피이드백 선형화를 위한 안정한 적응 신경회로망 구현

김 동 헌*○, 양 해 원*
* 한양대학교 전기공학과

Implementation of Stable Adaptive Neural Networks for Feedback Linearization

Dong-Hun Kim*○, Hai-Won Yang*
* Hanyang University

Abstract - For a class of single-input single-output continuous-time nonlinear systems, a multilayer neural network-based controller that feedback-linearizes the systm is presented. Control action is used to achieve tracking performance for a state-feedback linearizable but unknown nonlinear system. The multilayer neural network(NN) is used to approximate nonlinear continuous function to any desired degree of accuracy. The weight-update rule of multilayer neural network is derived to satisfy Lyapunov stability. It is shown that all the signals in the closed-loop system are uniformly bounded. Initialization of the network weights is straightforward.

1. Introduction

Nonlinear control has received a lot of attraction for the past two decades. Controlling nonlinear systems by 'feedback linearization' has experienced a growing popularity. Unfortunately, this type of control requires exact dynamic equations of systems[1]-[4]. Thus, recently many AI theories have been used to control nonlinear systems without such an priori knowledge around nonlinear function of plant[5]-[9]. But, generally, if they are used for approximation, it is difficult to control nonlinear systems precisely. Because there are function approximation error and higher terms coming from Taylor series expansion. Accordingly, dead-zone technique was used for cancelling them on condition of sufficiantly small approximation error[6], but it has many restrictions practically. Therefore, the object of this paper is to design stable controller and compensate for them. In order to attain this object, we introduce control input which consists of a feedback linearization portion provided by two NNs and a robustifying portion that ensures stability. NNs weights are updated on-line according to a rule which satisfies Lyapunov stability.

This paper is organized as follows. In section 2, we define plant dynamic equations for building up this theory and illustrate the concepts of input-output feedback linearization. In section 3, we design control input. Section 4 presents adaptive law by Lyapunov stability to update NNs weights. In section 5, we analyze convergence and stability of the closed-loop system. Finally, simulation results are demonstrated.

2. Problem Statement and Feedback Linearization

Consider a state-feedback linearizable system having a state-space representation in the controllability canonical form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f(\underline{x}) + g(\underline{x})u \\ y &= x_1 \end{aligned} \tag{2.1}$$

or equivalently of the form

$$x^{(n)} = f(\underline{x}) + g(\underline{x})u, \quad y = x \tag{2.2}$$

where f and g are unknown continuous functions, u and y are the input and output of the system, respectively, and $\underline{x} = [x_1\ x_2 \cdots x_n]^T = [x\ \dot{x} \cdots x^{(n-1)}]^T$ is the state vector of the system.

Assumptions :

(1) $f(\underline{x})$ and $g(\underline{x})$ are bounded and uniformly continuous functions.

(2) $|f(\underline{x})| \le f_u(\underline{x})$ for all x, where $f_u(\underline{x})$ is known.

(3) $0 < g_l(\underline{x}) \le g(\underline{x}) \le g_u(\underline{x})$ for all x, where $g_l(\underline{x})$ and $g_u(\underline{x})$ are known.

(4) state variable $\underline{x}$ is available for measurement.

With these assumptions, the control law can be defined as

$$u = \frac{1}{g(\underline{x})}(-f(\underline{x}) + \nu) \tag{2.3}$$

and the system becomes $y^{(n)} = \nu$ where ν is a new control input. If the control goal is for the plant ouput y to track a reference trajectory y_m, the control input ν can be defined as

$$\nu = y_m^{(n)} + k_1 e^{(n-1)} + \cdots + k_{n-1}\dot{e} + k_n e = \underline{k}^T \underline{e} + y_m^{(n)} \tag{2.4}$$

where $\underline{e} = [e\ \dot{e} \cdots e^{(n-1)}]^T$, $\underline{k} = [k_n\ k_{n-1} \cdots k_1]^T$ and $k_1, \cdots, k_{n-1}, k_n$ are chosen such that the polynomial $s^n + k_1 s^{n-1} + \cdots + k_n$ is Hurwitz. Then the control input(2.4) results in the error equation $e^{(n)} + k_1 e^{(n-1)} + \cdots + k_n e = 0$ with $e(t) = y_m(t) - y(t)$ being the tracking error. It is clear that e will approach zero.

