Transformation of neural network weight trajectories on 2D plane for learning type neural network direct controller

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

Abstract: Through the simulation of tracking method of neural network weight change on 2D plane, we noticed that it was hard for untrained users to observe the neural network weight performance on 2D plane in some cases. To overcome this problem, this paper applied a transformation of the neural network weight trajectories on 2D plane to the learning type neural network direct controller. The simulation results confirmed that if the trajectory of the neural network weight change on 2D plane had the simple structure, we could easily determine whether the leaning of the neural network was terminated. However, if it had more complex structure, we could not determine. The proposed transformation of the neural network weight trajectories to one dimensional values was useful for such cases

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. Although many neural network controllers have been proposed, we still have to tune several parameters of neural networks in order to obtain a better neural network learning performance in practical applications. A tracking method of neural network weight change has been proposed for better parameter tuning of the neural network controllers. This tuning method derives a weight vector from whole neural network weights. We can calculate an angle between this weight vector and a standard vector. The neural network weight change can be directly drawn on 2D plane by the use of the norm of the weight vector and the above angle. Drawn trajectories on 2D plane are not affected by the plant dynamics in comparison with the observation of the squared error between the neural network output and the plant output.

We have applied our tracking method to several types of the neural network controllers and confirmed its usefulness.[1][2 [3][4] However, we noticed through the simulation that it was hard for untrained users to observe the neural network weight performance on 2D plane in some cases. To overcome this problem, this paper proposes a transformation of the trajectories on 2D plane to one dimensional values. This transformation is to observe the velocity of weight change on 2D plane or its components. That is, when this velocity is almost zero, we can easily determine that the neural network learning is terminated in comparison with the observation of only neural network weight trajectories on 2D plane. The basic idea of this transformation was introduced by one example of the application to the learning type feedforward feedback neural network controller in my previous paper.[4] However, in order to obtain more accurate examination, we should apply it to more simple neural network controller.

Thus, this paper applies a transformation of the neural network weight trajectories on 2D plane to the learning type neural network direct controller and examines details of its performance. The reason of the choice of the direct controller is that this type controller has simplest structure and we can eliminate the effect of the feedback loop. Simulation results introduce the cases such that it is hard for untrained users to observe the neural network weight performance on 2D plan. Next, we confirm that our transformation of the neural network weight trajectories is useful for these cases.

II. TRACKING METHOD OF NEURAL NETWORK WEIGHT CHANGE

This section briefly explains the tracking method of the neural network weight change. This tracking method is applied to the learning type neural network direct controller for the SISO plant. Here, an output layer of the neural network has one neuron, the weights between the output layer and a hidden layer can be expressed as a vector ω and the weights between the hidden layer and the input layer can be expressed as a matrix W. For simplicity, the neuron number of the input layer is equal to that of the hidden layer. That is, the weight matrix W is a square matrix. The tracking method uses the following steps.

(Tracking method of neural network weight change)

(1) We can derive one weight vector Γ from the neural network weight vector ω and weight matrix W as follows:

$$\Gamma^T = [\omega_1 \cdots \omega_n \; W_{11} \cdots W_{1n} W_{21} \cdots W_{2n} \cdots W_{n1} \cdots W_{nn}] \quad (1)$$

where n is the neuron number both the input layer and the hidden layer.

(2)We must define a standard vector Γ_0. Any vector which is of same order as that of the weight vector Γ, can be selected as the standard vector. For example, the weight vectors derived from the initial neural network weights, the final neural network weights and so on.

(3) We can calculate an inner product of the weight vector Γ and the standard vector Γ_0 because these vectors are of the same order. We can also calculate the angle between the weight vector Γ and the standard vector Γ_0 as follows:

$$X = |\Gamma| \cos\theta, \quad Y = |\Gamma| \sin\theta \quad (2)$$

$$\theta = \cos^{-1}\left(\frac{<\Gamma_0 \cdot \Gamma>}{|\Gamma_0| \cdot |\Gamma|}\right) \quad (3)$$

where $<\Gamma_0 \cdot \Gamma>$ is the inner product between the vector Γ_0 and the vector Γ, and $|\Gamma|$ is the norm of the vector Γ.

(4) We can draw a new weight performance on the 2D plane by the use of X and Y in equations (2) and (3).

