'94 KACC (1994. 10. 17 ~ 20)

Suboptimal Control for DC Servomotor Using Neural Network

Hiroaki Kawabata*, Masayuki Yoshizawa**, Keiji Konishi† and Yoji Takeda**

* Department of Communication Engineering, Faculty of Computer Science and System Engineering, Okayama Prefectural University
111, Kuboki, Soja, Okayama, 719-11 Japan
**Department of Electrical and Electronic Systems, College of Engineering, University of Osaka Prefecture
1-1, Gakuen-cho Sakai, Osaka, 593 Japan
†Department of Electrical Engineering, Nara National College of Technology, 22, Yata-cho, Yamato-kohriyama, Nara, 639-11 Japan

Abstract

This paper proposes a method of suboptimal control for DC servomotor using a neural network. First we consider a nonlinear observer which is constructed by using an approximated linear dynamics of the nonlinear system and a neural network. The reccurent neural network is used for the learning of the dynamical system. Next we consider the nonlinear observer. Then, we apply the observer output to nonlinear optimal regulator and confirm the effectiveness by applying the method to the inverse pendulum system.

1. Introduction

Over the past few years a considerable number of studies have been made on the neural networks applying to the various engineering problem,[1)~7)] especially to the control problem, and a large number of studies have been made on the nonlinear optimal control likewise. In this paper we propose a method of suboptimal control in nonlinear system with some unmeasured state variables. We meet some problems whenever we consider the nonlinear control problems. Here we consider the two problems among them.

One is the nonlinear observer, and the other is the nonlinear optimal regulator.For the observer we consider the nonlinear observer which is constructed by using an approximated linear dynamics, then we devise the nonlinear obsever which reflects the nonlinear characteristics. The reccurent neural network which is useful for the learning of the dynamical system or for the online learning is used.

For the regulator we use a nonlinear optimal regulator using a Lyapunov-like function that we previously proposed.[8)] We apply the nonlinear observer output to the nonlinear optimal regulator and confirm the effectiveness by applying the method to the inverse pendulum system.

2. Nonlinear System and observer

We consider a dynamical system in which the nonlinearity exists on the state vector.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + B\boldsymbol{u} \tag{1}$$

where $\boldsymbol{x}$ is an n-dimensional state vector and $\boldsymbol{u}$ is an r-dimensional control vector. $\boldsymbol{f}(\boldsymbol{x})$ is a nonlinear vector-valued function and B is an n × r matrix. We rewrite Eq.1 as follows;

$$\dot{\boldsymbol{x}} = A_i\boldsymbol{x} + B\boldsymbol{u} + (\boldsymbol{f}(\boldsymbol{x}) - A_i\boldsymbol{x}) \quad (i = 1, m) \tag{2}$$

where A_i is a linearized matrix which most suitably approximates the nonlinear function $\boldsymbol{f}(\boldsymbol{x})$ at the operating state.

We consider the nonlinear observer shown in Fig.1 in which the difference between the nonlinear term and linear one is compensated by using a neural network.

In Fig.1 the system dynamics is described by a discrete one. The neural network plays the role of compensator for the nonlinear factor. When a test input $\boldsymbol{u}_t$ is given, we assume that the estimated state by the linearized observer is presented by Eq.3.

$$\tilde{\boldsymbol{x}}_{t+1} = A_i\hat{\boldsymbol{x}}_t + \boldsymbol{b}\boldsymbol{u}_t \tag{3}$$

where $\hat{\boldsymbol{x}}_t$ shows the estimated state using the obsereved data until the time t.

