TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 1788 ~ 1798
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4448

An Intelligent Temperature Controller for the Insulation Tool Storeroom of Power System

Chunzhi Wang[1]**, Siwei Wei***[2]
[1]School of Computer Science, Hubei University of Technology
[2]School of Automation, Wuhan University of Technology
*Corresponding author, e-mail: chunzhiwang@vip.163.com

Abstract

The storeroom of live working insulated tools of power system has some special requirements such as temperature and humidity. In general, the temperature is stabilized at a particular value and the humidity is higher than a certain value. The storeroom temperature and humidity are mainly controlled by the heater, dehumidifier and air-conditioning compressor. In order to well maintain the stability of the temperature, fuzzy neural network is used to control temperature which makes parameters and rules optimized by the neural network learning. So the designed controller has strong self-adapting character in controlling process. In order to improve learning speed, GA algorithm is used to optimize those initial values of fuzzy reasoning networks. And in order to overcome the shortcoming of local optimization, designed learning parameters can be adjusted according to error and error change in training. Simulating experiment shows that the designed controller has better controlling effect than other conventional controller and which can be used in the storeroom of live working insulation tools in power system.

Keywords: *temperature control, fuzzy neural network, storeroom of live working insulated tool, network training*

1. Introduction

In order to make uniform management for live working insulated tools of power systems, it is needed to design a dedicated storeroom for live working tools. With the development of the power system, the requirement of environment for live working tools is more and more strict. In order to reduce the declining of insulation performance, environmental temperature and humidity should be stable. The stable and liable automatic monitoring system for storeroom to control the temperature and humidity automatically is needed [1].

Due to large delay links in the temperature control, simple PID control cannot get good control performance. Fuzzy control is widely used in the industry in recent years [2, 3]. But it's difficult to set fuzzy control parameters using traditional fuzzy control. Neural control is another novel control method [4, 5]. But it also has some problems in practice i.e. Different combinations of fuzzy logic and neural networks provide various ingredients for smart adaptive applications [6]. So the intelligent algorithm combining fuzzy and neural network is designed in this paper to control temperature and humidity. The control strategy based on fuzzy neural network is given for the live working tools storeroom. The experiment shows that the control effect is much better than traditional way and the control strategy can be used in other similar products.

2. Structure of Control System

It is essential to construct the control system modeling before design a temperature controller. The controlled equipment includes air conditioning compressor, heater and dehumidifier. Each part owns its respective control method. The collected data are used in simulation analysis experiment. The cooling (heating) amount of air conditioner compressor has a certain relationship and it is converted the frequency of the compressor speed to reach the heat requirement [7]. For heater, it just trims different power by outputting different pulse width signals if temperature is lower than a certain value. The dehumidifier works when the humidity is below the set value. Corresponding to the equipment, the controlled plant model is composed by three parts. Air conditioning model is equivalent to an integrator with an inertia series link. Its

Received August 23, 2013; Revised September 30, 2013; Accepted October 9, 2013

input is frequency and output is changing temperature. The heater model can be seen as a link with delay inertia [8]. Compared with the above models, the room model is more complex due to the mixed impact of outdoor temperature interference, heater, dehumidifier, heat sink, air thermal conductivity delay and so on [9]. All models are simulated and analyzed though the experiment data. It also can use neural fuzzy networks in modeling data. When the outdoor temperature is lower than the set temperature, the heater is working and in other case the heater doesn't work. So the heater and air conditioning work in parallel. The heater is working at low temperature and air-conditioning compressor is working at high temperature [10].

The needed heat of the room is calculated according to the difference between the detected value and the given value and then to control the air conditioning compressor and the heater. In the case that the environmental temperature is higher than set value, the control signal u1 from the air-conditioning compressor operation is given and which is change frequency f to regulate the compressor power P. In the case that the environmental temperature is lower than set value, the pulse width signal *u2* is output to control heater work. The control system structure is given as Figure 1.

