ADAPTIVE AND INTELLIGENT CONTROLLER USING NEURAL NETWORK

Tu Diep Cong Thanh * and Kyoung Kwan Ahn **

* Mechatronics Department, HCMC University of Technology, Viet Nam
** School of Mechanical and Automotive Engineering, University of Ulsan, Korea

ABSTRACT

Intelligent control techniques have emerged to overcome some deficiencies in conventional control method in dealing with complex real-world systems. These problems include knowledge adaptation, learning and expert knowledge incorporation. In this paper, a newly proposed intelligent controller which includes both neural network controller as compensator and an intelligent switching control algorithm based on learning vector quantization neural network (LVQNN) is used to control of complex dynamic systems. A superb mixture of conventional PID controller and the neural network's powerful capability of learning, adaptive and tackle nonlinearity bring us a good-tracking controller for such a kind of plants with high nonlinearity and hysteresis. In addition, with the greatly changing external environments, a learning vector quantization neural network (LVQNN) is applied as a supervisor of the conventional PID controller, which estimates the external environments and switches to the optimal gain of the PID controller.

Results of simulating on the complex dynamic systems such as pneumatic artificial muscle (PAM) manipulator show that the newly proposed intelligent controller presented in this study can making online control with better dynamic property, strong robustness and suitable for the control of various plants, including linear and nonlinear process and without regard to the severe change of external environments.

1. INTRODUCTION

Staring with linear control techniques, the strategy of PID control has been one of the sophisticated methods and most frequently used in the industry due to its simple architecture, easy tuning, cheap and excellent performance [1][2]. However, the requirement of control precision becomes higher and higher, as well as the plants become more and more complex. Hence, the conventional PID controller with fixed parameters may usually deteriorate the control performance. Various types of modified PID controllers have been developed such as an adaptive/self-tuning PID controller [3], self-tuning predictive PID controller [4], and so on. Though satisfactory performance can be obtained and the proposed controllers above provide better response, command following and greater bandwidth than the conventional PID control method, these controllers are limited because of the limitation of capability of learning algorithm and step by step tuning control parameters without automatically.

More recently neural networks have been used to implement intelligent control systems. It is anticipated that the combination will take advantage of simplicity of PID control and the neural network's powerful capability of learning, adaptability and tackling nonlinearity. There are multitude of PID controllers based on neural networks with various kinds of structures and learning algorithms. The position controller based on PID controller and neural networks was used [5]. Nonlinear PID controller using neural networks to improve dynamic properties of complex system was proposed by Matsukuma and his team [6]. Although these intelligent controllers can control the nonlinear systems with high performance, it is difficult to analyze the control systems and in particular, the external environment problems were assumed to be constant or slowly varying. With greatly changing external environments, an intelligent

PID controller with a neural supervisor had been tried [7]. However, any con troll algorithm introduced up to now was proved that the control performance becomes deteriorated with respect to the abruptly and greatly changing external environments.

To overcome these problems, a newly proposed intelligent controller which includes both neural network controller as compensator and an intelligent switching control algorithm based on learning vector quantization neural network (LVQNN) is used to control of complex dynamic systems without regard to greatly changing external environments.

A superb mixture of conventional PID controller and the neural network's powerful capability of learning, adaptive and tackle nonlinearity bring us a good-tracking controller for such a kind of plants, which are high nonlinearity and hysteresis. In addition, with the greatly changing external environments, a learning vector quantization neural network (LVQNN) is applied as a supervisor of the conventional PID controller, which estimates the external environments and switches to the optimal gain of the PID controller.

Results of simulating on the complex dynamic systems such as pneumatic artificial muscle (PAM) manipulator show that the newly proposed intelligent controller presented in this study can make online control with better dynamic property, strong robustness and suitable for the control of various plants, including linear and nonlinear process and without regard greatly changing external environments.