3. Control Input Design

If we know the exact form of the nonlinear functions, then the control is

$$u=\frac{1}{g(x)}(-f(x)+k^T e+y_m^{(n)}) \quad (3.1)$$

Since we assume that these functions are not exactly known, we shall choose control action

$$u_c=\frac{1}{\hat{g}(x,\theta_g)}(-\hat{f}(x,\theta_f)+k^T e+y_m^{(n)}) \quad (3.2)$$

where the estimates $\hat{f}(x,\theta_f)$ and $\hat{g}(x,\theta_g)$ will be constructed by NNs. However, (3.1) is not same as (3.2) because of approximation error. Thus, we solve this problem by appending auxiliary input u_s to the u_c. That is, the final control is

$$u=u_c+u_s \quad (3.3)$$

Applying (3.2) to (2.2) and after straightforward manipulation, we obtain the error equation

$$e^{(n)}=-k^T e+[\hat{f}(x,\theta_f)-f(x)]+[\hat{g}(x,\theta_g)-g(x)]u_c - g(x)u_s$$

or equivalently

$$\dot{e}=\Lambda_c e+b_c[(\hat{f}(x,\theta_f)-f(x))+(\hat{g}(x,\theta_g)-g(x))u_c - g(x)u_s] \quad (3.4)$$

where $\Lambda_c=\begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -k_n & -k_{n-1} & \cdots & \cdots & \cdots & \cdots & -k_1 \end{bmatrix}$, $b_c=\begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ 1 \end{bmatrix}$

Since Λ_c is a stable matrix, there exists a unique positive definite symmetric $n\times n$ matrix P which satisfies the Lyapunov equation [1].

$$\Lambda_c^T P+P\Lambda_c=-Q \quad (3.5)$$

where Q is an arbitrary $n\times n$ positive definite matrix.

Let $V_e=\frac{1}{2}e^T P e$, then using (3.4) and (3.5) we have

$$\dot{V}_e=-\frac{1}{2}e^T Q e+e^T P b_c[(\hat{f}(x,\theta_f)-f(x)) + (\hat{g}(x,\theta_g)-g(x))u_c-g(x)u_s]$$

$$\le -\frac{1}{2}e^T Q e+|e^T P b_c|\,[|\hat{f}(x,\theta_f)|+|f(x)| + |\hat{g}(x,\theta_g)u_c|+|g(x)u_c|] - e^T P b_c g(x)u_s \quad (3.6)$$

In order to design u_s such that the right-hand side of (3.6) is negative, we need to know the bounds of $f(x)$ and $g(x)$. We choose the auxiliary input u_s as

$$u_s=sat(e^T P b_c)\frac{1}{g_l(x)}[|\hat{f}(x,\theta_f)|+f_u(x) + |\hat{g}(x,\theta_g)u_c|+|g_u(x)u_c|] \quad (3.7)$$

where $sat(e^T P b_c)=\begin{cases} 1 & \text{if } e^T P b_c>\Delta \\ -1 & \text{if } e^T P b_c<-\Delta \\ \frac{e^T P b_c}{\Delta} & \text{otherwise} \end{cases}$

There exists $\Delta\le e^T P b_c\frac{g(x)}{g_l(x)}\frac{f_u(x)+|g_u(x)u_c|}{|f(x)|+|g(x)u_c|}$

Substituting (3.7) into (3.6), we obtain

$$\dot{V}_e\le[|\hat{f}(x,\theta_f)|+|f(x)| + |\hat{g}(x,\theta_g)u_c|+|g(x)u_c| - \frac{g}{g_l}(|\hat{f}(x,\theta_f)|+|f_u(x)| + |\hat{g}(x,\theta_g)u_c|+|g_u(x)u_c|)]$$

$$\le -\frac{1}{2}e^T Q e\le 0 \quad (3.8)$$

Thus, we can guarantee that $V_e\le\bar{V}<\infty$.