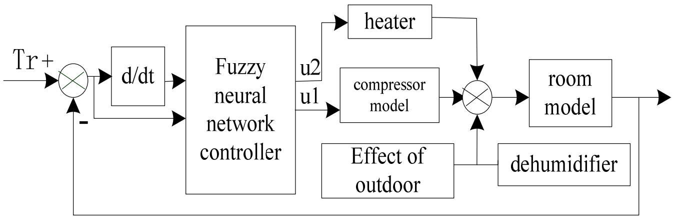

Figure 1. Structure of Storeroom Temperature Control

The error *e* is calculated according to the set temperature value Tr and the actually measured temperature value Ty. Then the error change rate *ec* is obtained. And these values are used as input signals of fuzzy neural network controller.

3. Design of FNNC

The controller has two working situations. When the outdoor temperature is higher than the set value, the air-conditioning operation and the controller calculates the amount of energy and calculates control output value for air conditioning. Here, the heater's controller output is zero. When the outdoor temperature is below than the set value, the controller calculates the control value for the heater. Here, the air conditioning compressor doesn't start and the control output is zero. Two control models are designed. The fuzzy nerve network structure is shown in Figure 2 which contains 4 layers.

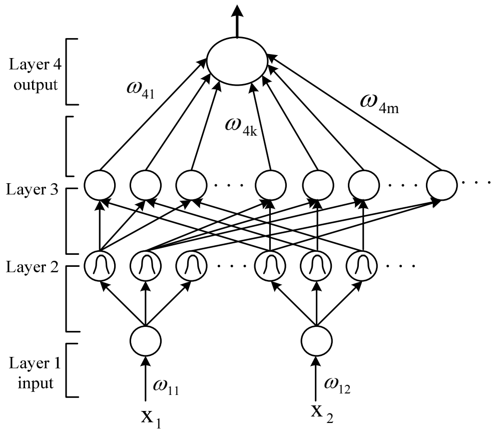

Figure 2. Structure of FNNC

The entire network combining input and output mapping relations are given as follow [11].

1) The first layer: Here, error e and error change rate ec are used as input value. ω_{11} and ω_{12} are weight coefficients. The relation between input and output is given as follow:

$$\begin{cases} I_i^1 = x_i \omega_{1i}\,, \quad i = 1{,}2 \\ O_i^1 = I_i^1 \end{cases} \tag{1}$$

2) The second layer: Nodes at layer 2 are term nodes that act as membership functions (MF) to represent the terms of the respective linguistic variable. Each input have 7 grades fuzzy linguistic variable, so there are 14 nodes in layer 2. Gaussian MF is used as transform function. The outputs of the layer are given as follow:

$$\begin{cases} I_{ij}^2 = -\dfrac{(O_i^1 - m_{ij})^2}{\sigma_{ij}^2}, \quad i = 1{,}2 \\ \mathrm{O}_{\mathrm{ij}}^2 = \mu_{\mathrm{i}}^{\mathrm{j}} = \exp(I_{ij}{}^2) \quad , \quad j = 1{,}2 \cdots 7 \end{cases} \tag{2}$$

3) The third layer: Each node at layer 3 represents fuzzy logic relationship. Thus all nodes of the layer form a fuzzy rule base. All rule nodes perform the fuzzy product operation, and there are 49 logic rules. The functions are given as follow:

$$\begin{cases} I_k^3 = \prod\limits_{i=1}^{2} \mu_i^j\,, \quad i = 1{,}2, \quad j = 1{,}2 \cdots 7 \\ \mathrm{O}_{\mathrm{k}}^3 = I_k^3 \quad , \quad k = 1{,}2 \cdots 7^2 \end{cases} \tag{3}$$

4) The fourth layer: In layer 4 nodes, the reasoning consequent of every reasoning rule is synthesized through connecting weight ω_{4k}. It is expressed as follow:

$$\begin{cases} I^4 = \sum\limits_{k=1}^{m} O_k^3 * \omega_{4k}\,, \quad m = 49 \\ O^4 = u = \dfrac{I^4}{\sum\limits_{k=1}^{m} O_k^3} \end{cases} \tag{4}$$

In above formula, m_{ij} and σ_{ij} are the center and the width of the Gaussian MF of the jth input term node of the ith input linguistic node. ω_{1i} and ω_{4k} are connecting weights.

