Intelligent PID Controller on Soft Computing

Honghua Xu, Xiaoqiang Di
Changchun University of Science and Technology, Changchun, China
Email :{honghuax,xiaoqiangd}@126.com

Huamin Yang, Guangcai Cui
Changchun University of Science and Technology, Changchun, China
Email : {huaminyan,guangcaicu}@yahoo.com

***Abstract*—The PID control algorithm is used for the control of almost all loops in the process industries, and is also the basis for many advanced control algorithms and strategies. For complex systems, controller's structure and parameters need to rely on experience and on line tuning to determine. Application of soft computing techniques can realize self-tuning of controller parameters. In this paper, the author applies fuzzy system, neural network technique and genetic algorithm to PID controller in order to achieve intelligent adjustment the controlled variables. Simulation results show that the application of soft computing techniques in the controller has better control quality.**

***Index Terms*—intelligent, fuzzy system, neural network, proportion integration differentiation, genetic algorithms**

I. Introduction

PID(Proportion Integration Differentiation) control is one of the best widely applied controlling methods in industrial producing process because it is simple and robust[1].

Conventional PID control is a most common method used in the process of manufacturing. It has found its application in many trades. It has many advantages such as ease of controlling, ease of realizing and debugging on the spot etc[2]. In the actual control process, the controlled process is nonlinear, time-varying and indefinite, especially for some more complicated systems. Conventional PID is hard to meet this requirement.

In the actual design of control system, there still exists trial and error method to solve[3]. With quick development of relative technology, control system needs higher norms and self-adaptation technique of PID control parameters in order to suit for more complicated situations and higher demand for norms. So the control engineers are on look for automatic intelligent tuning procedures.

In this paper, the parameters of PID controller are tuned for controlling some rock mechanics testing machine[4]. Conventional PID performance has been compared and analyzed with the intelligent tuning techniques like fuzzy system, neural network, and Genetic algorithm. Soft computing techniques based tuning methods have proved their excellence in giving better results by improving the steady state characteristics.

II. PID Control Mechanism

A. PID algorithm in analog system

In simulation system, the equation of PID algorithm[5] is:

$$\mathrm{U}(t)=K_P\left[e(t)+\frac{1}{T_I}\int e(t)dt+T_D\frac{de(t)}{dt}\right] \tag{1}$$

Where, $U(t)$ is the output signal of regulator; $e(t)$ is the error signal of regulator, which equals to the difference between measurement value and preset value; K_P is the proportional coefficient of regulator; T_I is the integral time of regulator; T_D is the differential time of regulator.

B. Digital PID Algorithm

Because computer control is a sampling control, it can only calculate control volume according to the error of the sampling moments[2,6]. Therefore, in the process of computer control, we must make dispersed treatment to equation (1) to use digital difference equation to take the place of differential equation in successive system. Then integral term and differential term can be expressed with summation and increment equation.

$$\int_0^n e(t)dt=\sum_{j=0}^{n}e(j)\Delta t=T\sum_{j=0}^{n}e(j) \tag{2}$$

$$\frac{de(t)}{dt}\approx\frac{e(k)-e(k-1)}{\Delta t}=\frac{e(k)-e(k-1)}{T} \tag{3}$$

We put (2) and (3) into (1) and then we can get dispersed PID equation.

$$\mathrm{U}(k)=K_P\left\{e(k)+\frac{T}{T_I}\sum_{j=0}^{k}e(j)+\frac{T_D}{T}\left[e(k)-e(k-1)\right]\right\} \tag{4}$$

Where, $\Delta t=T$ is the sampling period and T should be sufficiently small in order to obtain a definite precision; $e(k)$ is the error value of No.k sampling moment; $e(k-1)$ is the error value No.(k-1) sampling

doi:10.4304/jcp.7.10.2417-2424

moment; k is sampling serial number, k =0, 1, 3, …; $U(k)$ is the regulator output of No.k sampling moment.

From equation (4), if we want to get $\mathrm{U}(k)$, not only do we need the error signals $e(k)$ and $e(k-1)$, of this time and last time but we need plus previous error signals $e(j)$ from integral term, i.e. $\sum_{j=0}^{k} e(j)$. By this way, it is quite complicated to calculate, however, there needs much memory to store $e(j)$. So it is quite inconvenient to control directly with equation (4). Let's adjust it as follows:

$$\mathrm{U}(k-1)=K_P\left\{\mathrm{E}(k-1)+\frac{T}{T_I}\sum_{j=0}^{k-1}E(j)+\frac{T_D}{T}\left[E(k-1)-E(k-2)\right]\right\} \tag{5}$$

Subtract equation (5) from equation (4), we can get:

$$\begin{aligned}\mathrm{U}(k)=\mathrm{U}(k-1)+K_P\left[e(k)-e(k-1)\right]+\\ K_I e(k)+K_D\left[e(k)-2e(k-1)+e(k-2)\right]\end{aligned} \tag{6}$$

Where, $K_I=K_P\dfrac{T}{T_I}$ is integral coefficient;

$K_D=K_P\dfrac{T_D}{T}$ is differential coefficient.

