Discussion of Stability of Adaptive Type Neural Network Direct Controller and Its Folding Behavior

[1]Takayuki Yamada

[1]Ibaraki University, 4-12-1 Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan
(Tel: +81-294-38-5256, Fax: +81-294-38-5282)

[1]tyamada@mx.ibaraki.ac.jp

Abstract: This paper discusses stability of an adaptive type neural network direct controller in the viewpoint of its folding behavior. First, I discuss the stability for the nonlinear plant and the nonlinear neural network. This discussion confirms that we can include the plant Jacobian problem into the tuning problem of the parameter determining the neural network learning speed. This is because it was confirmed that the direct controller has the folding behavior and the sign of the slope of a part of plant inverse characteristics learned by the neural network dose not change. This means that the sign of the plant Jacobian dose not change. Next, I assume input output relations of the neural network are linear and present the detail of the stability condition. This assumption may not be practical, but it is helpful for understanding of the relationship between the plant Jacobian and the parameter tuning.

Keywords: Neural network, Controller, Learning, Adaptive

I. INTRODUCTION

Many studies have been undertaken in order to apply both the flexibility and the learning capability of neural networks to control systems. We seem to believe that a neural network learns an inverse dynamic of a target plant in a servo level neural network controller application. This is because the target plant dynamics can be cancelled by this learned inverse dynamics if the neural network can learn it. On the other hand, a practical plant generally has a nonlinear dynamics and it is mathematically expressed a many-to-one function whose more than one input values correspond to one output value. There is no inverse function of such function. For this problem, we confirmed that the neural network has the folding behavior in the neural network direct controller through the use of the sine wave as a nonlinear plant.[1] This behavior is that the neural network learns only one branch (; a part of the inverse characteristics and it can be expressed as a one-to-one function mathematically) of the inverse characteristics of the target plant in order to obtain whole plant output. When the neural network realizes such input output mapping, the whole inverse characteristics of the target plant seem to be folded into one branch of the inverse characteristics. This behavior means that we can realize an ideal control system if the neural network can learn only one inverse branch of the plant. Other remarkable point was also confirmed that the sign of the slope of one branch dose not change.[1] That is, the sign of the plant Jacobian (; the derivative of the output with regard to the input) of the branch learned by the neural network of the direct controller dose not change. This fact seems to realize the learning of the neural network direct controller without the direct object plant modeling. The complex structures of the forward and inverse modeling, the feedback error learning and so on are not necessary with regard to realize the neural network learning. However, our previous papers pointed out the possibility of this realization and we did not show the discussion about its stability.

Thus, this paper discusses the stability of the adaptive type neural network direct controller[2] in the viewpoint of the folding behavior. In the second chapter, I discuss the stability for the nonlinear plant and the nonlinear neural network. This discussion confirms that we can include the plant Jacobian problem into the tuning probelm of the parameter determining the neural network learning speed. This is because the sign of the slope of the learned branch dose not change and the sign of the plant Jacobian learned by the neural network of the direct controller dose not change.[1] In the third chapter, I assume the neural network is linear. This assumption may not be practical, but it is helpful for understanding of the relationship between the plant Jacobian and the parameter tuning. This is because this assumption can realize the more details of the stability analysis.

II. LEARNING OF NEURAL NETWORK DIRECT CONTROLLER AND FOLDING BEHAVIOR

Figure 1 shows a structure of the neural network direct controller. As shown in fig.1, the neural network output is the plant input and the learning of the neural network performs so as to minimize the error between the plant output and the desired value (teaching signal). If the learning of the direct controller performs well (equals that the plant output becomes to be the desired value), the neural network of the direct controller obtains a part of the inverse characteristics of the plant dynamics as mentioned in my previous paper.[1]

This paper selects the steepest descent method as the neural network learning rule. It is shown in the following equation.

$$\frac{dW}{dt} = -\eta \frac{\partial J}{\partial W} \tag{1}$$

Where W is the weight of the neural network, η is the parameter determining the learning speed and J is the cost function. This cost function is the square error between the plant output Y and the desired value Yd.

$$\frac{dJ}{dt} = \frac{\partial J}{\partial W}\frac{dW}{dt} \tag{2}$$

If the above equation (2) is satisfied, we can derive the following equation from eq.(1)

$$\frac{dJ}{dt} = -\eta \left(\frac{\partial J}{\partial W}\right)^2 \tag{3}$$

As shown in the above equation, since we can select that the parameter η is the positive value, the cost function J decreases as time progress. That is, the learning rule of the neural network is stable and the plant output becomes to be close to the desired value as time progress because the cost function J is the square error between the plant output and the desired value.

