

pH Neutralization Process Modeling Using GA Based Self-Organizing Polynomial Neural Network

Jong-Wook Park¹ and Dong W. Kim^{2*}

¹*Dept of Electronic Eng., Incheon National University, Incheon, South Korea, 402-752*

²*Dept. of Digital Electronics, Inha Technical College
100 Inha-ro, Nam-Gu, Incheon, South Korea, 402-752
dwnkim@inhatc.ac.kr*

Abstract

In this paper, GA-based SOPNN was constructed and applied to model a pH neutralization process. The pH neutralization process can be found in a variety of practical areas including wastewater treatment, biotechnology processing, and chemical processing. It is the measurement of the acidity or alkalinity of a solution containing a proportion of water. It is mathematically defined as the negative decimal logarithm of the hydrogen ion concentration $[H^+]$ in the solution. The strong nonlinearity in the pH process is characterized by its steady state titration curve. The performance of GA-based SOPNN was compared with other models including conventional polynomial neural network(PNN), and adaptive network based fuzzy inference system models. In all comparisons, the GA-based SOPNN demonstrated significantly improved performance. Moreover, the network complexity of conventional PNN could be significantly reduced.

Keywords: *pH neutralization process, genetic algorithm, self-organizing polynomial neural network*

1. Introduction

High performance process control and supervision often require accurate process models. Most processes are nonlinear and their model should be nonlinear [9]. The problem of regulating and controlling a pH process can be found in a variety of practical areas including waste water treatment, biotechnology processing, and chemical processing [4, 8]. Also there are many difficulties in controlling the pH process because of heavy nonlinearity and uncertainty. In many pH control strategies, having a good computational model is imperative for implementing an effective control. Recently, the issue of efficient techniques has been a focal point of a vast number of research endeavors in the area of the model identification of complex and nonlinear systems. The main research agendas include various important tasks such as an efficiency of learning, significant approximation and generalization abilities of the designed networks. While fuzzy sets, neural networks and evolutionary computing have augmented a field of modeling quite immensely, they have also gave rise to a number of new methodological issues and increased awareness about tradeoffs. When the dimensionality of the model goes up (say, the number of variables increases), so do the difficulties.

This paper is concerned with modeling and identification of pH process by genetic algorithm (GA) based self-organizing neural networks (SOPNN). There have been many reports on constructing pH models using intelligent systems such as fuzzy- neural approach [8]. Recently, a polynomial neural network (PNN) [1] has been applied and demonstrated improved performance over conventional statistical regression models or conventional neural network. The PNN provides an automated selection of essential input variables and builds hierarchical polynomial regressions 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 massive amount of data. But the PNN algorithm is a heuristic method and it does not guarantee that the obtained PNN model is the best one for modeling and prediction. Therefore, more attention must be paid to solve the drawbacks. Several methodologies to circumvent the drawbacks of the PNN stated earlier are presented by the author [10-14].

GA based SOPNN is another method to alleviate the handicap of the PNN [12-14]. The structure and parameter of the PNN are optimized and accomplished by using a genetic algorithm (GA) to optimize the training factors. The performance of GA based PNN is compared to conventional PNN, and adaptive network based fuzzy models. Model performance is evaluated with pH process data.

2. Experimental Study

The experimental study examined here is obtained from pH dynamics in a continuous stirred tank reactor (CSTR) and discussed. We apply GA based SOPNN to a highly nonlinear process of pH neutralization of a weak acid and a strong base. This model can be found in a variety of practical areas including wastewater treatment, biotechnology processing, and chemical processing [4-8].

pH is the measurement of the acidity or alkalinity of a solution containing a proportion of water. It is mathematically defined, for dilute solution, as the negative decimal logarithm of the hydrogen ion concentration $[H^+]$ in the solution, that is,

$$pH = -\log_{10}[H^+] \quad (1)$$

In the CSTR investigated, shown in Figure 1, acetic acid (HAC) of concentration C_a flows into the tank at flow rate F_a , and is neutralized by sodium hydroxide (NaOH) of concentration C_b which flows into the tank at rate F_b . The equations of the CSTR can be described as follows (here we assume that the tank is perfectly mixed and isothermal).