From equation (6), to get the output value $\mathrm{U}(k)$, it is ok to get $\mathrm{U}(k-1)$, $e(k)$, $e(k-1)$ and $e(k-2)$. It is much easier than using equation (4).

Although equation (6) is revised only a little bit, it brings the following advantages.

(1). Because the output is increment, it has small influence of malfunction.

(2). It's easy to realize the non-interfering switches of control patterns.

(3). No integral out of control. It's easy to get better regulation quality.

III. SOFT COMPUTING TECHNIQUES

A. PID Controller on Fuzzy System

Fuzzy system is a system based on knowledge or rules. Its core is knowledge base made up of so-called IF-THEN rules which come from experts or the knowledge in this field[7-9]. Then, we apply all these rules to single system. Different fuzzy systems should adapt to different combination rules.

1. fuzzy system

The basic structure of a pure fuzzy system is as shown in Figure 1. Rule base can be shown as a set of several IF-THEN rules. Through combing these IF-THEN rules, fuzzy inference engine will decide how to map the fuzzy set of input domain of discourse $U\subset R^n$ on the fuzzy set of output domain of discourse $V\subset R$.

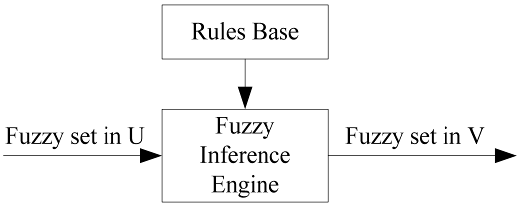


Figure 1. The structure of a pure fuzzy system.

The leading problem of a pure fuzzy system is that its input variable and output variable are all fuzzy sets, whereas, the input and output in engineering system are all real-valued variables[10,11].

To apply fuzzy system to engineering system, a simple method is to add a fuzzifier at the input end. Its purpose is to change the real-valued variable to a fuzzy set. At the output end, we add a defuzzifier, which is to switch the fuzzy set to real-valued variables. Figure 2 shows us a fuzzy system with a fuzzifier and a defuzzifier.

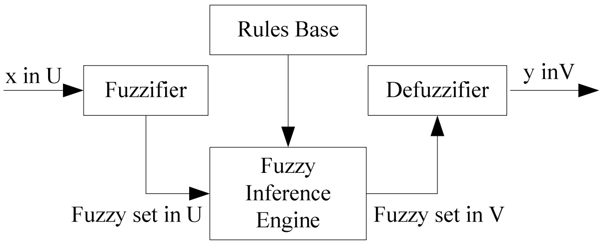


Figure 2. The structure of a fuzzy system with a fuzzifier and a defuzzifier.

Rule base is composed of fuzzy IF-THEN rule set. It is the core of a fuzzy system. Other components of a fuzzy system carry out these rules in a reasonable and efficient way. Rule base consists of the following IF-THEN rules.

$$R_u^{(l)}: \mathit{If}\ x_1\ \mathit{is}\ A_1^l\ \mathit{and}\ldots\mathit{and}\ x_n\ \mathit{is}\ A_n^l,\ \mathit{then}\ y\ \mathit{is}\ B^l \tag{7}$$

Where, A_i^l and B^l are fuzzy set of $U_i\subset R$ and $V\subset R$ respectively, $x=(x_1,x_2,\ldots,x_n)^T\in U$ and $y\in V$ are input and output variables of the fuzzy system.

To provide a general knowledge equation, equation (7) consists of the following rules:

(1). "incomplete rule"

$$\mathit{If}\ x_1\ \mathit{is}\ A_1^l\ \mathit{and}\ldots\mathit{and}\ x_m\ \mathit{is}\ A_m^i,\ \mathit{then}\ y\ \mathit{is}\ B^l\ \mathit{in}\ m<n$$

(2). "or rule"

$$\begin{aligned}&\mathit{If}\ x_1\ \mathit{is}\ A_1^l\ \mathit{and}\ldots\mathit{and}\ x_m\ \mathit{is}\ A_m^l\ \mathit{or}\ x_{m+1}\ \mathit{is}\ A_{m+1}^l\\ &\mathit{and}\ldots\mathit{and}\ x_n\ \mathit{is}\ A_n^l,\ \mathit{then}\ y\ \mathit{is}\ B^l\end{aligned}$$

(3). "single fuzzy statement"

$$y\ \mathit{is}\ B^l$$

(4). "stepwise variation rule"

$$\mathit{If}\ x\ \mathit{is\ smaller},\ \mathit{then}\ y\ \mathit{is\ bigger}$$

The function of a fuzzy inference engine is to compound the mapping of a fuzzy IF-THEN rule which exists in rule base. This mapping is from fuzzy set A' in

U on fuzzy set $B^{'}$ in V. Fuzzy inference engine is designed according to the following rules.