4. The Improved Learning Algorithm of FNNC

4.1. Learning Algorithm of FNNC

In order to obtain good network parameters, it is needed to training network parameters. Those parameters contain connecting weights, such as ω_{1i}, ω_{4k}, the center, width of the Gaussian MF, such as $\mathrm{m}_{\mathrm{ij}}, \sigma_{\mathrm{ij}}$. Here, the supervised gradient decent method is used as the learning algorithm of FNNC network. The energy function E is defined as follow:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_{ri} - y_i)^2 \tag{5}$$

Where, y_{ri} is given output value, y_i is real room temperature value, n is learning sample number. In order to achieve the aim of self-learning and self-adapt control performance, the learning algorithm based on back-propagation method is used to change connecting weights and MF parameters of network [12]. The training ways are given as follow:

$$\begin{cases} \omega(k+1) = \omega(k) + \eta(-\dfrac{\partial E}{\partial \omega}) + \alpha[\omega(k) - \omega(k-1)] \\ m_{ij}(k+1) = m_{ij}(k) + \eta(-\dfrac{\partial E}{\partial m_{ij}}) + \alpha[m_{ij}(k) - m_{ij}(k-1)] \\ \sigma_{ij}(k+1) = \sigma_{ij}(k) + \eta(-\dfrac{\partial E}{\partial \sigma_{ij}}) + \alpha[\sigma_{ij}(k) - \sigma_{ij}(k-1)] \end{cases} \tag{6}$$

Where, η is learning rate parameter, $\eta > 0$, α is smoothness factor, $0 < \alpha < 1$. Here, above differential coefficient, such as $\frac{\partial E}{\partial \omega}$, $\frac{\partial E}{\partial m_{ij}}$ and $\frac{\partial E}{\partial \sigma_{ij}}$, are solved through back-propagation method. So those parameters may be adjusted through above adjust algorithm. Where $\frac{\partial E}{\partial \omega}$ may be computed through formula (7)~(13).

$$\frac{\partial E}{\partial \omega_{4k}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial u}\frac{\partial u}{\partial \omega_{4k}} = -\sum_{i=1}^{n}(y_{ri} - y_i)(\frac{\partial y_i}{\partial u}) \cdot \frac{O_k^3}{\sum_{k=1}^{m} O_k^3} \tag{7}$$

Where $\frac{\partial E}{\partial \omega_{4k}}$ may be expressed as:

$$\frac{\partial E}{\partial \omega_{4k}} = \frac{\partial E}{\partial I_k^4}\frac{\partial I_k^4}{\partial \omega_{4k}} = -\delta_k^4 \cdot O_k^3 \tag{8}$$

In the output layer, the error term δ_k^4 is propagated through follow given formula.

$$\begin{cases} \delta_k^4 = -\dfrac{\partial E}{\partial I^4} = -\dfrac{\partial E}{\partial y_i}\dfrac{\partial y_i}{\partial O^4}\dfrac{\partial O^4}{\partial I^4} \\ \quad = \sum_{i=1}^{n}(y_{ri} - y_i) \cdot \dfrac{1}{\sum_{k=1}^{m} O_k^3} \cdot \dfrac{\partial y_i}{\partial u} \end{cases} \tag{9}$$

Based on the same back-propagation method, the error term of the third layer, the second layer and the first layer are given as follow.

$$\delta_k^3 = -\frac{\partial E}{\partial I_k^3} = -\frac{\partial E}{\partial I_k^4} \cdot \frac{\partial I_k^4}{\partial O_k^3} \cdot \frac{\partial O_k^3}{\partial I_k^3} = \delta_k^4 \cdot \omega_{4k} \tag{10}$$

$$\begin{cases} \delta_{ij}^2 = -\dfrac{\partial E}{\partial I_{ij}^2} = -\sum_{k=1}^{m} \dfrac{\partial E}{\partial I_k^3} \cdot \dfrac{\partial I_k^3}{\partial O_{ij}^2} \cdot \dfrac{\partial O_{ij}^2}{\partial I_{ij}^2} \\ \quad = \sum_{k=1}^{m} \delta_k^3 \cdot \dfrac{\partial I_k^3}{\partial O_{ij}^2} \cdot \dfrac{\partial O_{ij}^2}{\partial I_{ij}^2} = \sum_{k=1}^{m} \delta_k^3 \cdot s_{ij} \cdot O_{ij}^2 \end{cases} \tag{11}$$

