A Study on an Identification Procedure for Control of Nonlinear Plants Using Neural Networks

In Soo Lee and Gi Joon Jeon
Dept. of Electronics, Kyungpook National University, Taegu, KOREA and
Eng. Research Center for Adv. Control and Instr. of SNU by KOSEF

Abstract

A new learning method of both NNI and NNC by which the NNI identifies precisely the dynamic characteristics of the plant is proposed. For control of the nonlinear plant we use two neural networks, one for identification and the other for control. We define a closed loop error which depends on identification and control error. In the proposed learning method, the closed loop error is utilized to train the NNI and the NNC. Computer simulation results reveal that the NNC based on proposed method is insensitive to variations of the plant parameters.

I. Introduction

In the past three decades major advances have been made in adaptive control theory for identifying and controlling linear plants with varying parameters. But conventional adaptive control methods require the mathematical model describing the dynamics of the plant to be controlled. In practice, the exact representative model of the plant is difficult to obtain due to uncertainty and plant parameter variations. Recently, artificial neural networks are widely utilized for control of the nonlinear plants because of their learning ability and various fruitful results of the networks [1], [2].

The BP(backpropagation) algorithm which is based on the gradient descent method is widely used to train the MNN's(multilayered neural networks) to perform a desired task [3]. Nguyen and Widrow [4], Narendra and Parthasarathy [5] proposed indirect control method using two neural networks. But, the methods described in those paper require that NNI(neural network identifier) has been pre-trained as a model of the plant. Therefore, the NNC(neural network controller) is not adapted immediately when plant parameters are changed because of uncertainties or disturbances.

In the normal BP method, different errors are utilized to train the NNI and NNC. Namely, in indirect control of nonlinear plants using neural networks, weights of the NNI are adjusted by the identification error between the plant and the identification model output while those of the NNC are adjusted by backpropagating the control error between desired value and plant output through the NNI. This implies that when an indirect controller using neural networks is constructed the controller is learned by information of the NNI. Therefore, the NNI must be learned precisely so that characteristics of the plants are contained in the NNI as much as possible.

In this paper, we propose a new learning method of the NNI and NNC so that the NNI identified precisely the dynamic characteristics of the plant. We define a closed loop error, which depends on identification and control error, as difference between outputs of the reference model and two-network system(NNI plus NNC). In the proposed learning method, the identification error and the closed loop error are utilized to train the NNI, whereas the control error and the closed loop error are used to train the NNC. For control of the nonlinear plant we use two neural networks, one for identification and the other for control, and proposed NN control system is based on a framework of the MRC. An example is presented to illustrate the proposed learning method.

II. Problem statement

Consider the single-input single-output discrete-time nonlinear plant

$$y_p(t+1) = g[\, y_p(t),\, y_p(t-1),\, \cdots ,y_p(t-n+1),\; u(t),\, u(t-1),\, \cdots ,\, u(t-m+1)\,] \tag{1}$$

where $y_p(t)$ and $u(t)$ are the plant output and input at time t, respectively, n and m are nonnegative integers, and $g(\cdot)$ is some nonlinear function.

The reference model is described by the following transfer function

$$G_m(z) = k_m \frac{n_m(z)}{d_m(z)} \tag{2}$$

where $n_m(z)$ and $d_m(z)$ are monic coprime polynomials of degree m and n, respectively, and k_m is a constant. It is assumed that the reference model is stable, minimum phase. Also the reference input $r(t)$ to the model is specified and is assumed to be uniformly bounded. The control problem is to determine a bounded control input $u(t)$ to the plant so that the plant output $y_p(t)$ tracks the reference model output $y_m(t)$ as close as possible.

The application of conventional control methods to nonlinear plant described by equation (1) is difficult because it requires a mathematical model describing the plant to be controlled. To solve this control problem we use two MNN's, one for identification and the other for control. The basic structure of the MNN with feedforward connections is shown in Fig. 1.

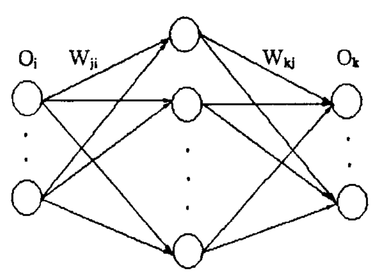


Fig.1 Multilayer neural network structure

III. Proposed learning method

The proposed neural network control architecture based on MRC is shown in Fig. 2. This architecture consists of two neural networks each of which performs a different task, one for identification and the other for control.