(1). In conformity with the experience of experts.

(2). High computer efficiency.

(3). Specific requirements of some features.

Product inference engine has the advantage of ease of computing and can directly reflect actual problems. Its equation is as shown in equation (8).

$$\mu_{B^{'}}(y) = \max_{l=1}^{M}[\sup_{x\in U}(\mu_{A^{'}}(x)\prod_{i=1}^{n}\mu_{A_i^l}(x_i)\mu_{B^{'}}(y))] \tag{8}$$

If you offer a fuzzy set $A^{'}$ in U, the inference engine can product a fuzzy $B^{'}$ in V.

A fuzzifier can be defined as a mapping of $x^* \in U \subset R^n$ on fuzzy set $A^{'}$ in U. The fuzzifier is designed according to the following rules.

(1). The fuzzifier should consider the fact that it is at clear point x^* that input occurs.

(2). If the input of a fuzzy system is interfered by noise, the fuzzifier needs the ability to overcome the influence of noise.

(3). The fuzzifier should contribute to simplify the computing of fuzzy inference engine.

The triangle fuzzifier maps $x^* \in U$ on fuzzy set $A^{'}$ in U. It has the following triangle membership function.

$$\mu_{A^{'}}(x) = \begin{cases} (1\text{-}\frac{|x_{1\text{-}}x_1^*|}{b_1})*...*(1\text{-}\frac{|x_{n\text{-}}x_n^*|}{b_n}) & in|x_{n\text{-}}x_n^*| \le b_i(i=1,2,\ldots,n) \\ 0 & others \end{cases} \tag{9}$$

Where, parameter $b_i(i=1,2,\ldots,n)$ is positive number, t-nom * usually chooses algebraic product operator or min operator.

A defuzzifier can be defined as a mapping of fuzzy set $\mathrm{B}^{'}$ (output of the fuzzy inference engine) in $\mathrm{V} \subset \mathrm{R}$ on $y^* \in V$. Defuzzifier methods should consider the following rules.

(1). Point y^* should visually represent $\mathrm{B}^{'}$.

(2). Ease of computing is quite important because of controller's real-time operation.

(3). Continuity is needed. Slight change of $\mathrm{B}^{'}$ shouldn't cause substantial variation of y^*.

The center average defuzzifier can be defined as the following equation (10).

$$y^* = \frac{\sum_{l=1}^{M}\bar{y}^l\omega_l}{\sum_{l=1}^{M}\omega_l} \tag{10}$$

Where, $\bar{y}^l$ is the center of the l th fuzzy set; ω_l is the height. The center average defuzzifier is ease of computing, reasonable and visual.

2. The structure of pid controller with fuzzy system

The PID controller with a fuzzy system takes error e and error variation ec as its input. It utilizes fuzzy system to tune PID parameters on line, which can satisfy the requirements of different e and ec to PID parameters' adaptation. Its structure is as shown in Figure3.

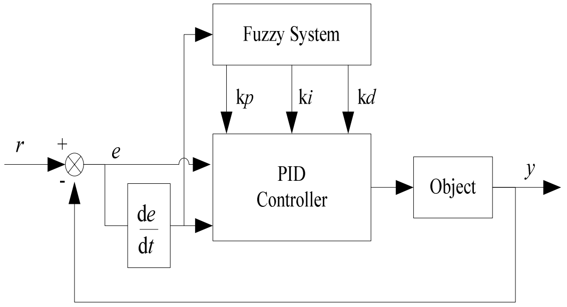


Figure 3. The structure of PID controller with a fuzzy system.

where the core of the PID controller is to build up fuzzy rule base on the basis of engineering technicians' knowledge and actual operation equation. The fuzzy sets of e and ec are {NB,NM,NS,O,PS,PM,PB} in which all the components stand for negative big, negative medium, negative small, zero, positive small, positive medium, positive big. The following fuzzy rules are established.

