Proceedings of The

IEEE International Conference on Industrial Technology (ICIT'96)

Sponsored by:
IEEE Industrial Electronics Society

Technically cosponsored by:
Society of Instrument and Control Engineers (Japan)
Tongji University
IEEE Robotics and Automation Society
IEEE Beijing Section

In cooperation with:
IEEE Power Electronics Society
Shanghai Jiaotong University

With support from :
National Natural Science Foundation of China
State Education Commission of China

2-6 December 1996
Tongji University Conference Center, Shanghai, China

96TH8151

Proceedings of The
IEEE International Conference
on Industrial Technology, 1996

Neuro-Based Optimal Regulator for a Class of System with Uncertainties

Bing Hong XU, Toshio TSUJI, Michio HATAGI and Makoto KANEKO
Faculty of Engineering, Hiroshima University
Higashi-Hiroshima, 739 JAPAN

Abstract–**This paper proposes a Neuro-Based Optimal Regulator (NBOR) for a class of system with uncertainties. In this paper, we show how the neural network output compensates the control input based on the Riccati equation and how the compensatory solution of the Riccati equation is estimated by the least-squares method. Then, the NBOR is applied to systems with uncertainties in order to illustrate its effectiveness and applicability.**

I. INTRODUCTION

The optimal regulator is usually designed for a mathematical model of a controlled system. However, the mathematical model is not exactly known in practical application of the optimal regulator. For the controlled system with linear uncertainties, the various kinds of the robust optimal regulator has been proposed [1]. It should be noted that the robust optimal regulator cannot work well if the nonlinear uncertainty of the controlled system exists.

For this problem, various regulators using neural networks have been proposed in recent years. Rovithakis and Christodoulou [2] linearized an unknown nonlinear dynamic system and used three neural networks for constituting a direct adaptive regulator. Levin and Narendra [3] presented a framework for the use of neural networks for identification and control of nonlinear dynamical systems. In this framework, an observer and a controller are consisted of neural networks. In the methods described above, the unknown part of the controlled system is identified by one neural network, and other neural networks are used as the regulator. Since the multiple neural networks must be trained, it requires a long time for computation and learning.

In this paper, a neuro-based optimal regulator (NBOR) for a class of dynamic system with uncertainties is proposed. The proposed method designs an optimal regulator for the linear known part and uses a neural network to identify the unknown part. At the same time, the neural network works as an adaptive compensator for the unknown part. First, we show how the neural network output compensates the control input based on the Riccati equation and how the compensatory solution of the Riccati equation is estimated by the least-squares method. Then, computer simulation is performed in order to illustrate the effectiveness and applicability of the NBOR.

II. NEURO-BASED OPTIMAL REGULATOR

A. System Formulation

We consider the following controlled system

$$\dot{x}(t) = A_L x(t) + \tilde{A}(x(t)) + Bu(t), \quad (1)$$
$$y(t) = Cx(t), \quad (2)$$

where $x(t) \in \Re^{n\times 1}$, $u(t) \in \Re^{m\times 1}$ and $y(t) \in \Re^{l\times 1}$ are the state, the input and the output, respectively; $A_L \in \Re^{n\times n}$, $B \in \Re^{n\times m}$, $C \in \Re^{l\times n}$ are the parameter matrices; and $\tilde{A}(\cdot)$ is the nonlinear function of the state $x(t)$.

The goal of the control is to find the optimal input that minimizes the quadratic performance index of the form:

$$J = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt, \quad (3)$$

where $Q \in \Re^{n\times n} \geq 0$, $R \in \Re^{m\times m} > 0$ are the weight matrices specified by the designer. The nonlinear regulator problem for the system (1) is very difficult and is approximately solved using one of the linearized techniques that synthesizes an observer and the linear optimal regulator. However, this approach cannot work well when (1) is not linearized appropriately, so that the nonlinear compensation is particularly required. In this paper, we propose the NBOR for solving the control problem.

