-133 -01), which is funded by MOCIE(Ministry of commerce, industry and energy).

REFERENCES

[1] V. Cherkassky, D. Gehring, and F. Mulier, "Comparison of adaptive methods for function estimation from samples", *IEEE Trans. Neural Networks*, Vol. 7, pp. 969 984, July 1996.

[2] J. A. Dicherson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules", *IEEE Trans. Syst., Man, Cybernetics*, Part B, Vol. 26, pp.542 560, Aug. 1996.

[3] A. G. Ivakhnenko, "Polynomial theory of complex systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-1, pp. 364-378, 1971.

[4] A. G. Ivakhnenko and H. R. Madala, *Inductive Learning Algorithms for Complex Systems Modeling*, CRC Press, London, 1994.

[5] S. K. Oh and W. Pedrycz, "The design of self-organizing Polynomial Neural Networks", *Information Science*, Vol. 141, pp. 237-258, 2002.

[6] S. K. Oh, W. Pedrycz, and B. J. Park, "Polynomial Neural Networks Architecture: Analysis and Design", *Computers and Electrical Engineering*, Vol. 29, pp. 703-725, 2003.

[7] S. K. Oh and W. Pedrycz, "Fuzzy Polynomial Neuron-Based Self-Organizing Neural Networks", *Int. J. of General Systems*, Vol. 32, pp. 237-250, 2003.

[8] D. E. Box and G. M. Jenkins, *Time Series Analysis, Forcasting and Control*, California: Holden Day, 1976.

[9] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic", *Fuzzy Sets Syst*, Vol. 90, pp. 111-117, 1997.

[10] Bezdek, J.C, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, Plenum (1981).

[11] Z. Michalewicz, "*Genetic Algorithms + Data Structures = Evolution Programs*", Springer-Verlag, Berlin Heidelberg, 1996.

[12] D. Jong, K. A., "*Are Genetic Algorithms Function Optimizers?*", Parallel Problem Solving from Nature 2, Manner, R. and Manderick, B. eds., North-Holland, Amsterdam.

[13] S. K. Oh and W. Pedrycz, "Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems", *Fuzzy sets and Systems*, Vol. 115, No. 2, pp. 205-230, 2000.

[14] E. T. Kim, et al, "A simple identified Sugeno-type fuzzy model via double clustering", *Information Science*, Vol. 110, pp. 25-39, 1998.

[15] Y. Lin, G. A. Cunningham III, "A new approach to fuzzy-neural modeling", *IEEE Trans. Fuzzy Systems,* Vol. 3, No. 2, pp. 190-197, 1995.

[16] H. S. Park, S. K. Oh, and Y. W. Yoon, "A New Modeling Approach to Fuzzy-Neural Networks Architecture", *Journal of Control, Automation and Systems Engineering*, Vol. 7, No. 8, pp. 664-674, Aug., 2001.

[17] S. K. Oh, W. Pedrycz, and D. W. Kim, "Hybrid Fuzzy Polynomial Neural Networks", *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems,* Vol. 10, No. 3, pp. 257-280, June, 2002.