If (e is NB) and (ec is NM) then (kp is PB)(ki is NB)(kd is NS)

If (e is NM) and (ec is PS) then (kp is PS)(ki is NS)(kd is NM)

……

If (e is PM) and (ec is NB) then (kp is PS)(ki is Z)(kd is PB)

If (e is PB) and (ec is PM) then (kp is NB)(ki is PB)(kd is PS)

B. PID Controller on RBFNN

Radial basis function neural network (RBFNN) is a three-layer feed forward[12]. With radical basis function network, we can construct a self-learning PID controller with parameters k_p, k_i and k_d.

1. Structure of Neural Network PID Controller

The PID control strategy based on neural network is shown in Figure 4. The regulator consists of two parts:

PID controller can directly closed-loop control the objects and three parameters k_p, k_i, k_d can be adjusted on line.

According to the operation condition of the system, neural network regulates the parameters of PID controller to optimize some performance index[13,14]. Through self-learning of neural network and adjustment of weight coefficient, neural network can obtain some optimized PID control parameter.

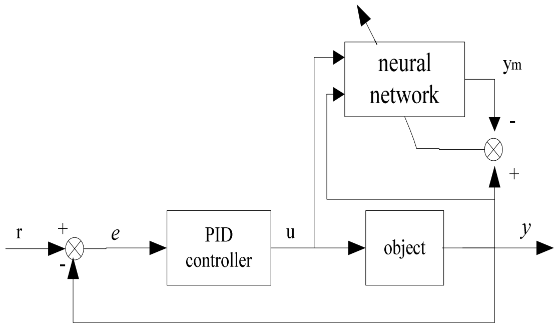


Figure 4. Structure of neural network PID controller

Where, r is the system input, and y is the system output. y_m is the output of neural network. u is the output of PID controller.

2. Neural Network Identification Algorithm

Radial basis function neural network consists of input layer, hidden layer and output layer, in which input layer consists of signal source nodes, the number of units in hidden layer is decided by the need of the described questions and output layer responds to the input layer[15]. The mapping from input to output is nonlinear, whereas the mapping from hidden layer space to output space is linear. Therefore, the learning speed will be much faster and at the same time local minimum problems will be avoided.

Suppose neural network is made up of n input nodes, m hidden nodes and one output node. In the structure of neural network, the input vector of network input layer can be expressed as $X=[x_1,x_2,...x_n]^T$, the radial basis vector of the network hidden layer can be expressed as $H=[h_1,h_2,h_j...h_m]^T$, where h_j is radial basis function may choose different algorithms. Here, we'd like to choose Gaussian function, i.e. (11).

$$h_j=\exp\left(-\frac{\left\|X-C_j\right\|^2}{2b_j^2}\right),\ j=1,2,...m \tag{11}$$

Where, C_j is the center vector of No.j node in network. The expression is as follows:

$$C_j=\left[c_{j1},c_{j2},...c_{ji}...c_{jn}\right]^T,(i=1,2,...n) \tag{12}$$

Suppose the base width vector of the network is $B=\left[b_1,b_2......b_m\right]^T$, b_j is the base width parameter of node j and b_j is greater than zero. The network weight vector is $W=\left[w_1,w_2...w_j...w_m\right]^T$. The output expression of identification network is as follows.

$$y_m(k)=\sum_{j=1}^{m}w_jh_j \tag{13}$$

In order to obtain better result of network identification, while revising weight coefficient, an inertia term should be added in. According to low gradient descent, the interactive algorithm of output weight, node center and node width parameter can be expressed as equations (14) (15) and (16)

$$\begin{aligned}&\Delta W_j=[y(k)-y_m(k)]h_j\\&W_j(k)=W_j(k-1)+\eta\Delta W_j+\alpha[w_j(k-1)-w_j(k-2)]\\&+\beta[W_j(k\text{-}2)\text{-}W_j(k\text{-}3)]\end{aligned} \tag{14}$$

$$\begin{aligned}&\Delta b_j=[y(k)-y_m(k)]w_jh_j\frac{\left\|X-C_j\right\|^2}{b_j^3}\\&b_j(k)=b_j(k-1)+\eta\Delta b_j+\alpha[b_j(k-1)-b_j(k-2)]\\&+\beta[b_j(k-2)-b_j(k-3)]\end{aligned} \tag{15}$$

$$\begin{aligned}&\Delta c_{ji}=[y(k)-y_m(k)]w_j\frac{X_j-C_{ji}}{b_j^2}\\&c_{ji}(k)=c_{ji}(k-1)+\eta\Delta c_{ji}+\alpha[c_{ji}(k-1)-c_{ji}(k-2)]\\&+\beta[c_{ji}(k-2)-c_{ji}(k-3)]\end{aligned} \tag{16}$$

Where, η is learning rate, and α , β is inertia coefficients whose value range is between 0 and 1.