Where, if $\mathrm{O}_{\mathrm{ij}}^2=\mu_i^j$ is an input of *k*th rule node, s_{ij} is computed using formula (12).

$$s_{ij}=\frac{\partial I_k^3}{\partial O_{ij}^2}=\prod_{\substack{j=1\\p\neq i}}^{7}\mu_p^j,\quad p=1,2\quad j=1,2\cdots 7 \tag{12}$$

If above condition cannot meet, s_{ij} is 0.

$$\delta_i^1=-\frac{\partial E}{\partial I_i^1}=-\frac{\partial E}{\partial I_{ij}^2}\cdot\frac{\partial I_{ij}^2}{\partial O_i^1}\cdot\frac{\partial O_i^1}{\partial I_i^1}=\sum_{j=1}^{7}\delta_{ij}^2\cdot\frac{2(O_i-m_{ij})}{\sigma_{ij}^{\ 2}} \tag{13}$$

The adjust formulas of weight may be shown as follow.

$$\omega_{4k}(k+1)=\omega_{4k}(k)+\eta\delta_k^4\cdot O_k^3+\alpha[\omega_{4k}(k)-\omega_{4k}(k-1)] \tag{14}$$

$$\omega_{1i}(k+1)=\omega_{1i}(k)+\eta\delta_i^1\cdot x_i+\alpha[\omega_{1i}(k)-\omega_{1i}(k-1)] \tag{15}$$

Center and width of Gauss function may be computed using formula (16).

$$\begin{cases} m_{ij}(k+1)=m_{ij}(k)+\eta(\delta_k^2\dfrac{2(x_i-m_{ij})}{\sigma_{ij}^{\ 2}})+\alpha[m_{ij}(k)-m_{ij}(k-1)] \\ \sigma_{ij}(k+1)=\sigma_{ij}(k)+\eta(\delta_k^2\dfrac{2(x_i-m_{ij})^2}{\sigma_{ij}^{\ 3}})+\alpha[\sigma_{ij}(k)-\sigma_{ij}(k-1)] \end{cases} \tag{16}$$

4.2. The Improved Learning Way

In the processing of the training of FNNC, the parameter of learning rate η has big effect to training speed. In real control, when work state having big varying, it is desired to adjust to stable state rapidly. But when adjusting error is less, error e, the change of control value Δu and temperature change value Δy would approach to 0, which makes learning process is unstable or appearing oscillation. Figure 3 comes from an instance that the improper learning rate brings oscillation around stable work-point.

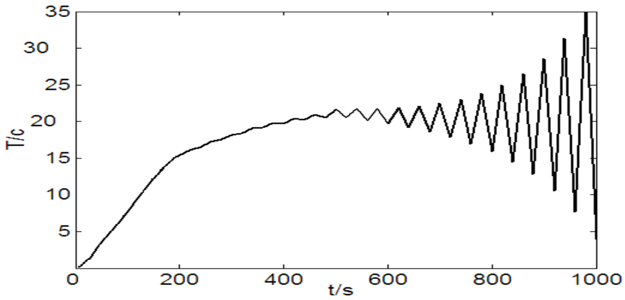

Figure 3. FNNC Brings Oscillation in Learning

So the improved learning algorithm is employed to train FNNC network. The learning process takes two stages. In training, learning rate η can change according to error and error change through proper setup. When error is bigger, error back-propagation method is used to train, and when error is less, the self-adapt learning method is used to train [13]. The self-adapt learning method is given in follow.