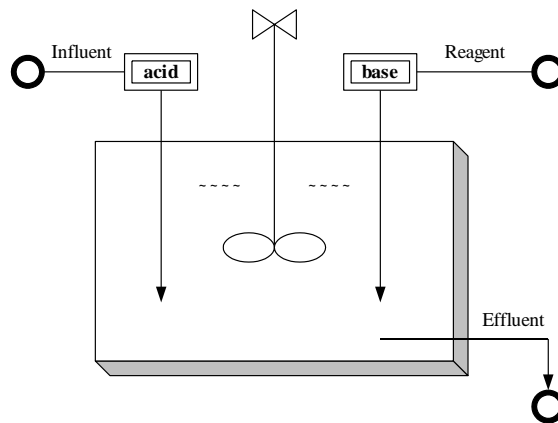


Figure 1. Continuous Tank Reactor for pH Neutralization

The process equations for the CSTR is given by

$$\frac{Vdw_a}{dt} = F_a C_a - (F_a + F_b)W_a \quad (2a)$$

$$\frac{Vdw_b}{dt} = F_b C_b - (F_a + F_b)W_b \quad (2b)$$

where the constant V is the volume of the content in the reactor, w_a and w_b are the concentrations of the acid and the base, respectively.

The above equation describes how the concentrations of w_a and w_b changes dynamically with time subject to the input streams F_a and F_b . To obtain the pH in the effluent, we need to find a relation between instantaneous concentrations w_a and w_b and pH values. This relationship can be described by a nonlinear algebra equation known as the titration or characteristic curve. Depending on the chemical species used, the titration curve varies.

Here we consider the case that a weak influent is neutralized by a strong reagent. The words strong and weak are used to characterize the degree of ionic dissociation in an aqueous solution. Strong reagents completely dissociate into their hydrogen or hydroxyl ions whereas weak reagents are only partially ionized.

Consider an acetic acid (weak acid) denoted by HAC being neutralized by a strong base NaOH(sodium hydroxide)in water. The reactions are



According to the electroneutrality condition, the sum of the charges of all ions in the solution must be zero, *i.e.*

$$[\text{Na}^+] + [\text{H}^+] = [\text{OH}^-] + [\text{AC}^-] \quad (4)$$

where the symbol $[X]$ denotes the concentration of the ion X.

On the other hand, the following equilibrium relationships hold for water and acetic acid:

$$K_a = [\text{AC}^-][\text{H}^+] / [\text{HAC}] \quad (5a)$$

$$K_w = [\text{H}^+][\text{OH}^-] \quad (5b)$$

where K_a and K_w are the dissociation constants of the acetic acid and water with $K_a = 1.76 \times 10^{-5}$ and $K_w = 10^{-14}$.

Defining $w_a = [\text{HAC}] + [\text{AC}^-]$ as the total acetate and $w_b = [\text{Na}^+]$ and inserting Eqs. (5 a) and (5 b) into Eq. (4), we have

$$[\text{H}^+]^3 + [\text{H}^+]^2 \{ K_a + w_b \} + [\text{H}^+] \{ K_a (w_b - w_a) - K_w \} - K_a K_w = 0 \quad (6)$$

Using Eq. (1), Eq. (6) becomes

$$w_b + 10^{-pH} - 10^{pH - pK_w} - \frac{w_a}{1 + 10^{pK_a - pH}} = 0 \quad (7)$$

where $pKa = -\log_{10} k_a$. The static relationship between base flow rate and pH in the reactor is plotted in Figure 2. It can be seen that the strong nonlinearity inherent in the pH process is characterized by its steady state titration curve.

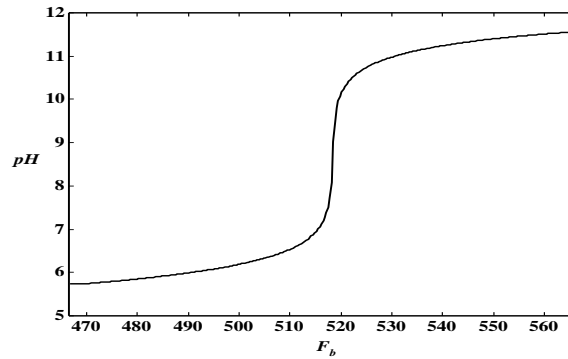


Figure 2. Titration Curve for pH Neutralization

We consider the weak acid-strong base neutralization process described by Eqs. (2 a), (2 b) and (7). By fixing the acid flow-rate F_a (81cc/min) at a specific value, the process is regarded as a single variable system with base flow-rate F_b and the pH in the effluent being the input and output, respectively. The (F_b, y_{pH}) data pairs were produced by using the process physical model with the parameter values given in Table 1.

