ISSN 1009 - 3095 *Journal of Zhejiang University(SCIENCE)* V.1, No.1, P.61 - 65, Jan. - Mar., 2000
http://www.zju.edu.cn; http://www.chinainfo.gov.cn/periodical

A REAL-TIME ADAPTIVE CONTROL ALGORITHM USING NEURAL NETS WITH PERTURBATION*

YANG Jian-gang(杨建刚)[1], WANG Kai(王凯)[1]
YANG Hua-yong(杨华勇)[2], ZHANG Jian-min(张健民)[2]

([1] *Insitiute of Artificial Intelligence, Dept. of Computer Science,* [2] *The State Key Laboratory of Fluid Power Transmission and Control, Yuquan Campus of Zhejiang University, Hangzhou, 310027, China*)

Received Dec. 12, 1998; revision accepted Apr. 15, 1999

Abstract: This paper proposes an adaptive algorithm of neural nets with a special perturbation for a real time velocity control system of a VVVF(Variable Voltage Variable Frequency) hydraulic elevator. The weight vector of the neural network is adaptively adjusted by the LMS (Least Mean Square) with perturbation, so it is not necessary to know the nonlinear continuous function of the control system. The nonlinear velocity control system is considered as the controller output function in an adaptive controller model. The experimental results obtained from the VVVF hydraulic elevator showed that the neural nets controller using the perturbation algorithm proposed are much stabler and faster in dynamic response compared with the conventional PID (Proportion-Integration-Derivation) controller.

Key words: neural nets, real-time control, VVVF hydraulic elevator
Document code: A **CLC number:** TP183

INTRODUCTION

After much research and some successful applications of Artificial Neural Network (ANN) to control systems, ANN has become one of the important tools to show the intelligence of the human brain. Since ANN can be used to describe any nonlinear continuous function, it has drawn much attention as a controller of nonlinear systems. ANN control is somehow similar to the adaptive control, which is however usually used for linear systems. The multi-layer neural network is very sluggish in the rate of learning so it can not meet the requirements of real-time control. Thus the neural network used for real-time application is often simple in its topology. For instance, the Cerebella Model Articulation Controller (CMAC) proposed by Albus(1975) was introduced to control the real time tracking of robot orbit with LMS algorithm by Miller et al. (1990). The CMAC was modified to achieve better performance in a real-time control system (Yang, 1993).

ANN is not suitable for use alone as a controller because of its features. But with its abilities of learning and nonlinear mapping, it is suitable for obtaining the control rule or some functions of the control system. The control model proposed in this paper uses a combination of the nonlinear algorithm similar to Adaptive Linear Neurons (Adaline) and PID. The real time learning algorithm is very important when ANN (except for its framework) is used in a controller. LMS is one of the most important learning rules in Perceptron and Adaline as an optimal algorithm. The conventional LMS algorithm is based on linear principles, while the LMS algorithm discussed in this paper is extended to nonlinear applications. When Adaline is used as a classifier, the convergent effect of LMS is not as good as expected. When used in a controller, the LMS algorithm, after suitable modification shows its special characteristics of simplicity in calculation and adaptation in control.

The hydraulic elevator, as a controlled plant widely used in low and medium height buildings, is a typical nonlinear and time-varying system that can work under variable load situations. The velocity of the modern hydraulic elevators is commonly controlled by the conventional closed-loop PID. Time delay, overshoot and vibration during the elevator operation, especially when

* Project (69775013) Supported by NSFC

the elevator starts or levels speed, severely affect the riding comfort of hydraulic elevators. The control algorithm proposed in this paper has been applied to control the velocity of most advanced VVVF hydraulic elevators. The experimental results are compared with those of the conventional PID controller.