2. INTELLIGENT CONTROL ALGORITHM

2.1 The overall control system

Figure 1 shows the overall structure of the newly proposed intelligent control algorithm. The proposed algorithm consists of a neural network controller, which is installed in parallel with conventional PID controller and an intelligent switching control algorithm in order to estimate the external environments and switch to the optimal gain of the PID controller.

A conventional PID control algorithm is applied in this paper as the basic controller. The controller output can be expressed in the time domain as:

$$u_f(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(t)dt + K_p T_d \frac{de(t)}{dt} \qquad (1)$$

Taking the Laplace transform of (1) yields:

$$U_f(s) = K_p E(s) + \frac{K_p}{T_i s}E(s) + K_p T_d sE(s) \qquad (2)$$

The resulting PID controller transfer function of:

$$\frac{U_f(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \qquad (3)$$

A typical real-time implementation at sampling sequence k can be expressed as:

$$u_f(k) = K_p e(k) + u(k-1) + \frac{K_p T}{T_i}e(k) + K_p T_d \frac{e(k)-e(k-1)}{T} \qquad (4)$$

$$e(k) = y(k) - x(k) \qquad (5)$$

where $u_f(k)$, $e(k)$, $y(k)$ and $x(k)$ are the output of conventional PID controller, the error between the desired set point and the output, the desired set point and the output, respectively.

From Fig. 1, the control input to plant can be computed as follow:

$$u(k) = u_f(k) + u_N(k) \qquad (6)$$

where $u_N(k)$ is the modify-output of neural network controller.

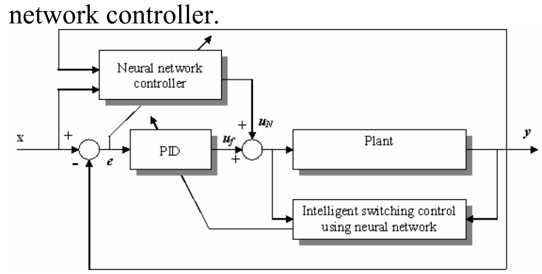

Fig. 1 Structure of the newly proposed intelligent control algorithm

2.2 Neural network controller

In order to overcome the limitation of the conventional PID controller and improve its property, a neural network controller is installed in parallel with conventional PID controller as compensator. Neural network controller can represent any nonlinear function, and has self-learning and parallel processing abilities as well as strong robustness and fault-tolerance, so it fits for online adaptive control with PID controller.

Conventional PID controller contributes to ensuring the stability of the system at the beginning of learning and neural networks controller adds the adaptability for variations of operational conditions. With the progress of learning, the output from linear controller decreases and the neural networks controller becomes to dominate the overall control system. The control error $e(k)$ is used as a teaching signal to be minimized.

2.2.1 Structure of neural network controller

Figure 2 shows the structure of neural network controller. The input layer has seven neurons including a neuron with output of –1 to set the bias value of each neuron in hidden layer. There are fourteen neurons including a neuron with –1 in hidden layer. All layers are connected in only the forward direction. The input to each neuron is given as the weighted sum of outputs from the previous layer. The output of each neuron is generated by linear function in the input layer, in hidden and output layers the sigmoid function is used.

$$f_{sigmoid}(x) = \frac{1}{1+e^{-x}} \tag{7}$$

2.2.2 Leaning algorithm

In Fig. 2, the following symbols are defined:

i_j^I : Input to the j[th] neuron in the input layer

o_j^I :Output from the j[th] neuron in the input layer

i_k^H : Input to the k[th] neuron in the hidden layer

o_k^H :Output of the k[th] neuron in the hidden layer

i^O : Input to the output layer

o^O : Output from the output layer

ω_{jk}^{IH} : Weight from the jth neuron in the input layer to the kth neuron in the hidden layer