Table 1. Parameters and Initial Values for pH Process

| Variables | Meaning | Initial setting |
|-----------|--------------------------------|-----------------------|
| V | Volume of tank | 1000 cc |
| F_a | Flow rate of acid | 81 cc/min |
| F_b | Flow rate of base | 515 cc/min |
| C_a | Concentration of acid in F_a | 0.32 mole/l |
| C_b | Concentration of base in F_b | 0.05 mole/l |
| K_a | Acid equilibrium constant | 1.76×10^{-5} |
| K_w | Water equilibrium constant | 1.0×10^{-14} |
| $W_a(0)$ | Concentration of acid | 0.0435 mole/l |
| $W_b(0)$ | Concentration of base | 0.0432 mole/l |

The base flow rate F_b was given by

$$F_b = 515 + 51.5 \sin(2\pi / 25) \quad \text{for } t \leq 150 \quad (8a)$$

$$F_b = 515 + 25.75 \sin(2\pi / 25) + 25.75 \sin(2\pi / 10) \quad \text{for } t > 150 \quad (8b)$$

For obtaining such a data pairs, we applied Newton-Raphson method that is given by Eq. (9).

$$pH_{i+1} = pH_i - \frac{f(pH_i)}{f'(pH_i)} \quad (9)$$

The system inputs of the GA based SOPNN structure consist of the delayed terms of $F_b(t)$ and $y_{pH}(t)$ which are input and output of the process, i. e.

$$\hat{y}_{pH}(t) = \varphi(F_b(t-3), F_b(t-2), F_b(t-1), y_{pH}(t-3), y_{pH}(t-2), y_{pH}(t-1)) \quad (10)$$

where, \hat{y}_{pH} and y_{pH} denote model output and the actual process output, respectively. 500 data pairs are generated from Eqs. (8a), (8b), and (9) where total data are used for training.

3. GA Based Self-organizing Polynomial Neural Network

The brief process to construct GA based SOPNN is presented in this section. Detailed descripts can be found in [12, 14]. When we design the SOPNN using GA, the most important consideration is the representation strategy, that is how to encode the key factors of the PNN into the chromosome. We employ a binary coding for the available design specification. We code the order and the inputs of each node in the PNN as a finite-length string. Our chromosomes are made of three sub-chromosomes. The first one is consisted of 2 bits for the order of polynomial (PD), the second one is consisted of 3 bits for the number of inputs of PD, and the last one is consisted of N bits which are equal to the number of entire input candidates in the current layer. These input candidates are the node outputs of the previous layer. The representation of binary chromosomes is illustrated in Figure 3.

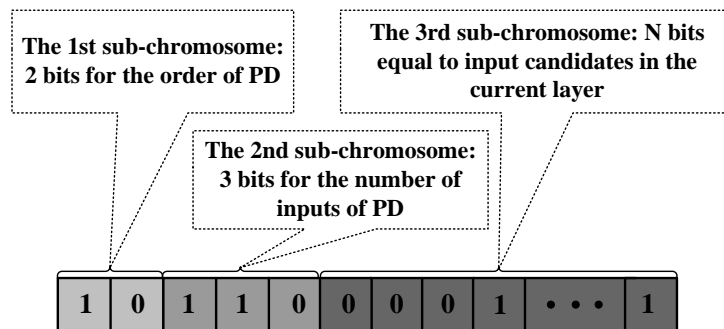


Figure 3. Structure of Binary Chromosome for a PD

The 1st sub-chromosome is made of 2 bits. It represents several types of order of PD. The relationship between bits in the 1st sub-chromosome and the order of PD is shown in Table 2. Thus, each node can exploit a different order of the polynomial. The 3rd sub-chromosome has N bits, which are concatenated a bit of 0's and 1's coding. The input candidate is represented by a 1 bit if it is chosen as input variable to the PD and by a 0 bit it is not chosen. This way solves the problem of which input variables to be chosen.