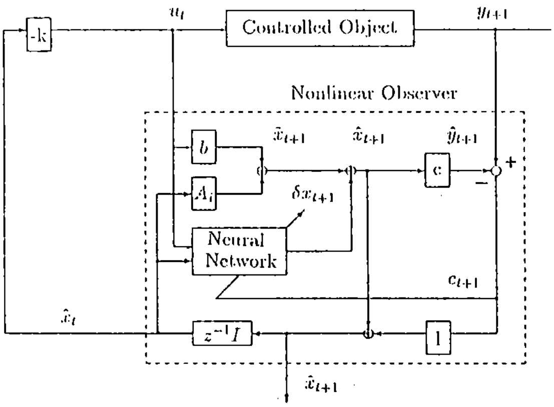

Fig.1 Nonlinear observer using linear approximation

When the output of controlled object, $\boldsymbol{y}_{t+1}$ is obtained, the estimated state $\hat{\boldsymbol{x}}_{t+1}$ is given by the output $\delta\boldsymbol{x}_{t+1}$ of the neural network.

$$\hat{\boldsymbol{x}}_{t+1} = \tilde{\boldsymbol{x}}_{t+1} + \delta\boldsymbol{x}_{t+1} \tag{4}$$

$$\hat{\boldsymbol{y}}_{t+1} = \boldsymbol{c}\hat{\boldsymbol{x}}_{t+1} \tag{5}$$

We define the error $\boldsymbol{e}_{t+1}$ between the output $\boldsymbol{y}_{t+1}$ and $\hat{\boldsymbol{y}}_{t+1}$

$$\boldsymbol{e}_{t+1} = \boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1} \tag{6}$$

Then, the estimated state of the controlled object is given by the following equation.

$$\hat{\boldsymbol{x}}_{t+1} = \tilde{\boldsymbol{x}}_{t+1} + \boldsymbol{l}(\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_t) \tag{7}$$

where $\boldsymbol{l}$ is the coefficient vector to hasten the convergence.

To derive the learning algorithm of the neural network, we define the next error function.

$$J_{t+1} = \frac{1}{2}(\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1})^2 = \frac{1}{2}(\boldsymbol{y}_{t+1} - \boldsymbol{c}\hat{\boldsymbol{x}}_{t+1})^2 \tag{8}$$

Then the modified amount of the coefficient is given by the following.

$$\delta w_{t+1} = -\alpha\frac{\partial J_{t+1}}{\partial w} = (\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1})\boldsymbol{c}\frac{\partial \hat{\boldsymbol{x}}_{t+1}}{\partial w} \tag{9}$$

where α is a learning constant. From the next relations we have Eq.13.

$$\frac{\partial \hat{\boldsymbol{x}}_{t+1}}{\partial w} = \frac{\partial \tilde{\boldsymbol{x}}_{t+1}}{\partial w} + \frac{\partial g(\hat{\boldsymbol{x}}_t, u_t)}{\partial w} \tag{10}$$

$$\frac{\partial \tilde{\boldsymbol{x}}_{t+1}}{\partial w} = A_i\frac{\partial \hat{\boldsymbol{x}}_t}{\partial w} + \boldsymbol{b}\frac{\partial u_t}{\partial w} \tag{11}$$

$$\frac{\partial \tilde{\boldsymbol{x}}_{t+1}}{\partial w} = A_i\frac{\partial}{\partial w}[\hat{\boldsymbol{x}}_t + \boldsymbol{l}(\boldsymbol{y}_t - \boldsymbol{c}\hat{x}_t)] = A_i(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{c})\frac{\partial \hat{x}_t}{\partial w} \tag{12}$$

$$\frac{\partial \hat{\boldsymbol{x}}_{t+1}}{\partial w} = A_i(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{c})\frac{\partial \hat{x}_t}{\partial w} + \frac{\partial g(\hat{\boldsymbol{x}}_t, u_t)}{\partial w} \tag{13}$$

Substituting the Eq.13 into Eq.9 we get the eventual modified amount of the coefficient

$$\begin{aligned}
&\delta w_{t+1} \\
&= (\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1})\boldsymbol{c}\frac{\partial \hat{\boldsymbol{x}}_{t+1}}{\partial w} \\
&= (\boldsymbol{y}_{t+1} - \hat{\boldsymbol{y}}_{t+1})\boldsymbol{c}A_i(\boldsymbol{I} - \boldsymbol{l}\boldsymbol{c})\frac{\partial \hat{x}_t}{\partial w} + \frac{\partial g(\hat{\boldsymbol{x}}_t, u_t)}{\partial w}
\end{aligned} \tag{14}$$

We can also consider the nonlinear observer illustrated in Fig.2.

