5th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2017)

A Study on PID Controller Parameter Optimization Based on Cell Membrane Computing

JiaChang Xu[1,a] ,YouRui Huang[2,b] and GuangYu Xu[3,c]

[1,2] School of Electrial and Information Engineering, Anhui University of Science and Technology,
Huainan 232001,Anhui, China

[1,3] School of computer science and engineering,Anhui University of Science and Technology,
Huainan 232001,Anhui, China

[a] jcxu@aust.edu.cn, [b] hyr628@163.com, [c] gyxu@aust.edu.cn

Keywords: PID Controller; Optimizing Control; BP Neural Network; Membrane Computing

Abstract: It is less likely to construct an exact mathematical mode as actual industrial processes are featured with non-linearity and uncertainty, which further causes the conventional PID controller to fail to achieve an ideal control effect. In this paper, a PID controller parameter optimization method was presented based on nested membrane algorithm. It focused on comparative studies on PID controller parameter optimization of BP neural network. In addition, an objective of optimization was realized through experimental simulation. The relevant experimental results indicated that such an algorithm not only improves stability of PID control system, but enhances characteristics of it, such as convergence and fast response, etc..

1. Introduction

PID controller is a typical proportional, integrational and derivative controller. Thanks to good robustness, high reliability and its adaptability to diverse working conditions, the PID controller is able to control engineering applications easily [1]. While performance of the PID controller mainly depends on controller parameter settings, different controlled objects and control parameters have different impacts on the system, which is especially obvious as far as values taken for PID parameters are concerned. Specific to PID parameter optimization, masses of researches have been carried out by many experts and scholars. Huang Yourui et al. put forward a self-tuning PID controller based on genetic neural network to improve robustness and response speed of the control algorithm. In order to elevate search speed and convergence, Jiang Jinlong et al. present a PID parameter tuning method based on genetic algorithm and an optimization strategy. Regarding Li Junyong et al., a technology combining the modified genetic algorithm with fuzzy control is adopted to obtain a better control effect by PID tuning parameter optimization. In allusion to problems of PID parameter tuning, a PID parameter tuning approach was further proposed in this paper based on cell membrane computing [2] to construct a cell membrane computing model and realize PID controller parameter optimization by virtue of a monolayer membrane algorithm. Then, it was compared with BP neural network algorithm based parameter tuning, which proved that the approach presented in this paper had certain superiority.

2. PID Controller Algorithm and Design Based on BP Neural Network

2.1 PID Controller Algorithm Based on BP Neural Network

Non-linear expressiveness of the neural network can implement optimization of PID controller by means of system learning so as to achieve three optimal parameters for such a controller. Besides, a PID controller parameter self-learning mechanism was established based on BP neural network. Regarding PID control system based on BP neural network, the diagram is given in Fig. 2-1. According to performance states of the control system, PID controller parameters were adjusted so that the system could obtain an optimal performance state and outputs from the output layer corresponded to three parameters of the controller described above. Through self-adaptive learning of the neural network, the associated weighting coefficient was adjusted with an aim to make outputs of the neural network correspond to parameters of the PID controller in an optimal state[3].

