Proceedings of the 13th KACC, October 1998

Identification of Fuzzy Systems by means of the Extended GMDH Algorithm

Chun Seong Park, Jae Ho Choi and Sung Kwun Oh
Dept. of Control & Instrumentation Engineering, Wonkwang Univ., Iksan, Korea
Tel : +82-653-850-6342
Fax : +82-653-853-2196
Email : ohsk@wonnms.wonkwang.ac.kr

Abstract

A new design methology is proposed to identify the structure and parameters of fuzzy model using PNN and a fuzzy inference method. The PNN is the extended structure of the GMDH(Group Method of Data Handling), and uses several types of polynomials such as linear, quadratic and cubic besides the biquadratic polynomial used in the GMDH. The FPNN(Fuzzy Polynomial Neural Networks) algorithm uses PNN(Polynomial Neural networks) structure and a fuzzy inference method. In the fuzzy inference method, the simplified and regression polynomial inference methods are used. Here a regression polynomial inference is based on consequence of fuzzy rules with a polynomial equations such as linear, quadratic and cubic equation. Each node of the FPNN is defined as fuzzy rules and its structure is a kind of neuro-fuzzy architecture. In this paper, we will consider a model that combines the advantage of both FPNN and PNN. Also we use the training and testing data set to obtain a balance between the approximation and generalization of process model. Several numerical examples are used to evaluate the performance of the our proposed model.

1. Introduction

Recently, many researchers have had much interest in various method for system modeling. Among them, mathematical modeling methods such as regression techniques were widely used to identify and predict the linear systems based on input-output data. However, the mathematical models to express dynamic analysis of nonlinear real system, have had lots of problems in the selection, of variables constructing the model among many input-output variables, and of model structure.

In general, higher-order equations require too many data against estimating all system parameters in mathematical models. To solve the problem, the PNN based on GMDH was first introduced by A. G. Ivakhnenko[1]. The GMDH has been used to synthesize the PNN-the building blocks of modeling methodology. This approximation technique based on the perceptron principle with a neural network-type architecture has been applied to modeling, identification and prediction to the input-output relationship of a nonlinear process system with limited data sets.

In this paper, FPNN algorithm is proposed to estimate the structure and parameters of fuzzy model, using the PNN based on the GMDH algorithm and the fuzzy modeling method. The new algorithm fuses the PNN algorithm and fuzzy inference by replacement of each neuron of the PNN with fuzzy implications rules, in order to model the nonlinear process system with limited data sets, namely, to identify the premise structure and parameters of fuzzy implications rules. The premise fuzzy membership of each input variable uses Gaussian functions obtained by heuristics. The consequence utilizes the simplified inference consisting of regression polynomial. The optimal consequence parameters are obtained by least square method. We also will consider a extended FPNN that combines the advantages of both FPNN and PNN. And we use the training and testing data set to obtain a balance between the approximation and the generalization of process model by means of using external criterion of the testing data.

2. The Structure and Algorithm of GMDH

2.1 The Structure of GMDH

We start by computing the regression equations for each pair of input variables x_i and x_j and output y of the object system which desires to modeling.

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j \qquad (1)$$

This will give us $m(m-1)/2$ high-order variables for predicting the output y in place of the original m variables $x_1, x_2, \cdots, x_m$. After finding these regression equations from a set of input-output observations, we then find out which ones to save. This will give us a collection of quadratic regression models which best predictly. We now use each of the quadratic equation that we have just computed and generate new independent observations which will replace the original obsevations of the variables $x_1, x_2, \cdots, x_m$. From these new independent variables we will combine them exactly as we did before. That is, we compute all of the quadratic regression equations of y versus these new variables. This will give us a new collection of $m(m-1)/2$ regression equation for predicting y from the new variables, which in turn are estimates of y from the previous equations. We now merely select the best of these new estimates, generate new independent variables from selected equations to replace the old, and combine all pair of these new variables. We continue this process until the regression equation begin to have a poorer predictability power than did the previous ones. In other words, the model will start to become overfitted. After stopping we pick the best of the quadratic polynomials in that generation. What we have is an estimate of y as the quadratic of two variables, which are themselves quadratic of two more variables. In other words, if we were to make the necessary algebraic substitutions, we would arrive at a very complicated polynomial of the form which is known as the Ivakhneko polynomial.

$$y = a + \sum_{i=1}^{m} b_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} d_{ijk} x_i x_j x_k + \cdots \qquad (2)$$

where, $X(x_1, x_2, \cdots, x_m)$ is input variables vector and $A(a, b_i, c_{ij}, d_{ij}, \cdots)$ is vector of coefficients or weight of the Ivakhnenko polynomials. Components of the input vector X can be independent variables, functional forms or finite difference terms. The Algorithm allows to find simultaneously the structure of model and modelled system ouput on the values of most significant inputs of the system.