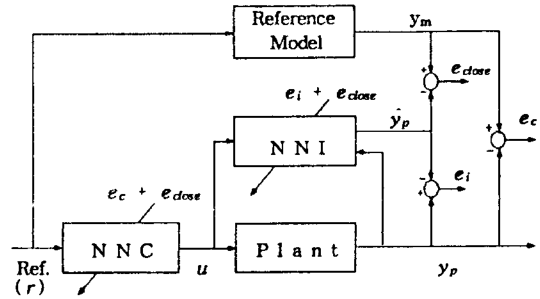


Fig.2 Proposed architecture of neural network control system

In the normal BP method, the NNI and the NNC are trained to minimize the identification error and the control error respectively. We define a closed loop error which depends on identification and control error. In the proposed learning method, the NNI is trained by minimizing a combined identification error (identification error plus the closed loop error) while the NNC is trained by minimizig the combined control error(control error plus the closed loop error).

1. Identification strategy

The problem of identification consists of setting up a suitably parameterized NNI and adjusting the weights of the NNI to minimize the cost function defined as follows:

$$E_I = \frac{1}{2} \sum_{k=1}^{N} [(y_p^k(t) - \hat{y}_p^k(t))^2 + (y_m^k(t) - \hat{y}_p^k(t))^2] \tag{3}$$

where subscript I means identification and N is the number of output nodes. $y_m(t)$, $y_p(t)$ and $\hat{y}_p(t)$ are outputs of the reference model, plant and the NNI, respectively. Combined identification error, identification error and closed loop error, is defined as follows:

$$e_{NNI}(t) = e_i(t) + e_{close}(t) \tag{4}$$

$$e_i(t) = y_p^k(t) - \hat{y}_p^k(t) \tag{5}$$

$$e_{close}(t) = y_m^k(t) - \hat{y}_p^k(t) \tag{6}$$

The first term on the right-hand side of (3), which is the identification error as in (5), corresponds exactly to a cost function in the normal BP algorithm. The second term is the closed loop error which is difference between reference model output and two-network system(control plus identification network) output.

In the following, the subscripts k, j and i refer to any node in the output, hidden, and input layers, respectively. Error signal at the k-th output node is obtained as

$$\delta_{Ik} = -\frac{\partial E_I}{\partial net_k} = -\frac{\partial E_I}{\partial \hat{y}_p^k(t)} \cdot \frac{\partial \hat{y}_p^k(t)}{\partial net_k} \quad (7)$$
$$= [(y_p^k(t) - \hat{y}_p^k(t)) + (y_m^k(t) - \hat{y}_p^k(t))] \cdot \dot{f}_k(net_k)$$

where $f_k(\cdot)$ and $\dot{f}_k(\cdot)$ are the sigmoid activation function and its derivative respectively. The incremental change of output weights in steepest descent method and the updated weights are computed as follows:

$$\Delta w_{kj} = -\eta \cdot \frac{\partial E_I}{\partial w_{kj}} = \eta \cdot \delta_{Ik} \cdot O_{Ij} \quad (8)$$

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj} + \alpha \cdot (w_{kj}(t) - w_{kj}(t-1)) \quad (9)$$

where η is the learning rate, α is the momentum term to accerlate learning speed. O_{Ij} is the j-th actual output in the hidden layer and w is the weight vector. Error signal at the j-th hidden node and the updated weights are obtained as follows:

$$\delta_{Ij} = -\frac{\partial E_I}{\partial net_j} = -\frac{\partial E_I}{\partial O_{Ij}} \cdot \frac{\partial O_{Ij}}{\partial net_j} \quad (10)$$
$$= \sum_{k=1}^{N} \delta_{Ik} \cdot w_{kj} \cdot \dot{f}_k(net_j)$$

$$\Delta w_{ji} = -\eta \cdot \frac{\partial E_I}{\partial w_{ji}} = \eta \cdot \delta_{Ij} \cdot O_{Ii} \quad (11)$$

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} + \alpha \cdot (w_{ji}(t) - w_{ji}(t-1)) \quad (12)$$

2. Control strategy

Weights of the NNI are adjusted by the combined identification error while those of the NNC are adjusted by backpropagating the combined control error through the NNI. Training procedures of the two neural networks are done simultaneously and the weights of the NNI are fixed for training of the NNC. To train the NNC, we define the cost function as follows:

$$E_C = \frac{1}{2} \sum_{k=1}^{N} [(y_m^k(t) - y_p^k(t))^2 + (y_m^k(t) - \hat{y}_p^k(t))^2] \quad (13)$$