4. Adaptive Law Design

In NNs, we define optimal weights[9]

$$\theta_f^*=\arg\min_{\theta_f\in\Omega_f}[\max_{x\in U_c}|\hat{f}(x,\theta_f)-f(x)|]$$

$$\theta_g^*=\arg\min_{\theta_g\in\Omega_g}[\max_{x\in U_c}|\hat{g}(x,\theta_g)-g(x)|] \quad (4.1)$$

where $\Omega_f=\{\theta_f:|\theta_f|\le M_f\}$, $\Omega_g=\{\theta_g:|\theta_g|\le M_g\}$ and M_f and M_g are positive constants.

Taking the Taylor series expansions of $\hat{f}(x,\theta_f^*)$ and $\hat{g}(x,\theta_g^*)$ around θ_f and θ_g, we have

$$\hat{f}(x,\theta_f)-\hat{f}(x,\theta_f^*)=\phi_f^T(\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f})+O(|\phi_f|^2)$$

$$\hat{g}(x,\theta_g)-\hat{g}(x,\theta_g^*)=\phi_g^T(\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g})+O(|\phi_g|^2) \quad (4.2)$$

where $\phi_f=\theta_f-\theta_f^*$, $\phi_g=\theta_g-\theta_g^*$ and $O(|\phi_f|^2)$ and $O(|\phi_g|^2)$ are higher-order terms.

Substituting (4.2) into (3.4), we have

$$\dot{e}=\Lambda_c e-b_c g(x)u_s+b_c v+b_c[\phi_f^T(\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f}) + \phi_g^T(\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g})u_c] \quad (4.3)$$

where $v=(\hat{f}(x,\theta_f^*)-f(x))+(\hat{g}(x,\theta_g^*)-g(x))u_c + O(|\phi_f|^2)+O(|\phi_g|^2)u_c$ (4.4)

If we use projection method so that θ_f and θ_g are bounded, then we have the following adaptive law[11].

$$\dot{\theta}_f=-\gamma_1 e^T P b_c\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f} \quad (4.5)$$

if ($|\theta_f|<M_f$) or ($|\theta_f|=M_f$ and $e^T P b_c\,\theta_f^T\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f}\ge 0$)

$$=-\gamma_1 e^T P b_c\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f} + \gamma_1 e^T P b_c\frac{\theta_f}{|\theta_f|^2}\theta_f^T\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f} \quad (4.6)$$

if ($|\theta_f|=M_f$ and $e^T P b_c\,\theta_f^T\frac{\partial\hat{f}(x,\theta_f)}{\partial\theta_f}<0$)

$$\dot{\theta}_g=-\gamma_2 e^T P b_c\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g}u_c \quad (4.7)$$

if ($|\theta_g|<M_g$) or ($|\theta_g|=M_g$ and $e^T P b_c\,\theta_g^T\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g}u_c\ge 0$)

$$=-\gamma_2 e^T P b_c\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g}u_c + \gamma_2 e^T P b_c\frac{\theta_g}{|\theta_g|^2}\theta_g^T\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g}u_c \quad (4.8)$$

if ($|\theta_g|=M_g$ and $e^T P b_c\,\theta_g^T\frac{\partial\hat{g}(x,\theta_g)}{\partial\theta_g}u_c<0$)

To investigate the usefulness of the above mentioned adaptive law we consider the Lyapunov function candidate

$$V=\frac{1}{2}\,e^TPe+\frac{1}{2\gamma_1}\,\phi_f^T\,\phi_f+\frac{1}{2\gamma_2}\,\phi_g^T\,\phi_g \tag{4.9}$$