ALGORITHM SCHEME

In the model of adaptive nonlinear neural unit as shown in Fig. 1 (modified from Adaline), $X_k = (x_{1k}, x_{2k}, \cdots, x_{nk})$ is the input vector, and $W_k = (w_{1k}, w_{2k}, \cdots, w_{nk})$ is the weight vector. The output of neural nets s_k is the inner product of X_k and W_k, i.e.

$$s_k = X_k W_k^T \tag{1}$$

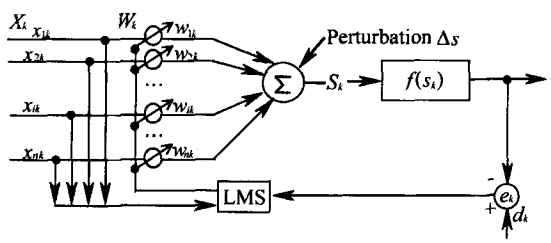

Fig. 1 Model of nonlinear Adaline

The output of the system is $y_k = f(s_k)$, and $f(\cdot)$ is a nonlinear continuous function. If $f(\cdot)$ is a threshold function, the model shown in Fig. 1 degenerates back to a standard Adaline.

1. Perturbation algorithm

The basic of LMS is based upon the gradient calculation of the Mean Square Error (MSE) to obtain accurate estimation of the weight vector W^*, so we have

$$W_{k+1} = W_k + \mu(-\nabla_k) \tag{2}$$

Where μ is the learning rate. The measurement of error is defined as:

$$\tilde{e}_k \triangleq d_k - y_k \tag{3}$$

Where d_k is the desired output, y_k is the actual system output, i.e., $y_k = f(s_k)$.

The instantaneous gradient to kth input vector X_k is:

$$\hat{\nabla}_k = \frac{\partial \tilde{e}_k^2}{\partial W_k} = 2\tilde{e}_k \frac{\partial \tilde{e}_k^2}{\partial W_k}$$

$$= -2\tilde{e}_k f'(s_k) X_k \tag{4}$$

Eq. (4) is an adaptive learning rule based upon a gradient algorithm. If $f(\cdot)$ is selected as a nonlinear Sigmoid function, this algorithm is called the LMS-BP rule. The aim here is to obtain the gradient estimation of the weight vector of the unknown $f(\cdot)$ function, or the perturbation algorithm.

The principle of the algorithm is as follows: if the derivative of function $f(\cdot)$ in the Eq. (4) can be replaced by the system output and the error, function $f(\cdot)$, can be taken as the plant controlled. Hence the plant can be adaptively controlled with the LMS.

Eq. (4) can be rewritten as

$$\hat{\nabla}_k = \frac{\partial \tilde{e}_k^2}{\partial W_k} = \frac{\partial \tilde{e}_k^2}{\partial s_k} \cdot \frac{\partial s_k}{\partial W_k}$$

$$= \frac{\partial \tilde{e}_k^2}{\partial s_k} \cdot X_k = 2\tilde{e}_k \frac{\partial \tilde{e}_k}{\partial s_k} \cdot X_k \tag{5}$$

A small perturbation signal Δs_k is added to the output of neural network s_k as shown in Fig. 1, so

$$\frac{\partial \tilde{e}_k}{\partial s_k} \approx \frac{\Delta \tilde{e}_k}{\Delta s_k} \tag{6}$$

Considering the effect of perturbation (Δs), we have

$$\hat{\nabla}_k = 2\tilde{e}_k \frac{\partial \tilde{e}_k}{\partial s_k} \cdot X_k = 2\tilde{e}_k \left(\frac{\Delta \tilde{e}_k}{\Delta s_k}\right) X_k \tag{7}$$

Combining Equations (2) and (7), we have

$$W_{k+1} = W_k - 2\mu\tilde{e}_k \left(\frac{\Delta \tilde{e}_k}{\Delta s_k}\right) X_k \tag{8}$$

Since $\Delta\tilde{e} = -\Delta y_k$

Eq. (8) can be rewritten as:

$$W_{k+1} = W_k + 2\mu\tilde{e}_k \left(\frac{\Delta y_k}{\Delta s_k}\right) X_k \tag{9}$$

In Eq. (8) or (9), there is no need to know the derivative of the function $f(\cdot)$. This is a great advantage for the real time control.

2. Transformation of the perturbation

The model shown in Fig. 1 is a single layer of neural nets. The perturbation signal (Δs_k) should be added according to the updating of the weight vector in the iterative processing. The perturbation signal (Δs_k) can be obtained from the change of the network input vector.

