Remarks on tracking method of neural network weight change 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

A neural network usually learns so as to minimize a scalar value such as a cost function. This scalar value is useful for a confirmation of neural network learning performance. However, this confirmation may not be correct for neural network controllers because a plant dynamics affects the cost function. This paper proposes a new tracking method of neural network weight change. The proposed tracking method can provide a new confirmation of the neural network learning performance.

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][2]. A neural network controller is usually designed so as to minimize the error between a plant output (or neural network output) and a desired output (teaching signal). For this aim, neural network learning rules are designed to change neural network weights whose number reaches into thousands or tens of thousands in some applications. The reason to use these huge number of weights is that a biological neural network has huge number of neurons and it is proved that more neurons realize more accurate nonlinear mapping capability of the neural network. As mentioned above, the essence of neural network learning is nothing but the change of the neural network weights. However, in order to examine the performance of the neural network learning, most researchers use a cost function (squared error between the desired output and the neural network output (or the plant output)). This is because it is not practical to examine the huge number of the neural network weights and the cost function is a scalar value which is easily dealt with. However, the neural network weight change may not be reflected in the cost function. This problem is especially serious in neural network controller applications. This is because the performance of the cost function is affected by dynamics of the plant. This fact leads that more accurate examination of the neural network controller learning performance requires to track the neural network weight change directly.

This paper proposes a new tracking method of the neural network weight change. A leaning type neural network direct controller[3] for a second order discrete time plant is selected in order to examine the proposed tracking method and its simulation results show the usefulness of the proposed method.

2. Tracking method of neural network weight change

This section proposes the tracking method of the neural network weight change and its application to the learning type neural network direct controller. For our tracking method, first, one weight vector is derived from the neural network weights. Next, we calculate an inner product of this weight vector and a standard vector. Any vector, which has same order as that of the weight vector, can be selected as this standard vector, for example, the weight vectors derived from the initial neural network weights, the final neural network weights and so on. We can also calculate an angle between the weight vector and the standard vector. The track of the neural network weight change can be drawn on a 2D plane through the use of these calculated inner product and angle. This track does not show whole neural network weight change, but it is not affect by the plant dynamics and it can show an another characteristic of the neural network learning performance. We can realize the new examination of the neural network learning through the use of the proposed tracking method or its combination with the cost function.

In order to verify the usefulness of the tracking method, we applied it to the learning type direct controller.

A reason of this selection is that the direct controller is simplest. The another reason is that the cost function of the learning type is the sum of the squared error at each sampling time and the plant dynamics less affect it in comparison with an adaptive type. This fact is useful to examine the proposed tracking method effectiveness. We also select a discrete time SISO (single input and single output) plant as an object plant because it is simplest and useful for a practical controller application. When we use above selections, 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 an input layer can be expressed as a matrix W. To simplify, the neuron number of the input layer is equal to that of the hidden layer. That is, the weight matrix W is the square matrix.

We can derive a new weight vector Γ from these neural network weight vector and matrix 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. When we define the standard vector Γ_0 , the track of the neural network weight change on the 2D plane can be expressed as the following equations.

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

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

Where $<\Gamma_0 \;\; \Gamma>$ is the inner product between the vector Γ_0 and the vector Γ, and $|\Gamma|$ is the norm of the vector Γ. As mentioned above, we can draw a new weight performance on the 2D plane by use of X and Y in equations (2) and (3). The plant dynamics does not affect this weight performance directly.

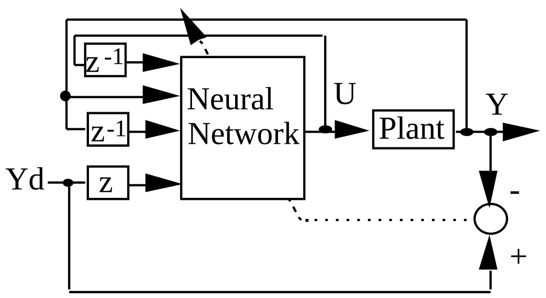

Fig.1 Block diagram of learning type neural network direct controller for second order discrete time plant.