Differentiating (4.9) with respect to time and using (3.5) and (4.3) yields

$$\dot V=-\frac{1}{2}\,e^TQe-g(x)\,e^TP\,b_cu_s+\,e^TP\,b_cv$$
$$+\frac{1}{\gamma_1}\,\phi_f^T[\,\dot\theta_f+\gamma_1\,e^TP\,b_c\frac{\partial\hat f(x,\,\theta_f)}{\partial\,\theta_f}]$$
$$+\frac{1}{\gamma_2}\,\phi_g^T[\,\dot\theta_g+\gamma_2\,e^TP\,b_c\frac{\partial\hat g(x,\,\theta_g)}{\partial\,\theta_g}u_c] \tag{4.10}$$

Substituting (4.5)-(4.8) into (4.10), we have

$$\dot V=-\frac{1}{2}\,e^TQe-g(x)\,e^TP\,b_cu_s+\,e^TP\,b_cv$$
$$+I_1\,e^TP\,b_c\frac{\phi_f^T\,\theta_f}{|\,\theta_f|^2}\,\theta_f^T\frac{\partial\hat f(x,\,\theta_f)}{\partial\,\theta_f}$$
$$+I_1\,e^TP\,b_c\frac{\phi_g^T\,\theta_g}{|\,\theta_g|^2}\,\theta_g^T\frac{\partial\hat g(x,\,\theta_g)}{\partial\,\theta_g}u_c \tag{4.11}$$

where $I_1=0$ in case of (4.5),(4.7) and $I_1=1$ in case of (4.6),(4.8). Now we show that the last two terms of (4.11) are negative. Though (4.11) results in (4.12) directly in case of $I_1=0$, the case of $I_1=1$ becomes

$$\phi_f^T\,\theta_f=(\,\theta_f-\,\theta_f^*)^T\,\theta_f=\frac{1}{2}[|\,\theta_f|^2-|\,\theta_f^*|^2+|\,\theta_f-\,\theta_f^*|^2]\ge 0$$

$$\phi_g^T\,\theta_g=(\,\theta_g-\,\theta_g^*)^T\,\theta_g=\frac{1}{2}[|\,\theta_g|^2-|\,\theta_g^*|^2+|\,\theta_g-\,\theta_g^*|^2]\ge 0$$

Therefore, the term with I_1 is nonpositive and we have

$$\dot V\le-\frac{1}{2}\,e^TQe-g(x)\,e^TP\,b_cu_s+\,e^TP\,b_cv \tag{4.12}$$

From (3.7) and $g(x)>0$, we have $g(x)\,e^TP\,b_cu_s\ge 0$; therefore, (4.12) can be further simplified to

$$\dot V\le-\frac{1}{2}\,e^TQe+\,e^TP\,b_cv$$
$$\le-\frac{\lambda_{Qmin}-1}{2}|e|^2-\frac{1}{2}[|e|^2-2\,e^TP\,b_cv+|P\,b_cv|^2]$$
$$+\frac{1}{2}|P\,b_cv|^2$$
$$\le-\frac{\lambda_{Qmin}-1}{2}|e|^2+\frac{1}{2}|P\,b_cv|^2 \tag{4.13}$$

where λ_{Qmin} is the minimum eigenvalue of Q.

5. Convergence Analysis

This section presents that all the signals are bounded and closed-loop systems are stable.

Theorem: The overall control scheme guarantees the following properties:

(1) $|\,\theta_f|\le M_f,\ |\,\theta_g|\le M_g,\ \forall\ \theta_g\ge\varepsilon,$

$$|\,x(t)|\le|\,y_m|+\sqrt{\frac{2V}{\lambda_{Pmin}}}, \tag{5.1}$$

$$\text{and, }|u(t)|\le\frac{1}{\varepsilon}(M_f+|\,y_m^{(n)}|+|k|\sqrt{\frac{2V}{\lambda_{Pmin}}})$$
$$+\frac{1}{g_l(x)}[M_f+|f_u(x)|$$
$$+\frac{1}{\varepsilon}(M_g+g_u(x))(M_f+|y_m^{(n)}|+|k|\sqrt{\frac{2V}{\lambda_{Pmin}}})] \tag{5.2}$$

where λ_{Pmin} is the minimum eigenvalue of P, and $y_m=[y_m\ \dot y_m\cdots y_m^{(n-1)}]^T$.