2.2 The improved GMDH Algorithms

We will select the model of the desired system by the selection of input variables, the structure and parameters of model and external criterion. And we will take final optimal model by heuristic self-organizing model estimation of the complex systems. Searching for the optimal configuration in the space of all possible polynomial neural networks is untractable and requires a set of hruristics. A series if heuristics is, therefore, defined to prune the search as follows;

Step 1. Begining the GMDH algorithm by collecting regression-type data. that is, n observation of the original m input variables $x_1, x_2, \cdots, x_m$ and output variables y.

Step 2. Constructing new variables $z_1, z_2, \cdots, z_k$: select the two variables x_p and x_q in the previous layer's output variables, and construct the regression equations for each pair of input polynomial as follows.

$$z_k = c_0 + c_1x_p + c_2x_p + c_3x_p^2 + c_4x_p^2 + c_5x_px_q \qquad (3)$$

where, $k = 1, 2, \cdots, n(n-1)/2$ and $c_0, c_1, \cdots, c_5$ is the coefficients. Also we can use the regression polynomials such as Table 1. in place of eqn. (3).

Table 1. The node's regression equations considered for polynomial neural network synthesis

	1	2	3
1	linear	bilinear	trilinear
2	quadratic	biquadratic	triquadratic
3	cubic	bicubic	tricubic

where,

$$\text{trilinear} = C_0 + C_1x_1 + C_2x_2 + C_3x_3$$
$$\text{triquardratic} = \text{trilinear} + C_4x_1x_2 + C_5x_1x_3 + C_6x_2x_3 + C_7x_1^2 + C_8x_2^2 + C_9x_3^2$$
$$\text{tricubic} = \text{triquadratic} + C_{10}x_1x_2x_3 + C_{11}x_1^3 + C_{12}x_2^3 + C_{13}x_3^3$$

Step 3. Estimate the coefficients of each regression polynomial : In the bilinear regression polynomial, we compute the coefficients $c_0, c_1, \cdots, c_5$ by the standard least square method in the training data.

$$E = \sum_{i=1}^{N_{tr}} (y_i - z_{ki})^2 \qquad (4)$$

Step 4. Compute and generate new independent observations by training data samples : We substitute computed coefficients $c_0, c_1, \cdots, c_5$ into inplace of previous coefficients of the regression polynomial eqn. (3). We can compute the identification errors for the regression polynomial eqn. (3) which is replace the testing data instead of training. For calculated $E_1, E_2, \cdots, E_{n(n-1)/2}$ whose number is $n(n-1)/2$, a set of the best W variables can be selected and then continue full sorting procedure for this set only, and we only select the $E_1, E_2, \cdots, E_w$ and replace to ordered identification errors from the smallest value toward largest value. The remaining models whose number are $n(n-1)/2$ - W is abandoned.

Step 5. Stop the GMDH algorithms if E_l is satisfied equation (5).

$$E_1 \geqq E_* \qquad (5)$$

where, E_* is the E_l of prevoius layer for testing data.

Stop 6. In the case of being not satisfied eqn. (5), construct the new input-output data pairs corresponding to $x_{1i} = z_{1i}$, $x_{2i} = z_{2i}$, $\cdots$, $x_{wi} = z_{wi}$ and go to Step 2.

3. The sturcture and Algorithm of FPNN

In this section, a FPNN algorithm is proposed to estimate the structure and parameters of fuzzy model, using the PNN based on the GMDH algorithm and the fuzzy modeling mothod. The new algorithm fuses the PNN algorithm and fuzzy inference by replacement of each neuron of the PNN with fuzzy implications rules, in order to model the nonlinear process system with limited data sets. namely, to identify the premise structure and parameters of fuzzy implications rules.

The premise of fuzzy rules in each node is expressed by Gaussian functions obtained from heuristics. The consequences of rules in each node are expressed by regression polynomial equation. in the FPNN. If premise input variables and parameters are given, the optimal consequence parameters which minimizes performance index can be determined by least-square method in a similar way of fuzzy system. The structure of FPNN is almost like those of the PNN except using the fuzzy system in each node.