Assume the network has n neural units, and the input vector $X_k = (x_{1k}, x_{2k}, \cdots, x_{nk})$ is a unit matrix. Let the unit i be equal to zero before updating the weights in every step, or only one unit is cleared with zero, i.e. let $x_{ik} = 0$. The input vector being perturbed is $X_k^* = (x_{1k}, x_{2k}, \cdots, x_{ik} = 0, \cdots, x_{nk})$. The perturbation unit is indicated by the pointer ii which equals the mode of k divided by n, k is the kth discrete time instant. The algorithm is described as follows

```
.....
k = k + 1
ii = MOD(k, n)
i = ii
X[i] = 0
.....
```

The above algorithm is called a perturbation planner, which adds perturbation to every unit of the network one by one according to the period of the system control. The perturbation period of every neural unit is n, and the network output is

$$s_{k+1} = X_k^* W_k^T + X_k \Delta W_k \qquad (10)$$

EXPERIMENTS AND DISCUSSION

1. Scheme of VVVF hydraulic elevator

A series of experiments were carried out on the up speed control of a VVVF hydraulic elevator controlled by a hydrostatic system, or a variable flow hydraulic system, which just change the shaft speed of a fixed displacement pump by varying the voltage and frequency of the electric motor. Fig.2 shows the VVVF hydraulic elevator system consisting of hydraulic cylinder, pump, cabin velocity transducer, the neural controller proposed, etc. On receiving the selected floor command, the neural controller as shown in Fig.1 transmits the signal of the ideal elevator velocity to the VVVF converter to regulate the AC voltage and frequency of the motor driving the hydraulic pump providing the pressurized output flow passing through the check valve to the cylinder connected with the cabin. The velocity of the cabin measured by an optical encoder is fed back to the neural controller. The comparison between the ideal and measured speeds of the car is made continuously, the controller corrects the errors between these two speeds and sends the car running according to the ideal speed desired.

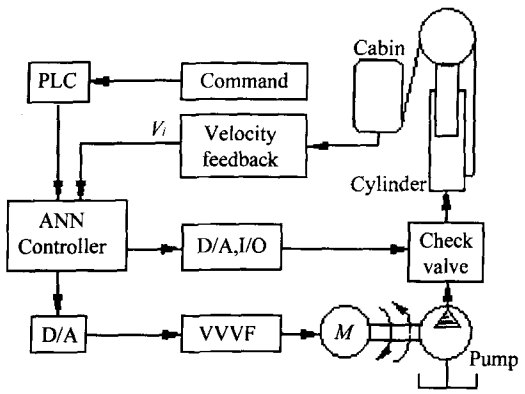

Fig.2 Schematic diagram of hydraulic elevator

2. Real-time adaptive controller using neural nets

A real-time adaptive controller using neural nets was developed to control the up speed of the VVVF hydraulic elevator (Fig. 3).

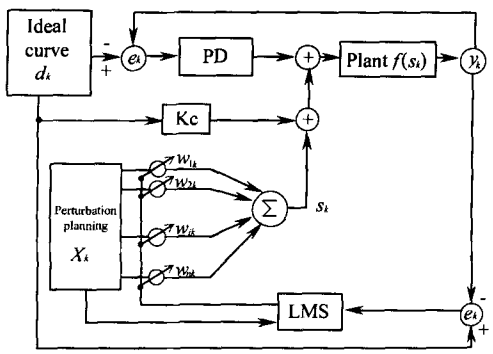

Fig.3 Schematic diagram of real time controller using neural nets

The controller proposed consists of a PD control link which enlarges the two closed feedback error loops of the output. When an external disturbance exists in the hydraulic system, if the number of neural units is not big enough, a static error in the output will be generated by the control algorithm, this could be overcome by increasing compensation to the static error, Kc.

3. Simulation

The approximate model for the up speed of the elevator is

$$Y(s) = \frac{1}{(0.3s+1)(s^2+2\times 19\times 0.06s+19^2)} \quad (11)$$

This model's simulation results using the controller shown in Fig. 3 are shown in Fig. 4, 5 and 6. In Fig. 4, curve 1 is the step response of a PID controller, and curve 2 is that of the neural controller with perturbation and 30 neural units. It can be seen that the dynamic performance of the proposed controller was much better than that of the PID controller.