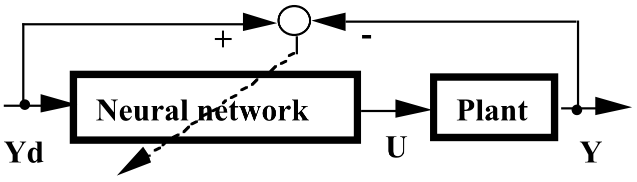

Fig.1 Scheme of neural network direct controller.

If we use an analog hardware as the neural network, we may realize the learning rule of eq.(2). However, most of the neural network controllers use the computer software since it is useful. In this case, the neural network learning rule is expressed as the following equation.

$$W(k+1) = W(k-d) - \eta_0 \frac{\partial J(k)}{\partial W(k-d)} \tag{4}$$

Where k is the sampling number and d is the dead time of the plant. Since this paper selects the adaptive type neural network, the weight of the neural network is changed at every several sampling time as shown in eq.(4). The cost function J(k) is defined as follows;

$$J(k) = \frac{1}{2}\varepsilon^2(k) \tag{5}$$

$$\varepsilon(k) = Yd(k) - Y(k) \tag{6}$$

Where Y is the plant output, Yd is the desired value and ε is the output error between them. Here, when the following eqs.(7) and (8) are assumed, we can obtain eq.(9) from eq.(4).

$$\frac{dW(k-d)}{dt} \cong w(k+1) - w(k-d) \tag{7}$$

$$\frac{dJ(k)}{dt} \cong \frac{\partial J(k)}{\partial W(k-d)}\frac{dW(k-d)}{dt} \tag{8}$$

$$\frac{dJ(k)}{dt} \cong -\eta_0 \left(\frac{\partial J(k)}{\partial W(k-d)}\right)^2 \tag{9}$$

As shown here, although we select the discrete time control system, if the eqs.(7) and (8) are satisfied, the plant output becomes to be close to the desired value as learning progress.

However, it is difficult to realize the learning rule eq.(4) in the direct controller when the plant characteristics is unknown. It is because $(\partial J(k)/\partial W(k\text{-}d))$ can not be calculated. As the practical solution, we can use the following equation in the replace of eq.(4).

$$W(k+1) = W(k-d) + \eta\, \varepsilon(k) \frac{\partial U(k-d)}{\partial W(k-d)} \tag{10}$$

Where U is the plant input. It equals the neural network output in the neural network direct controller as shown in fig.1. We must tune the parameter η in the replace of η_0 in eq.(4) and we can calculate eq.(10). It is because $(\partial U(k\text{-}d)/\partial W(k\text{-}d))$ can be determined by the neural network structure and this is known. The meaning of eq.(10) is explained in the following eq.(11) if eq.(12) is satisfied.

$$\eta = \eta_0 \frac{\partial Y(k)}{\partial U(k-d)} \tag{11}$$

$$\frac{\partial J(k)}{\partial W(k-d)} = \frac{\partial J(k)}{\partial Y(k)}\frac{\partial Y(k)}{\partial U(k-d)}\frac{\partial U(k-d)}{\partial W(k-d)} \tag{12}$$

$$\frac{\partial J(k)}{\partial Y(k)} = -\varepsilon(k) \tag{13}$$

From the similar way of eqs.(7)(8) and (10), we can obtain the following equation.

$$\frac{dJ(k)}{dt} = \eta\, \varepsilon(k) \frac{\partial J(k)}{\partial W(k-d)}\frac{\partial U(k-d)}{\partial W(k-d)} \tag{14}$$

From eqs(11)-(14), the following equation is obtained.

$$\frac{dJ(k)}{dt} = -\eta \frac{\partial Y(k)}{\partial U(k-d)} \left(\varepsilon(k) \frac{\partial U(k-d)}{\partial W(k-d)}\right)^2 \tag{15}$$

Where ($\partial Y(k)/\partial U(k-d)$) is the derivative of the plant output Y with regard to the plant input U and it is called the plant Jacobian. As shown in eq.(15), if $\eta(\partial Y(k)/\partial U(k-d))$ is positive, the cost function J decreases as time progress and the plant output becomes to be close to the desired value. That is, the neural network learning is stable and it performs well,