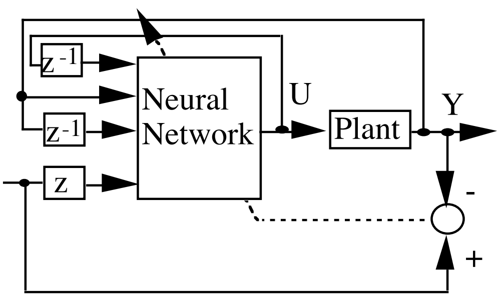

Fig.1 Block diagram of neural network direct controller for second order discrete time plant.

III. SIMULATION

This paper applies the tracking method of the neural network weight change to the learning type neural network direct controller. The simulated plant is follows:

$$Y(k) = -a_1 Y(k-1) - a_2 Y(k-2) + U(k-1) + bU(k-2) - a_3 Y(k-3) + C_{non} Y^2(k-1) \quad (4)$$

where Y(k) is the plant output, U(k) is the plant input, k is the sampling number, a_1, a_2 and b are the plant parameters, a_3 is the parasite term, and C_{non} is a 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. The output error ε and the cost function J are defined as follows:

$$\varepsilon(k) = Yd(k) - Y(k) \quad (5)$$

$$J(p) = \frac{1}{2} \sum_{k=1}^{\rho} \varepsilon^2(k) \quad (6)$$

where p is the trial number and ρ is the sampling number within one trial period.

For this simulated plant, the neuron number n in both the input and the hidden layers is 4. The neural network input vector I is defined as the following equation.

$$I^T(k) = [Y_d(k+1)\; Y(k)\; Y(k-1)\; U(k-1)] \quad (7)$$

We select the following sigmoid function f(x) as the input-output relation of the hidden layer neuron.

$$f(x) = \frac{X_g\{1 - \exp(-4x/X_g)\}}{2\{1 + \exp(-4x/X_g)\}} \quad (8)$$

where Xg is the parameter which defines the sigmoid function shape. The neural network output U(k) is composed as follows:

$$U(k) = \omega^T(p) f\{W(p) I(k)\} \quad (9)$$

The block diagram of the learning type neural network direct controller is shown in Fig.1. 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

$$W_{ij}(p+1) = W_{ij}(p) + \eta \sum_{k=1}^{\rho} [\varepsilon(k)\omega_i(p) I_j(k-1) f'\{\sum_{j=1}^{n} W_{ij}(p) I_j(k-1)\}] \quad (10)$$

$$\omega_i(p+1) = \omega_i(p) + \eta f[\sum_{j=1}^{n} \{W_{ij}(p) \sum_{k=1}^{\rho} (\varepsilon(k) I_j(k-1))\}] \quad (11)$$

where η is the parameter to determine the neural network learning speed. We select the weight vector derived from the initial neural network weights as the standard vector Γ_0 of the equations (2) and (3)

In order to realize the easier expression of the neural network weight performance for untrained users, this paper proposes a transformation of the trajectories on 2D plane to one dimensional values as follows:

$$Vwx(p) = X(p+1) - X(p) \quad (12)$$

$$Vwy(p) = Y(p+1) - Y(p) \quad (13)$$

$$Vw(p) = \sqrt{(Vwx(p))^2 + (Vwy(p))^2} \quad (14)$$

As shown in eq.(14), Vw in eq.(14) is the velocity of neural network weight change on 2D plane. Vwx and Vwy show the X and Y components of the velocity Vw respectively That is, when the velocity Vw is almost zero or both its X and Y components Vwx and Vwy are almost zero, we can easily determine that the neural network learning is terminated. Eqs.(12)-(13) is our proposed method.

Fig.2 shows the example 1 of the trajectory of the neural network weight change on 2D plane. It can be drawn through the use of eqs.(2) and (3). It has the simple structure and we can easily observe that the neural network weights convergence as shown here. Figs.3 and 4 show the example 2 of the trajectory of the neural network weight change on 2D plane and its expansion respectively. This example of the trajectory has more complex structure and it is difficult to observe the neural network weight performance on 2D plane. If we can expand the example 2 as shown in fig.4, we can observe the fine structure of the neural network weight performance, but this work requires big effort for the untrained users.

Fig.5 shows the velocity Vw of the neural network weight change on 2D plan for example 2. As shown here, Vw has the simple structure and the untrained users can easily determine whether the learning of the neural network is terminated or not. Vw is similar to the cost funtion J, but they are different because the cost function is affected by the plant dynamics. We can not observe the large vibration of the neural network weight change on 2D plane through the use of Vw. If this information is important for some applications, Vwx and Vwy in eqs.(12) and (13) are useful.

