Remarks on Folding Behavior of Mapping Capability of Neural Network Direct Controller for Many-to-one Plant

Takayuki Yamada

Department of Computer and Information Sciences

Faculty of Engineering

Ibaraki University
4-12-1 Nakanarusawa, Hitachi, Ibaraki, 316-8511, Japan

Abstract: We believe that a neural network learns an inverse dynamic of a target plant in a servo level neural network controller application. However, a practical plant generally has a nonlinear dynamics and there is no inverse dynamics of it usually. This paper studied this problem through the use of the neural network direct controller. Simulation results confirmed that the neural network learned the branch of the inverse characteristics (; a part of the inverse characteristics and it can be expressed as a one-to-one function mathematically) and if the neural network learned only one branch, the plant output matched with the desired value in whole region. Through the simulation, the whole inverse characteristics of the plant seems to be folded into one branch when the neural network learns this branch. This behavior is called the neural network folding behavior in this paper.

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 are believing that a neural network learns an inverse dynamic of a target plant in a servo level neural network controller application.[1][2] This is because the target plant dynamics can be cancelled by this learned inverse dynamics if the neural network can learn it. That is, this cancellation means that a plant output completely matches with a desired value and we can realize an ideal control system. 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 which correspond to one output value. It is well known that there is no inverse function of such many-to-one function. However, many practical neural network controller applications have been successfully reported. Does the neural network learn the inverse dynamics of the plant on the practical applications? For this question, we studied that the neural network obtains the branch of the inverse characteristics of the plant (; a part of the inverse characteristics and it can be expressed as a one-to-one function mathematically) if the learned region is restricted to the one-to-one region.[3] If the learned region is not restricted, what happen? When we apply the neural network to the practical applications, the plant characteristics are unknown usually. Such restriction is impossible. For this question, we studied that the neural network learned the mean characteristics of the several plant inverse branches for off-line learning neural network controller.[3] However we did not study the detail of the on-line neural network controller for this problem yet.

Thus, this paper studies the above question through the use of a neural network direct controller with online learning. The reason to use this type controller is that it is simplest among the servo level neural network controllers. The sine function is selected as a target plant. This plant is static, but it is suitable for a basic study. Simulation results confirm folding behavior of the neural network mapping capability. This behavior is that the neural network learns only one branch of the inverse characteristics of the target plant in order to obtain whole plant output. This fact means that we can realize an ideal control system if the neural network can learn only one inverse branch of the plant. This is because the plant output can match with the desired value. 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.

II. NEURAL NETWORK STRUCTURE FOR TEST OF FOLDING BEHAVIOR

This section explains a neural network structure for the test of folding behavior. For this test, a following sine function is selected as a target plant.

$$Y(k) = \sin(U(k)) \tag{1}$$

where Y is the plant output, U is the plant input and k is the sampling number. The reason of this selection is that the sine function has smooth feature and it is easily mapped by a neural network. The reason of the discrete time system selection is to examine the discrete time control system as a future study. Since this paper selects the direct controller, the plant input U is composed of the following equation.

$$U(k) = \sum_{i=1}^{n} [\omega_i(p)\ f\{W_i(p)Y_d(k) + T_i(k)\}] \tag{2}$$

where ω_i is the i th element of the weight vector between the hidden layer and the output layer, Wi is the i th element of the weight vector between the input layer and the hidden layer, Ti is the i th element of the offset values added to the hidden layer neuron, n is the neuron number of the hidden layer, p is the learning number and f is the sigmoid function expressed by the following equation.

$$f(x)=\frac{X_g\{1-\exp(-4x/X_g)\}}{2\{1+\exp(-4x/X_g)\}} \tag{3}$$

where x is the input of the sigmoid function and Xg is the parameter which defines the sigmoid function shape. The scheme of the neural network direct controller and the structure of above neural network are shown in Fig.1 and Fig.2 respectively.

The output error ε and the cost function J are defined as follows:

$$\varepsilon(k)=Yd(k)-Y(k) \tag{4}$$

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

where Yd is the desired value for the control system and ρ is the sampling number within one learning period. The learning rule of this neural network controller is designed so as to minimize the cost function J. When we apply the δ rule to this learning rule, it is expressed as

$$\omega_i(p) = \omega_i(p+1) - \eta \frac{\partial J(p)}{\partial \omega_i(p)} \tag{6}$$