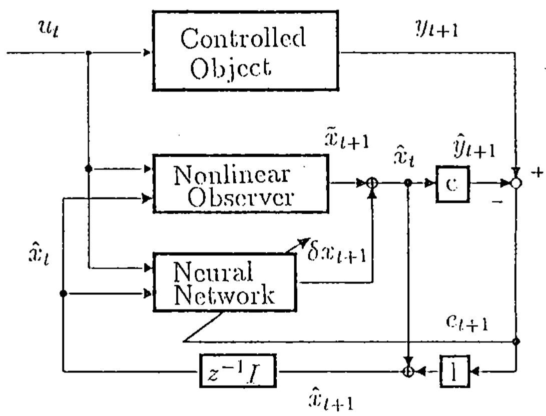

Fig.2 Nonlinear observer

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}) + B\boldsymbol{u} + K(\boldsymbol{y} - C\hat{\boldsymbol{x}}) \tag{15}$$

where $\boldsymbol{y} = C\boldsymbol{x}$ is the output vector which is observed.

Provided the error vector $\boldsymbol{e}$

$$\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}} \tag{16}$$

$$\dot{\boldsymbol{e}} = KC\boldsymbol{e} + f(\hat{\boldsymbol{x}}) - f(\hat{\boldsymbol{x}}) \tag{17}$$

By compensating the difference of nonlinear term and linear one by the neural network, the nonlinear observer is constructed.

3. Reccurent Neural Network

When we apply a neural network to the control problem of a dynamic system, we meet the problem that a conventional neural network is able to learn only the input-output relations as an algebraic mapping. To learn the dynamics of the system, we use the neural network which has the reccurent connection between the hidden units. We call it the reccurent neural network. In the neural network shown in Fig.3, the units of input layer get the external input.

$$I_k(t) = u_k(t) \tag{18}$$

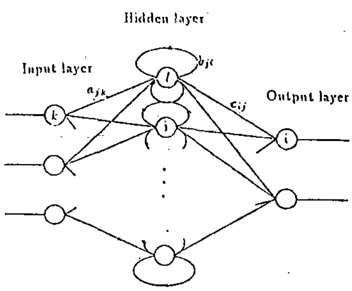

Fig.3 Reccurent neural network

The units of the middle layer get the signals from the input units and the signals from the one step former middle layer, and send the output $H_j(t)$.

$$T_{Hj}(t) = \sum_{k=1}^{m} a_{jk}u_k(t) + \sum_{i=1}^{r} b_{jl}H_l(t-1) \tag{19}$$

$$H_j(t) = f_H(T_{Hj}(t)) \tag{20}$$

The output unit get the output $T_{Oi}(t)$ from the middle units as an input, and send $y_i(t)$ as an output of the neural network.

$$T_{Hj}(t) = \sum_{j=1}^{n} c_{ij}H_j(t) \tag{21}$$

$$y_i(t) = O_i(t) = f_o(T_{oi}(t)) \tag{22}$$

The relation of the inputs and outputs in the whole neural network is given as follows.

$$y_i(t) = f_o(\sum_{j=1}^{n} c_{ij}f_H(\sum_{k=1}^{m} a_{jk}u_k(t) + \sum_{i=1}^{r} b_{jl}H_l(t-1))) \tag{23}$$

To operate the neural network, we must calculate the derivative of the error function for the coupling coefficient, and there are two methods for it. In the method to have the derivative of the error function for the coupling coefficient using directly the derivative of complex function, we get it by using the next derivative of the input for the coupling coefficient.