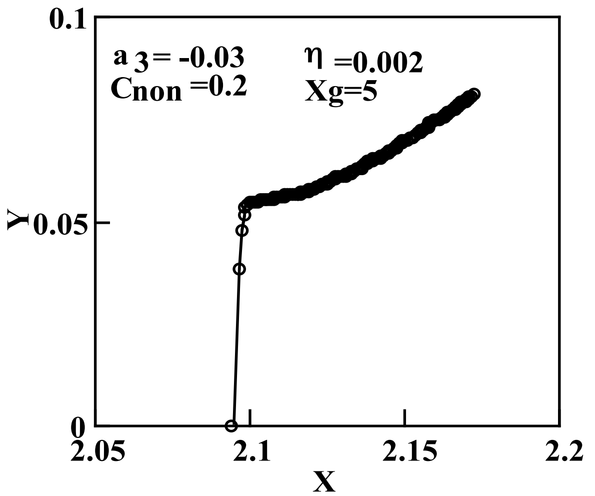

Fig.2 Example 1 of trajectory of neural network weight change.

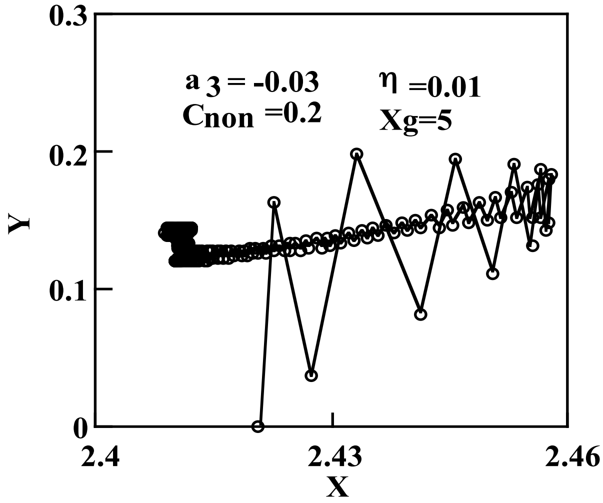

Fig.3 Example 2 of trajectory of neural network weight change.

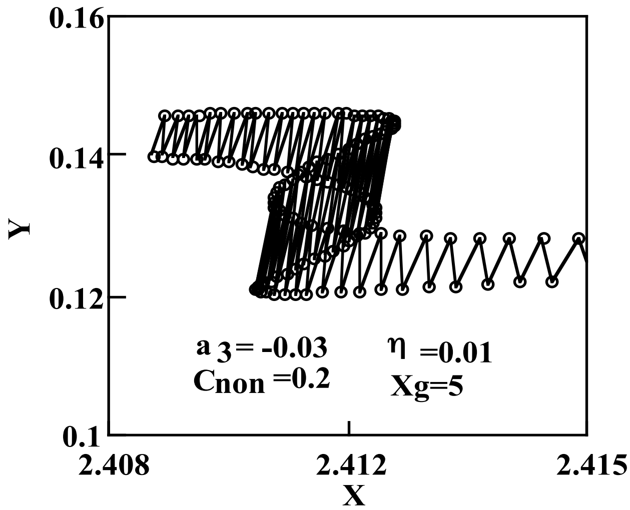

Fig.4 Expansion of example 2.

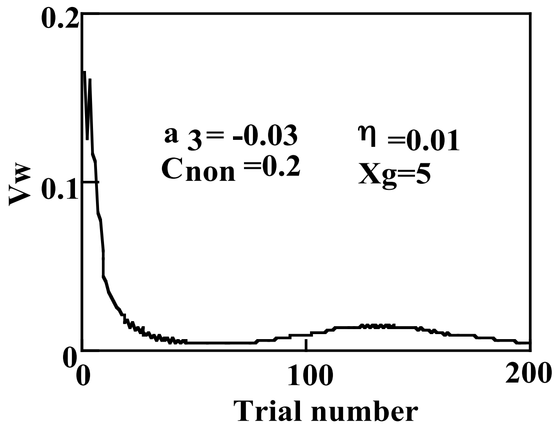

Fig.5 Vw for example 2.

Figs.6 and 7 show Vwx and Vwy for example 2 respectively. As shown in these figures, we can observe the vibration of the neural network weight change and we can easily determine whether the learning of the neural network is terminated or not. Fig.8 shows the example 3 of the trajectory of the neural network weight change on 2D plane. It is another example which has more complex structure. However, Vw, Vwx and Vwy are also useful for this more complex structure.