ω_k^{HO} : Weight from the kth neuron in the hidden layer to the output layer

The modify-output of neural network controller can be expressed as following equation

$$u_N = K_n\left(o^O - 0.5\right) \tag{8}$$

K_n : Proportional gain of the output of neural network controller

The operation of each neuron is described as:

$$o_j^I = i_j^I \tag{9}$$

$$o_k^H = f_{sigmoid}(i_k^H), \quad i_k^H = \sum_j \omega_{jk}^{IH} o_j^I \tag{10}$$

$$o^O = f_{sigmoid}(i^O), \quad i^O = \sum_k \omega_k^{HO} o_k^H \tag{11}$$

The leaning process is based on the back propagation algorithm, which minimizes E given by:

$$E = \frac{1}{2}(x-y)^2 = \frac{1}{2}e^2 \tag{12}$$

The weights are updated by the following increments to minimize E:

$$\Delta\omega_{jk}^{IH} = -\eta \times \frac{\partial E}{\partial \omega_{jk}^{IH}} \tag{13}$$

$$\Delta\omega_k^{HO} = -\eta \times \frac{\partial E}{\partial \omega_k^{HO}} \tag{14}$$

where $\eta > 0$ is learning rate to determine the speed of leaning.

$\frac{\partial E}{\partial \omega_k^{HO}}$ in Eq. (14) can be calculated by:

$$\frac{\partial E}{\partial \omega_k^{HO}} = \frac{\partial E}{\partial i^O}\frac{\partial i^O}{\partial \omega_k^{HO}} \tag{15}$$

$$\frac{\partial i^O}{\partial \omega_k^{HO}} = \frac{\partial}{\partial \omega_k^{HO}}\left(\sum_k \omega_k^{HO} o_k^H\right) = o_k^H \tag{16}$$

$$\frac{\partial E}{\partial i^O} = -\delta^O \tag{17}$$

$$\frac{\partial E}{\partial \omega_k^{HO}} = -\delta^O \times o_k^H \tag{18}$$

δ^O is called a generalized error calculated by:

$$\delta^O = -\frac{\partial E}{\partial y}\frac{\partial y}{\partial o^O}\frac{\partial o^O}{\partial i^O} \tag{19}$$

$$\frac{\partial E}{\partial y} = \frac{\partial}{\partial y}\left(\frac{1}{2}(x-y)^2\right) = -e \tag{20}$$

$$\frac{\partial o^O}{\partial i^O} = \frac{\partial f_{sigmoid}(i^O)}{\partial i^O} = f'_{sigmoid}(i^O) \tag{21}$$

The dynamic of the controlled plant is not considered to calculate $\frac{\partial y}{\partial o^O}$ assumed to be constant.

$$\frac{\partial y}{\partial o^O} = C = const \tag{22}$$

The increment of weight can be written as:

$$\Delta\omega_k^{HO} = -\eta \times \frac{\partial E}{\partial \omega_k^{HO}} = \eta \times \delta^O \times o_k^H \tag{23}$$

Consequently, the weight is updated by:

$$\omega_k^{HO} = \omega_k^{HO} + \eta \times \delta^O \times o_k^H = \omega_k^{HO} + \eta \times e \times C \times f_{sigmoid}^{'}(i^O) \times o_k^H \quad (24)$$

The update equation, Eq. (25) of the weight ω_{jk}^{IH} can be derived in the same manner.

$$\omega_{jk}^{IH} = \omega_{jk}^{IH} + \eta \times \delta_k^H \times o_j^I \quad (25)$$

where,

$$\delta_k^H = -\frac{\partial E}{\partial y}\frac{\partial y}{\partial o^O}\frac{\partial o^O}{\partial i^O}\frac{\partial i^O}{\partial o_k^H}\frac{\partial o_k^H}{\partial i_k^H} = e \times C \times f_{sigmoid}^{'}(i^O) \times \omega_k^{HO} \times f_{sigmoid}^{'}(i_k^H) \quad (26)$$

With the learning of the neural network and the decreasing of the error, the neural networks works more and more effective until it completely compensates the deficiency of the conventional PID controller. The structure and the learning algorithm of the network are relative simple and the physical meaning of the input and outputs is clear. The effectiveness of the proposed controller is investigated through the simulation of the complex dynamic systems such as PAM manipulator.