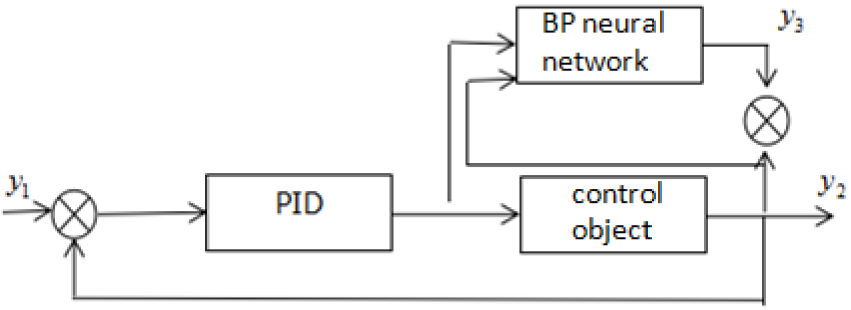


Fig. 2-1 PID Control System Block Diagram Based on BP Neural Network

In a three-layer BP neural network presented in Fig. 2-2,

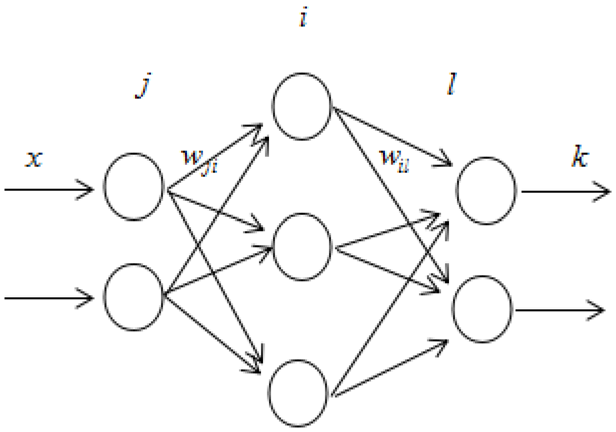


Fig. 2-2 Structure Diagram of BP Network

Input of the input layer can be expressed in

$$I^{(1)}{}_j = x_{(j)} \qquad \text{(j=1,2,...,n)} \tag{2.1}$$

Input and output of the hidden layer are denoted as follows.

$$net_i^{(2)}(k) = \sum_{i=1}^{n} w_{ij}^{(2)} I_j^{(1)} \qquad \text{(i=1,2,...m)} \tag{2.2}$$

$$I_i^{(2)} = f(net_i^{(2)}(k))$$

Where, $w_{ij}^{(2)}$ is a weighting coefficient of the hidden layer; and, superscripts (1), (2) and (3) respectively refer to input, hidden and output layers of such a three-layer neural network. Moreover, activation function of neuron in hidden layer is,

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.3}$$

Input and output of the output layer are denoted by

$$net_l^{(3)}(k) = \sum_{i=0}^{m} w_{ij}^{(3)} o_i^{(2)}(k)$$

$$o_l^{(3)}(k) = h(net_l^{(3)}(k)) \qquad (2.4)$$

Where, value of l is 1, 2 or 3; and, $o_l^{(3)}(k)$ stands for PID controller parameter k_p, k_i or k_d separately. Besides, function of the output layer is,

$$h(x) = \frac{e^x}{e^x + e^{-x}} \qquad (2.5)$$

In line with PID control theory, system performance function is expressed in an equation below.

$$e(k) = \frac{1}{2}(in(k) - out(k))^2 \qquad (2.6)$$

According to gradient descent algorithm principle, η refers to learning speed and a to inertia coefficient. Then, a fast search convergence item $e(k)$ described as follows.

$$\Delta w_{li}^{(3)}(k) = -\eta \frac{\partial e(k)}{\partial w_{li}^{(3)}} + a\Delta\partial w_{li}^{(3)}(k-1) \qquad (2.7)$$

By further differential of the formula, another equation below can be obtained by derivation.

$$\frac{\partial net_l^{(3)}(k)}{\partial w_{li}^{(3)}} = o_i^{(2)}(k) \qquad (2.8)$$

Depending on the above equations, learning algorithms for output and hidden layers are denoted as follows.

$$\begin{cases} \Delta w_{li}^{(3)}(k) = a\Delta w_{li}^{(3)}(k-1) + \eta\delta_l^{(3)} o_i^{(2)}(k) \\ \delta_l^{(3)} = e(k) g'(net_l^{(3)}(k)) \\ \Delta w_{ij}^{(2)}(k) = a\Delta w_{ij}^{(2)}(k-1) + \eta\delta_i^{(2)} o_j^{(1)}(k) \\ \delta_i^{(2)} = f'(net_i^{(2)}(k)) \sum_{l=1}^{3} \delta_l^{(3)} w_{li}^{(3)}(k) \end{cases} \qquad (2.9)$$

2.2 PID Controller Design Based on BP Neural Network

It is assumed that the BP neural network has a three-layer structure and PID controller of such a network can be designed as follows according to the algorithm described above.

1) Nodal values of the input and the hidden layers were determined together with initial weights of all layers so as to set both the learning rate and the inertia coefficient.

2)Sampling was fulfilled for $in(k)$ and $out(k)$; and, the corresponding error was computed to be $e(k) = in(k) - out(k)$.

3) Inputs and outputs in all layers of the neural network were figured out; and, those of the output layers are PID controller parameters k_p, k_i and k_d.

4) Corresponding to three parameters of the PID controller, formula for output is given below.

$$e(k) = e(k-1) + k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2)) \tag{2.10}$$

3. PID Controller Algorithm and Design Based on Cell Membrane Computing

3.1 Membrane Computing

A membrane system is a distributed and parallel computingmodel that is inspired by the structure and functionality ofliving cells. The system includes a hierarchy of membranes that are located within a main membrane, which is called the skinmembrane. Each membrane contains some objects and a set of transformation and communication rules.