When error is less, quick learning speed brings oscillation easily. In order to improve training speed, dynamic adjusting value $\Delta_i^{\,j}(k)$ is employed to modify training rate. This adaptive update-value $\Delta_i^{\,j}(k)$ evolves based on the error energy function E. Here, the training process of ω_i^j is used as an example to introduce the self-adapt algorithm. The study algorithm is:

```
When training weights {
if $(\frac{dE}{d\omega_{ij}}(k-1)*\frac{dE}{d\omega_{ij}}(k)>0)$ then{
$\Delta_{ij}(k)=\min(\Delta_{ij}(k-1)*\eta^{+},\Delta_{\max})$
$\Delta\omega_{ij}(k)=-\frac{dE}{d\omega_{ij}}(k)*\Delta_{ij}(k)$
$\omega_{ij}(k+1)=\omega_{ij}(k)+\Delta\omega_{ij}(k)+\alpha[\omega(k)-\omega(k-1)]$
}
Else if $(\frac{dE}{d\omega_{ij}}(k-1)*\frac{dE}{d\omega_{ij}}(k)<0)$ then{
$\Delta_{ij}(k)=\max(\Delta_{ij}(k-1)*\eta^{-},\Delta_{\min})$
$\Delta\omega_{ij}(k)=-\frac{dE}{d\omega_{ij}}(k)*\Delta_{ij}(k)$
$\omega_{ij}(k+1)=\omega_{ij}(k)+\Delta\omega_{ij}(k)+\alpha[\omega(k)-\omega(k-1)]$
$\frac{dE}{d\omega_{ij}}(k)=0)$
}
Else if $(\frac{dE}{d\omega_{ij}}(k-1)*\frac{dE}{d\omega_{ij}}(k)=0)$ then{
if $(\frac{dE}{d\omega_{ij}}(k)\neq 0)$ then{
$\Delta\omega_{ij}(k)=-\frac{dE}{d\omega_{ij}}(k)*\Delta_{ij}(k)$
$\omega_{ij}(k+1)=\omega_{ij}(k)+\Delta\omega_{ij}(k)+\alpha[\omega(k)-\omega(k-1)]$
}
else $\omega_{ij}(k+1)=\omega_{ij}(k)+\Delta_{ij}(k-1)+\alpha[\omega(k)-\omega(k-1)]$
}
```

In training, the choice of initial value $\Delta_{ij}(0)$ is very important, which affects training process. So $\Delta_{ij}(0)$ is chose properly. The choice of η^{-} and η^{+} is important also. When the change of previous parameters is biggish and cannot confirm the change value using gradient information, the learning parameter chooses η^{-}. The value of η^{+} should be bigger than the value of η^{-}, which make training speed quick and accelerating convergence [14].

In the training process of FNNC, the choice of those initial values of parameters is a very important work. Here, GA algorithm is used to optimize those initial values of fuzzy reasoning networks.

4.3. The Design of Genetic Algorithms

GAs is used to optimize some parameters of FNNC, such as ω_{1i} and ω_{4k}. Combining the character of the control system, a quick learning GAs is put forward to train parameters of FNNC. The improved GAs can avoid the shortcoming of general GAs and give optimized result.

Before training, some GA parameters need to set, such as: popsize, lchrom, crossover probability Pc, mutation probability Pm, iterative number Gen, maxgen, maxruns [15].

The optimizing processes of GAs is given as follows.

1) Encoding: Encoding is the first and an important part of a GAs process, because problem related to information is encoded in a structure called chromosome or string. So the ranges of variation for the different variables are selected through a careful study. This paper adopts decimal code method.

In first training stage, 51 parameters need being trained. Five bits code is adopted here. So each individual is formed by 255 bits gene code of chromosome [16].

2) Setting of initial population: The individuals are selected randomly in a scope.

3) Fitness function: Fitness function indicates individual quality. Evaluation function is a main source to provide the mechanism for evaluating the status of each chromosome and is an important link between GAs and the system. GAs fitness is switched over *J(e)*. Fitness can be defined such as follow formula.

$$J(e)=\frac{1}{2}\sum_{l=1}^{r}(\overline{y_l}-y_l)^2 \tag{17}$$

$$\begin{cases} F(e)=C_{\max}-J(e) & J(e)<C_{\max} \\ F(e)=0 & J(e)\geq C_{\max} \end{cases} \tag{18}$$

Where $C_{\max}$ is 4 times the maximum of *J(e)* index.