2.3 An intelligent switching control algorithm

Problems with control the complex dynamic systems without regard greatly changing external environments is briefly discussed in this section. The variation external environments must be recognized for an intelligent control of the complex dynamic systems. Here, the learning vector quantization neural network (LVQNN) is proposed as a supervisor of the intelligent switching control algorithm.

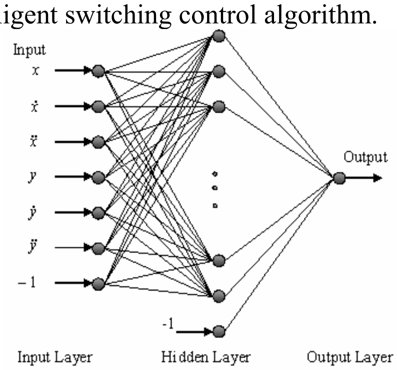

Fig. 2 Structure of neural network controller

2.3.1 Structure of the neural classifier

According to the learning process, neural networks are divided into two kinds: supervised and unsupervised. The difference between them lies in how the networks are trained to recognize and categorize objects. The LVQNN is a supervised learning algorithm, which was developed by Kohonen and is based on the self-organizing map (SOM) or Kohonen feature map. The LVQNN methods are simple and effective adaptive learning techniques. They rely on the nearest neighbor classification model and are strongly related to condensing methods, where only a reduced number of prototypes are kept from a whole set of samples. This condensed set of prototypes is then used to classify unknown samples using the nearest neighbor rule. The LVQNN has a competitive and linear layer in the first and second layer, respectively. The competitive layer learns to classify the input vectors and the linear layer transforms the competitive layer's classes into the target classes defined by the user. Figure 3 shows the architecture of the LVQNN, where P, y, W1, W2, R, S1, S2, and T denote input vector, output vector, weight of the competitive layer, weight of the linear layer, number of neurons of the input layer, competitive layer, linear and target layer, respectively. In the learning process, the weights of the LVQNN are updated by the following Kohonen learning rule if the input vector belongs to the same category.

$$\Delta W_1(i,j) = \lambda a_1(i)(p(j) - W_1(i,j)) \quad (27)$$

If the input vector belongs to a different category, the weights of the LVQNN are updated by the following rule:

$$\Delta W_1(i,j) = -\lambda a_1(i)(p(j) - W_1(i,j)) \quad (28)$$

where λ is the learning ratio and $a_1(i)$ is the output of the competitive layer.

2.3.2 Data generation for the training of the LVQNN

In the design of the LVQNN, it was very important to identify what input to select and how many sequences of data to use. Generally the training result was better according to the increase of the number of input vectors, but it took more calculation time and the starting time of the recognition of inertia load was later. In our simulation, in order to recognize the variation external environments, the control input and system response are utilized to input vectors as shown in Fig. 4. The output of the LVQNN is an integer value, which is represented for the recognized-class. In our research works, 3 kinds of the environments are used, which are variation from high stiffness to low stiffness, and are called environment 1, environment 2 and environment 3, respectively. With respect to each environment, the outputs of the LVQNN are also called class 1, class 2, and class 3, respectively.

To obtain the learning data for the LVQNN, a series of experiments were conducted under 3 different external environments. With each environment, it just only has one PID controller, which is suitable to. That means there are 3 controllers (PID Controller 1, PID Controller 2 and PID Controller 3), which are suitable with 3 kinds of the environments one by one. And then, the generation of training data is shown in Fig. 5 and 6, which correspond to the control input to the system, and system response, respectively. To obtain the generation of training data, the control parameters of the PID controllers are obtained through trial-and-error, which are shown in Table 1. From Table 1, it was understood that the proportional, integral and derivative control gains were increasing in accordance with a decrease in the stiffness of the external environments.