B. Optimal Regulator

First, we divide the matrix $A_L = A_{L_n} + \Delta_{A_L}$ of (1) into the known parameter matrix $A_{Ln} \in \Re^{n\times n}$ and the uncertainties matrix $\Delta_{A_L} \in \Re^{n\times n}$, and assume that the nonlinear function $\tilde{A}(x(t))$ is approximately described as

$$\tilde{A}(x(t)) \approx A^* x(t) + \Delta_{A^*} x(t) \quad (4)$$

near the operating point of the controlled system. Here, $A^* \in \Re^{n\times n}$ represents the linearized parameter and $\Delta_{A^*} \in \Re^{n\times n}$ represents the unknown linearized modeling error.

Then the system (1) becomes

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (5)$$
$$A = A_n + \Delta_A, \quad (6)$$
$$A_n = A_{Ln} + A^*, \quad (7)$$

0-7803-3104-4

where $\Delta_A = \Delta_{A_L} + \Delta_{A^*}$. Δ_A and $A_n \in \Re^{n\times n}$ are respectively the system uncertainty and the nominal parameter. Also A_{Ln} and $\Delta_{A_L} \in \Re^{n\times n}$ are the known part of the parameter A_L and the parameter uncertainty. The system (5) is assumed to be controllable and observable.

The optimal control input $u^*(t)$ that minimizes the performance index J of (3) is given as

$$u^*(t) = -R^{-1}B^T Px(t), \tag{8}$$

where $P \in \Re^{n\times n}$ is the unique solution of the Riccati equation

$$PA + A^T P - PBR^{-1}B^T P + Q = 0. \tag{9}$$

Here the solution P is assumed as

$$P = P_n + \Delta_P, \tag{10}$$

where $P_n \in \Re^{n\times n}$, $\Delta_P \in \Re^{n\times n}$ are respectively the solution for the known linear part of the system (5) and the compensatory solution of the Riccati equation.

Substituting (6), (10) into (9), we have

$$(P_n + \Delta_P)(A_n + \Delta_A) + (A_n + \Delta_A)^T(P_n + \Delta_P)$$
$$-(P_n + \Delta_P)BR^{-1}B^T(P_n + \Delta_P) + Q = 0. \tag{11}$$

If the quadratic terms $\Delta_P\Delta_A$, $\Delta_A^T\Delta_P$, $\Delta_P\Delta_P$ are sufficiently small, (11) can be divided into the following two equations:

$$P_nA_n + A_n^TP_n - P_nBR^{-1}B^TP_n + Q = 0, \tag{12}$$
$$\Delta_PA_n + P_n\Delta_A + \Delta_A^TP_n + A_n^T\Delta_P$$
$$-\Delta_PBR^{-1}B^TP_n - P_nBR^{-1}B^T\Delta_P = 0. \tag{13}$$

In order to compute the optimal control input $u^*(t)$ of (8), the solution P of the Riccati equation has to be known. Although P_n can be obtained from (12), the compensatory solution Δ_P cannot be directly computed from (13) since Δ_A is unknown.

C. *Estimation of Compensatory Solution*

In this section, we use the least-squares estimation for the compensatory solution of the Riccati equation. From (13), we have the following form:

$$\Delta_P\mathcal{D} + \mathcal{F}\Delta_P = \Delta_A^TP_n + P_n\Delta_A, \tag{14}$$

$$\mathcal{D} = BR^{-1}B^TP_n - A_n, \mathcal{F} = P_nBR^{-1}B^T - A_n^T.$$

Deriving the quadratic form of the state $x(t)$ for both sides of (14) yields

$$x^T(t)\mathcal{K}x(t) = \Delta_x^T(t)P_nx(t) + x^T(t)P_n\Delta_x(t), \tag{15}$$
$$\Delta_x(t) = \Delta_Ax(t), \tag{16}$$
$$\mathcal{K} = \Delta_P\mathcal{D} + \mathcal{F}\Delta_P. \tag{17}$$

where $\Delta_x(t)$ is the uncertain state occurred by the uncertainty Δ_A.