In order to conquer the slow training speed problem of general GAs, the aim function is divided into three portions to restrict the contribution of different control error.

$$J1=\frac{1}{2}\sum_{l=1}^{r1}(\overline{y_l}-y_l)^2 \tag{19}$$

$$J2=\frac{1}{2}\sum_{l=r1+1}^{r2}(\overline{y_l}-y_l)^2 \tag{20}$$

$$J3=\sum_{l=r2}^{r}u^2 \tag{21}$$

$$J_{Aim}=J1*a1+J2*a2+J3*a3 \tag{22}$$

Here, $a1$, $a2$ and $a3$ are parameters.

4) Selection operator: Selection operator is the process that high fitness individuals are chosen to replace the inferior individuals as the new population. At first, every individual's fitness and proportions of every individual's fitness are calculated. Then, selecting probability of every individual are decided according to the percent of individual's fitness. Here, simulating anneal GAs is used to extend properly the shares of each individual. The way is given as follow:

$$f_i = \frac{e^{f_i/T}}{\sum_{i=1}^{M} e^{f_i/T}} \tag{23}$$

$$T = T_0(0.99^{g-1}) \tag{24}$$

Here, f_i is the share degree of i individual. M is totality of all individuals. g is the data of the generation. T is temperature and T_0 is beginning temperature.

5) Crossover operations: Crossover operator has the ability of global search and it is the main operator in genetic algorithm. Crossover is achieved in three stages. The first stage is matching. In the second stage, a crossover point is determined in each of the individuals. In the final stage, two parts of the individuals are replaced with each other. Mutation means a random change in the information of a chromosome that does not depend on a reason.

6) Mutation operator: Individual is judged whether meeting requirement. If it is satisfied, those shows that the population convergence and stop calculating. The code parameter of optimum individual is used as optimum parameter. Otherwise, return to *4)* and continue operating, until having individual meeting the precision request [17-19].

5. Simulation Experimentation

The designed control system is used to control storeroom of live working insulated tool. The time constant of storeroom's thermal inertia is assumed as Ty=310. The heat delay of air is assumed asτ=18. In order to illuminate the performance of the designed FNNC system, comparison experiment with traditional PID and fuzzy controller is made. In the following contrast experiment, some typical conditions are used to simulate the control process.

5.1. Simulating to the System in High Temperature

When the environment temperature is higher than the set temperature, the air-conditioning works. The designed fuzzy neural network calculates the control values to control the air conditioning. Here, the out room temperature is 40℃. The controlled room temperature is different from the temperature of outside storeroom and the requested temperature is 25℃. The simulation time is 5000s. The run result is given in Figure 4.

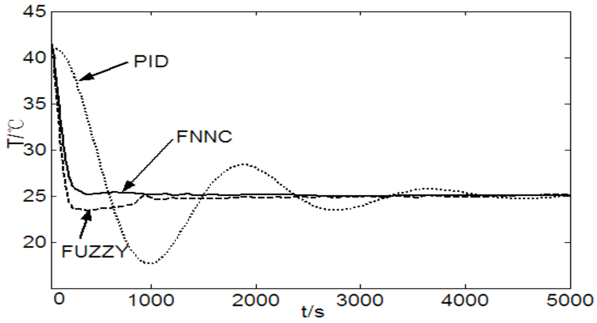

Figure 4. The Running Result of Temperature Change

From above run result, we can see that the FNNC has better control effect than other control methods such as PID control or fuzzy control. FNNC control system has quick regulating speed and small error in steady state because of its quick learning capacity.