$$W_i(p) = W_i(p+1) - \eta \frac{\partial J(p)}{\partial W_i(p)} \tag{7}$$

$$T_i(p) = T_i(p+1) - \eta \frac{\partial J(p)}{\partial T_i(p)} \tag{8}$$

$$\frac{\partial J(p)}{\partial \omega_i(p)} = -\sum_{k=0}^{\rho} \{(\varepsilon(k)\ f(W_i(p)I_j(k-1) + T_i(p))\cos(Uk)\} \tag{9}$$

$$\frac{\partial J(p)}{\partial W_i(p)} = -\sum_{k=0}^{\rho} \{(\varepsilon(k)\ \omega_i(p) f'(W_i(p)Y_d(k) + T_i(p))Y_d(k)\cos(Uk)\} \tag{10}$$

$$\frac{\partial J(p)}{\partial T_i(p)} = -\sum_{k=0}^{\rho} \{(\varepsilon(k)\ \omega_i(p) f'(W_i(p)Y_d(k) + T_i(p))\cos(Uk)\} \tag{11}$$

where f' is the derivative of the sigmoid function and η is the parameter to determine the neural network learning speed.

III. SIMULATION

This paper selects the following sine wave as a desired value for control.

$$Y_d(k) = \sin(X_t(k)) \tag{12}$$

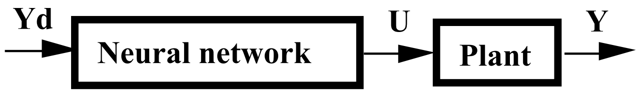

Fig.1 Scheme of neural network direct controller.

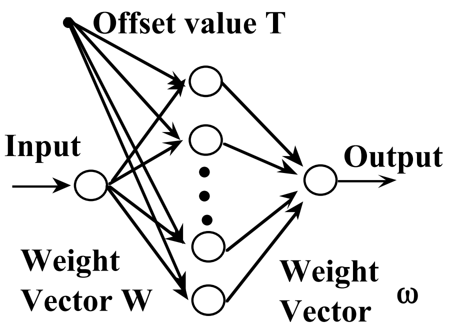

Fig.2 Structure of neural network.

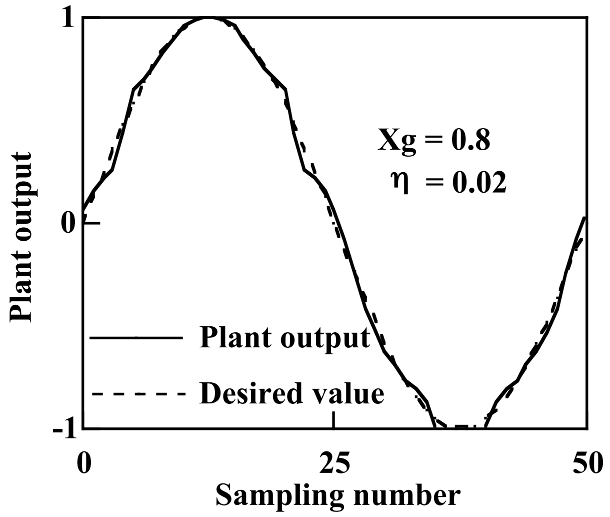

Fig.3 Learning result of example 1.

where Xt is the signal for the generation of the desired value. The sampling number within one learning period ρ =50 is also selected.

Figure 3 shows the learning result of the example 1. The solid line in this figure is the plant output and the broken line is the desired value. As shown here, the output error remains between the plant output and the desired value, but it is small and the plant output well match with the desired value. This fact means that the neural network learning well performs and our neural network can map the sine wave. Figure 4 shows the neural network output (equals the plant input). The solid line shows the neural network output. As shown in this figure, it is restricted from π/2 to 3π/2 although the plant output matches with the sine wave in one cycle as shown in fig.3. If the neural network can map the inverse characteristic of the sine wave from 0 to 2π, the neural network output should be 0 to 2π as shown in the broken line in fig.4. Fig.5 shows the neural network input-output relation of the example 1. The solid line is the learned neural network input-output relation. The broken line is the inverse characteristics of the plant from 0 to 2π. As shown in this figure, the neural network learns a part of the whole inverse characteristics and the learned part is restricted from π/2 to 3π/2. In this restricted region, the inverse characteristics is expressed as the one-to-one function. Such region is called the branch of the inverse characteristics in this paper. In other word, if the neural network learns only one branch, the plant output matches with the desired value in whole region. This feature of the neural network is called the folding behavior in this paper. This is because the whole inverse characteristics seems to be fold into one branch.