If the plant is linear system, it is well known that the plant Jacobian ($\partial Y(k)/\partial U(k-d)$ is constant. It is not difficult to tune the parameter η for the linear plant. However, if the plant is nonlinear, the plant Jacobian is not constant and it may be changed from negative to positive or from positive to negative in the neural network learning progress. Thus, the learning rule eq.(10) seems to be not practical because it seems to be difficult to tune the parameter η. However, it was confirmed that the neural network direct controller has the folding behavior.[1] It is not necessary for neural network learning of the direct controller to obtain whole plant inverse characteristics. If the neural network obtains only one branch (a part of the plant inverse characteristics and it is mathematically expressed as one to one function), the plant output matches with the desired value for whole region. The plant Jacobian in eq.(15) is referred to the branch which the neural network must learn. This is because the plant is controlled within this branch. The sign of its slope is not changed because the branch is expressed as one to one function and it monotonously increases or decreases. From eq.(15), the choice of the parameter η is positive or negative to stabilize the neural network learning. It is not so difficult to tune the parameter η.

As shown in eq.(15), the plant Jacobian effect can be included into the tuning of the parameter η and we can use the learning rule eq.(10). However, the assumptions eq.(7) and (8) seem to be strained. In particular, eq.(7) may not be satisfied around the weights which the neural network converge to. In the next section, I discuss the stability condition of the adaptive type neural network direct controller. This discussion uses the assumption that the input output relation of the neural network is linear. It is because it is difficult for the nonlinear neural network to analyze its stability condition, but the discussion of the next session is helpful to understand the plant Jacobian effect and the meaning of the learning rule eq.(10).

III. STABILITY CONDISION OF ADAPTIVE TYPE NEURAL NETWORK DIRECT CONTROLLER

Previous section shows the discussion of the stability of the neural network direct controller and its folding behavior, but the assumption of eqs.(7) & (8) seems to be strained. In order to discuss its stability condition, this section selects a three layer neural network whose neurons have a liner input output relation. This linear assumption may not be practical, but it is helpful for understanding of the neural network stability condition. The neural network is expressed as the following equation by use of this liner assumption.

$$U(k) = \omega^T(k)\ W(k)\ I(k) \tag{16}$$

Where ω is the weight vector between the hidden layer and the output layer, W is the weight matrix between the input layer and the hidden layer and I is the input vector of the neural network.

The following nonlinear plant is selected in this paper.

$$Y(k) = f(Y(k\text{-}d),\bullet\bullet\bullet,Y(k-d-n),U(k-d),\bullet\bullet\bullet,U(k-d-m)) \tag{17}$$

Where n & m are the plant orders and f is the nonlinear function which expresses the plant nonlinearity. The learning rule is derived as the following equations based on eqs.(10)-(13) and the linear assumption.

$$W(k+1) = W(k-d) + \eta\,\varepsilon(k)\omega(k-d)I^T(k-d) \tag{18}$$

$$\omega(k+1) = \omega(k-d) + \eta\,\varepsilon(k)W(k-d)I(k-d) \tag{19}$$

Here, we define the parameter error vector ζ as the following equation.

$$\zeta^T(k) = \omega_0^T\, W_0 - \omega^T(k)\ W(k) \tag{20}$$

Where ω_0 and W_0 are the weight vector and the weight matrix when the neural network is finished to converge.

The following discussion is described about the local stability when the neural network almost converges to the weight vector ω_0 and matrix W_0. From this assumption, the output error is expressed as follows;

$$\varepsilon(k+d) \cong \frac{\partial Y(k+d)}{\partial U(k)}\,\Delta U(k) \tag{21}$$

Where $\Delta U(k)$ is the difference between the present plant input and the plant input when the neural network is finished to converge. This $\Delta U(k)$ is sufficiently small because of the local stability. We assume the follows;

$$\begin{aligned}\Delta U(k) &\cong (\omega_0 - \omega(k))^T\,(W_0 - W(k))\ I(k)\\ &\cong (\omega_0^T W_0 - \omega_0^T W(k) - \omega^T(k)\ W_0 + \omega^T(k)\ W(k))\ I(k)\end{aligned} \tag{22}$$

From the local stability, we assume that both ω_0 and W_0 are almost equal to ω and W for the second and third terms of eq.(22). From eqs.(21)(22) and this assumption, we can obtain the following equation.

$$\varepsilon(k+d) = \frac{\partial Y(k+d)}{\partial U(k)} \zeta^T(k) I(k) \tag{23}$$

From the local stability, the output error ε is sufficiently small and ε^2 is almost 0. From this assumption and eqs.(18)(19) & (23), we obtain the following equation.