5.2. Simulating to the System in Low Temperature

When the environment temperature is lower than the set temperature, the heater works and the neural network outputs the control signal to regulate pulse width to change heater power. In the follow experiment, the regulated room temperature is 25℃ and the outside temperature is 0℃. The experiment is done in the following case that the door is opened in beginning and is shut after 500s. The control result is shown in Figure 5

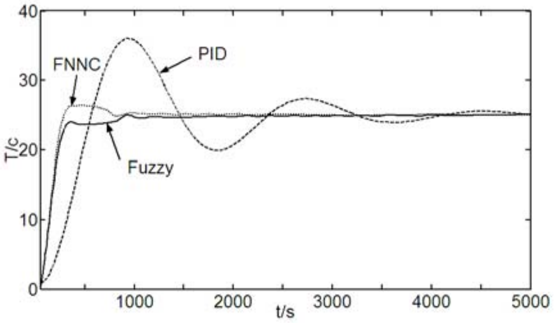

Figure 5. The Controlling Result in Low Temperature

From above figure, FNNC can control heater performs better than other control ways. The control process has small shaking which makes device working steady and has energy saving effect.

5.3. Simulating to the System with Interference

The controlled room has many troubled factors, such as opening door, starting or stopping equipment and temperature change of out room. In the follow experiment, the regulated room temperature is 25□ and the out of room temperature is 35℃. The door is opened in beginning and is shut after 500s. The control result is shown in Figure 6

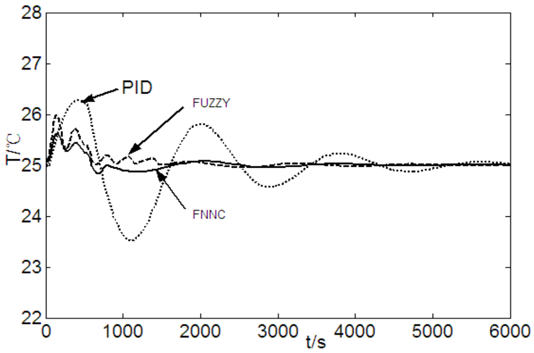

Figure 6. The Controlling Result when Door being Opened

When the room environment is complicated, PID control system is impressible, which has slow regulating speed. FNNC has least dithering and quick regulating speed, which shows FNNC has better control effect than other conventional control strategy.

5.4. Simulating to the System having Big Change

In real control, some room environment is complicated and rooms model changes sometimes from one step to two steps. Figure 7 is an illustration that the controlled system has complicated environment.

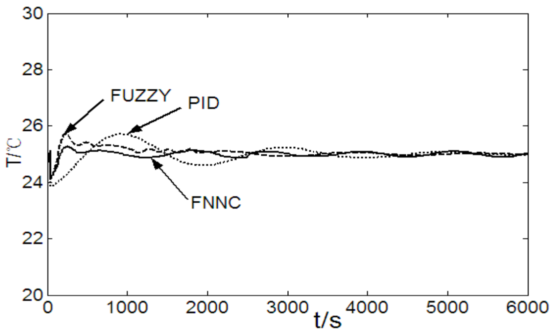

Figure 7. The Controlling Result when Object having Big Change

Above figure shows that PID controller has bad control effect when rooms model changes from one step to two steps and Fuzzy control has bigger error in beginning. FNNC has least error and quick regulating speed, which shows FNNC is not sensitive to room model change.

6. Conclusion

In order to well manage live working tools for power system, it is needed to design dedicated warehouse and to keep the room temperature stable. So the fuzzy neural network is used as introduced according to the specific requirements. Fuzzy reasoning parameters of the controller and reasoning rules can be trained through neural network which solves the difficult problem in traditional fuzzy control. Different controller based on fuzzy neural network is designed according to the specific control circumstances. GA algorithm is used to optimize initial values of the network to improve the training speed and quality. In the training, the training step can change according to the magnitude of the error. So the network has good adaptability. The experimental results show that the designed temperature controller for live working warehouse works well. The research on FNNC arithmetic is an important researching problem in control system and a good control strategy is put forward in the fields of room temperature control.

Acknowledgements

This paper is partially supported by National Natural Science Foundation of China, Grant: No. 61170135 and the Twilight Plan Project of Wuhan City in China (No. 201050231084).