3.1 The structure of FPNN

The structure of FPNN model is shown in Figure 1. And the structure of Extended FPNN model is shown in figure 2. In the FPNN model, each node of the neural networks are implicated to fuzzy inference system. The outputs of each node are used for as indepedent input variables in the next layer of FPNN structure. A consequent part of fuzzy if-then rules is used to Structure 1,2,3,4, that is, utilizes the simplified fuzzy inference consisting of constants and the linear fuzzy inference consisting of regression polynomials. In the extended FPNN model, only first layer of the neural network is carried out fuzzy inference method, remaining all layers of neural networks are achieved to PNN method used to the GMDH algorithms. each nodes which fused the fuzzy inference methods in the first layer generate new independent observations which will replace the original observations of the variables. From these new variables we will combine them exactly as we did before. That is, we compute all of the regression polynomial equations of output versus these new variables by using the improved GMDH algorithms. In the PNN algorithm, we will use the regression polynomial which has the Structure 2,3,4.

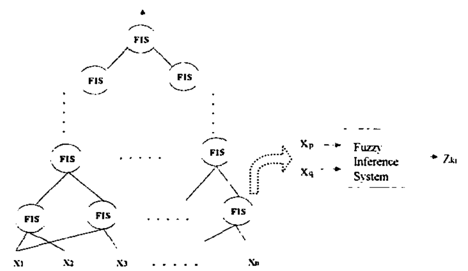


Fig 1. The architecture of FPNN model

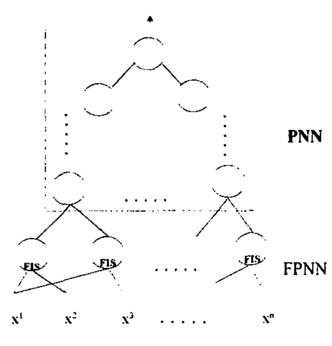


Fig 2. The architecture of extended FPNN model

3.2 FPNN by simplified inference

The consequence part of the simplified inference mechanism where the rules have constant conclusion part is given as follows.

$$R^j : IF\ x_1\ is\ A_{j1} \cdots x_k\ is\ A_{jk}\ Then\ a_{j0} \quad (6)$$

$$y^* = \frac{\sum_{j=1}^{n} w_{ji}\, y_j}{\sum_{j=1}^{n} w_{ji}} = \frac{\sum_{j=1}^{n} w_{ji}\, a_{j0}}{\sum_{j=1}^{n} w_{ji}} \quad (7)$$

If premise input variables and parameters are given in consequence parameter identification, the optimal consequence parameters which minimize PI can be determined. PI is a criterion which uses the mean squared differences between the output data of original system and the output data model.

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

The consequence patrameters of fuzzy model can be estimated by the least squared method is determined as follows.

$$\hat{a} = (X^T X)^{-1} X^T Y \quad (9)$$

3.3 FPNN by regression polynomial inference

The consequence is expressed by linear, quadratic and cubic polynomial equations as shown in Table 1. We wll use the regression polynomial inference as eqn. (10).

$$R^j : IF\ x_1\ is\ A_{j1} \cdots x_k\ is\ A_{jk}\ Then\ y = f_j(x_1, \cdots, x_k) \quad (10)$$

$$y^* = \frac{\sum_{j=1}^{n} w_{ji}\, y_j}{\sum_{j=1}^{n} w_{ji}} = \frac{\sum_{j=1}^{n} w_{ji}\, f_j(x_1, \cdots, x_k)}{\sum_{j=1}^{n} w_{ji}}$$

In this paper, consequence structure in the fuzzy inference method which has each two variables regression polynomial equation is the same as bellow structure 1,2,3,4. The Structure 1. is the simplified fuzzy inference form and the Structure 2. is the same as linear fuzzy inference's consequence form. We will construct FPNN model by using the Structure 1,2,3,4 such as regression polynomial equations.

Structure 1 $= a_{j0}$

Structure 2 $= a_{j0} + a_{j1}x_1 + a_{j2}x_2$

Structure 3 $= a_{j0} + a_{j1}x_1 + a_{j2}x_2 + a_{j3}x_1x_2$

Structure 4 $= a_{j0} + a_{j1}x_1 + a_{j2}x_2 + a_{j3}x_1^2 + a_{j4}x_2^2 + a_{j5}x_1x_2$

4. Simulations and Results

In this section, Mackey-Glass time series data[6], Box and Jenkins gas furnace data[7] and the sewage treatment process of a nonlinear system are considered for the purpose of evaluation the performance of the FPNN and extended FPNN. Here PI and E_PI denote the values of the performance index for the training and testing data.

4.1 Mackey-Glass time series

Mackey-Glass time seriesis is a time series to used as generalized examples to the performance index of the prediction algorithm[6]. We will demonstrate how the proposed FPNN model can be employed to predict future values of a chaotic time series. The Mackey-Glass time series is generated by the following delay differential equation[6].