Combined control error, control error and closed loop error are defined as follows:

$$e_{NNC}(t) = e_c(t) + e_{close}(t) \quad (14)$$
$$e_c(t) = y_m^k(t) - y_p^k(t) \quad (15)$$
$$e_{close}(t) = y_m^k(t) - \hat{y}_p^k(t) \quad (16)$$

where N is the number of output nodes, $y_m(t)$, the reference model output, $y_p(t)$, the plant output and $\hat{y}_p(t)$, the NNI output. The first term on the right-hand side of (13), which is the control error as in equation (15), equals to the cost function in the normal BP algorithm. Closed loop error is difference between reference model output and two-network system(control plus identification networks) output. Actual plant output $y_p(t)$ is unknown. Therefore, there is no way to obtain partial derivatives of the output analytically. To overcome this problem we rewrite the equation (15) as

$$e_c(t) = y_m^k(t) - e_I(t) - \hat{y}_p^k(t) \quad (17)$$

Error signals at the output and hidden layers in the NNI are obtained from the equation (13) and (17) as follows:

$$\delta_{Ik} = [(y_m^k(t) - y_p^k(t)) + (y_m^k(t) - \hat{y}_p^k(t))] \cdot \dot{f}_k(net_k) \quad (18)$$

$$\delta_{Ij} = \sum_{k=1}^{N} \delta_{Ik} \cdot w_{kj} \cdot \dot{f}_j(net_j) \quad (19)$$

These are backpropagated to the output layer of the NNC and error signal to train the NNC at the output layer is obtained as

$$\delta_{Ck} = -\frac{\partial E_C}{\partial net_k} = -\frac{\partial E_C}{\partial U_k} \cdot \frac{\partial U_k}{\partial net_k} \quad (20)$$
$$= \sum_{j=1}^{J} \delta_{Ij} \cdot w_{ji} \cdot \dot{f}_k(net_k)$$

where subscript C means control, J is the number of hidden nodes and U_k is control signal at the output layer in the NNC. The incremental change of output weights in steepest descent method and the updated weights are computed by the following equations:

$$\Delta v_{kj} = -\eta \cdot \frac{\partial E_C}{\partial v_{kj}} = \eta \cdot \delta_{Ck} \cdot O_{Cj} \quad (21)$$

$$v_{kj}(t+1) = v_{kj}(t) + \Delta v_{kj} + \alpha \cdot (v_{kj}(t) - v_{kj}(t-1)) \quad (22)$$

where v is the weight vector of the NNC. Error signal at the j-th hidden node and the updated weights are obtained from the following equations:

$$\delta_{Cj} = -\frac{\partial E_C}{\partial net_j} = -\frac{\partial E_C}{\partial O_{Cj}} \cdot \frac{\partial O_{Cj}}{\partial net_j} = \sum_{k=1}^{K} \delta_{Ck} \cdot v_{kj} \cdot \dot{f}_k(net_j) \quad (23)$$

$$\Delta v_{ji} = -\eta \cdot \frac{\partial E_C}{\partial v_{ji}} = \eta \cdot \delta_{Cj} \cdot O_{Ci} \quad (24)$$

$$v_{ji}(t+1) = v_{ji}(t) + \Delta v_{ji} + \alpha \cdot (v_{ji}(t) - v_{ji}(t-1)) \quad (25)$$

Ⅳ. Simulation results

In this example the nonlinear plant is given by

$$y_p(t+1) = \left[\frac{y_p(t)}{1+y_p(t)^2}\right] + u^3(t) \quad (26)$$

where $y_p(t)$ and $u(t)$ are the plant output and input, respectively. The reference model and reference input adopted from the reference [2] are written as

$$y_m(t+1) = 0.6y_m(t) + r(t) \quad (27)$$

$$r(t) = \sin\left(\frac{2\pi t}{25}\right) + \sin\left(\frac{2\pi t}{10}\right) \quad (28)$$

where $y_m(t)$ is the output of the reference model and $r(t)$ is reference input. Both NNI and NNC consist of 2 inputs, 1 hidden layer with 30 nodes, and a single output node, identically. The weights of the NNI are adjusted at every time step using a learning rate, $\eta = 0.2$, momentum term, $\alpha = 0.1$ and those of the NNC are adjusted at every time step using $\eta = 0.08$, $\alpha = 0.1$. Training continued for 20000 time steps altogether. When the normal BP method and proposed learning method have been applied to the control of above plant, the output of the plant as well as that of the reference model are shown in parts (a) and (b) of Fig. 3, respectively.