3.1.1 Monolayer Membrane Optimization Algorithm

In Literature [4], P system of a monolayer membrane structure is combined together with evolution rules of the quantum-inspired evolutionary algorithm to come up with a quantum-inspired evolutionary membrane algorithm based on membrane computing. Moreover, experimental simulation was also completed in conjunction with the knapsack problem. It was demonstrated that such a newly proposed algorithm has a better performance than a nested system membrane optimization algorithm. Regarding the quantum-inspired evolutionary algorithm that utilizes a monolayer membrane structure, its basic thoughts are to adopt evolution rules of such an algorithm to achieve optimization in various basic membranes; and, after the completion of elementary membrane optimizing, such an algorithm can be further used to convey optimal individuals in all elementary membranes into surface membrane. In this manner, a global optimal solution to this algorithm in a condition of current iterations is acquired. In the end, such an optimal solution is given back into all elementary membranes by means of exchange rules to affect next-generation evolution of individuals related to such an algorithm[4].

3.1.2 Nested Membrane Optimization Algorithm

In a nested membrane structure, each area contains a sub-problem and some candidate solutions to problems to be solved. Moreover, operation rules of candidate solutions are changed into those between adjacent membranes. Execution steps of this algorithm are as follows.

Step 1

Subalgorithm for each membrane was utilized to generate a candidate solution set of the corresponding area.

Step 2

Computing began from a membrane of the outermost layer. Optimal solutions generated by each membrane were conveyed to an inner membrane adjacent to it, while the worst solution to the adjacent outer membrane.

Step 3

In each layer of membrane, the solution set figured out by the subalgorithm served as a candidate solution set of the next generation evolution on the premise of excluding optimal and worst solutions; and, optimal solutions conveyed from an adjacent area were added into the corresponding membrane.

Step 4

The above computational processes were repeated and candidate solution sets of all membranes updated up until termination conditions of the algorithm could be satisfied; finally, an optimal result was output from a membrane of the innermost layer.

To be specific, subalgorithms can be freely collocated according to practical problems to be dealt with. For example, while the outer membrane takes advantage of an algorithm with a strong global searching ability, that with a strong local searching ability is used for inner membrane; in this way, a good complementary effect is achieved. Reasonable selection of various subalgorithms is beneficial to effectively improve performance of membrane optimization algorithms. After a membrane optimization algorithm in embryo has been put forward, there are three major directions of modified researches, including hierarchical structure and characteristics of the membrane, evolution rules and inter-membrane communication mechanism.

3.1.3 Mixed Membrane Optimization Algorithm

Inspired by membrane computing, some principal operation rules and computing mechanisms of evolutionary computation were brought into the rule by adopting the membrane algorithm proposed by Nishida as the basic framework; and, static or dynamic membrane structure of P system was employed. Let’s take DNA computing for example. Double strands of DNA were introduced into the membrane algorithm to construct a dyon system for a similar membrane structure and put forward a membrane computing optimization algorithm featured with information redundancy. Furthermore, two sub-systems have been established for such an algorithm and both have a strong point of information backup. As a result, anti-premature ability of such an algorithm was also improved. Then, according to characteristics of the cell itself, another membrane computing optimization algorithm based on relaxation and contraction rules was come up with successively. By comparing optimal solutions to different areas, those relatively worse were abandoned while areas preferable expanded, so as to accelerate convergence rate of this algorithm.

3.2 PID Controller (MPIDNN) Design Based on Nested Membrane Algorithm

If the controlled member of MPIDNN is a complex non-linear strongly coupled 3-input-3-output system, transfer function of the system is that shown in Equation (3.1). Error computational formula (i.e., performance index function) can be figured out according to Equation (3.2).

$$y_h(k) = ay_h(k-1) + u_i(k-1)/[1+u_i(k-1)] + bu_i(k-1)^3 + cu_{i+1}(k-1) + dy_{h+1}(k-1) \tag{3.1}$$

where, a, b, c∈(0.1,0.5), u^i refer to the node of network layer, x_1, x_2 and x_3 to indicators of controlled quantity, r_1, r_2 and r_3 to control rules of the controller, y_1, y_2 and y_3 to current values of the controlled quantity, h to labels of various inputs and outputs, $x_h(k)$ to the control objective, and, $y_h(k)$ to the actual output. For the purpose of implementing non-linear dissymmetrical control, an MPIDNN control system was set up and its structure chart has been presented in Fig. 3-1.