3. Simulation

To verify the usefulness of the proposed tracking method, it is applied to the learning type neural network direct controller for the second order discrete time plant. The simulated plant is follows:

$$Y(k) = -a_1Y(k-1) - a_2Y(k-2) + U(k-1) + bU(k-2) - a_3Y(k-3) + C_{non}Y^2(k-1) \quad (4)$$

Where Y(k) is the plant output, U(k) is the plant input, k is the sampling number, a_1, a_2 & 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. The output error ε and the cost function J(p) of the trial number p are defined as follows:

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

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

where ρ is the sampling number within one trial period. In this simulation, ρ=300 is selected.

For this simulated plant, the neuron number n in both the input and 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.

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

Where Xg is the parameter which defines the sigmoid function shape. The plant input U(k) equals the neural network output 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 shown in the following equations.

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

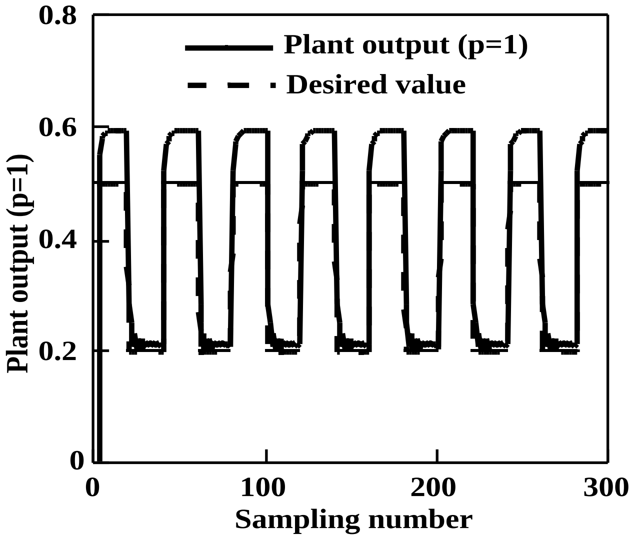

Fig.2 Plant output (p=1).

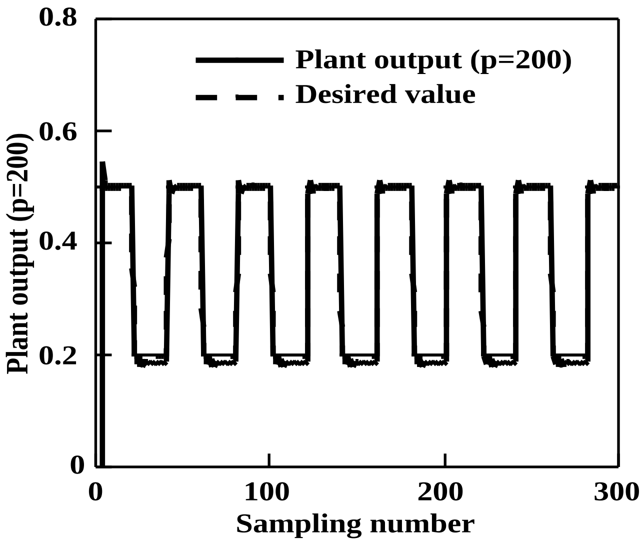

Fig.3 Plant output (p=200).

