Remarks on adaptive type neural network direct controller with separate learning rule of each layer

Takayuki Yamada
Department of Computer and Information Sciences
Faculty of Engineering
Ibaraki University
4-12-1 Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan

Abstract

This paper proposes a new learning rule of multi-layer neural network controllers in order to eliminate an inference of neural network weights of each layer and to discuss stability condition of nonlinear three layer neural network controllers. This learning rule is that the neural network weights between one layer and next layer are only changed at same time and other neural network weights are not changed, and after some sampling numbers, the neural network weights between other layer and next layer are changed. The proposed learning rule is applied to an adaptive type neural network direct controller and a simulation result shows it performed well.

1. Introduction

Many studies have been undertaken in order to apply both the flexibility and the learning capability of neural networks to control systems.[1]-[3] However, there are few attempts to clear the stability conditions of neural network control systems. Among these attempts, we proved local stability condition of a three layer neural network direct controller whose neurons have a linear input output relation.[1][2] We also confirmed that the interference between neural network weight learning of each layer causes more difficulty to analyze the stability condition in comparison with the stability analysis of conventional adaptive controllers in this study. This interference is caused by an usual neural network learning rule so as to change whole neural network weights at same time. In other words, the usual neural network learning rule has good efficiency of the neural network weight convergence, but it causes difficult stability analysis of the neural network controllers. One of simple methods to avoid this interference is to use a two layer neural network as a controller. However, the nonlinear mapping capability of such neural network is limited because it was proved that a nonlinear three layer neural network was able to approximate any continuous nonlinear functions. That is, this method means that such neural network controllers have less nonlinear mapping capability in comparison with the nonlinear three layer neural network. Another method to avoid above problem is to use a linear three layer neural network. However, the linear three layer neural network can not realize sufficient approximation for nonlinear functions. This also means that such neural network controllers have less nonlinear mapping capability.

Thus, we propose a new neural network learning rule for three layer nonlinear neural network controllers in this paper. It is called separate learning rule of each layer in this paper. This learning rule is that the neural network weights between one layer and next layer are only changed at same time and other neural network weights are not changed, and after some sampling numbers, the neural network weights between other layer and next layer are changed. That is, we can eliminate the interference between layers and realize easier discussion about the nonlinear neural network controller stability although the separate learning rule has slow convergence in comparison with the usual neural network learning rule. We can realize the same nonlinear mapping capability because the neural network has the nonlinear three layer structure. The proposed learning rule is applied for an adaptive type neural network direct controller and the discussion about its stability condition is presented. A simulation result of the adaptive type neural network direct controller with the separate learning rule shows that the proposed learning rule can be realized.

2. Interference between neural network weights

This section explains the structure of the adaptive type neural network direct controller briefly.[1][2] The interference between neural network weight learning of each layer is focused and it is discussed why the stability analysis of the nonlinear neural network controllers with the three layer structure is difficult in comparison with that of controllers with linear two layer structure such as conventional adaptive controllers. Based on this discussion, necessity of the separate learning rule of each layer is confirmed.

The following object plant is selected in this paper.

$$Y(k) = f(Y(k\text{-}d)\bullet\bullet\bullet Y(k\text{-}d-n), U(k-d)\bullet\bullet\bullet U(k-d-m)) \qquad (1)$$

Where $Y(k)$ is the plant output, $U(k)$ is the plant input, k is the sampling number, d is the dead time, n&m are the plant orders and f is the nonlinear function which expresses a nonlinear characteristic of the plant. The output error $\varepsilon(k)$ is defined by the following equation.

$$\varepsilon(k) = Yd(k) - Y(k) \qquad (2)$$

Where Yd is the desired value. Figure 1 shows the scheme of the direct controller. As shown here, the plant input is only composed of the neural network output in the direct controller. From eqs.(1) and (2), the neural network input vector I is selected as follows;

$$I^T(k) = [Yd(k+d), Y(k)\bullet\bullet\bullet Y(k-n), U(k-1)\bullet\bullet\bullet U(k-m)] \qquad (3)$$

To simplify, the following discussion selects that the neural network has three layers, its output neuron is one and both the number of neurons in the input and hidden layers are the same as the order of the input vector I. When the adaptive type learning is selected, the usual neural network learning rule is expressed as the following equation through the use of the δ rule.