Figure 6 shows the learning result of the example 2. The solid line and the broken line are the plant output and the desired value respectively. As shown here, the neural network learning also performs well. Figure 7 shows the neural network output. The solid line is the neural network output and the broken line is the ideal characteristics if the neural network can map the whole inverse characteristics of the plant. As shown here, the neural network output is restricted from 3π/2 to 5π/2 although the plant output is similar to that of the example 1. Figure 8 shows the neural network input-output relation of the example 2. The solid line is the neural network input-output relation and the broken line is the ideal characteristics if the neural network can map the whole inverse characteristics of the plant. As shown here, the neural network input-output relation is restricted from 3π/2 to 5π/2. This fact means that the neural network learns the different branch from that of fig.5 of the example 1. The difference between the examples 1 and 2 is only initial neural network weight. That is, the initial neural network weight determines which branch is learned by the neural network and if the neural network learns any branches, the plant output can be matched with the desired value.

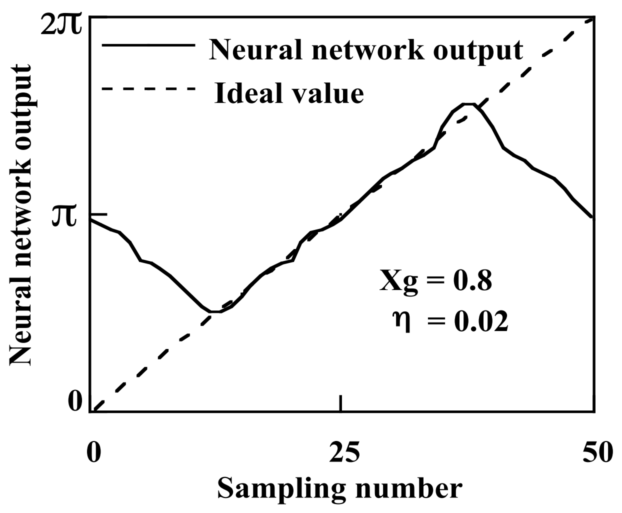

Fig.4 Neural network output of example 1.

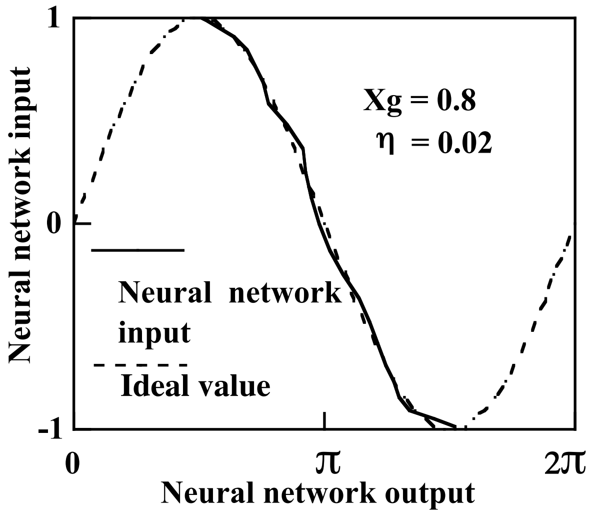

Fig.5 Neural network input-output relation of example 1.

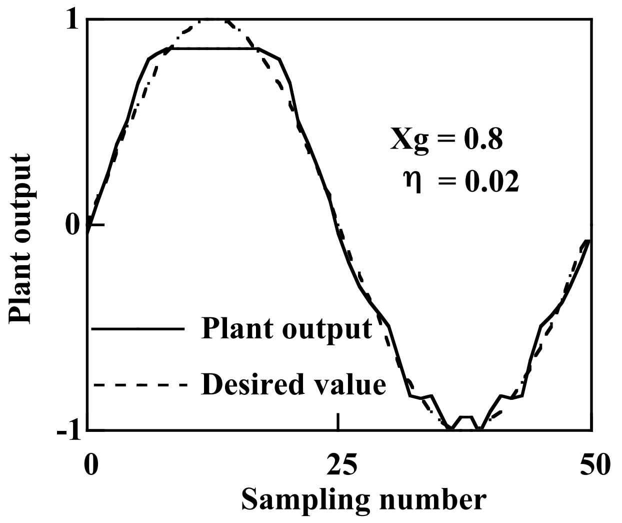

Fig.6 Learning result of example 2.