Table 2. Relationship between Bits in the 1st Sub-Chromosome and Order of PD

| Bits in the 1st sub-chromosome | Order of polynomial(PD) |
|--------------------------------|-----------------------------|
| 00 | Type 1 – Linear |
| 01 | Type 2 – Quadratic |
| 10 | |
| 11 | Type 3 – Modified quadratic |

Table 3. Relationship between Bits in the 2nd Sub-chromosome and Number of inputs to PD

| Bits in the 2nd sub-chromosome | Number of inputs to PD |
|--------------------------------|------------------------|
| 000 | 1 |
| 001 | 2 |
| 010 | 2 |
| 011 | 3 |
| 100 | 3 |
| 101 | 4 |
| 110 | 4 |
| 111 | 5 |

If many input candidates are chosen for model design by the 3rd sub-chromosome, the modeling is computationally complex, and normally requires a lot of time to achieve good results. In addition, it causes improper results. Good approximation performance does not necessarily guarantee good generalization capability. To overcome this drawback, we introduce the 2nd sub-chromosome into the chromosome. The 2nd sub-chromosome is consisted of 3 bits and represents the number of input variables to be selected. The number based on the 2nd sub-chromosome is shown in the Table 3. Input variables for each node are selected among entire input candidates as many as the number represented in the 2nd sub-chromosome. Designer must determine the maximum number in consideration of the characteristic of system, design specification, and some prior knowledge of model. With this method we can solve the problem which is the requirement of a lot of computing time.

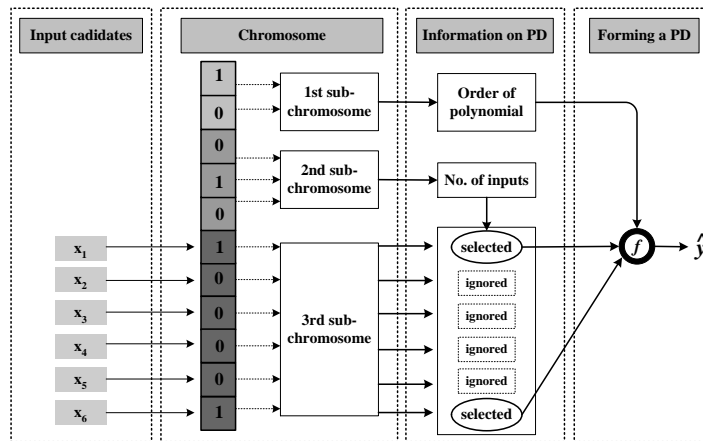


Figure 4. Example of PD Whose Various Pieces of Required Information are Obtained from its Chromosome

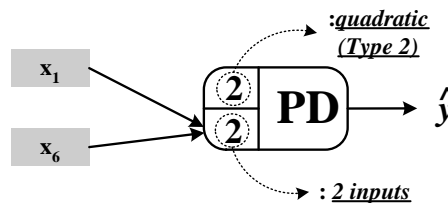


Figure 5. Node with PD Corresponding to Chromosome in Figure 4

The relationship between chromosome and information on PD is shown in Figure 4. The PD corresponding to the chromosome in Figure 4 is described briefly as Figure 5. Figure 4 shows an example of PD. The various pieces of required information are obtained its chromosome. The 1st sub-chromosome shows that the polynomial order is Type 2 (quadratic form). The 2nd sub-chromosome shows two input variables to this node. The 3rd sub-chromosome tells that x_1 and x_6 are selected as input variables. The node with PD corresponding to Figure 4 is shown in Figure 5. Thus, the output of this PD \hat{y} can be expressed as (11).

$$\hat{y} = f(x_1, x_6) = c_0 + c_1x_1 + c_2x_6 + c_3x_1^2 + c_4x_6^2 + c_5x_1x_6 \quad (11)$$

where coefficients c_0, c_1, \dots, c_5 are evaluated using the training data set by means of the standard least squared error (LSE).