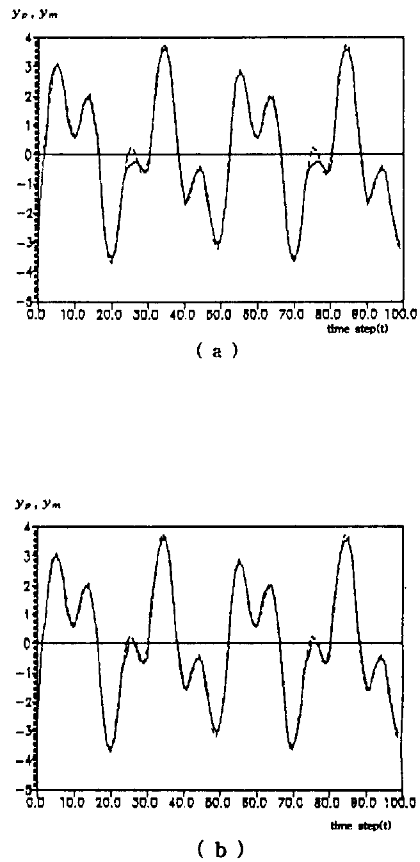
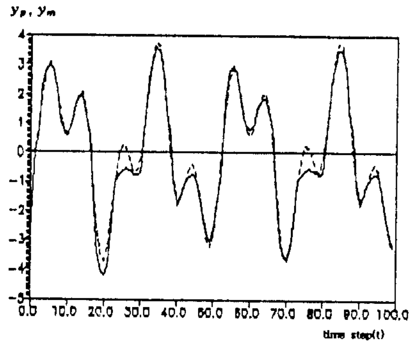


Fig. 3. Outputs of the plant(solid lines) and the reference model(dashed lines) after 20000 time-steps (a) By normal BP algorithm, (b) by Proposed algorithm

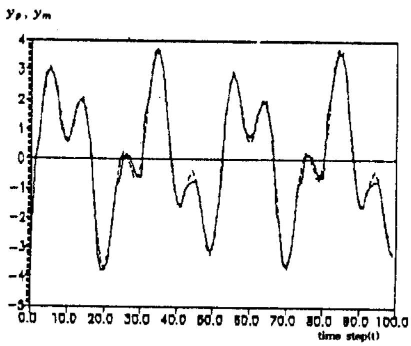
The above results indicate that the proposed learning method is better than normal BP method. We also examined the sensitivity of the trained controllers to parameter variations of the plant. The previously trained NNC and NNI were used to generate control signals for the following plant. Notice that all the coefficients of the original system are unities.

$$y_p(t+1) = \left[\frac{1.3y_p(t)}{1+0.7y_p(t)^2}\right] + 1.2u^3(t) \quad (29)$$

The results shown in parts (a) and (b) of Fig. 4 illustrate that the NNC trained by the proposed learning method, is less sensitive to variations in the plant parameters than the NNC trained by the normal BP algorithm.



(a)



(b)

Fig. 4. Outputs of the plant(solid lines) and the reference model(dashed lines) when the parameters of the plant change at 16 time-step after learning by (a) Normal BP algorithm, (b) Proposed algorithm

V. Conclusion

A new learning method by which the NNI identifies precisely the dynamic characteristics of the plant is presented. In the proposed approach, we introduce a closed loop error and both weights of the NNI and those of the NNC are adjusted by the closed loop error. And the identification phase and the control phase are carried out simultaneously. The simulation results illustrate that the NNC trained by proposed learning method is less sensitive to variations in the plant parameters than NNC trained by the normal BP method. Therefore, we conclude that proposed learning method is superior to the normal BP method. The stability of the overall system, selection of the initial weights, the number of hidden layers and the number of nodes need to be further investigated.

VI. References

[1] M.M. Gupta, D.H. Rao, and J. Gao, "Learning and adaptation in neural control of higher-order linear systems," *American Control Conference* (*ACC*), pp. 3044-3048, 1992.

[2] M.M. Gupta, D.H. Rao, and H.G. Wood, "Learning and adaptive neural controller," *the Int. Joint Conference on Neural Networks* (*IJCNN*), Singapore, pp. 2380-2385, Nov. 1991.

[3] K.S. Narendra, and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 252-262, Mar. 1991.

[4] D.H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Contr. Syst. Mag.*, vol. 10, pp. 18-23, Apr. 1990.

[5] K.S. Narendra, and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, 1990.