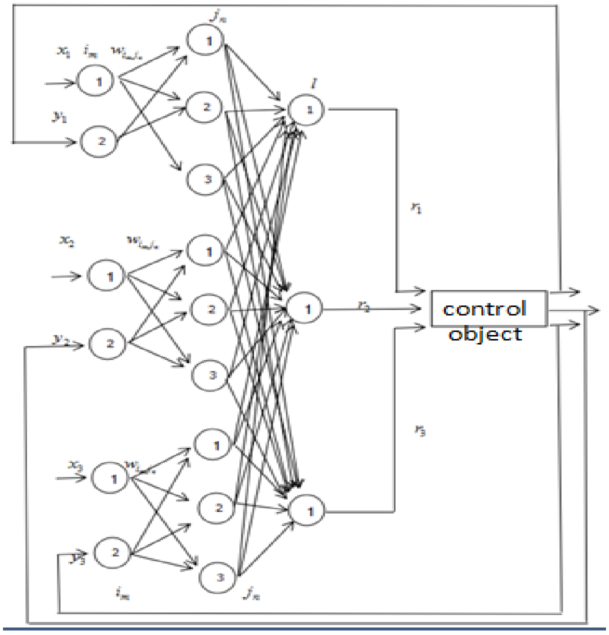
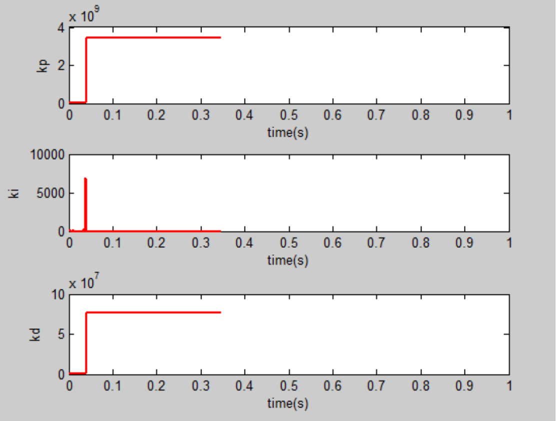


Fig. 3-1 MPIDNN Structure Chart

4. Simulation Results

Combining Fig. 3-1, the controlled quantity is initialized to [0 0 0], the control objective to[0.5 0.4 0.4], and, the interval between control time to 0.001s; in addition, the number of network structures is 3 and that of output neurons is 3 likewise. Learning rate of the network weight is 0.06 and upper limit of input is 1 (p=1). For that beyond such an upper limit, the corresponding actual output is -1 (p=-1). In terms of such an algorithm, the quantity of elementary membranes was set at 10, while executions of outer and inner membranes at 50and 15 respectively. On this basis, control strategies based on BP algorithm and monolayer membrane algorithm were separately used to perform simulation tests. Relevant simulation results obtained have been shown in Figs. 4-1, 4-2, 4-3 and 4-5.



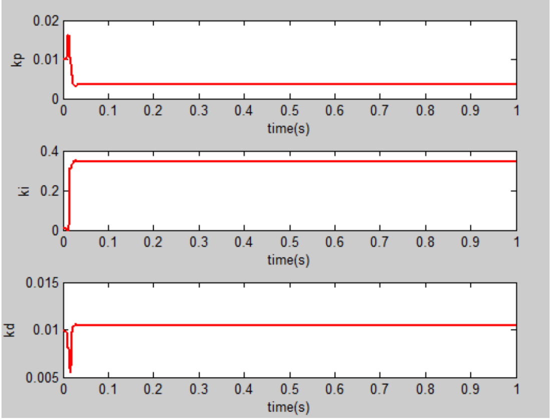
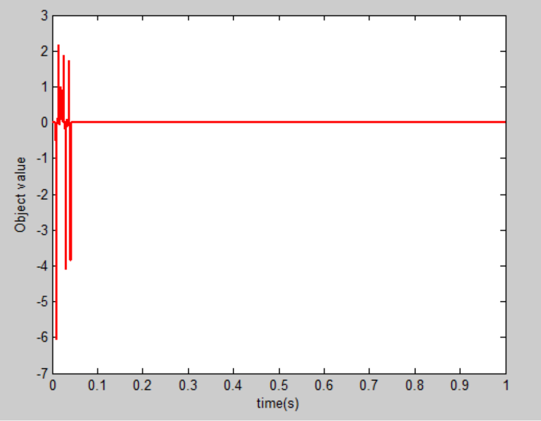