The sensitivity information algorithm of the output of the controlled object to the control input is as follows.

$$\frac{\partial y(k)}{\partial u(k)}\approx\frac{\partial y_m(k)}{\partial u(k)}=\sum_{j=1}^{m}w_jh_j\frac{c_{1j}-x_1}{b_j^2} \tag{17}$$

In the actual application, x_1 is the input term of identification network and its value is $u(k)$.

3. PID Parameter Tuning Principle online

According to Figure 4, control error is:

$$e(k)=r(k)-y(k) \tag{18}$$

Suppose there are parameters c_1,c_2,c_3 .

$$c_1=e(k)-e(k-1)$$
$$c_2=e(k)$$
$$c_3=e(k)-2e(k-1)+e(k-2)$$

PID control strategy uses equation (6) to calculate.

$$u(k)=u(k-1)+k_pc_1+k_ic_2+k_dc_3$$

(19)

Neural network tuning target is :

$$E(k)=\frac{1}{2}e^2(k)$$

Low gradient descent should be adopted to adjust parameters k_p , k_i and k_d . Its calculation process is as follows:

$$\Delta k_p=-\eta\frac{\partial E}{\partial k_p}=-\eta\frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial k_p}=\eta e(k)\frac{\partial y}{\partial u}c_1 \tag{20}$$

$$\Delta k_i = -\eta \frac{\partial E}{\partial k_i} = -\eta \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial k_i} = \eta e(k)\frac{\partial y}{\partial u}c_2 \tag{21}$$

$$\Delta k_d = -\eta \frac{\partial E}{\partial k_d} = -\eta \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial k_d} = \eta e(k)\frac{\partial y}{\partial u}c_3 \tag{22}$$

Where, $\frac{\partial y}{\partial u}$ is information of controlled object which could be obtained through neural network identification.

C. PID Controller on GA

1. Genetic algorithms

Genetic algorithms (GA), which imitate genetic mechanism and the theory of evolution, is the most optimized parallel random searching method[16,17]. GA introduces the rule of the theory of evolution" survival of the fittest" to the coded string group composed of the most optimized parameters. According to adaptation value functions, GA which selects units through copying, crossing and varying retains high adaptation value units which form new units that inherit the information from previous generation and optimize the previous generation in turn. Repeatedly in this way, the adaptation ability of units is continuously improved till a satisfactory situation. Its algorithm is quite easy and can be concurrently dealt with to get overall optimization.

PID controller to which GA is applied can employ GA to optimize parameters. This method is the one which doesn't need any initial information and can find the most optimized and the most efficient combination method.

The elements for GA:

(1) Chromosome coded method. Basic GA employs fixed length binit to mark units in the group and its allele is composed of symbol set {0.1}.

(2) Unit's sufficiency evaluation. The directly proportional probability of GA and unit sufficiency can decide how much probability it will be from the units of current group to the group of the next generation. To correctly calculate the probability, all the sufficiency must be firstly defined from objective function to unit sufficiency.

(3) Genetic operator. Basic GA employs genetic operators which include selection operator which selects operation ratio, one-point crossover operator which employs crossover operation, simple mutation operator which employs mutation operation or uniform mutation operator.

(4) The operation of basic GA demands predetermined parameters:
M: group size i.e. the number of units in the group.
G: the last evolution algebra of GA.
P_c: crossover probability.
P_m: mutation probability.

2. PID setting principle based on GA

(1). Parameter determination and representation.

At first parameter scope should be determined and this scope should be given by users. Next, in terms of precision, coding should be made. Binary character string should be chosen to represent each parameter and relation should be set up. At last, binary character string should be attached to form a long binary character string which is that object that GA may operate.

(2). Selecting initial population.

Because all the processes can be realized through programming, initial populations are generated by computer random. As for binary coding, firstly evenly distributed random numbers between 0 and 1 are generated[18]. Then 0 stands for the generated numbers between 0 and 0.5; 1 stands for the generated numbers between 0.5 and 1. Besides, the size of population is decided by the complexity of computer.

(3). Fitting function determination.

Under constraint conditions, common optimizing algorithm can obtain a group of satisfactory parameters. In the process of designing, the best parameter should be chosen from this group of parameters. There are three aspects we use to measure the indicators of the controlling system, i.e. stability, accuracy and rapidity. Rise time reflects the rapidity of the system. The shorter the rise time spends, the faster the controlling carries out.

(4). GA operation.

At first, we should employ sufficiency ration to copy, i.e. through fitting function we can get fitting value to get corresponding copy probability of each string. The product of copy probability and the number of strings of each generation is the number of strings copied in the next generation. The bigger copy probability is, the more offspring the next generation will have. On the contrary, it will die out.