With regard to other remarkable feature of the neural network, the branches monotonously increase or decrease. This fact means that the derivative of the plant characteristics learned by the neural network also monotonously increases or decreases. As shown in eqs.(9)-(11), the leaning rules in this paper use the derivative of the plant characteristics. However, it may not be necessary. The reason of the requirement of the plant derivative is that we believe the neural network should converge to the whole plant inverse characteristics and its derivative changes in whole region usually. The value of this derivative is the petty problem. This is because it relates to the convergence speed and we can compensate through the use of the parameter η tuning in the learning rules. The sign of its derivative is the serious problem. This is because this relates to which the neural network converges or not. However, the above simulation results confirm that the sign of the slope of the plant inverse characteristics learned by the neural network is constant. This is because the signs of the branch derivatives are constant. That is, if the parameter η in eqs.(6)-(8) has the constant sign (plus or minus) and the small value, we can expect that the neural network converge to either branch although the derivatives of the plant inverse characteristics are removed in the learning rule. As shown in above simulation results, if the neural network converges to either branch, the plant output matches with the desired value and we can obtain the ideal control characteristics in whole region. This expectation will be shown in my future work.

IV. CONCLUSION

This paper studied what characteristics did the neural network learn through the use of the neural network direct controller. This is because we expect that the neural network learns the inverse dynamics of the object plant, but such inverse dynamics does not exist for the usual nonlinear plant. Simulation results confirmed that the neural network learned the branch of the inverse characteristics (; a part of the inverse characteristics and it can be expressed as a one-to-one function mathematically) and if the neural network learned only one branch, the plant output matched with the desired value. Through the simulation, the whole inverse characteristics of the plant seems to be folded into one branch when the neural network learns this branch. This behavior is called the neural network folding behavior in this paper.

ACKNOWLEDGMENT
The author wishes to express his thanks to Mr. Junpei Oota, graduated student, Ibaraki University, for his programming and simulation.

REFERENCES

[1]K.S.Narendra and K.Parthictsarathy, "Identification and Control of Dynamics Systems Using Neural Networks", IEEE Transactions on Neural Networks, 1-1, pp.4-27(1990)
[2]M.Kawato, "Computational Schemes and Neural Network Models for Formation and Control of Multijoint Arm Trajectory", Neural Networks for Control (W.T.Miller, R.S.Sutton, P.J.Werbos ed.), MIT Press, pp.197-228(1990)
[3]T.Yamada, "Remarks on Neural Network Controller for Inverse Dynamics of Many-to-One Plant", IAS'95 (1995 International IEEE/IAS Conference on Industrial Automation and Control; Emerging Technologies), Taipei, Taiwan, R.O.C., May 22-27,(1995)

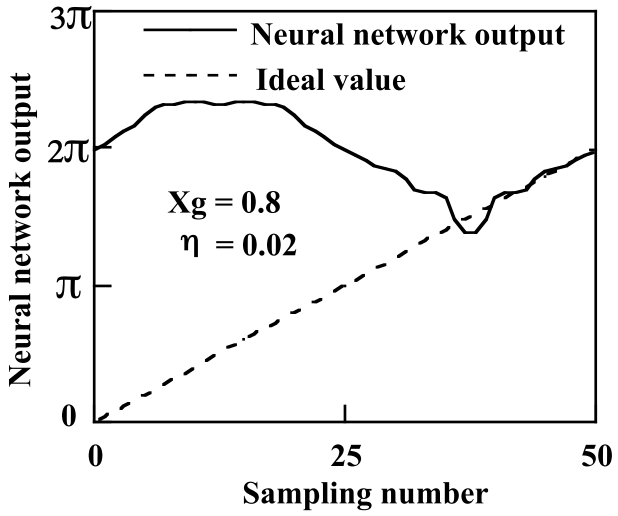

Fig.7 Neural network output of example 2.

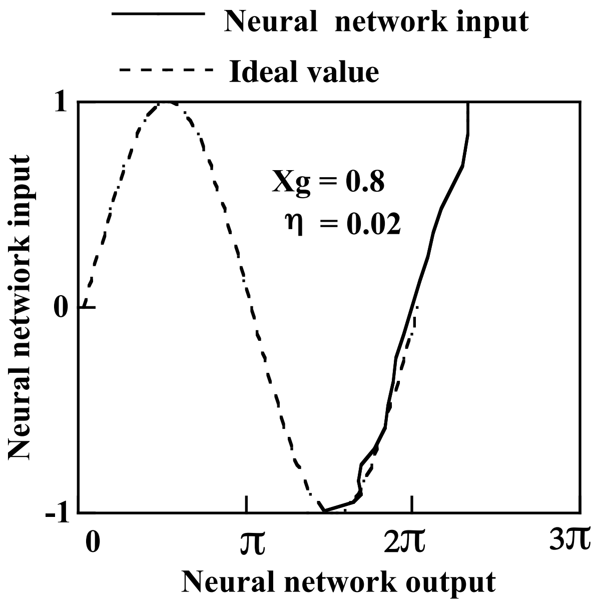

Fig.8 Neural network input-output relation of example 2