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t) \quad (11)$$

we extracted 1000 input-output data pair of the following: $[x(t-18), x(t-12), x(t-6), x(t); x(t+6)]$ where $t = 118$ to 1117. The first 500 input-output data pairs used for training the proposed model, the remaining 500 pairs as the testing data sets were used for validating the identified model. To simulate the prediction capability of the FPNN model, we was used the RMSE as the well-known performance index.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i) - \hat{x}(i))^2} \quad (12)$$

Table 2 are listed the comparison of prediction capability through each the fuzzy resoning method and each number of the fuzzy input variables in the proposed FPNN model for Mackey-Glass time series. Table 3 also is compared our proposed FPNN with other fuzzy modeling methods. As the results to predicted future variable *x(t+6)*, we simulated our FPNN model, which is the linear polynomial fuzzy inference.

Fig 3. Prediction results for Training data

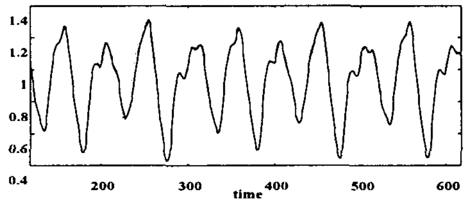


Fig 4. Errors between real and predicted value

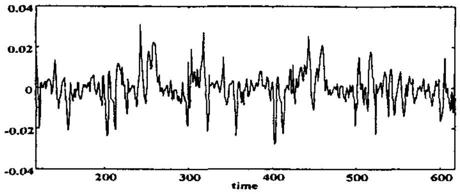


Table 2. Identification errors for FPNN

Structure	Rule No.	PI	E PI
Structure 2	2 × 2	0.0140	0.0158
	2 × 3	0.0139	0.0144
	3 × 2	0.0126	0.0134
	3 × 3	0.0114	0.0158
Structure 3	2 × 2	0.0130	0.0146
	2 × 3	0.0125	0.0164
	3 × 2	0.0116	0.0136
	3 × 3	0.0143	0.0157

Table 3. Generalization prediction error

Model	Rules No.	RMSE
Wang	7	0.0447
Wang	11	0.0280
Wang	15	0.0194
Wang	19	0.0160
Wang	23	0.0137
Wang	27	0.0117
Wang	31	0.0107
Jang[5]	16	0.007
Our Model	12	0.0130

4.2 Gas Furnace

In this section, We apply FPNN Algorithm to nonlinear system identification, using the well-known Box and Jenkins gas furnace[7] data as the training and testing data set. This is a time series data set for a gas furnace process with gas flow rate *u(t)* as the furnace input and CO_2 concentration *y(t)* as the furnace output. We want to extract a dynamic process model to predict *y(t)* using 6 candidate inputs to FPNN; [*u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1)*; y(t)]. We use the first 145 data points as the training data set for a approximation of nonlinear system, and remaining 145 as the testing data set for a generalization.

Table 4 are listed according to the fuzzy inference method and the number of fuzzy input variables for FPNN model. The prediction results for extended FPNN are listed to Table 4. The identification error (or performance index) of each model is compared with other fuzzy modeling method in Table 6. Figure 5 is the structure of extended FPNN model.

Table 4. Identification errors for FPNN

Structure	Rule No.	PI	E PI
Structure 1	2×2	0.3319	5.4133
	2×3	0.2554	3.4667
	3×2	0.1883	4.7114
Structure 2	2×2	0.0149	0.1942
	2×3	0.0139	0.3013
	3×2	0.0143	0.5978

Table 5. Identification errors for extended FPNN

FPNN	PNN	PI	E PI
Structure 1	Linear	0.3466	5.1027
	Bilinear	0.3587	5.0457
	Biquadratic	0.3169	3.9618
Structure 2	Linear	0.0169	0.1263
	Bilinear	0.0169	0.1263
	Biquadratic	0.0167	0.1279

Table 6. Generalization identification error

Model	PI
Tong's model	0.469
Pedrycz's model	0.320
Xu's model	0.328
Sugeno and Yasukawa's model	0.190
Sugeno and Tanaka's model	0.068
Park's model[8]	0.055
Our model (Bilinear)	0.0414
Our model (Biquadratic)	0.0342

Fig 5. Structure of extended FPNN (linear polynomial)