Through copy, crossover and mutation, initial population obtains a new population, which can be taken into fitting function after decoding. Then we should check if it satisfies termination condition. If not, the above operation will repeat till making it.

In order to obtain satisfactory dynamic characteristics of transient process, performance indicator of error absolute value time should be adopted as the minimum objective function of parameter selection. To avoid excessive control energy, quadratic term of control input should be added to objective function. The following equation is chosen as the most optimized indicator of parameter selection.

$$J = \text{ò}_0^{\text{¥}} (w_1 | e(t) | + w_2 u^2(t)) d_t + w_3\ t_u \tag{23}$$

Where, $e(t)$ is the system error. $u(t)$ is controller output. t_u is rise time. w_1, w_2, w_3 are weights.

To avoid overshoot, punitive function is adopted, i.e. once overshooting, the overshoot is considered as a term of the most optimized indicator. At that time, the most optimized indicator is as follows:

$$J = \int_0^{\infty} (w_1 |e(t)| + w_2 u^2(t) + w_4 |e(t)|) d_t + w_3 \ t_u$$
$$if .e(t) < 0 \qquad (24)$$

Where, w_4 is a weight and $w_4 >> w_1$.

IV. Experiment's Result and Analysis

Some rock mechanics testing machine has the functions of real-time control, monitoring and feedback. Its control process is nonlinear and highly complicated. The control system asks for realizing automated control with high accuracy and high reliability to the testing machine.

The transfer function of the rock mechanics testing machine system is equation below:

$$G(s) = \frac{0.4}{0.015s^3 + 0.523s^2 + 0.12s}$$

Here, traditional PID control algorithm and intelligent PID control algorithm based on soft computing are adopted to respectively make simulation experiment and analysis to the system.

The executive body of the system adopts driving step motor to control. Its control algorithm is as shown in equation (6). The process of control algorithm is as follows:

(1) initializing parameter.
(2) sample $r(k)$ and $y(k)$.
(3) calculating error value.
(4) calculating controller output;
(5) renewing parameters and returning to procedure (2).

Controller parameters are set as follows:

k_p =20, k_i =0.2 and k_d =8. The simulation result is shown in Figure 5.

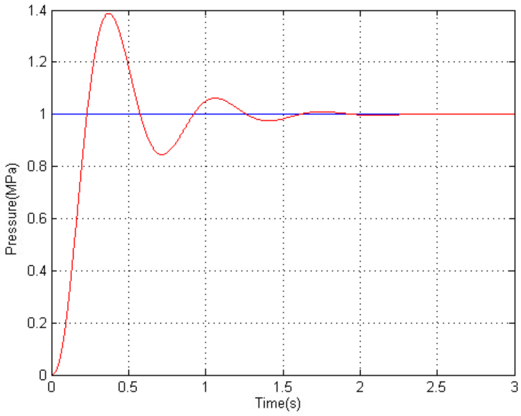


Figure 5. The result of conventional PID.

During intelligent tuning, we define the variation scopes of system error e and error variation rate ec as the domain of discourse on fuzzy set and then decide its fuzzy subsets and their membership. While controlling on line, control system together fuzzy system's handling and computing finishes proofreading to PID parameters on line. Its working flow is as follows:

(1) Taking current sampling value.
(2) Computing $e(k)$, $ec(k)$ and $e(k-1)$.
(3) Fuzzifying $e(k)$ and $ec(k)$.
(4) Fuzzy adaptation to Δk_p, Δk_i and Δk_d.
(5) Computing current k_p, k_i and k_d.
(6) Effect of PID controller's output on the controlled system.

The simulation result is shown in Figure 6.

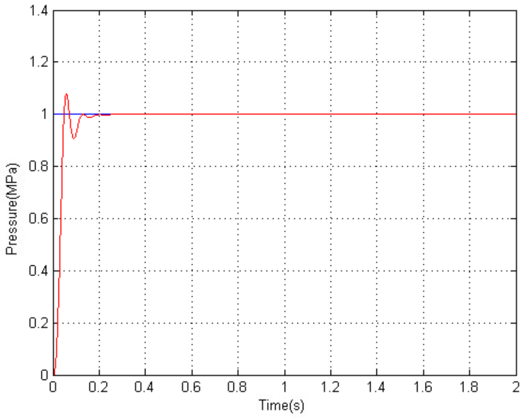


Figure 6. The result of PID controller on fuzzy system.