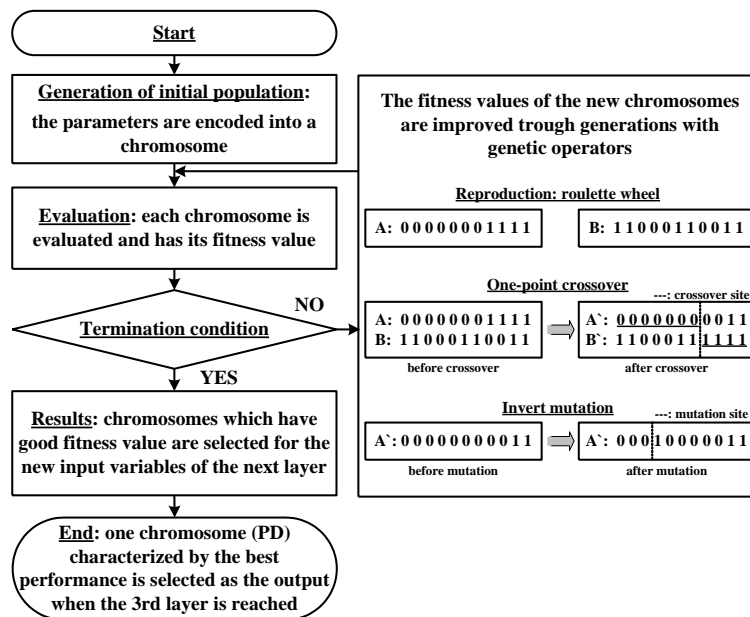


Figure 6. Block Diagram of the Design Procedure of GA Based SOPNN

The polynomial function, PD, is formed automatically according to the information of sub-chromosomes. The design procedure of GA-based SOPNN is shown in Figure 6. At the beginning of the process, the initial populations comprise a set of chromosomes that are scattered all over the search space. The populations are all randomly initialized. Thus, the use of heuristic knowledge is minimized. The assignment of the fitness in GA serves as guidance to lead the search toward the optimal solution. After each of the chromosomes is evaluated and associated with a fitness, the current population undergoes the reproduction process to create the next generation of population. The roulette-wheel selection scheme is used to determine the members of the new generation of population. After the new group of population is built, the mating pool is formed and the crossover is carried out. The crossover proceeds in three steps. First, two newly reproduced strings are selected from the mating pool produced by reproduction. Second, a position (one point) along the two strings is selected uniformly at random. The third step is to exchange all characters following the crossing site. We use one-point crossover operator with a crossover probability of P_c (0.85). This is then followed by the mutation operation. The mutation is the occasional alteration of a value at a particular bit position (we flip the states of a bit from 0 to 1 or vice versa). The mutation serves as an insurance policy which would recover the loss of a particular piece of information (any simple bit). The mutation rate used is fixed at 0.05 (P_m). Generally, after these three operations, the overall fitness of the population improves. Each of the population generated then goes through a series

of evaluation, reproduction, crossover, and mutation, and the procedure is repeated until a termination condition is reached. After the evolution process, the final generation of population consists of highly fit bits that provide optimal solutions. After the termination condition is satisfied, one chromosome (PD) with the best performance in the final generation of population is selected as the output PD. All remaining other chromosomes are discarded and all the nodes that do not have influence on this output PD in the previous layers are also removed. By doing this, the GA-based SOPNN model is obtained.

4. Results

The pH process is modeled by using GA based SOPNN. The design parameters of the model are shown in Table 4. Considering the design parameters of GA-based SOPNN, these are the same as Table 4.

Table 4. Design Parameters of GA-Based SOPNN for Modeling

| Parameters | 1st layer | 2nd layer | 3rd layer |
|----------------------------|-----------|-----------|-----------|
| Maximum generations | 40 | 60 | 80 |
| Population size:(w) | 20:(15) | 60:(50) | 80 |
| String length | 8 | 20 | 55 |
| Crossover rate (P_c) | 0.85 | | |
| Mutation rate (P_m) | 0.05 | | |
| Weighting factor: θ | 0.1~0.9 | | |
| Type (order) | 1~3 | | |

w : the number of chosen nodes whose outputs are used as inputs to the next layer

As seen from Table 4, in the 1st layer, 20 chromosomes are generated and evolved until 40 generations. Those coefficients of each of the resulting 20 nodes (PDs) are estimated with the training data and by the LSE. According to the fitness values computed by cost function, the PDs are then selected by a predetermined number w from the node of the largest fitness. As indicated in Table 4, the value of w is different from each layer. The chosen PDs (w nodes) must be preserved for the design of the next layer and the outputs of the preserved PDs serve as the inputs to the next layer. This procedure is repeated for the 2nd layer and the 3rd layer.