$$W(k+1) = W(k-d) - \eta' \frac{\partial J(k)}{\partial W(k-d)} \qquad J(k) = \frac{1}{2}\varepsilon^2(k)$$
$$\omega(k+1) = \omega(k-d) - \eta' \frac{\partial J(k)}{\partial \omega(k-d)} \qquad (4)$$

Where W is the weight matrix composed of the neural network weights between the input and hidden layers , ω is the weight vector composed of the neural network weights between the hidden and output layers, J is the cost function and η' is the parameter determining convergence speed. When the neural network is linear, we derive the following learning rule from eq.(4).

$$W(k+1) = W(k-d) - \eta\varepsilon(k)\omega(k-d)I^T(k-d)$$
$$\omega(k+1) = \omega(k-d) - \eta\varepsilon(k)W(k-d)I(k-d)$$
$$\eta = \eta'\left(\frac{\partial Y(k)}{\partial U(k-d)}\right) \qquad (5)$$

As shown in eq.(5), the learning of the weight matrix W uses the weight vector ω and the learning of the weight vector ω uses the weight matrix W. This fact causes the interference between the neural network weight learning of each layer and the difficulty of the stability analysis in comparison with that of the linear two layer structure controllers such as the conventional adaptive controller. The separate learning rule is proposed as one solution of above problem in the next section.

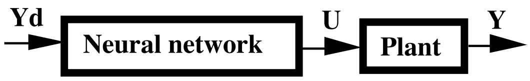

Fig.1 Scheme of direct controller.

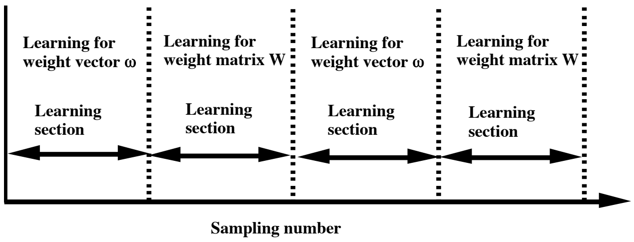

Fig.2 Scheme of separate learning of each layer.

3. Separate learning rule of each layer

This section proposes the separate learning rule of each layer as a solution to overcome the interference between the neural network weights. Figure 2 shows its scheme. This learning rule is that the neural network weights between one layer and next layer are only changed and other neural network weights are not changed within some sampling numbers which is called learning section. For this example, the weight vector ω is selected in the first learning section. That is, the learning rule for the weight ω of eq.(4) is only used. In the second learning section, the neural network weights between other layer and the next layer are changed. For this example, the weight matrix W is selected and the learning rule for the weight matrix W of eq.(4) is only used. As mention above, the neural network weights of each layer are independently changed. That is , we can eliminate the interference of the neural network weights. The nonlinear mapping capability is not reduced because our neural network has the three layer structure.

The stability condition of the weight vector ω is briefly discussed in the neighborhood of the converged vector. The following equation is obtained from the separate learning rule and eq.(4).

$$\omega_0 - \omega(k+1) = \omega_0 - \omega(k-d) - \eta' \frac{\partial Y(k)}{\partial U(k-d)} Sg(WcI(k-d)) \tag{6}$$

Where ω_0 is the converged vector within one learning section and Sg is the sigmoid function. When the weight vector ω is changed, the weight matrix W is the constant matrix Wc. Form eq.(6), the following stability condition of the weight vector ω is obtained when the Taylor expansion of the output error with regard to the weight vector ω is used. (Details are mentioned in the Appendix)

$$0 \le \eta(\frac{\partial Y(k)}{\partial U(k-d)})\lambda_0 \le 2 \qquad \eta = \eta'(\frac{\partial Y(k)}{\partial U(k-d)}) \tag{7}$$

Where λ_0 is the maximum eigen value of the following matrix P.

$$P = Sg(WcI(k-d))\{Sg(WcI(k-d))\}^T \tag{8}$$

The stability condition of the weight matrix W can be obtained in the similar method if we can use some assumptions. (Details are mentioned in the Appendix)