(2) $$\int_0^t|e(\tau)|^2d\tau\le a+b\int_0^t|v(\tau)|^2d\tau \tag{5.3}$$

where a and b are constants.

(3) If v is square integrable, that is, $\int_0^\infty|v(t)|^2dt<\infty$, then

$$\lim_{t\to\infty}|\,e(t)|=0\ . \tag{5.4}$$

We omit the proof of this theorem owing to the limitation of space.

The structure of the overall system for feedback linearization using NNs is shown in Fig.1.

6. Simulation

We use our two adaptive NN controllers to control an inverted pendulum to track a sine-wave trajectory. Let $x_1=\theta$ and $x_2=\dot\theta$. The dynamic equations of the inverted pendulum system[1] are

$$\dot x_1=x_2$$

$$\dot x_2=\frac{g\sin x_1-\dfrac{ml\,x_2^2\cos x_1\sin x_1}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2x_1}{m_c+m}\right)}+\frac{\dfrac{\cos x_1}{m_c+m}}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2x_1}{m_c+m}\right)}u$$

$$y=x_1$$

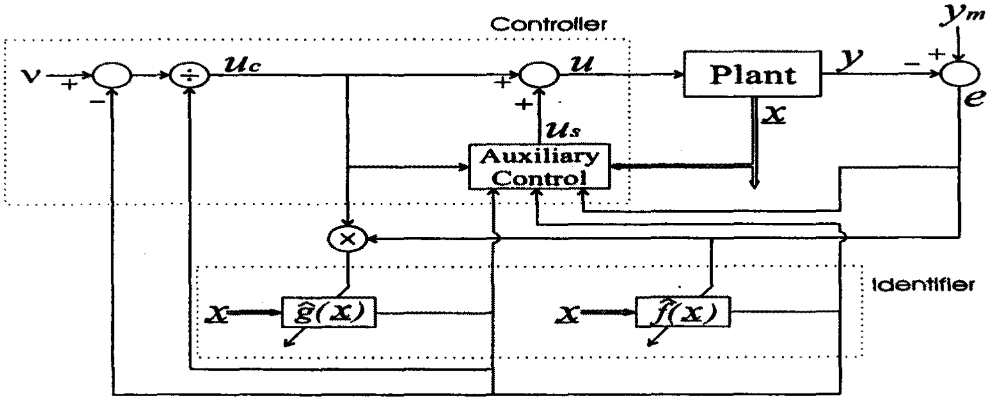

Fig. 1. The structure of the overall system

where $m_c = 1\,kg$, $m = 0.1kg$, $l = 0.5m$.

Design parameters are set to $k_1 = 3$, $k_2 = 1$, $M_f = 10.0$, $M_g = 2.58$, $\Delta = 0.1$, $Q = \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}$ and $P = \begin{bmatrix} 8 & 2 \\ 2 & 1 \end{bmatrix}$. Sampling time is $5\,ms$. Supposing that $|x_1| \le \frac{\pi}{6}$ and $|u| \le 100$, we have $f_u(x) = 15.78$, $g_u(x) = 1.46$ and $g_l(x) = 1.12$. We see that the range of $f(x)$ is much larger than that of $g(x)$; therefore, we choose $\gamma_1 = 140$ and $\gamma_2 = 1.2$. But if γ_1 and γ_2 are chosen smaller, then actual output converges slower towards desired output. Also, u_s becomes greater than u_c. The NN is a one hidden layer with 10 hidden neurons plus bias. Initial conditions of the NN are $\underline{\theta}_f(0) = 0.001$ and $\underline{\theta}_g(0) = 0.5$, so that $\widehat{g}(0) > 0$ and $x(0) = (\frac{\pi}{60}, 0)^T$. The desired trajectory is defined as $y_m(t) = \frac{\pi}{30} \sin t$. Actual and desired outputs are shown in Fig.2. Actual output converges very quickly towards desired output. Fig.3 shows the state $x_2(t)$ and its desired value $\dot{y}_m(t) = \frac{\pi}{30} \cos t$, and Fig.4 shows the control input.