$$B_{jpq}(t) = \frac{\partial T_{Hj}(t)}{\partial b_{pq}}, A_{jpq}(t) = \frac{\partial T_{Hj}(t)}{\partial a_{pq}} \tag{24}$$

that is, the same error signal given in the hierarchical neural network.

$$\frac{\partial E_r(t)}{\partial T_{oi}(t)} = f_o'(T_{oi}(t))\frac{\partial E_r(t)}{\partial x_j(t)} = \hat{\delta}_{oi}(t) \tag{25}$$

$$\frac{\partial E_r(t)}{\partial T_{Hj}(t)} = f_H'(T_{Hj}(t))\sum_{i=1}^{n}\hat{\delta}_{oi}(t)c_{ij} \tag{26}$$

By using the above equation.

$$\frac{\partial E_r(t)}{\partial c_{pq}(t)} = \hat{\delta}_{op}(t)H_q(t) \tag{27}$$

$$\frac{\partial E_r(t)}{\partial b_{pq}(t)} = \sum_{j=1}^{r}\hat{\delta}_{Hj}(t)B_{jpq}(t) \tag{28}$$

$$\frac{\partial E_r(t)}{\partial a_{pq}(t)} = \sum_{j=1}^{r}\hat{\delta}_{Hj}(t)A_{jpq}(t) \tag{29}$$

And we have the following equations.

$$B_{jpq}(t) = H_q(t-1)\delta_{pj} + \sum_{l=1}^{r} b_{jl}f_H'(T_{Hl}(t-1))B_{lpq}(t-1) \tag{30}$$

$$A_{jpq}(t) = u_q\delta_{pj} + \sum_{l=1}^{r} b_{jl}f_H'(T_{Hl}(t-1))A_{lpq}(t-1) \tag{31}$$

This method is the most effective for the online learning neural network.

4. Optimal Regulator using Lyapunov-like Function

Although differntial equations describing the actual behavior of a dynamical system are generally nonlinear, the linear optimal control is effective when the operation of the system is restricted to a small region around a chosen operating point. But, in an inherent nonlinear system which shows strong nonlinearity, the optimal control often fails, and it is desirable to design the nonlinear control method for the wide range operation. Here we show a method of suboptimal control for a nonlinear dynamical system using a neural network.[8] We rewrite the next nonlinear system equation.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + B\boldsymbol{u} \tag{32}$$

We consider the performance criterion Eq.33.

$$J = \int_0^\infty [q(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{u}^t R\boldsymbol{u}]dt \tag{33}$$

$$V(\boldsymbol{x}, t) = \min_{\boldsymbol{u}} \int_t^\infty [q(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{u}^t R\boldsymbol{u}]dt \tag{34}$$

If we assume that the minimum value of Eq.34 is finite, the steady state Hamilton-Jacobi-Bellman equation becomes the following equation.

$$\min_{\boldsymbol{u}}[q(\boldsymbol{x} + \frac{1}{2}\boldsymbol{u}^T R\boldsymbol{u} + V_{\boldsymbol{x}}(\boldsymbol{x}, t)^T\{\boldsymbol{f}(\boldsymbol{x}) + B\boldsymbol{u}\}] = 0 \tag{35}$$

Then, the optimal control which satisfies Eq.35 is given by Eq.36.

$$\boldsymbol{u} = -R^{-1}B^T V_{\boldsymbol{x}}(\boldsymbol{x}, t) \tag{36}$$

where $V_{\boldsymbol{x}}(\boldsymbol{x}, t)$ is the gradient of Lyapunov-like function $V(\boldsymbol{x})$.