Fig. 4-1 Parameter Optimization Curves of BP Neural Network

Fig. 4-2 Parameter Optimization Curves of MPIDNN



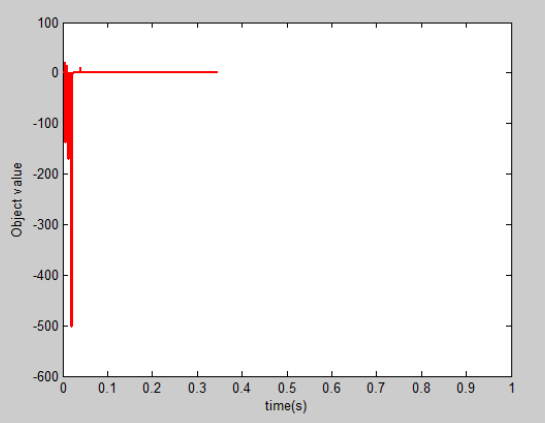


Fig. 4-3 Controlled Object Information Identification of BP Neural Network

Fig. 4-4 Controlled Object Information Identification of MPIDNN

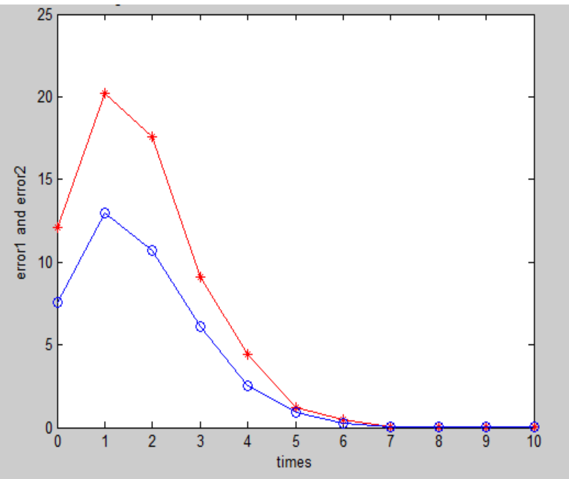


Fig. 4-5 Error Comparison Curves for BP (o) and MPIDNN (×)

5. Brief Summary

Dependent on simulation results, two control systems are able to implement the corresponding control under effects of two diverse algorithms despite that control effects of them are different from each other to a certain extent. In detail, the BP algorithm generates a relatively worse effect as it is much more likely for it to run into local optimum. With regard to MPIDNN, both ability and efficiency of global searching are enhanced thanks to the introduction of a membrane algorithm. Additionally, convergence speed of MPIDNN as well as its control efficiency and accuracy are all improved provided that a requirement of obtaining the minimum error has been satisfied. In this way, MPIDNN achieves a better control results.

6.Acknowledgements

This work was financially supported by The Natural Science Foundation of China(61471004)

References

[1] Liu Jinkun. MATLAB Simulation under Advanced PID Control. Beijing: Publishing House of Electronics Industry, 2011.3.

[2] Liu Gexiang, et al.. Membrane Computing: Theory & Application. Beijing: Science Press, 2012.12.

[3] Wang Tao. Controller Study and Design Based on Membrane Computing Optimization Algorithm. Xihua University, 2012.

[4] Zhang G X, Gheorghe M, Wu C Z. A quantum-inspired evolutionary algorithm based in P systems for knapsack problem [J]. Fundamenta Informaticae, 2008, 87(1): 93-116.

[5] Huang Liang. A Study on Membrane Computing Optimization Methods[D]. Zhejiang: College of Information Science and Engineering, Zhejiang University, 2007.

[6] M. El-Abd, An improved global-best harmony search algorithm, Appl. Math.Comput. 222 (2013) 94–106.

[7] C. Wang, Y. Huang, Self-adaptive harmony search algorithm for optimization,Expert Syst. Appl. 37 (2010) 2826–2837.

[8] M. Al-Betar, I. Doush, A. Khader, M. Awadallah, Novel selection schemes for harmony search, Appl. Math. Comput. 218 (2011) 6095–6117.

[9] M. Al-Betar, A. Khader, Z. Geem, I. Doush, M. Awadallah, An analysis of selection methods in memory consideration for harmony search, Appl. Math.Comput. 219 (2013) 10753–10767.

[10] G. Paun, G. Rozenberg, A. Salomaa (Eds.), Handbook of Membrane Computing,Oxford University Press, Oxford, 2010.

[11] A. Maroosi, R.C. Muniyandi, E. Sundararajan, A.M. Zin, Parallel and distributed computing models on a graphics processing unit to accelerate simulation ofmembrane systems, Simul. Modell. Pract. Theory 47 (2014) 60–78.