Almost perfect tracking is obtained in less than 1 s.

7. Conclusion

We proposed in this paper a learning algorithm for multilayer feedforward NN used in indirect adaptive control. This algorithm takes explicitly into account the approximation error of the network and the error due to the Taylor's series expansion. Simulation results show the effectiveness of the approach. Further studies are in progress to cancel the hypothesis that v is square integrable.

8. References

[1] J. E. Slotine, W. Li., Nonlinear Applied Control, Englewood Cliffs, NJ, Prentice Hall, 1991.

[2] Isidori A., Nonlinear Control Systems: An Introduction, Springer-Verlag, Berlin, 1995.

[3] R. Marino, P. Tomei, Nonlinear Control Design: Geometric, Adaptive and Robust, Prentice Hall, 1995.

[4] Hassan K. Khalil, Nonlinear Systems, Prentice Hall, 1996.

[5] L. X. Wang, "Stable Adaptive Fuzzy Control of Nonlinear Syatems", IEEE Trans. on Fuzzy Systems, vol. 1, no. 2, pp. 146-155, 1993.

[6] F. C. Chen, C. C. Liu, "Adaptive Control of Nonlinear Continuous-Time Systems Using Neural Networks-general relative degree and MIMO cases" Int. J. Control, vol. 58, no. 2, pp. 317-335, 1993.

[7] A. Yesildirek, F. L. Lewis, "Feedback Linearization Using Neural Networks", Automatica, vol. 31, no. 11, pp. 1659-1664, 1995.

[8] E. T. Hancock, F. Fallside, "Stable Control of Nonlinear Systems Using Neural Networks", Int. J. of Robust and Nonlinear Control, vol. 2, pp. 63-86, 1992.

[9] D. Lamy, "Stable Direct Adaptive Neurocontrol of Nonlinear System", IEEE conf. on systems, man and cybernetics, vancouver, pp. 2176-2181, 1995.

[10] K. S. Narendra, A. M. Annaswamy, Stable Adaptive Systems, Prentice Hall, 1989.

[11] P. A. Ioannou, J. Sun, Robust Adaptive Control, Prentice Hall, 1996.

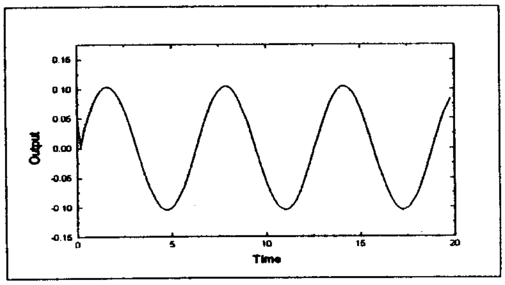

Fig. 2. actual and desired states : x_1(solid line), y_m(dotted line)

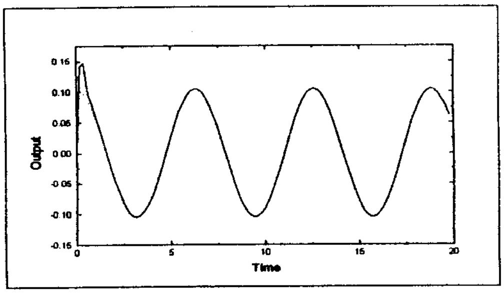

Fig. 3. actual and desired states : x_2(soild line), $\dot{y}_m$(dotted line)

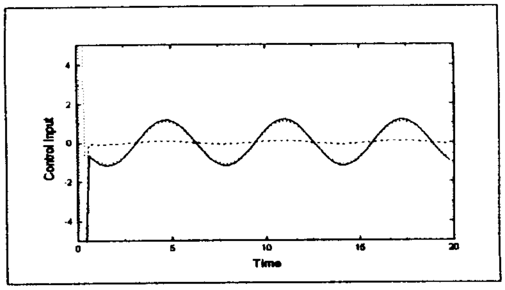

Fig. 4. control input : u(solid line), u_c (dotted line), u_s (dashed line)