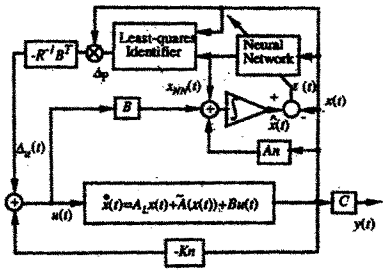

Fig. 1: Block diagram of Neuro-Based Optimal Regulator

Developing (15) into a linear equation and arranging the linear equation due to the unknown compensatory solution Δ_P, we can obtain

$$\omega(t)\theta = \lambda(t), \tag{18}$$
$$\theta(t) = [\vartheta_{11}, \cdots, \vartheta_{nn}]^T \in \Re^{n^2\times 1}, \tag{19}$$
$$\omega(t) = [\omega_{11}(t), \cdots, \omega_{nn}(t)] \in \Re^{1\times n^2}, \tag{20}$$

$$\omega_{kh}(t) = \sum_{i=1}^{n} x_i(t)d_{hi}x_k(t) + \sum_{j=1}^{n} x_h(t)f_{jk}x_j(t)$$
$$(k, h = 1, 2, \cdots, n), \tag{21}$$

$$\lambda(t) = \sum_{j=1}^{n}\sum_{i=1}^{n} \Delta_{x_j}(t)p_{ji}x_i(t) + \sum_{j=1}^{n}\sum_{i=1}^{n} x_j(t)p_{ji}\Delta_{x_i}(t), \tag{22}$$

where p_{ij}, d_{hi}, f_{jk} are the elements of the known matrices P_n, $\mathcal{D}$, $\mathcal{F}$, respectively; ϑ_{ij} is the element of the unknown vector $\theta(t)$; and $\theta(t) = \text{cs}\Delta_P$ is the expanded form of the column of the matrix Δ_P.

With the sampling time step, we can get $s(\geq n^2)$ sets of the sequential data and have the following matrix equation from (18).

$$\Omega(t)\theta(t) = \Lambda(t), \tag{23}$$
$$\Lambda(t) = [\lambda(t - \Delta t), \cdots, \lambda(t - s\Delta t)]^T \in \Re^{s\times 1}, \tag{24}$$
$$\Omega(t) = [\omega(t - \Delta t), \cdots, \omega(t - s\Delta t)]^T \in \Re^{s\times n^2} \tag{25}$$

If $\lambda(t)$ of (22) is obtained, then $\theta(t)$ can be computed by the least-squares method, that is

$$\theta(t) = \Omega^+(t)\Lambda(t), \tag{26}$$

where $\Omega^+(t)$ represents the pseudo inverse matrix.

Consequently, when $\lambda(t)$ of (22) can be obtained, the compensatory solution Δ_P can be estimated. However, $\Delta_x(t)$ included in (22) cannot be computed from (16), since Δ_A is unknown. Therefore the neural network is introduced for solving the problem.

D. *NBOR Scheme*

Fig. 1 shows the block diagram of the NBOR. The identification system shown in Fig. 1 is described as

$$\hat{x}(t) = A_nx(t) + Bu(t) + x_{NN}(t), \tag{27}$$

where $\hat{x}(t) \in \Re^{n\times 1}$ and $x_{NN}(t) \in \Re^{n\times 1}$ are the predicted state of the identification system and the output of the neural network, respectively.

Substituting (10) into (8), we have the optimal control input $u^*(t)$ as

$$\begin{aligned} u^*(t) &= u_n(t) + \Delta_u(t), &(28)\\ u_n(t) &= -K_n x(t), &(29)\\ K_n &= R^{-1}B^T P_n, &(30)\\ \Delta_u(t) &= -R^{-1}B^T \Delta_P x(t), &(31)\end{aligned}$$

where $K_n \in \Re^{m\times n}$ is the feedback gain of the linear optimal regulator. The identified state error $\epsilon(t) \in \Re^{n\times 1}$ between $\hat{x}(t)$ and $x(t)$ is defined as

$$\begin{aligned} \epsilon(t) &= \hat{x}(t) - x(t)\\ &= \int_0^t \zeta(\tau)d\tau, &(32)\\ \zeta(\tau) &= [x_{NN}(\tau) - \Delta_x(\tau)], &(33)\end{aligned}$$

where $\zeta(\tau) \in \Re^{n\times 1}$ represents the error between the neural network output and the uncertain state. However, even if the identified error $\epsilon(t)$ becomes zero, the integrand $\zeta(\tau)$ of (32) may not be zero. So, we define the energy function $E(t)$ for training the neural network as