2.3.3 Training process of the LVQNN

The learning vector quantization neural network (LVQNN) is a method for training competitive layers in a supervised manner. A competitive layer will automatically learn to classify input vectors. However, the classes that the competitive layer finds are dependent only on the distance between input vectors. If two input vectors are very similar, the competitive layer probably will put them into the same class. Thus, the LVQNN can classify any set of input vectors, not just linearly separable sets of input vectors. The only requirement is that the competitive layer must have enough neurons, and each class must be assigned enough competitive neurons.

A total of 9 simulation cases were carried out to prepare for the generation of training data for the LVQNN. In the training stage of LVQNN, the number of input vectors were adjusted from 4 to 22 with 10 steps and the number of neurons in the competitive layer were adjusted from 10 to 28 with 10 steps, as shown in Table 2, in order to obtain the optimal weight of the LVQNN. To investigate the classification ability of the LVQNN, the same input vectors, which were used in the learning stage, were re-entered into the LVQNN and the learning success rate was calculated. Here, the learning success rate defines the percentage of success of the LVQNN learning, where success means that the output of the LVQNN was equal to the target class with respect to the same input vectors.

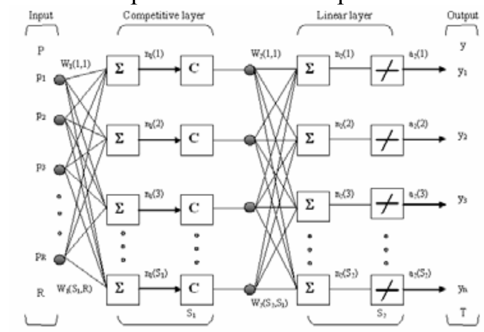

Fig. 3 Structure of the LVQNN

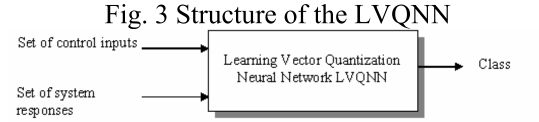

Fig. 4 Learning data for the LVQNN

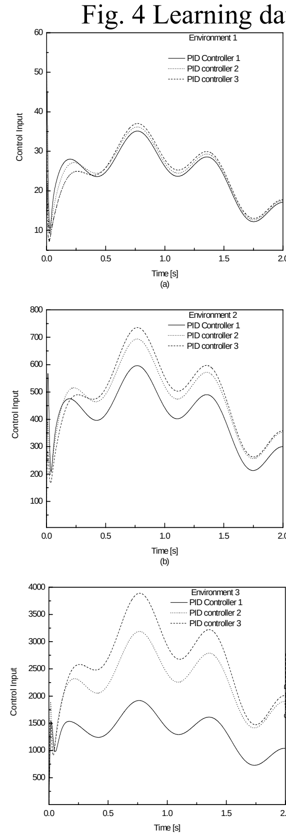

Fig. 5 Simulation results for learning data generation of control input

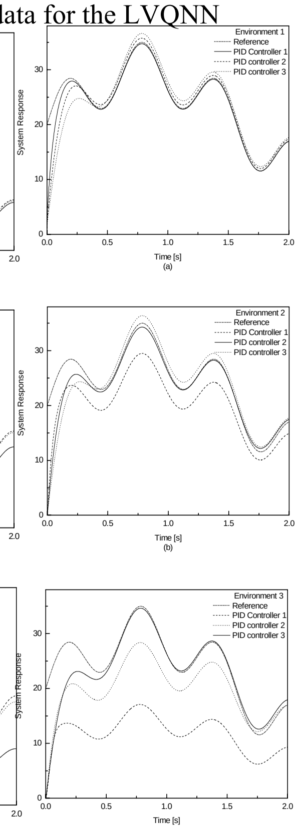

Fig. 6 Simulation results for learning data generation of system response

target classes by using a competitive layer and the classes that the competitive layer found were dependent only on the distance between input vectors, a high learning success rate was realized when the input vectors were distributed widely.