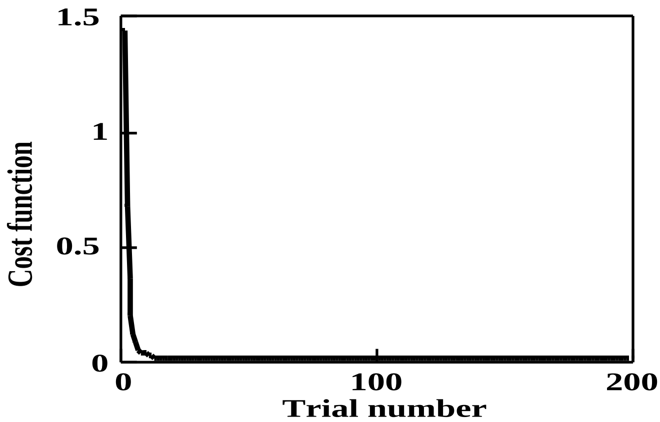

Fig.4 Cost function.

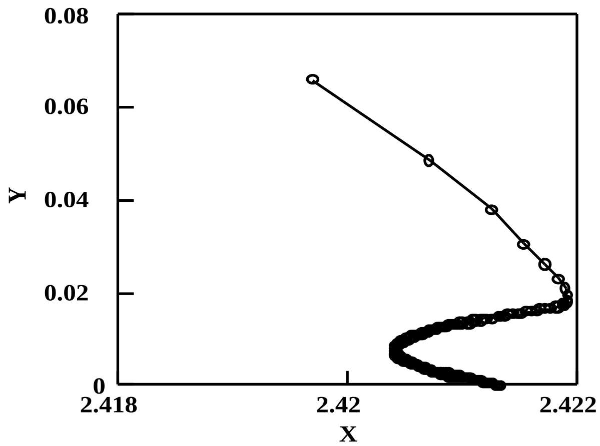

Fig.5 Track of neural network weight change (p=1~200).

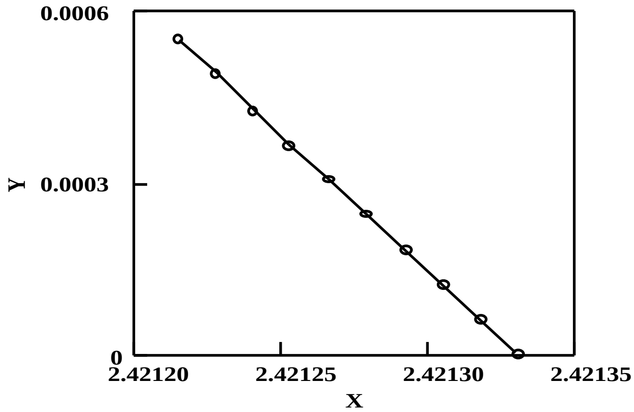

Fig.6 Track of neural network weight change (p=191~200).

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

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

Figure 2 shows an example of the plant output (p=1) using an initial neural network weight. The solid line and the dotted line show the plant output and the desired value respectively. Figure 3 shows the final plant output (p=200). As shown in these figures, the neural network learning well performs and the plant output converges to the desired value. Figure 4 shows the cost function with regard to the trial number. As shown here, it appears that the neural network weights do not change after several tens of trials and the neural network learning is completely finished. Figure 5 shows the track of the neural network weight change (p=1~200) through the use of the proposed tracking method. As shown in this figure, the neural network weights change relatively large from the first trail p=1 to six trials p=6. After p=6, they change continuously and this change is not finished at p=200. To verify this, figure 6 shows the expansion of fig.5 which is the track of the neural network weight change (p=191~200). As shown here, we confirm that the neural network weights continuously change at p=200. That is, the neural network learning is not finished yet at p=200. This fact can not be observed by use of the cost function shown in fig.4 and the proposed tracking method is useful for the neural network performance examination. Figure 7 shows an another example of the cost function. In this example, it also appears that the neural network learning is finished within tree trials. Figure 8 and 9 shows the track of the neural network weight change (p=1~200) and its expansion (p=191~200) respectively. The neural network weights change continuously and the neural network learning is not finished yet.

As mentioned above, the proposed tracking method is useful to track the neural network weight change on 2D plane. This track shows another feature of the neural network learning performance.