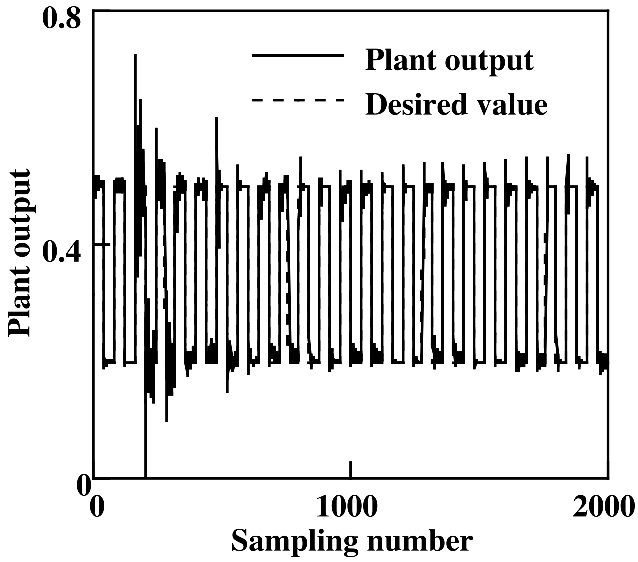

Fig.3 Simulation result.

4. Simulation

This section shows a simulation result of the adaptive type direct controller with the proposed separate learning rule of each layer. The simulated plant is follows: A simulated plant is selected as follows;

$$Y(k) = - a_1 Y(k-1) - a_2 Y(k-2) \\ +U(k-1) +bU(k-2) -a_3 Y(k-3)+ C_{non} Y^2(k-1) \tag{9}$$

Where a_1, a_2 & b are the plant parameters, a_3 is the parasite term and C_{non} is the nonlinear term. For this simulation, a_1=-1.3, a_2=0.3, b=0.7, a_3=-0.03 and C_{non}=0.2 are selected. The rectangular wave is also selected as the desired value Yd. We select the following sigmoid function f(x) as the input output relation of the hidden layer neuron.

$$f(x)=\frac{X_g\{1-\exp(-4x/X_g)\}}{2\{1+\exp(-4x/X_g)\}} \tag{10}$$

Where Xg is the parameter determining the sigmoid function shape. Xg=0.5 is selected in this simulation.

Figure 3 shows the simulation result where η=1.5 and the learning section is 2 cycle of the desired value. The solid line is the plant output Y and the broken line is the desired value Yd. As shown here, the output error suddenly increases after 2 cycle of the desire value, but after that, the plant output converges with the desired value as learning progresses. This result shows that the proposed learning rule performs well.

4. Conclusion

This paper proposed the new learning rule of the multi-layer neural network controller in order to eliminate the inference of the neural network weights of each layer . We can discuss the stability condition of the nonlinear three layer neural network controller through the use of this learning rule. It was applied to the adaptive type neural network direct controller and the simulation result showed that it performed well.

Acknowledgment

The author wishes to express his thanks to Mr.Masaki Kobayashi, graduated student, Ibaraki University, for his programming and simulation.

References

[1]T.Yamada and T.Yabuta, “Some Remarks on Characteristics of Direct Neuro-Controller with Regard to Adaptive Control”, Transactions of the Society of Instrument and Control Engineers, Vol.27, No.7, pp.784-791,1991 (in Japanese)

[2]T.Yamada and T.Yabuta, “Adaptive Neural Network Controllers for Dynamics Systems”, Journal of Robotics and Mechatronics Vol.2, No.4, pp.29-41, 1989

[3]K.S.Narendra and K.Parthictsarathy, “Identification and Control of Dynamics System Using Neural Networks”,

IEEE Transaction on Neural Networks, Vol.1, No.1, pp.4-27, 1990

Appendix

First, this appendix discusses the stability condition of the weight vector ω. The following parameter error vector ζ is defined.

$$\zeta(k) = \omega_0 - \omega(k) \tag{a-1}$$

It is assumed that $\varepsilon(k)=0$, when $\omega(k)=\omega_0$. From eq.(6) and the first order Taylar expansion of the output error $\varepsilon(k)$ with regard to the weight vector $\omega(k\text{-}d)$, Following equations are obtained.