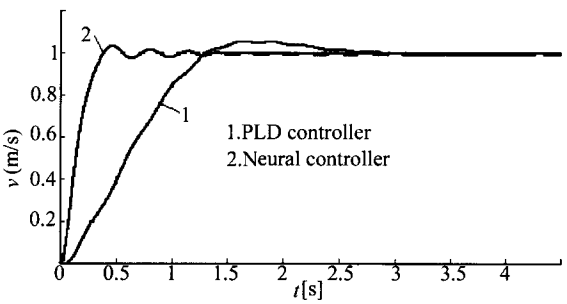

Fig.4 Step responses of PID and the perturbation algorithm

In order to investigate the effects on the control performance due to variation of the learning rate, μ, in the perturbation algorithm, a group of simulation results corresponding to different steps were obtained as shown in Fig. 5, with $\mu = 0.2$ for curve 1, $\mu = 0.15$ for curve 2, $\mu = 0.1$ for curve 3, $\mu = 0.05$ for curve 4, and with proportional coefficient $K_p = 7$ and the differential constant $T_i = 0.03$ for PD link parameters.

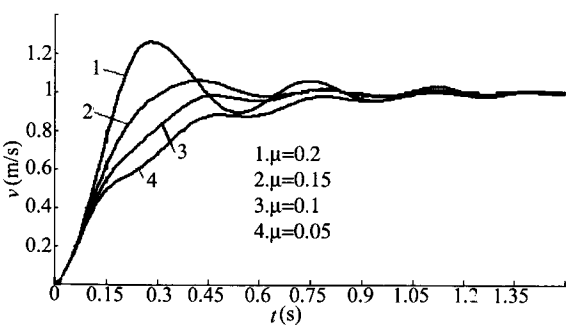

Fig.5 The effect of learning rate on control performance

The effect of the number of neural units on control performance is shown in Fig. 6, in which the learning rate is set at $\mu = 0.08$, and curve 1, 2, 3 and 4 corresponding to $n = 35$, $n = 30$, $n = 25$ and $n = 20$ respectively, while the PD link parameters remain the same, i.e. $K_p = 7$ and $T_i = 0.03$.

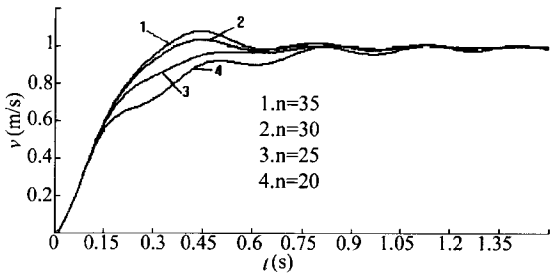

Fig.6 The effect of the number of neural units on control performance

The above simulation results show that the number of neural units and the learning rate affected the control performance; and that the learning rate μ is approximately inversely proportional to the number of neural units n as shown in Fig. 7. When the number of neural units was small, the step response of the network was slow, whereas when the number of neural units was large, the calculated time increased linearly. This is a disadvantage for real-time control. If the change of the units number is very small, the control performance will not be very much affected. In the experiments, the learning rate was selected to be between 0.5 and 0.05, and the units number to be between 15 to 30.

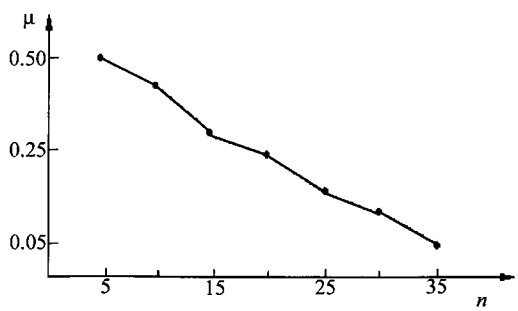

Fig.7 Relationship between number of neural units n and learning rate μ

4. Experimental results

A series of experiments were carried out to compare the performance of a PID controller withthat of the proposed neural controller for the VVVF hydraulic elevator in the State Key Laboratory of Fluid Power Transmission and Control in Zhejiang University. Fig. 8 shows the up speed