5. Numerical Simulations

5.1 Example System

We consider a simple nonlinear descrete system as an example.

$$\begin{bmatrix} x_{m1,t+1} \\ x_{m2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_t \tag{37}$$

$$x_{1,t+1} = x_{m1,t+1} + \varepsilon x_{m1,t+1}^2 exp(-x_{m2,t+1}^2) \tag{38}$$

$$x_{2,t+1} = x_{m2,t+1} + \varepsilon x_{m2,t+1}^2 exp(-x_{m1,t+1}^2) \tag{39}$$

$$y_{t+1} = \boldsymbol{c}\boldsymbol{x}_{t+1}, \quad \boldsymbol{c} = [0, \quad 1] \tag{40}$$

$$\boldsymbol{x}_{t+1} = [x_{1,t+1}, \quad x_{2,t+1}]^T \tag{41}$$

We use the reccurent neural network as the nonlinear observer in the above system. The network has two neurans in output layer, eight neurons in middle layer and three neurons in input layer. The outputs of the neurons in each layer are given by the following respectively.

$$f_I(x) = x \tag{42}$$

$$f_H(x) = \frac{1}{1 + exp(-x)} \tag{43}$$

$$f_O(x) = x \tag{44}$$

The relations between the inputs and outputs are given by the following.

$$g_i(t) = \sum_{j=1}^{8} c_{ij} H_j(t) + \theta_i \tag{45}$$

$$H_j(t) = f_H(T_{H_j}(t)) \tag{46}$$

$$T_{H_j}(t) = \sum_{k=1}^{3} a_{jk} I_k(t) + \sum_{l=1}^{8} b_{jl} H_l(t-1) + \nu_j \tag{47}$$

The derivative of the output for each coupling coefficient is given as follows.

We get the derivative for the coupling c_{ij} between middle layer and output layer.

$$\frac{\partial g_i}{\partial c_{pq}} = H_q(t)\delta_{pi} \tag{48}$$

For the coupling b_{jl} between middle layer and middle layer.

$$\begin{aligned} \frac{\partial g_i}{\partial b_{pq}} &= \sum_{j=1}^{8} c_{ij} \frac{\partial H_j(t)}{\partial b_{pq}} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t)) \frac{\partial T_{H_j}(t)}{\partial b_{pq}} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t)) B_{jpq}(t) \end{aligned} \tag{49}$$

$$\begin{aligned} B_{jpq}(t) &= H_q(t-1)\delta_{pj} + \sum_{j=1}^{8} b_{jl} \frac{\partial H_l(t-1)}{\partial T_{H_l}(t-1)} \frac{\partial T_{H_l}(t-1)}{\partial b_{pq}} \\ &= H_q(t-1)\delta_{pj} + \sum_{j=1}^{8} b_{jl} f'_H(T_{H_l}(t-1)) B_{lpq}(t-1) \end{aligned} \tag{50}$$

For the coupling a_{jk} between input layer and middle layer.

$$\begin{aligned} \frac{\partial g_i}{\partial a_{pq}} &= \sum_{j=1}^{8} c_{ij} \frac{\partial H_j(t)}{\partial a_{pq}} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t)) \frac{\partial T_{H_j}(t)}{\partial a_{pq}} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t)) A_{jpq}(t) \end{aligned} \tag{51}$$

$$\begin{aligned} A_{jpq}(t) &= I_q(t)\delta_{pj} + \sum_{j=1}^{8} b_{jl} \frac{\partial H_l(t-1)}{\partial T_{H_l}(t-1)} \frac{\partial T_{H_l}(t-1)}{\partial a_{pq}} \\ &= I_q(t)\delta_{pj} + \sum_{j=1}^{8} b_{jl} f'_H(T_{H_l}(t-1)) A_{lpq}(t-1) \end{aligned} \tag{52}$$

For the offset θ_i of output layer.

$$\frac{\partial g_i}{\partial \theta_p} = \delta_{pi} \tag{53}$$

For the offset ν_j of hidden layer.