Figure 7 and 8 illustrate the trend of the performance index and fitness function values produced in successive generations of the GA, respectively. Figure 9 shows the performance comparison of the proposed method with conventional PNN. In Figure 10, the proposed GA-based SOPNN model with 3 layers are depicted. From the Figure 9 and 10, we can see that the GA-based SOPNN has more good performance than conventional PNN.

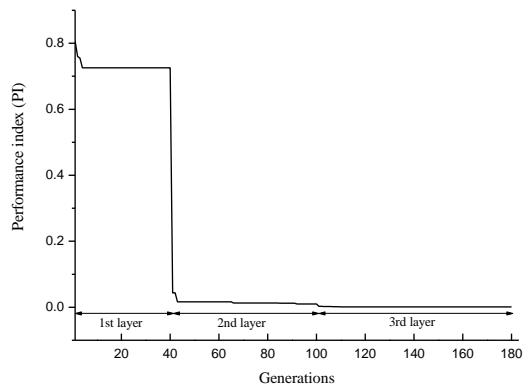


Figure 7. Trend of Performance Index Values with Respect to Generations through Layers

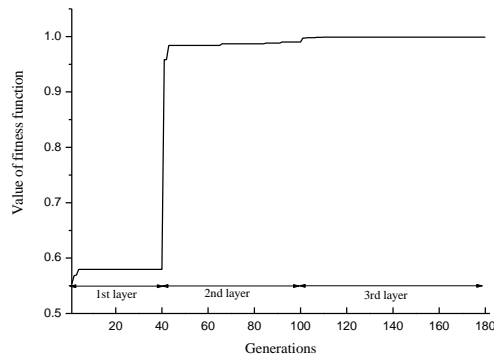


Figure 8. Values of the Fitness Function with Respect to the Successive Generations

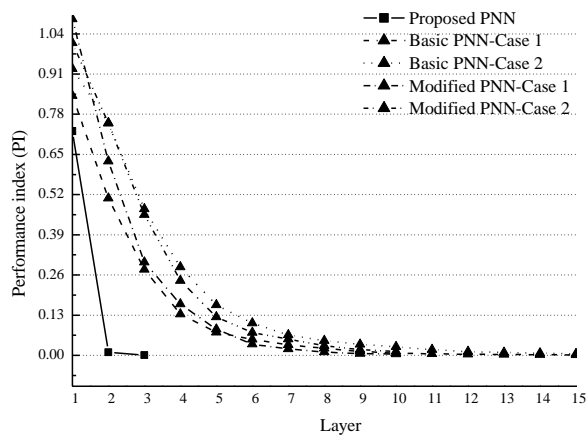


Figure 9. Performance Comparison of the Proposed Method with Conventional PNN

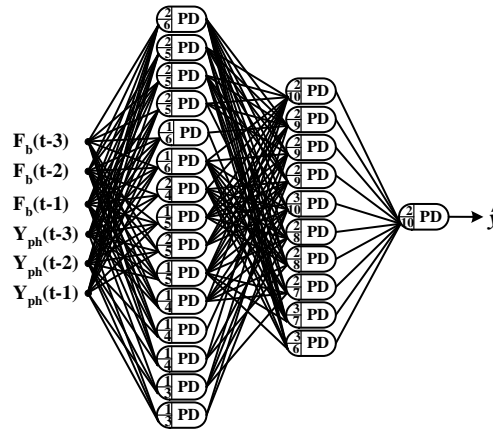


Figure 10. Structure of the GA-Based SOPNN Model with 3 Layers

Table 5 provides a comparative analysis of fuzzy models and conventional PNN with 4 cases. Considering a performance index, Mean square error (MSE) is used. From the Table 5, it becomes obvious that GA based architecture outperforms other models.