References

[1] Khayyam Humid, Caraway Jamal, J azar, Reza N. Intelligent energy management control of vehicle air conditioning system coupled with engine. *Applied Thermal Engineering.* 2012; 48: 211-224.

[2] Hai-bo LIN. *A Kind of Intelligent Temperature Controller Use PID Algorithm to Realize*. International Conference on Electrical and Control Engineering. ICECE. 2011; 2838-2840.

[3] Li, Zheng. *Intelligent fuzzy immune PID controller design for multivariable process control system*. Proceedings of the World Congress on Intelligent Control and Automation (WCICA). 2010; 4953-4957.

[4] Wang Juan, Song Ai Juan, Zhu Qun Xi. Adaptive PID temperature controller based on fuzzy logic. *Applied Mechanics and Materials*. 2011; 55: 1455-1458.

[5] Duan, Ying-Hong. The design of predictive fuzzy-PID controller in temperature control system of electrical heating furnace. *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2010; 6329 LNCS: 259-265.

[6] Wang Shuqing, Zhang Zipeng, Xue Liqin. Research on the intelligent fuzzy nerve networks controller for air-conditions. *Journal of Computational Information Systems*. 2010; 6(4): 1259-1226.

[7] Vasu Murthy B, Pavan Kumar YV, Ratna Kumari UV. *Fuzzy logic intelligent controlling concepts in industrial furnace temperature process control.* Proceedings of 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies, ICACCCT. 2012; 353-358.

[8] Miao Qiuhua, Jiao Peigang, Liang Jie. Research on fuzzy control system for air-condition in bus. *Advanced Materials Research.* 2012; 457-458: 998-1001.

[9] Pamela D, Jebarajan T. Design of intelligent controller for temperature process. *Communications in Computer and Information Science*. 2012; 350 CCIS: 278-284.

[10] Man-chen XIONG, Ling-long WANG. *Intelligent Fuzzy- PID temperature controller design of drying system*. International Conference on Information Management, Innovation Management and Industrial Engineering. 2012; 3: 54-57.

[11] José Angel Barrios, Miguel Torres-Alvarado, Alberto Cavazos. Neural, fuzzy and Grey-Box modelling for entry temperature prediction in a hot strip mill. 2012; 39(3): 3374-3384.

[12] Soyguder, Servet. *Intelligent system based on wavelet decomposition and neural network for predicting of fan speed for energy saving in HVAC system*. Energy And Buildings. 2011; 43(4): 814-822.

[13] Gao Y, Meng J. Modelling control and stability analysis of nonlinear systems using generalized FNN. *International Journal of Systems Science*. 2003; 34(6): 427-438.

[14] Sung-Kwun Oh, Witold Pedrycz, Byoung-Jun Park. Self-organizing neurofuzzy networks in modeling software data. *Fuzzy Sets and Systems*. 2004; 145(3): 165-181

[15] Rak A, Aliev, Bijan Fazlollahi, Rustam M Vahidov. Genetic algorithm-based learning of fuzzy neural networks. Part1: *Feed-forward fuzzy neural networks, Fuzzy Sets and Systems*. 2001; 118: 351-358.

[16] Sung-Kwun Oh, Witold Pedrycz. Genetic optimization-driven multi-layer hybrid fuzzy neural networks. *Simulation Modeling Practice and Theory*. 2005; 14: 597-613.

[17] Qu Sun, Renhou Li, Ping'an Zhang. Stable and optimal adaptive fuzzy control of complex systems using fuzzy dynamic model. *Fuzzy Sets and Systems*. 2003; 133: 1-17.

[18] Zhou Jianguo, Zhang Fen. Sulfur Dioxide Emission Combination Prediction Model of China Thermal Power Industry. *TELKOMNIKA Indonesia Journal of Electrical Engineering*. 2013; 11(3): 1157-1165.

[19] Xiujuan Dong, Shengtao Li, Nan Jiang, Yuanwei Jing. Non-certainty Equivalent Adaptive Exciting Control of Multi-machine Power Systems. *TELKOMNIKA Indonesia Journal of Electrical Engineering*. 2013; 11(7): 4060-4068.