$$\begin{aligned}\frac{\partial g_i}{\partial \nu_p} &= \sum_{j=1}^{8} c_{ij}\frac{\partial H_j(t)}{\partial \nu_p} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t))\frac{\partial T_{H_j}(t)}{\partial \nu_p} \\ &= \sum_{j=1}^{8} c_{ij} f'_H(T_{H_j}(t))C_{jp}(t) \qquad (54)\end{aligned}$$

$$\begin{aligned}C_{jp}(t) &= \delta_{pj} + \sum_{j=1}^{8} b_{jl}\frac{\partial H_l(t-1)}{\partial T_{H_l}(t-1)}\frac{\partial T_{H_l}(t-1)}{\partial \nu_p} \\ &= \delta_{pj} + \sum_{j=1}^{8} b_{jl} f'_H(T_{H_l}(t-1))C_{lp}(t-1) \qquad (55)\end{aligned}$$

Fig.4 shows the comparison among the output of the observer using a linear approximation, the output of the nonlinear observer and the output of the original system(plant). It is observed that the output of the nonlinear observer using the neural network tracks well the output of the original system.

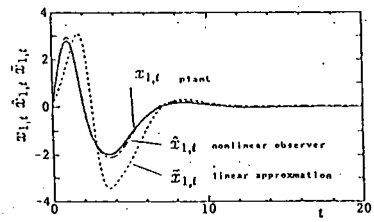

Fig.4 Comparison in the outputs of observers

Similarly Fig.5 shows the comparison among the output of the nonlinear observer using a reccurent neural network and the output of the nonlinear observer using a conventional neural network. It is seen that the nonlinear observer using the reccurent neural network is suitable for the case of dynamical and online estimation.

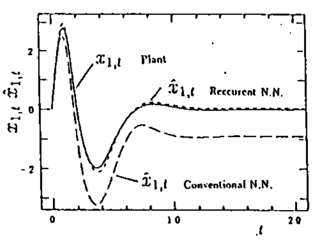

Fig.5 Comparison in the outputs of observers

5.2 DC servomotor with an inverted pendulum

Next we consider a DC servomotor with a inverted pendulum as an example which has a nonlinear load characteristics. Invert pendulum is well known as a load with a nonlinear load characteristics.

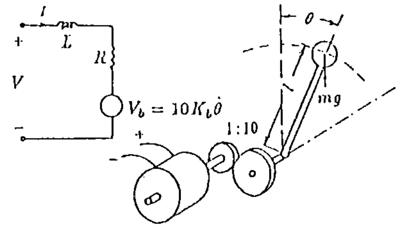

Fig.5 Inverted pendulum system with a DC servomotor

The system equations of the inverted pendulum shown in Fig.5 are given as follows.

$$ml^2\ddot{\theta} = lmg\sin\theta - T_p \qquad (56)$$

$$T_p = 10K_m I \qquad (57)$$

$$L\dot{I} + RI + V_b = V \qquad (58)$$

$$V_b = 10K_b\dot{\theta} \qquad (59)$$

where L and R are the inductance and resistance of the armature winding of the DC motor. The induced Voltage V_b is proportional to the angular velocity, where K_b is the induced voltage constant. Regarding θ, $\dot{\theta}$ and I as state variables, and voltage V as control variable, we let $x_1 = \theta$, $x_2 = \dot{\theta}$, $x_3 = I$ and $u = V$. Then we obtain the following simultanious differential equations system.

$$\begin{aligned}\dot{x}_1 &= x_2 \qquad (60)\\ \dot{x}_2 &= \frac{g}{l}\sin x_1 + 10\frac{K_m}{l^2 m}x_3 \qquad (61)\\ \dot{x}_3 &= -10\frac{K_b}{L}x_2 - \frac{R}{L}x_3 + \frac{1}{L}u \qquad (62)\end{aligned}$$

We have numerical simulations for the case $g = 9.8m/s^2, m = 0.5kg, l = 0.3m, K_m = 0.01Nm/A, K_b = 0.02Vs/rad, L = 20mH and R = 0.1\Omega$. Then the following equations are obtained.

$$\begin{aligned}\dot{x}_1 &= x_2 \qquad (63)\\ \dot{x}_2 &= 32.67\sin x_1 + 2.22x_3 \qquad (64)\\ \dot{x}_3 &= -x_2 - 5.0x_3 + 50.0u \qquad (65)\end{aligned}$$

In the simulations, we set the desired state to an unstable equilibrium point(inveted pendulum). Fig.6

shows the output of nonlinear observer when an appropriate test input. In the case the oscilation's amplitude is small, the output of nonlinear observer converges to the real output. Fig.7 shows the output in the case the oscilation's amplitude is comparatively large. In this case the output of observer doesn't converge to the real output.