$$\begin{aligned} E(t) &= \frac{1}{2}\dot{\epsilon}^T(t)\dot{\epsilon}(t) + \frac{1}{2}\epsilon^T(t)\epsilon(t)\\ &= \frac{1}{2}\zeta^T(t)\zeta(t) + \frac{1}{2}\epsilon^T(t)\epsilon(t)\\ &= E^{(1)}(t) + E^{(2)}(t), &(34)\\ E^{(1)}(t) &= \frac{1}{2}\zeta^T(t)\zeta(t), &(35)\\ E^{(2)}(t) &= \frac{1}{2}\epsilon^T(t)\epsilon(t). &(36)\end{aligned}$$

When $E(t)$ becomes zero, the output $x_{NN}(t)$ of the neural network agrees with the state uncertainties $\Delta_x(t)$, that is,

$$x_{NN}(t) = \Delta_x(t). \quad (37)$$

We can see that the control input shown in Fig. 1 will be gradually close to the optimal control input of (8) as the neural network trains.

III. NEURAL NETWORK SCHEME

A multi-layer neural network used in the proposed regulator is shown in Fig. 2. The numbers of the units of the input layer, the hidden layer and the output layer are n, p and n, respectively. In Fig. 2, w_{ij} represents the weight that connects the unit j of the input layer to the unit i of the hidden layer; v_{ki} represents the weight that connects the unit i of the hidden layer to the unit k of the output layer. The weight matrices are represented as $W(t) \in \Re^{p\times n}$ and $V(t) \in \Re^{n\times p}$, respectively. Also, the input and output vectors of the neural network are represented as $x(t)$, $x_{NN}(t)$, respectively.

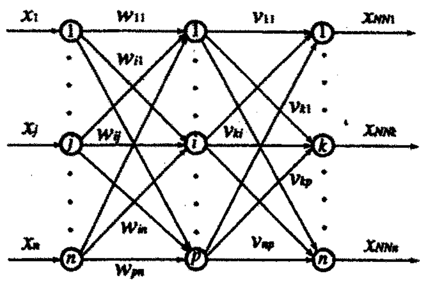

Fig. 2: Three-layer neural network

Let the unit j's output of the input layer be $I_j = x_j(t)$ $(j = 1, \cdots, n)$, the unit i's output of the hidden layer be $H_i = \sigma(s_i)$, $s_i = \sum_{j=1}^{n} w_{ij} I_j$, where the sigmoid function $\sigma(\cdot)$ is defined as $\sigma(\mu) \equiv \frac{1}{\gamma} tanh(\gamma\mu)$. Here, γ is the positive parameter related with the shape of the sigmoid function. Also, let the unit k's output of the output layer be $O_k = \sigma(\kappa_k)$, $\kappa_k = \sum_{i=1}^{p} v_{ki} H_i$.

In the training process, the energy function of (34) is minimized by changing the weights w_{ij} and v_{ki}. According to the back-propagation algorithm, the weight updating rules can be described as

$$v_{kj}(t+\Delta t) = v_{kj}(t) - \eta\left[\frac{\partial E^{(1)}(t)}{\partial v_{kj}(t)} + \frac{\partial E^{(2)}(t)}{\partial v_{kj}(t)}\right], \quad (38)$$

$$w_{ij}(t+\Delta t) = w_{ij}(t) - \eta\left[\frac{\partial E^{(1)}(t)}{\partial w_{ij}(t)} + \frac{\partial E^{(2)}(t)}{\partial w_{ij}(t)}\right], \quad (39)$$

where $\eta > 0$ is the learning rate; and Δt is the time interval of the network learning. The function $E^{(1)}(t)$ of (34) for $\zeta(t) = \dot{\epsilon}(t)$ is rewritten as

$$\begin{aligned} E^{(1)}(t) &= \frac{1}{2}\zeta^T(t)\zeta(t)\\ &= \frac{1}{2}\sum_{q=1}^{n}[x_{NN_q}(t) - \Delta_{x_q}(t)]^2, &(40)\end{aligned}$$

where $x_{NN_q}(t)$, $\Delta_{x_q}(t)$ are respectively the elements of the neural network output $x_{NN}(t)$ and the uncertain state $\Delta_x(t)$.