From Fig. 7, it was also understood that the optimal number of input vectors and neurons of the competitive layer were chosen to be 14 and 20, respectively and the maximum training success rate was 97[%], which was enough for recognition of the external environments.

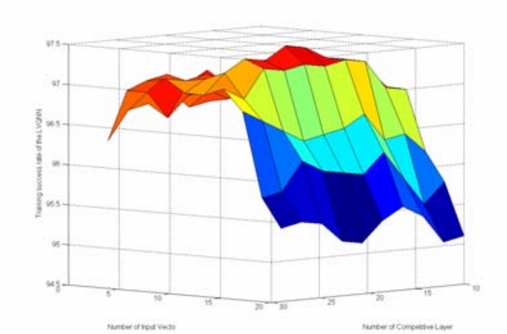

Fig. 7 Training success rate of the LVQNN

2.4 Proposition of the smooth switching algorithm

If the external environment was different from the previous training condition, the output of the LVQNN may have belonged to the mixed classes with different ratios in each case (i.e. if the external environment between environment 1 and environment 2, it may have belonged to 1 or 2 class). Therefore the following switching algorithm was proposed to apply to the abrupt change of class recognition result. The switching algorithm is described by the following equation:

$$class(k) = \alpha \times class(k-1) + (1-\alpha) \times class(k) \tag{29}$$

where k is the discrete sequence, α is the forgetting factor and $class(k)$ is the output of the LVQNN at the k time sequence.

3. SIMULATION RESULTS

To investigate the newly proposed intelligent control algorithm, the simulation on the complex dynamic systems such as pneumatic artificial muscle manipulator is carried out. As a novel actuator, which has been regarded during the decades as an interesting alternative to hydraulic and electronic actuators, the PAM actuator has been applied to many industrial applications as well as researching on modeling and control.

Among previous works, as done by Osuka and his team [8], the nominal plant model of PAM manipulator was obtained as follow:

$$G(s) = \frac{889.27}{s^2 + 23.374 + 889.27} \tag{30}$$

In my study, 3 kinds of environments with variation stiffness were assumed as below:

$$G_k(s) = \frac{889.27 \times k}{s^2 + 23.374 + 889.27} \tag{31}$$

where $k=1$, $k=0.1$ and $k=0.01$ with respect to high stiffness, normal stiffness and low stiffness, respectively.

Firstly, the effectiveness of newly proposed intelligent control algorithm is demonstrated through simulation with respect to high stiffness environment. In simulation, the proportional gain of output of neural network controller, K_n, and learning rate of neural network controller, η, are set to be 1100 and 0.01, respectively. These control parameters are obtained through trial-and-error. The initial values of weights from the input layer to the hidden layer, ω_{jk}^{IH}, and that of weights from the hidden layer to the output layer, ω_k^{HO}, of the neural network are given by random numbers from –0.1 to 0.1. As also, the control parameters of PID controller 1 are used in this case. Figure 8 shows the comparison between conventional PID controller 1 and the proposed controller in case considering the effectiveness of neural network controller as compensator. That means, in this case, the effectiveness of an intelligent switching control algorithm is not applied yet. From Fig. 8, it is clear that the complex dynamics, high nonlinearity and hysteresis have been handled. The system response with the proposed control algorithm is very agreement with the desired set point. In addition, it is obvious that the proposed controller plays the main role at the beginning of the control process. After the neural network controller is consistently trained through error, it gradually compensates the deficiency of the conventional PID controller. This is a controlling and learning process with the ability of adapting the changing of the complex dynamic systems such as PAM manipulator.

Figure 9 shows the simulation results of system response with variation external environments (k=1, k=0.1 and k=0.01), where the PID control gains were fixed and the same as that of the high stiffness environment. From Fig. 9, it was understood that the system response became worse according to the decrease of the stiffness and it was requested that the control parameters of PID controller be adjusted according to the change of the external environments.