$$\begin{aligned}&\omega^T(k+1)\,W(k+1)\\&= \omega^T(k-d)\,W(k-d) + \eta\,\frac{\partial Y(k)}{\partial U(k-d)}\,\zeta^T(k-d)\,\xi(k-d)\end{aligned} \tag{24}$$

$$\xi(k) = I(k)\,\omega^T(k)\,\omega(k)\,I^T(k) + I(k)\,I^T(k)\,W^T(k)\,W(k) \tag{25}$$

From eqs.(20) and (24), the following equation is derived.

$$\zeta^T(k+1) = \zeta^T(k-d)(E - \eta\,\frac{\partial Y(k)}{\partial U(k-d)}\,\xi(k-d)) \tag{26}$$

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

$$\begin{aligned}\Delta\varphi &= \varphi(k+1) - \varphi(k-d)\\ &= \zeta^T(k-d)Q\zeta(k-d)\end{aligned} \tag{27}$$

$$Q = -2\eta\,(\frac{\partial Y(k)}{\partial U(k-d)})\,\xi(k-d) + \eta^2(\frac{\partial Y(k)}{\partial U(k-d)})^2\,\xi(k-d)\xi^T(k-d) \tag{28}$$

Since ξ defined by eq.(25) is the real symmetric matrix whose eigen values are not negative, there is a real orthogonal matrix V so as to $\xi=V^{-1}\beta V$ where β is a diagonal matrix whose diagonal elements are the eigen values of ξ. From eqs.(27) and (28), the following equation is derived.

$$Q = V^{-1}(\eta^2(\frac{\partial Y(k)}{\partial U(k-d)})^2\beta^2 - 2\,\eta\,\frac{\partial Y(k)}{\partial U(k-d)}\,\beta)V \tag{29}$$

When λ_i is defined by the eigen value of β, λ_i is not negative and the rank of β is 1. That is, the positive λ_i is 1 and this is the maximum eigen value λ_0 which is not 0. From eqs.(27)-(29), when the following equation is satisfied, Δϕ is not positive and the neural network controller is stable.

$$0 \le \eta(\frac{\partial Y(k)}{\partial U(k-d)})\lambda_0 \le 2 \tag{30}$$

As shown in eq.(30), the plant Jacobian ($\partial Y(k)/\partial U(k-d)$) is related to the stable condition. Its sign is important because the parameter η must be changed if its sign is changed in order to stabilize the neural network direct controller. However, my previous paper confirmed that the neural network of the direct controller has the folding behavior. It learns only one branch of the plant inverse in order to express whole plant characteristic. This branch is expressed as one to one function mathematically. The sign of the plant Jacobian is not changed. This is because the branch learned by the neural network monotonously increases or decreases. From eq.(30), it is not necessary to change the sign of the parameter η because λ_0 in eq.(30) is positive.

There is a maximum of absolute value of the plant Jacobian ($\partial Y(k)/\partial U(k-d)$) if the branches learned by the neural network are continuous. If neither this maximum nor λ_0 are not 0, we obtain the following stability condition.

$$0 \le \eta \le \frac{2}{P\lambda_0}\quad (0<P)\,,\qquad \frac{2}{P\lambda_0} \le \eta \le 0\quad (0>P)$$
$$P = \left|\frac{\partial Y(k)}{\partial U(k-d)}\right|_{Max} \tag{31}$$

where P is the maximum of absolute value of the plant Jacobian ($\partial Y(k)/\partial U(k-d)$). As shown in eq.(31), if we tune the parameter η so as to satisfy eq.(31), we can stabilize the neural network direct controller. This is because the maximum eigen value λ_0 is positive and it can be calculated within the neural network learning.

IV. CONCLUSION

This paper discusses the stability of the adaptive type neural network direct controller in the viewpoint of the folding behavior. First, I discuss the stability for the nonlinear plant and the nonlinear neural network. This discussion confirms that we can include the plant Jacobian problem into the tuning problem of the parameter determining the neural network learning speed. This is because the neural network direct controller has the folding behavior. However, the used assumption may be strained around the converged neural network weights. Thus, I assume the input output relation of the neural network is linear and present the more detail of the stability condition of the adaptive type neural network controller.

REFERENCES

[1] Takayuki Yamada, "Remarks on Folding Behavior of Mapping Capability of Neural Network Direct Controller for Many-to-one Plant", Proceedings of AROB 16th '11 (The Sixteenth International Symposium on Artificial Life and Robotics 2011), pp.881-884(2011)
[2] 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)