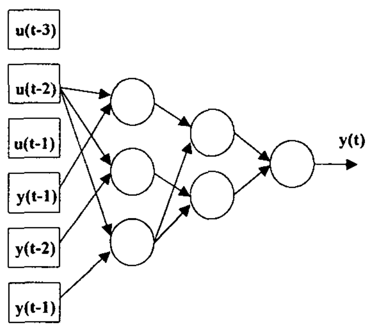
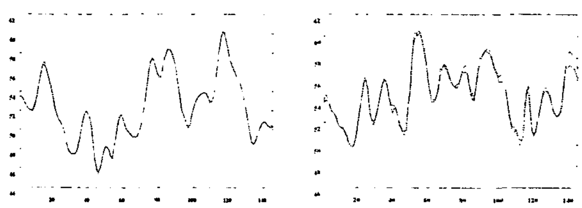


Fig 6. The Prediction results of FPNN



(a) training data (b) testing data

4.3 Sewage treatment process

A sewage treatment generally uses the activated sludge process which consisted of sand basin, primary sedimentation basin, aeration tank and final sedimentation basin. Suspended solid included in sewage is sedimented by gravity in sand and primary sedimentation basins. We really measured input variables such as Mixed Liquid Suspended(MLSS), Waste Slage Ratio(WSR), Recycled sludge ratio Set Point(RRSP) and Dissolved Oxygen Set Point(DOSP), and output variables such as Efficient Suspended Solids(ESS). The FPNN modeling is carried out using the 52 pairs of input-output data obtained from the activated sludge process. We also simulated the modeling using the first 26 data pairs as the training procedure and the remaining 26 data pairs as the testing.

Table 7. Identification errors for FPNN model

Structure	Rules No.	PI	E_PI
Structure 1	2×2	10.6442	9.1129
	2×3	6.7071	8.8620
	3×2	6.8771	13.1938
Structure 2	2×2	7.9196	26.915

Table 8. Identification errors for extended FPNN

FPNN	PNN	PI	E_PI
Structure 1	Linear	12.5055	12.5467
	Bilinear	11.7633	12.7282
	Biquadratic	6.1408	10.2891
Structure 2	Linear	6.3793	20.8947
	Bilinear	5.2839	21.3807
	Biquadratic	3.3711	28.1672

Table 6. Comparison of identification errors

Model			PI	E_PI
Oh's model [9]	Triangular	Simplified	12.802	15.915
	Triangular	Linear	6.396	54.233
Our model	FPNN	Simplified	6.7071	8.8620
		Linear	7.9196	13.1938
	Extended FPNN	Simplified	6.1408	10.2891
		Linear	6.3793	20.8947

5. Conclusions

In this paper, the FPNN was proposed to model fused GMDH algorithm to fuzzy inference method, in order to model the complex nonlinear system. As this data samples is divided into parts for model construction and evaluation of input-output data. We were carried out design to the optimal FPNN model through the harmony balance between approximation and generalization. FPNN model has more performance in linear reasoning method than simplified reasoning method. In the case of the sewage treatment process, The FPNN model causes over-fitting for training data and more large error for testing data. However, the extended FPNN not causes over-fitting to complex nonlinear systems.

6. Reference

[1] A.G. Ivahnenko, "The group method of data handlling ; a rival of method of stochastic approximation ", *Soviet Automatic Control*, 1-3, pp.43-55, 1968.

[2] I. Hayashi and H. Tanaka , "The Fuzzy GMDH algorithm by possibility models and its application", *Fuzzy Sets and Systems*, 36, pp.245-258, 1990.

[3] Hideo Tanaka, Katsunori and Hisao Ishibuchi, "GMDH by If-Then Rules with Certainty Factorsl", *Fifth IFSA World Conference*, pp.802-805, 1993.

[4] L.X. Wang and J.M. Mendel, "Generating fuzzy rules from numerical data, with applications," *IEEE Trans. on Systems, Man, and Cybern.*, 22 no. 6, pp. 1414-1427, 1992.

[5] J.R. Jang and C. Sun, "Predicting Chaotic Time Series with Fuzzy If-Then Rules," *2nd IEEE Inter. Conf. Fuzzy Systems*, San Francisco, pp. 1079-1084, 1993.

[6] M.C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems. *Science*, 197:287-289, July 1977.

[7] G.E.P. Box and F.M. Jenkins, *Time Series Analysis : Forecating and Control*, 2nd ed. Holden-Day, 1976

[8] E. Kim, M. Park, S. Ji and M. Park, "A New Approach to Fuzzy Modeling", *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 3, pp. 328-337, 1997.

[9] C.S. Park, S.K. Oh, and W. Pedrycz, "Fuzzy Identification by means of Auto-Tuning Algorithm and Weighting Factor", *The Third Asian Fuzzy Systems Symposium*, PP.701-706, 1998.