Next, simulations were carried out to verify the effectiveness of the proposed intelligent control algorithm. In this case, the proportional gain of output of neural network controller, K_n, with respect to 3 kinds of the external environments from high stiffness to low stiffness are set to be 1100, 7500, and 45000, respectively. As also, proposition of the smooth switching algorithm is applied in this situation. And the forgetting factor, α, is set to be 0.6. These control parameters are obtained through trial-and-error. In order to demonstrate the effectiveness of the newly proposed intelligent control algorithm, the initial control parameters of PID controller 1 are used. That means the control parameters of PID controller 1 are set for all simulation without regard the external environments. After a few milliseconds, when the data is enough for recognizing the external environment by the LVQNN, the control parameters of PID will be auto-tuning by proposition smooth switching algorithm and the result from recognition class of the LVQNN. The simulation results are shown in Fig. 10, 11 and 12, which correspond to the high stiffness environment, normal stiffness environment, and low stiffness environment, respectively. In these figures, we show system response, control input, output of neural network controller and output of the LVQNN in case proposition smooth switching algorithm is applied, respectively. The number of the input vector was 14, which included 7 control inputs and 7 system response outputs. From these simulation results, particularly in the output of the LVQNN, it was verified that the external inertial load was almost exactly recognized to the correct class and an accurate control performance was obtained without regard the greatly changing external environments.

The simulation results, which the external environment is between class 2 and class 3 (k=0.004), are shown in Fig.13. In this case, the proportional gain of output of neural network controller is $K_n = 1000$. From Fig. 13, the class number calculated from the output of the LVQNN was between 2 and 3, which proved that the external environment was between k=0.1 and k=0.01. In Fig. 14, 15 and 16, simulations were conducted to compare the system response with respect to 3 different external environments (k=0.1, k=0.01 and k=0.04) with and without the proposed intelligent control algorithm using neural networks. As also, in these figures, the comparison between proposed controller and the conventional PID controller with respect to correctly of that environment. From the simulation results, it was found that the system response became worse according to decrease in the stiffness of external environment without auto-tuning adaptively control parameters of PID controller. On the contrary, the system response was almost the same in any case by using the newly proposed intelligent control algorithm. To compare with the conventional PID controller with respect to correctly of that environment, it was also verified that the proposed method was very effective in the accurate control of the PAM manipulator.

4. CONCLUSION

In this study, the newly proposed intelligent control algorithm using neural network are given. It is strongly recommended that the proposed control algorithm is very effective in both handling the high nonlinearity, hysteresis and without regard the greatly changing external environments.

The newly intelligent controller presented can making online control with better dynamic property, strong robustness and suitable for the control of various kinds of complex dynamic systems.

A more essential factor is that the proposed controller is easy applied to both accurate position control and force-control of various plants, including linear and nonlinear process and without regard greatly changing external environments.

Table 1 Optimal parameters of the PID controller

Environments	Kp	Ki	Kd
1	100	10	5
2	300	200	30
3	1200	1000	200

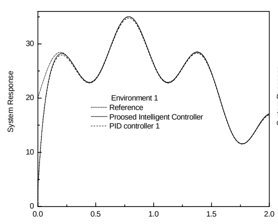

Fig. 8 Comparison of the simulation results with and without neural network controller

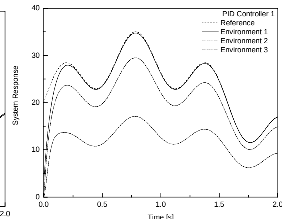

Fig. 9 Simulation results of system response with variation external environments