By (32), (40), $\partial E^{(1)}(t)/\partial v_{ki}(t)$ becomes

$$\frac{\partial E^{(1)}(t)}{\partial v_{ki}(t)} = \dot{\epsilon}_k(t)\frac{\partial x_{NN_k}(t)}{\partial v_{ki}(t)} = \dot{\epsilon}_k(t)\sigma'(\kappa_k)H_i, \quad (41)$$

where $\dot{\epsilon}_k(t)$ represents the element of the vector $\dot{\epsilon}(t)$ and $\sigma'(\cdot)$ represents the derivative of $\sigma(\cdot)$.

Also, $\partial E^{(1)}(t)/\partial w_{ij}(t)$ can be written as

$$\begin{aligned} \frac{\partial E^{(1)}(t)}{\partial w_{ij}(t)} &= \sum_{q=1}^{n}\dot{\epsilon}_q(t)\frac{\partial x_{NN_q}(t)}{\partial w_{ij}(t)}\\ &= \sum_{q=1}^{n}\dot{\epsilon}_q(t)\sigma'(\kappa_q)v_{qi}\sigma'(s_i)x_j(t). &(42)\end{aligned}$$

On the other hand, the function $E^{(2)}(t)$ of (34) for the identified error $\epsilon(t)$ is given as

$$E^{(2)}(t) = \frac{1}{2}\epsilon^T(t)\epsilon(t) = \frac{1}{2}\sum_{k=1}^{n}[\hat{x}_k(t) - x_k(t)]^2, \quad (43)$$

and $\partial E^{(2)}(t)/\partial v_{ki}(t)$ can be described by

$$\frac{\partial E^{(2)}(t)}{\partial v_{ki}(t)} = \epsilon_k(t)\frac{\partial \epsilon_k(t)}{\partial x_{NN_k}(t)}\frac{\partial x_{NN_k}(t)}{\partial v_{ki}(t)}, \quad (44)$$

where $\epsilon_k(t)$ is the element of the vector $\epsilon(t)$. The partial derivative $\partial\epsilon_k(t)/\partial x_{NN_k}(t)$ can be approximated as

$$\frac{\partial \epsilon_k(t)}{\partial x_{NN_k}(t)} \approx \frac{\Delta\epsilon_k(t)}{\Delta x_{NN_k}(t)}. \quad (45)$$

If $x_{NN_k}(t)$ is changed by $\Delta x_{NN_k}(t)$, the variation $\Delta\epsilon_k(t)$ of $\epsilon_k(t)$ becomes

$$\begin{aligned}\Delta\epsilon_k(t) &\approx \sum_{j=0}^{N_t}[x_{NN_k}(j\Delta t_s) - \Delta_{x_k}(j\Delta t_s) + \Delta x_{NN_k}(t)]\Delta t_s \\ &\quad - \sum_{j=0}^{N_t}[x_{NN_k}(j\Delta t_s) - \Delta_{x_k}(j\Delta t_s)]\Delta t_s \\ &= \Delta x_{NN_k}(t)\Delta t_s. \end{aligned} \quad (46)$$

Therefore, we can approximate $\epsilon_k(t)/\partial x_{NN_k}(t)$ as

$$\frac{\partial\epsilon_k(t)}{\partial x_{NN_k}(t)} \approx \Delta t_s, \quad (47)$$

where Δt_s is the small sampling time and $t = N_t\Delta t_s$. Substituting (47) into (44) yields

$$\frac{\partial E^{(2)}(t)}{\partial v_{ki}(t)} \approx \epsilon_k(t)\Delta t_s\frac{\partial x_{NN_k}(t)}{\partial v_{ki}(t)}. \quad (48)$$

Also, $\partial E^{(2)}(t)/\partial w_{ij}(t)$ can be given by

$$\frac{\partial E^{(2)}(t)}{\partial w_{ij}(t)} \approx \sum_{q=1}^{n}\epsilon_q(t)\Delta t_s\frac{\partial x_{NN_q}(t)}{\partial w_{ij}(t)}. \quad (49)$$