Table 5. Performance Comparison of Various Identification Models

| Model | | MSE |
|--------------------|---------------------|--------|
| | | PI |
| Nie's model[8] | USOCPN | 0.230 |
| | SSOCPN | 0.012 |
| Basic SOPNN [1] | Case 1 (15th layer) | 0.0015 |
| | Case 2 (15th layer) | 0.0052 |
| Modified SOPNN [1] | Case 1 (10th layer) | 0.0039 |
| | Case 2 (10th layer) | 0.0124 |
| GA-based SOPNN | | 0.0008 |

5. Conclusion

In this paper, GA-based SOPNN was constructed and applied to model a pH neutralization process. The strong nonlinearity in the pH process is characterized by its steady state titration curve. The performance of GA-based SOPNN was compared with other models including conventional polynomial neural network (PNN), and adaptive network based fuzzy inference system models. In all comparisons, the GA-based SOPNN demonstrated significantly improved performance. Moreover, the network complexity of conventional PNN could be significantly reduced.

Acknowledgment

This work was supported by the Incheon National University Research Grant in 2014.

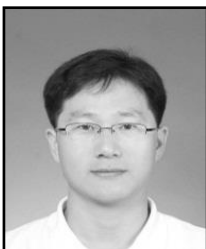
References

- [1] S. K. Oh and W. Pedrycz, "The design of self-organizing Polynomial Neural Networks", *Inf. Sci.* vol. 141, (2002), pp. 237-258.
- [2] Y. Shi, R. Eberhart and Y. Chen, "Implementation of Evolutionary Fuzzy Systems", *IEEE Trans. Syst. Man Cybern.*, vol. 7, (1999), pp. 109-119.
- [3] K. Kristinnson and G. A. Dumont, "System identification and control using genetic algorithms", *IEEE Trans. Syst., Man, Cybern.*, vol. 22, (1992), pp. 1033-1046.
- [4] F. G. Shinskey, "pH and pION Control in Process and Waste Streams", (1973); Wiley, New York.
- [5] R. C. Hall and D. E. Seberg, "Modeling and Self-Tuning Control of a Multivariable pH Neutralization Process", *Proc. ACC*, (1989).
- [6] T. J. McAvoy, "Time optimal and Ziegler-Nichols control", *Ind. Engrg. Chem. Process Des. Develop.*, vol. 11, (1972), pp. 71-78.
- [7] T. J. McAvoy, E. Hsu and S. Lowenthal, "Dynamics of pH in controlled stirred tank reactor", *Ind. Engrg. Chem. Process Des. Develop.*, vol. 11, (1972), pp. 68-70.
- [8] J. Nie, A. P. Loh and C. C. Hang, "Modeling pH neutralization processes using fuzzy-neural approaches", *Fuzzy Sets and Systems*, vol. 78, (1996), pp. 5-22.
- [9] J. Zhang and J. Morris, "Process modeling and fault diagnosis using fuzzy neural networks", *Fuzzy Sets and Systems*, vol. 79, (1996), pp. 127-140.
- [10] D. Kim and G. T. Park, "Advanced self-organizing polynomial neural network", *Neural Comput. & Applic.*, vol. 16, (2007), pp. 443-452.
- [11] D. Kim, S. J. Seo and G. T. Park, "Hybrid GMDH-type modeling for nonlinear systems: Synergism to intelligent identification", *Advances in Eng., Software*, vol. 40, (2009), pp. 1087-1094.
- [12] D. Kim and G. T. Park, "A New Design of Polynomial Neural Networks in the Framework of Genetic Algorithms", *IEICE Trans. Inf.&Syst.*, vol. E89-D, no. 8, (2006), pp. 2429-2438.
- [13] D. Kim, B. Kim, I. J. Shim and G. T. Park, "Plasma Etch Prediction Using Genetic Algorithm Based Polynomial Neural Network", *Surface Engineering*, vol. 20, no. 1, (2004), pp. 31-36.
- [14] D. W. Kim, "Adaptive Learning Networks in Evolutionary Framework", *Journal of Platform Technology*, vol. 2, no. 3, (2014), pp. 35-54.

Authors



Jong W. Park, He received the Ph.D. degree in Electronic Engineering from Inha University in 1985. He is professor of Incheon National University. He had been Director of Incheon Technology Innovation Center from 1999 to 2003. His research interests include robotics and motion estimation of robot vision system.



Dong W. Kim, He received the Ph.D. degree in Electrical Engineering from Korea University in 2007. He is associated professor of Inha Technical College. His research interests include robotics, advanced robot design, evolutionary multi-mobile robot system, humanoid robot, soft computing and their application to control.