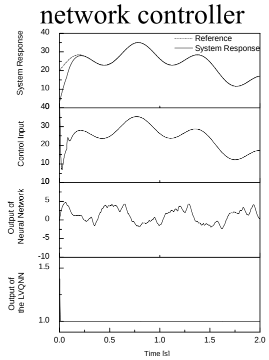

Fig. 10 Simulation results with respect to external environment 1

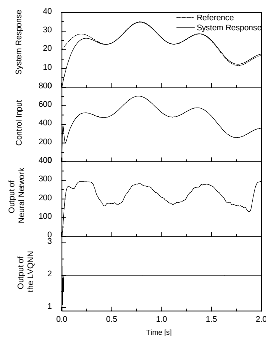

Fig. 11 Simulation results with respect to external environment 2

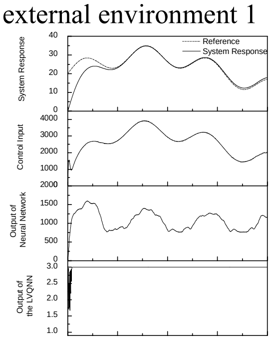

Fig. 12 Simulation results with respect to external environment 3

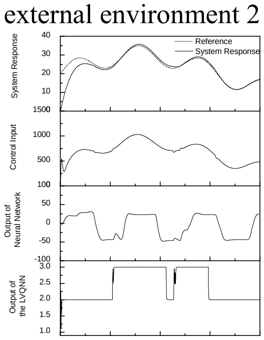

Fig. 13 Simulation results with respect to external environment between 2 and 3

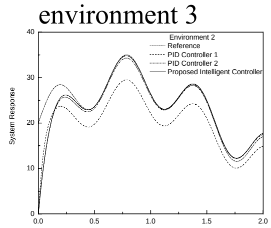

Fig. 14 Comparison of the simulation results with and without proposed intelligent controller with respect to environment 2

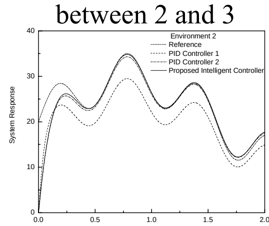

Fig. 15 Comparison of the simulation results with and without proposed intelligent controller with respect to environment 3

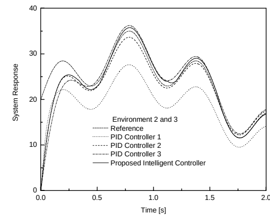

Fig. 16 Comparison of the simulation results with and without proposed intelligent controller with respect to environment between 2 and 3

REFERENCES

1. Bennett, S., "Development of the PID controller," in IEEE, Control Systems Magazine, Vol. 13 (1993), pp. 58~62.
2. Hamdan, M. and Zhiqiang Gao,"A novel PID controller for pneumatic proportional valves with hysteresis," in IEEE Int., Conf., Industry Applications, Vol. 2 (2000), pp. 1198~1201.
3. Grassi, E., Tsakalis, K.S., Dash, S., Gaikwad, S.V., and Stein, G., "Adaptive/self-tuning PID control by frequency loop-shaping," in Proc., IEEE Int., Conf., Decision and Control, Vol. 2 (2000), pp. 1099~1101.
4.Vega, P., Prada, C., Aleixander, V., "Self-tuning predictive PID controller," in IEE Proc., Vol. 3 (1991), pp. 303~311.
5. Choi, G.S., Lee, H.K., and Choi, G.H., 1998, "A study on tracking position control of pneumatic actuators using neural netw," in Proc., IEEE Int., Conf., Industrial Electronics Society, Vol. 3 (1998), pp. 1749~1753.
6. Matsukuma, T., Fujiwara, A., Namba, M., and Ishida, Y., "Non-linear PID controller using neural networks," Int., Conf., Neural Networks, Vol. 2 (1997) , pp. 811~814.
7. Oki, T., Yamamoto, T., Kaneda, N., and Omatsu, S., 1997, "An intelligent PID controller with a neural supervisor," in IEEE Int., Conf., Systems, Man, and Cybernetics, Vol. 5 (1997), pp. 4477~4482.
8. Osuka, K., Kimura, T., Ono, T., "H∞ control of a certain nonlinear actuator," in Proc., IEEE Int., Conf., Decision and Control, Honolulu, Hawai, Vol. 1 (1990), pp. 370~371.