$$\zeta^T(k+1) = \zeta^T(k-d)[E - \eta' g^2(k) Sg(WcI(k-d))\{Sg(WcI(k-d))\}^T]$$
$$\varepsilon(k) = g(k)\ Sg(I^T(k-d)Wc^T) \qquad g(k) = \frac{\partial Y(k)}{\partial U(k-d)} \tag{a-2}$$

Where E is the identity matrix. When $\varphi(k)=\zeta^T(k)\zeta(k)$ is selected as a candidate of the Lyapunov function, we can obtain the following equation.

$$\begin{aligned} \Delta\varphi &= \varphi(k+1) - \varphi(k-d) \\ &= \zeta^T(k-d) Q \zeta(k-d) \\ Q &= -2\eta' g^2(k) P + \eta'^2 g^4(k) P^2 \end{aligned} \tag{a-3}$$

Since P defined by eq.(8) is the real symmetric matrix whose eigen values are not negative, there is a real orthogonal matrix V so as to $P=V^{-1}\beta V$ where β is a diagonal matrix whose diagonal elements are the eigen values of P. From eq.(a-3), the following equation is derived.

$$Q = V^{-1}(\eta'^2 g^4(k)\beta^2 - 2\eta' g^2(k)\beta)V \tag{a-4}$$

When λi is defined by the eigen value of β, λi is not negative and the rank of β is 1. That is, the positive λi is 1 and this is the maximum eigen value λ_0. From eqs.(3)(4), when the following equation is satisfied, $\Delta\varphi$ is not positive and the neural network controller is stable with regard to the weight vector ω learning.

$$0 \le \eta\left(\frac{\partial Y(k)}{\partial U(k-d)}\right)\lambda_0 \le 2 \qquad \eta = \eta'\left(\frac{\partial Y(k)}{\partial U(k-d)}\right) \tag{7}$$

Next, the stability condition of the weight matrix W is discussed. When the weight matrix W is changed, the weight vector $\omega(k)$ is constant vector whose symbol is ωc. When the chain rule is use, the following equation is obtained from eqs.(2) and (4).

$$\frac{\partial J(k)}{\partial W(k-d)} = -\varepsilon(k)\frac{\partial Y(k)}{\partial U(k-d)}\frac{\partial U(k-d)}{\partial W(k-d)} \tag{a-5}$$

From the neural network structure, the following equation is obtained.

$$\frac{\partial U(k)}{\partial W_{ij}(k)} = \omega c_i Sg'\{\sum_{j=1}^{n} W_{ij}(k) I_j(k))\} I_j(k) \tag{a-6}$$

Where Sg' is the derivative of the sigmoid function Sg with regard to its input and n is the number of the input and hidden layers. We can define the diagonal matrix Γ whose iith diagonal element is follows;

$$Sg'\{\sum_{j=1}^{n} W_{ij}(k) I_j(k))\} \tag{a-7}$$

From eqs.(4) and (a-5)-(a-7), the learning rule of the weight matrix W is expressed as the following equation.

$$W(k+1) = W(k-d) + \eta'\varepsilon(k)g(k)\Gamma(k-d)\omega c I^T(k-d) \tag{a-8}$$

Here, when the input vector I is continuous, we can derive the following equation from eq.(8).

$$\begin{aligned} &W(k+1)I(k+1) \cong W(k+d)I(k-d) \\ &= W(k-d)I(k-d) + \eta'\varepsilon(k)g(k)\Gamma(k-d)\omega c I^T(k-d)I(k-d) \end{aligned} \tag{a-9}$$

The following equation is the definition of the parameter error vector ζ for the weight matrix W.

$$\zeta(k) = W_0 I(k) - W(k)I(k) \tag{a-10}$$

Where W_0 is the converged weight matrix W within one learning section. From eqs.(4)(a-9)(a-10) and the first order Taylar expansion of output error $\varepsilon(k)$ with regard to $W(k\text{-}d)I(k\text{-}d)$, we can obtain the following equation.

$$\begin{aligned} \zeta^T(k+1) = \zeta^T(k-d)\{&E - \eta' g^2(k)\Gamma(k-d) \\ &\times \omega c\, \omega c^T \Gamma(k-d) I^T(k-d) I(k-d)\} \end{aligned} \tag{a-11}$$

From eq.(a-11), we can obtain the stability condition of the weight matrix W in the same way.