As a result, the updating rules of (38), (39) reduce to the following form

$$v_{ki}(t+\Delta t) \approx v_{ki}(t) - \eta[\dot{\epsilon}_k(t) + \epsilon_k(t)\Delta t_s]\frac{\partial x_{NN_k}(t)}{\partial v_{ki}(t)}, \quad (50)$$

$$w_{ij}(t+\Delta t) \approx w_{ij}(t) - \eta\sum_{q=1}^{n}[\dot{\epsilon}_q(t) + \epsilon_q(t)\Delta t_s]\frac{\partial x_{NN_q}(t)}{\partial w_{ij}(t)}. \quad (51)$$

Since (50) and (51) mean that the error signal for training the neural network is the weighted sum of $\epsilon(t)$ and $\dot{\epsilon}(t)$, this learning rule is corresponding to the PD control rule of the feedback control system. Therefore, it is called as a *PD* updating rule in this paper.

It should be noted that, if the energy functions of (34) are modified as

$$E^{(1)}(t) = \frac{1}{2}\dot{\epsilon}^T(t)B_l\dot{\epsilon}(t), E^{(2)}(t) = \frac{1}{2}\epsilon^T(t)K_l\epsilon(t), \quad (52)$$

the *PD* learning can be adjusted by the learning gains $B_l \in \Re^{n\times n}$, $K_l \in \Re^{n\times n}$ defined as the positive definite matrix.

IV. COMPUTER SIMULATION

In order to show the effectiveness and the applicability of the NBOR, we apply the NBOR and the Linear Quadratic Regulator (LQR) to the following simulation system with uncertainties:

$$\begin{aligned}\dot{x}(t) &= (A_n + \Delta_A)x(t) + Bu(t) \quad &(53)\\ y(t) &= Cx(t), \quad &(54)\end{aligned}$$

where

$$A_n = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}, \Delta_A = \begin{bmatrix} 0.1 & 0.08 \\ 0.05 & 0.15 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and the initial state $x(0)$ and the final state $x^*(t)$ are $x(0) = [1.0\ 2.0]^T$ and $x^*(t) = [0.0\ 0.0]^T$, respectively. The weight matrices of the quadratic performance index of (3) are designed as

$$Q = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}, R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

For A_n, B of (53), the solution P_n of the Riccati equation and the feedback gain K_n are obtained as

$$P_n = \begin{bmatrix} 0.0766 & 0.0630 \\ 0.0630 & 0.1395 \end{bmatrix}, K_n = \begin{bmatrix} 0.1531 & 0.1259 \\ 0.1259 & 0.2791 \end{bmatrix}.$$

On the other hand, for $A_n + \Delta_A$ and B, the solution P^* of the Riccati equation and the gain K^* are given by

$$P^* = \begin{bmatrix} 0.0928 & 0.0882 \\ 0.0882 & 0.1817 \end{bmatrix}, K^* = \begin{bmatrix} 0.1855 & 0.1764 \\ 0.1764 & 0.3634 \end{bmatrix}.$$

The responses of the state $x(t)$ and the output $y(t)$ using K_n and K^* are shown in Fig. 3 as the thin line (LQR) and the dotted line (DRE), respectively, where the reference signal of the unit step function is used. From Fig. 3, we can see that the error between the thin line and the dotted line is not negligible.

In the NBOR, the four-layer neural network is used. Corresponding to the system (53), the unit numbers of the input layer and the output layer are 2, and the unit number of each hidden layer is 10. The weight initial value of the neural network is chosen as the uniform random