IV. CONCLUSION

This paper applied the transformation of the neural network weight trajectories on 2D plane to the learning type neural network direct controller and the simulation results examine details of its performance. They confirmed that if the trajectory of the neural network weight change on 2D plane had the simple structure, we could easily determine whether the leaning of the neural network was terminated. However, if it had more complex structure, we could not determine. The transformation of the neural network weight trajectories to one dimensional values was useful for such cases. This is because these one dimensional values have simple structure and the untrained users can easily observe the neural network weight performance through the use of these values.

ACKNOWLEDGMENT

The author wishes to express his thanks to Mr.Taka-aki Kuji graduated student, Ibaraki University, for his programming and simulation.

REFERENCES

[1] Takayuki Yamada, "Remarks on tracking method of neural network weight change for learning type neural network direct controller", Proceedings of AROB 10th '05 (The Tenth International Symposium on Artificial Life and Robotics 2005), pp.624-627(2005)
[2] Takayuki Yamada, "Remarks on tracking method of neural network weight change for adaptive type neural network direct controller", Proceedings of AROB 11th '06 (The Eleventh International Symposium on Artificial Life and Robotics 2006), GS19-3 (2006)
[3]Takayuki Yamada, "Remarks on tracking method of neural network weight change for adaptive type neural network feedforward feedback controller", Proceeding of AROB13th '08 (The Thirteenth International Symposium on Artificial Life and Robotics), pp.559-562(2008)
[4] Takayuki Yamada, "Remarks on tracking method of neural network weight change for learning type neural network feedforward feedback controller", Proceeding of AROB14th '09 (The Fourteenth International Symposium on Artificial Life and Robotics), pp.389-392(2009)

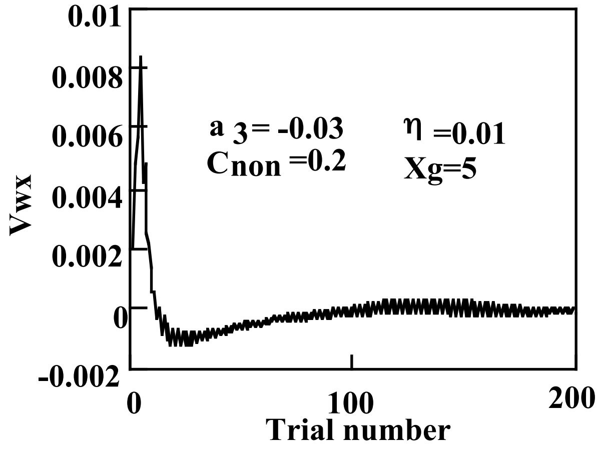

Fig.6 Vwx for example 2.

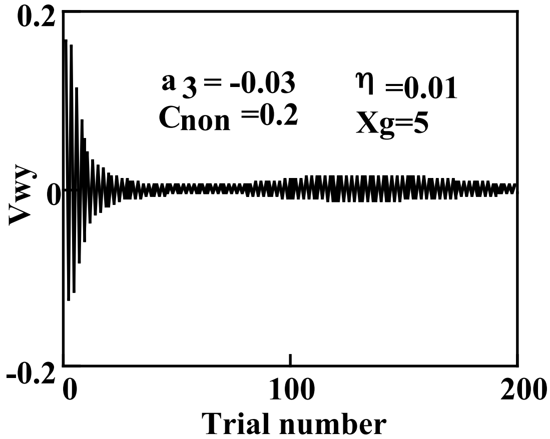

Fig.7 Vwy for example 2.

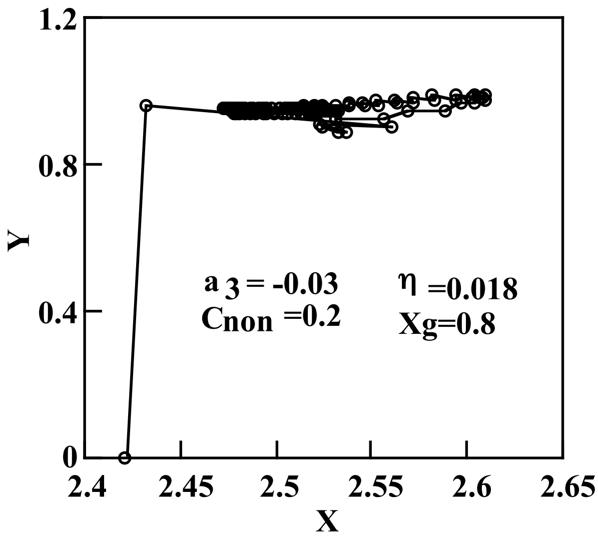

Fig.8 Example 3 of trajectory of neural network weight change.