4. Conclusion

This paper proposed the new tracking method of the neural network weight change. It was applied to the learning type neural network direct controller and simulated. The simulation results showed the usefulness of the proposed tracking method and it could be observed that the neural network weights were continuously changed in some cases although the neural network learning appeared to be finished. The combination of the cost function and the proposed tracking method is useful to examine the neural network learning performance more accurately.

Acknowledgment

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

References

[1]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

[2]D.Psaltis, A.Sideris and A.Yamamura, "Neural Network Controller", Proceedings of 1987 IEEE International Conference on Neural Networks, Vol.IV, pp.551-558, San Diego, 1987

[3]Takayuki Yamada and Tetsuro Yabuta, "Nonlinear Neural Network Controller for Dynamic System", Proceedings of IECON'90 (16th Annual Conference of IEEE Industrial Electronics Society), pp.1244-1249(1990)pp.295-302, 1994

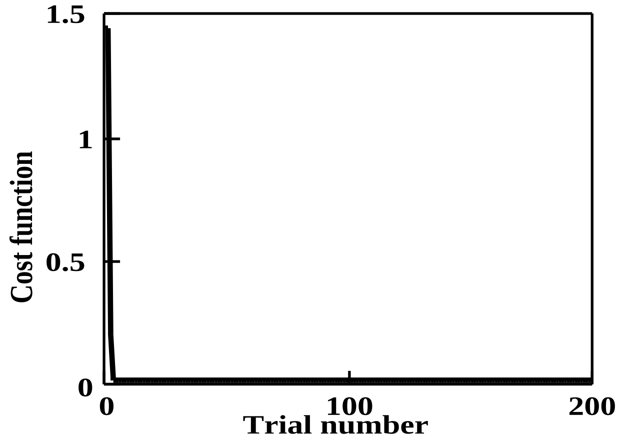

Fig.7 Cost function.

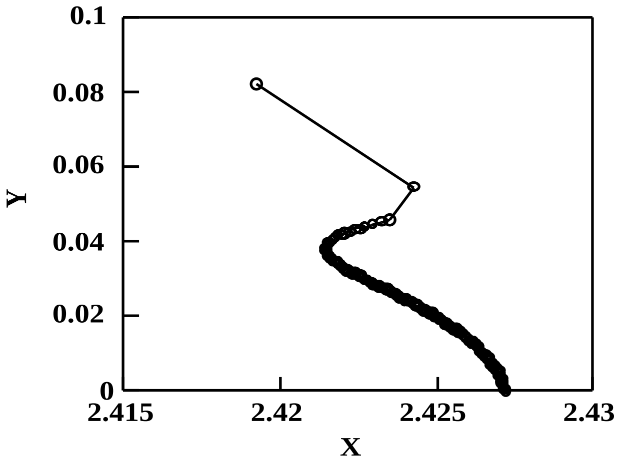

Fig.8 Track of neural network weight change (p=1~200).

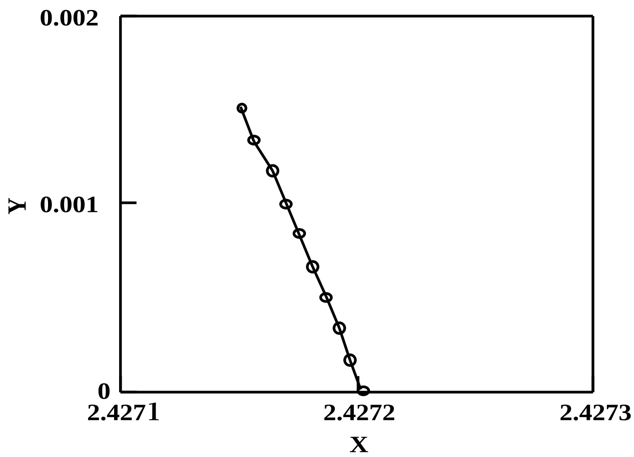

Fig.9 Track of neural network weight change (p=191~200).