The procedures of control strategy using RBFNN are as follows:

(1) Making sure the number of network input nodes, the number of hidden layers, learning rate and inertia coefficient to get center vector of hidden nodes, base width vector of the network and the initial value of weight vector.
(2) Sampling to get input r and output y, and calculating error e in terms of equation (18).
(3) Calculating the output u of regulator according to equation (19) to control the controlled object real time.
(4) Calculating network output ym and adjusting center vector of network hidden nodes and base width vector and weight vector to obtain network identification information.
(5) Adjusting parameters of regulator in terms of equations (20) (21) and (22).
(6) Going back to procedure (2).

The structure of neural network is 3-6-1. The three inputs of network identification are $u(k)$, $y(k)$ and $y(k-1)$. Learning rate η =0.25. Inertia coefficients α =0.05, β =0.01. The simulation result is shown in Figure 7.

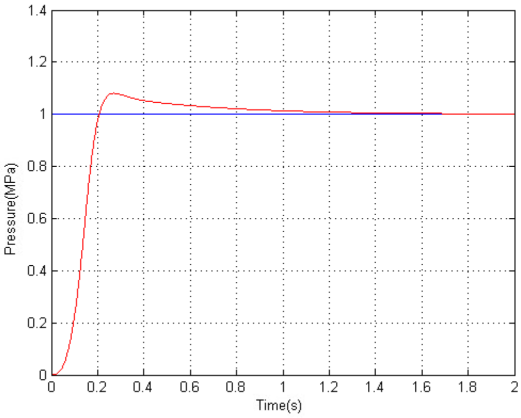


Figure 7. The result of PID controller on RBFNN.

The specific procedures of optimizing PID parameters using GA are as follows:

(1) making sure the scope and coding length of each parameter and coding.
(2) generating initial population made up of n random units.
(3) decoding respective parameter value from the population and then using this parameter value to get cost function value and fitting function value.
(4) using copy, crossover and mutation operators to operate population to produce a new population.
(5) operating procedures (3) and (4) till parameter convergence or achieving desired indicator.

Adopting binary coding is to avoid using the length of 10 bits binary coding string to mark three decision variables k_p, k_i and k_d. The equation (24) is used to select the most optimized indicator. The number of samples used in GA is Size=30. Crossover probability and mutation probability are Pc=0.60, Pm=0.001-[1:1:Size]x0.001/Size. The value span of parameter k_p is [0.20]. The value span of k_i and k_d is [0.1], w_1=0.99, w_2=0.001, w_3=1.0, w_4=100.

The simulation result is shown in Figure 8.

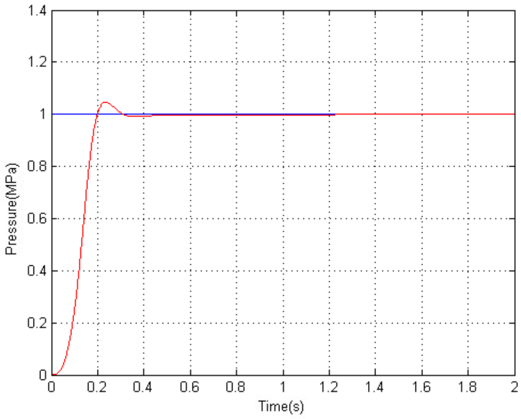


Figure 8. The result of PID controller on GA.

From above Figures, traditional control algorithm needs longer tuning time, rise time and larger overshoot. The controller employing soft computing technique to realize can obtain better tuning performance for the system tuning. The specific comparison is shown in tableI.

TABLE I.
COMPARISON OF STEADY STATE RESPONSES

	Peak over shoot (%)	Settling time (sec)	Rise time (sec)
Con.	39.6	2.52	0.362
FS	12.7	0.26	0.067
NN	8.2	1.7	0.261
GA	6.7	1.23	0.236

V. Conclusion

The PID control algorithm is the basis for many advanced control algorithms and strategies in the process industries. In order to use a controller, it must first be tuned to the system. This tuning synchronizes the controller with the controlled variable, thus allowing the process to be kept at its desired operating condition. Conventional PID control method can be used only when the process mode is of a certain type. The study on PID controller using soft computing technique embodies its excellence. Soft computing has the features of intelligence and self-adaptation, so there is no need for the controlled objects constructing precise model. The system has stronger capacity of resisting disturbance. For complicated controlled objects, the system can obtain higher control quality, hence robustness.

While solving complicated actual problems, the system may combine knowledge, techniques and methods of various sources. Various computing models of soft computing technique aren't mutually exclusive and teamwork will own bigger excellence. The systems employing different computing techniques are regarded as complementary mixed intelligent systems, which will be a further study content.