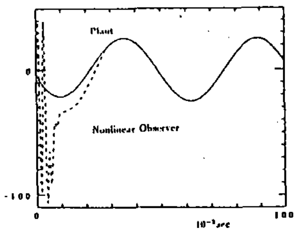

Fig. 6 Output response of Nonlinear Observer

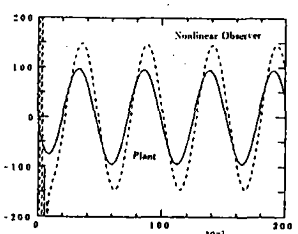

Fig. 7 Output response of Nonlinear Observer

We set the Lyapunov-like function $V(\boldsymbol{x})$ and the weight coefficient matrix R in Eq.36 as follows.

$$V(\boldsymbol{x}) = (x_1, x_2, x_3) \begin{pmatrix} 96.73 & 16.53 & 0.66 \\ 16.57 & 2.93 & 0.11 \\ 0.66 & 0.11 & 0.02 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{66}$$

$$\boldsymbol{R} = \boldsymbol{I} \tag{67}$$

Fig.8 shows the case the nonlinear optimal regulator using Eq.36 is added to the input.

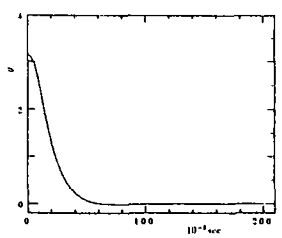

Fig. 8 Output response using Nonlinear Optimal Regulator

6. Conclusion

We have proposed a method of suboptimal control in nonlinear system using a neural network. It seems that there exist a limit in the nonlinear observer which is constructed by using an approximated linear dynamics. Although we have proposed the method of nonlinear observer, in the strong nonlinear range, it doesn't give the good performance. There is room for improvement. So far as the inverse pendulum system is concerned, as nonlinear optimal regulator is effective, the nonlinear observer which works in the wide range is the problem remained.

References

1) H. Kinjo, S. Omatsu and T. Yamamoto,"Suboptimal Control for a Nonlinear System Using Neural Network", T. IEE Japan, Vol.113-C, No.6, 394 (1993)

2) D. Psaltis, A. Sideris and A. A. Yamamura: "A Multilayered Neural Network Controller", IEEE Control Syst. Mag., 8, No.2, 17(1988)

3) M. Khalid and S. Omatsu: "A Neural Network Controller for a Temperature Control System", IEEE Control Syst. Mag., 12, No.3, 58(1992)

4) K.S. Narendora and K. Parthasarathy: "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Trans. Neural Networks, 1, No.1, 4 (1990)

5) M. I. Jordan: "Generic Constraints on Underspecified Target Trajectories", Proceedings of International Joint Conference on Neural Networks, 1, 217, Washington D. C. (1989)

6) Y. Iiguni, H. Sakai and H. Tokumaru: "A Nonlinear Regulator Design in the Presence of System Uncertainties Using Multilayered Neural Networks", IEEE Trans. Neural Networks, 2, 410 (1991)

7) Y. Yokohama, et al: "Plant Identification and Synthesis of Optimal Control", Trans.SICE,Vol.29.No.3,340 (1993)

8) H. Kawaba, et al: "A Method of Nonlinear Optimal Regulator Using a Lyapunov-like Function", Proceedings of '90 KACC, 1060 (1990)

9) R. J. Williams and D. Zipser: "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks", Neural Computation, 1, 270 (1989)