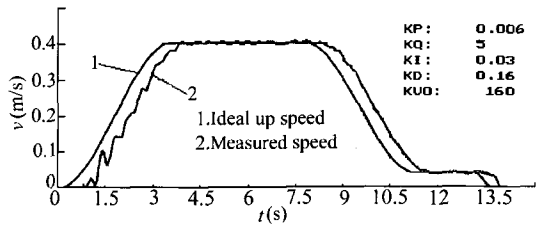

Fig.8 Velocity tracking curve of PID control

results of the test elevator using a PID controller, and the (a), (b), (c) of Fig. 9 show the neural controller with perturbation and 30, 20, 5 neural units respectively. The initial weights of each net unit was zero. In these figures, curve 1 is the ideal up speed and curve 2 is the measured speed. In the experiment, while the number of ANN units was selected to be from 5 to 30, better control performances could be obtained only by adjusting the value of the learning rate.

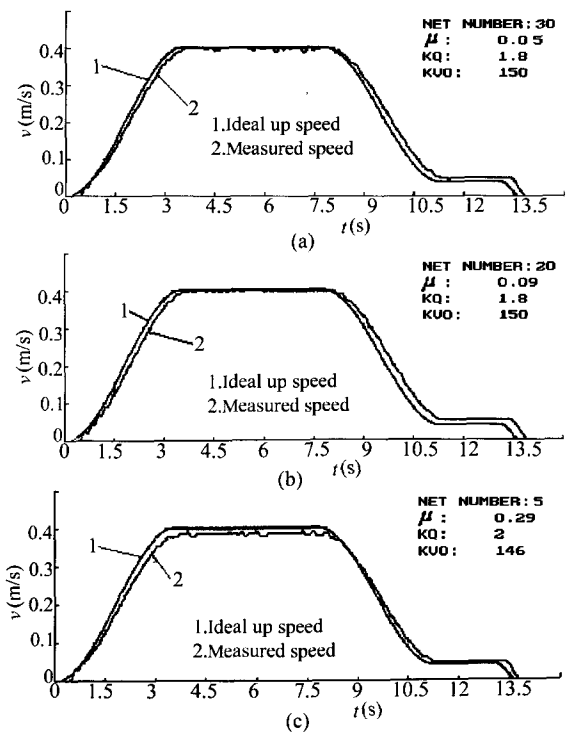

Fig.9 Velocity tracking curve of ANN control with perturbation
(a) unit = 30; (b) unit = 20; (c) unit = 5

It is obvious from these measured results that the elevator velocity using the proposed algorithm was much better in terms of the smooth ride, response time and stability, particularly at the start of the system compared with those of the PID controller.

CONCLUSION

An adaptive algorithm of neural nets with special perturbation for a real time velocity control system of a VVVF hydraulic elevator was developed in this study, where the nonlinear plant to be controlled was considered as a nonlinear output function of the ANN controller constructed for this study. The weights of neural nets were adaptively adjusted using the gradient method with perturbation algorithm in order to avoid the otherwise necessary requirement that the output function of the nonlinear plant be considered. A real-time adaptive ANN controller with perturbation was developed for the nonlinear hydraulic elevator velocity control system. The simulation and experimental results showed that the ANN real-time controller proposed has obviously simple scheme and fast dynamic response. It can be concluded from the study that the controller developed can suitably be used to control the VVVF hydraulic elevator velocity and can also be used as a reference in the development of ANN controllers for other similar control systems.

References

Albus, J. S., 1975. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Journal of Dynamic System, Measurement and Control, Trans. ASME*, **97**(3): 220 - 227.

Miller, W. T., Hewes, R. P., Glanz, F. H., et al., 1990. Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller. *IEEE Trans. on Robotics and Automation*, **6**(1): 1 - 9.

Yang, Jiangang, 1993, Real time dynamic control of a double inverted pendulum using ameliorated CMAC network. The 1st Congress of Post-Doctoral of China , National Defense Industry Press, Beijing, p.358 - 361.