Acknowledgment

The authors would like to acknowledge the financial support of The Scientific and Technological Key Support Projects with the project number 20090307.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J.Astrom and T.H.agglund. PID Controllers: Theory, design and tuning. Instrument society of America, 2nd edition, 1995.
[3] Kiam Heong Ang, Gregory Chong: "PID Control System Analysis, Design, and Technology" IEEE Transactions on

Control Systems Techonology, Vol.13, No.4, July 2005 pp. 559-576.

[4] Honghua Xu, Guangcai Cui, Ji Li. FuzzyPID Control Technology Application in Rock-testing System. Journal of changchun University of Science and technology. 2005,28(3), pp. 42-45.

[5] Neuron Chip Distributed Communication and Control Processors. Motolora Inc., 1994.

[6] Qing-Guo Wang,Tong-Heng Lee: "PID Tuning for Improved Performance" IEEE Transactions on Controll Systems Technology, Vol.7, No.4 July 1999, pp. 457-465.

[7] Gravel, A. and H. Mackenberg, Mathematical analysis of the Sugeno controller leading to general design rules. Fuzzy Sets and Systems, 85, pp. 165-175, 1995.

[8] Kadmiry B.,Bergsten P. "Robust Fuzzy Gain Scheduled visual-servoing with Sampling Time Uncertainties" IEEE Int. Symposium on Intelligent Control (CSS / ISIC), Taiwan 2004.

[9] Z. Y. Zhao, M. Tomizuka, and S. Isaka, .Fuzzy gain scheduling of PID controllers,. IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 5, pp. 1392.1398, 1993.

[10] Honghua Xu, Guang Dong, Weili Shi. "An Intelligent PID Controller Based on Fuzzy System", IEEE Conf. ICEEE2010, vol. 5, pp. 3117-3121, 2010.

[11] Di Nola, A., S. Sessa, W. Pedrycz, and E. Sanchez, Fuzzy Relation Equations and Their Applications to Knowledge Engineering, Kluwer, Boston, 1989.

[12] Honghua Xu, Yinghua Tian. "An Intelligent PID Controller on RBFNN", IEEE Conf. Electrical Engineering and Automation Control, vol. 4, pp. 1256-1259, 2011.

[13] Cheeseman, P., "An inquiry into computer understanding," (with discussions) Computational Intelligence, 4., pp.58-142, 1988.

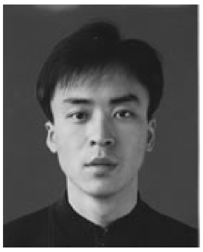
[14] Bonadeo N H, Erland J, Gammon D, et al. Coherent Optical control of the Quantum State of a Single Quantum Dot. Science, 282: pp.1473-1475, 1998.

[15] Karayiannis, N. B. Reformulated radial basis neural networks trained by gradient descent. IEEE Trans. on Neural Networks. 10(3): pp. 657-671, 1999.

[16] Jukka Lieslehto "PID controller tuning using Evolutionary programming" American Control Conference, VA June pp. 25-27, 2001

[17] Hang, C. C. and K. K. Sin, "A comparative performance study of PID auto-tuners", IEEE Control Syst. Msg., vol. 11, pp. 41-47, 1991.

[18] T. Ota and S. Omatu, "Tuning of the PID control gains by GA," in Proc. IEEE Conf. Emerging Technol. Factory Automation, Kauai, HI, Nov., pp. 272–274, 1996.



Honghua Xu graduated from Changchun University of Science and Technology in 2004. He is a lecturer at Changchun University of Science and Technology and mainly engaged in teaching and research work. He has participated in some provincial and ministerial research projects. He has presented many papers in journals and international conferences. Main areas of research are computer simulation and artificial intelligence.

Xiaoqiang Di graduated from Changchun University of Science and Technology in 2002. He is a lecturer at Changchun University of Science and Technology and mainly engaged in teaching and research work. He has participated in some provincial and ministerial research projects. He has presented many papers in journals and international conferences. Main areas of research are image processing and artificial intelligence.

Huamin Yang graduated from Dalian University of Technology in 1985. He is a professor and doctoral tutor at Changchun University of Science and Technology and mainly engaged in teaching and research work. He is a member of the Computer Federation of Province. He has chaired many provincial and national research projects, including some of projects received awards. He has presented more than 70 papers in journals and international conferences. Main areas of research are computer simulation, image processing and artificial intelligence.

Guangcai Cui graduated from Beijing Institute of Technology in 1986. He is a professor and doctoral tutor at Changchun University of Science and Technology and mainly engaged in teaching and research work. He is a member of the Computer Federation of Province. He has chaired several provincial research projects, including some of projects received awards. He has presented more than 40 papers in journals and international conferences. Main areas of research are artificial intelligence and Chinese information processing.