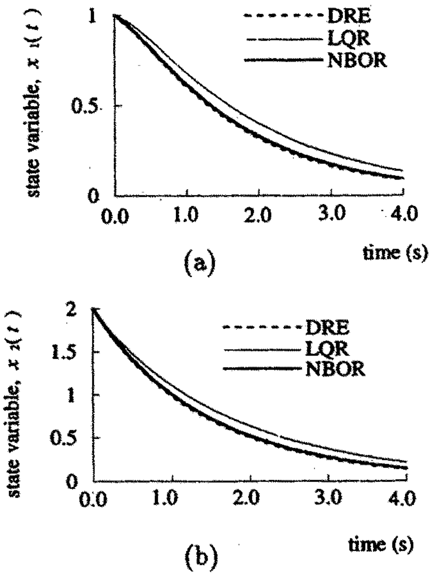

Fig. 3: Comparison of the simulation results

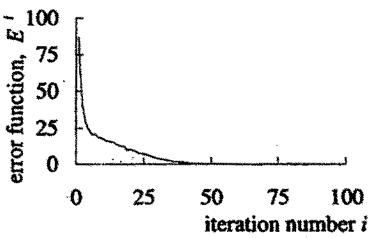

Fig. 4: The error function during the learning iteration

number in [-0.1, +0.1] and $\gamma = 1.0$ in the sigmoid function is used. The learning rate is η=0.0005. Also the sampling time is $\Delta t = 0.002$ s and the control time T=4 s for one iteration of the reference signal of the unit step function. The response of the state $x(t)$ and the output $y(t)$ during the 100th learning iteration using the NBOR is shown in Fig. 3 as the solid line (NBOR). From Fig. 3, the response under the NBOR can almost achieve the desired response for the system with uncertainties.

Fig. 4 shows the relationship between the learning iteration and the identified error of the NBOR. In order to evaluate the identified error, the error function E^i is defined as

$$E^i = \sum_{j=1}^{500} E\{[j + 500(i-1)]\Delta t\}, \tag{55}$$

where $E(\cdot)$ is defined as (3) and i is the iteration number of the reference signal. From Fig. 4, the identification system is approaching the system (53) according to the learning of the neural network.

The NBOR can adaptively control the controlled system with uncertainties, and automatically realize the identification system of the controlled system simultaneously. Therefore, we also investigate the identification ability of the NBOR. The error between the state $x(t)$ of (53) and the predicted state $\hat{x}(t)$ of the identification system using the neural network after 100 learning iterations is shown in Fig. 5 as the dotted line.

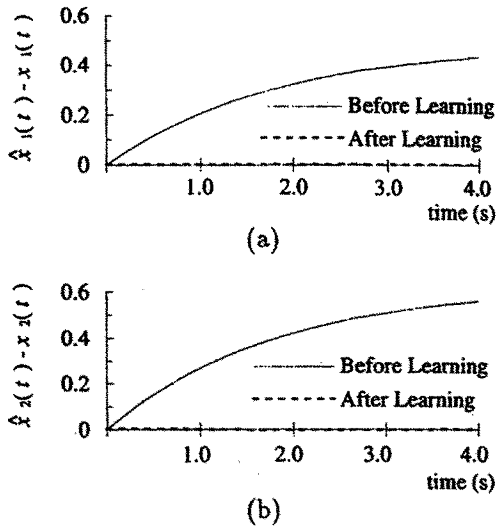

Fig. 5: Identification ability of the proposed method

V. CONCLUSIONS

In this paper, the NBOR is proposed for a class of system with uncertainties. In the NBOR scheme, the neural network can identify the unknown part of the controlled system and compensate the control input from the linear optimal regulator simultaneously. By the training process of the neural network, the control input can be adaptively modified and the identification system is automatically realized. In order to illustrate the applicability of the NBOR, we plan to apply it to a control problem of a manipulator in future.

References

[1] J. Douglas and M. Athans, "Robust Linear Quadratic Design with Real Parameter Uncertainty", *IEEE Trans. On Automatic Control*, AC Vol.39-1,1994, pp.107-111.

[2] G.A. Rovithakis and M. A. Christodoulou, "Direct Adaptive Regulation of Unknown Nonlinear dynamical Systems via Dynamic Neural Networks", *IEEE Trans. On Systems, Man, and Cybernetics*, SMC Vol.25-12, 1995, pp.1578-1594.

[3] A.U.Levin and K.S.Narendra, "Control of Nonlinear Dynamical Systems Using Neural Networks-Part II: Observability, Identification, and Control", *IEEE Trans. On Neural Networks*, Vol.7, No.1, 1996, pp.30-42.