Proceedings of the 11th
KACC, October 1996

Robust Control for External Input Perturbation using Second Order Derivative of Universal Learning Network

○Masanao OHBAYASHI*, Kotaro HIRASAWA*
* Dep. of Electrical and Electronic Systems Eng., Kyushu University, Higashiku, Fukuoka 812, JAPAN
Tel: +81-92-642-3959; Fax: +81-92-642-3962; E-mail:oba@scel.ees.kyushu-u.ac.jp

Abstract: This paper proposes a robust control method using Universal Learning Network(U.L.N.) and second order derivatives of U.L.N.. Robust control considered here is defined as follows. Even if external input (equal to reference input in this paper) to the system at control stage changes awfully from that at learning stage,the system can be controlled so as to maintain a good performance. In order to realize such a robust control, a new term concerning the perturbation is added to a usual criterion function. And parameter variables are adjusted so as to minimize the above mentioned criterion function using the second order derivative of the criterion function with respect to the parameters.

Keywords Universal learning network, Neural network, Robust control, Second order derivative, External input perturbation

1 Introduction

Universal Learning Network(U.L.N.) and a computing method for its higher order derivatives have been proposed [1][2], which can be used as a fundametal tool in modelling and control of large-scale complicated systems such as economic, social and living systems as well as industrial plants. In case of designing a control system using U.L.N., the system to be controlled and the controller are both constructed by U.L.N., and the controller is best tuned through learning to minimize a criterion function which is assumed to be function of the target value of system node output, actual value of system node output and output of the controller.

U.L.N. has the same generalization ability as Neural Network(N.N.). So the controller constructed by U.L.N. is able to control the system in a favorable way under the conditions different from those of the control system at learning stage. But stability and performance can not be realized sufficiently under the conditions much different from those at learning stage. One of such conditions is perturbation of the initial values of the system [3], another is perturbation of the system parameters [4]. The difference between reference input at control stage and that at learning stage is considered in this paper.

Finally it is shown that the controller constructed by the proposed method works in an effective way through a simulation study of a nonlinear crane system.

2 Structure of Universal Learning Network

Basic structure of U.L.N. which consists of nonlinearly operated nodes and branches that may have arbitrary time delays is shown in **Fig.1**.

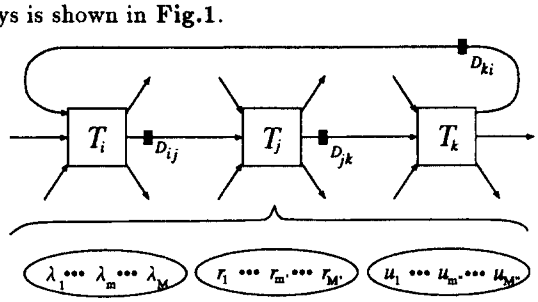

Fig.1 Structure of Universal Learning Network

Basic equation of U.L.N. is represented by Eq.(1):

$$h(T_j,t) = O_j(\{h(T_i,t-D_{ij}|i \in I)\},\{\lambda_m(t)\},\{r_{m'}(t)\},\{u_{m''}(t)\}), \quad (1)$$

where

$h(T_j,t)$	:	output of T_j node at time t, $(j \in R)$
$\lambda_m(t)$	:	mth parameter variable at time t,
$r_{m'}(t)$	:	m'th external input variable at time t,
$u_{m''}(t)$	:	m''th control variable at time t,
O_j	:	nonlinear function of T_j node,
D_{ij}	:	time delay from T_i node to T_j node,
I	:	set of nodes whose output are connected to T_j node,
R	:	set of suffixes for nodes,
T	:	set of sampling times.

Let a criterion function be written in Eq.(2):

$$E = E(\{h(T_r,s)\},\{u_{m''}(s)\},\{\lambda_m(s)\}) \quad (2)$$

$r \in R_o$	R_o	:	set of suffixes for nodes related with evaluation,
$m'' \in M_o''$	M_o''	:	set of suffixes for control variables related with evaluation,
$m \in M_o$	M_o	:	set of suffixes for parameter variables related with evaluation,
$s \in S_o$	S_o	:	set of sampling times related with evaluation.

In the following chapters,a computing method for derivative of criterion function E with respect to parameter variable $\lambda_m(t_0)$ is presented,which is essential to design a robust control system using U.L.N.[1][2].

3 Computation of Higher Order Derivative

3.1 First Order Derivative

First order derivative of E with respect to parameter $\lambda_1(t_0)$ can be written in the form of Eq.(3),assuming t_0 to be designated sampling time,

$$\frac{\partial^\dagger E}{\partial \lambda_1(t_0)} = \sum_{r\in R_o}\sum_{s\in S_o}\left(\frac{\partial E}{\partial h(T_r,s)}\frac{\partial^\dagger h(T_r,s)}{\partial \lambda_1(t_0)}\right) + \frac{\partial E}{\partial \lambda_1(t_0)} \quad (3)$$

As $\frac{\partial E}{\partial h(T_r,s)}$ and $\frac{\partial E}{\partial \lambda_1(t_0)}$ can be calculated easily from Eq.(2), it is a matter of importance to calculate $\frac{\partial^\dagger h(T_r,s)}{\partial \lambda_1(t_0)}$. Here † denotes the ordered derivative proposed by Werbos [5].

Generally, $\frac{\partial^\dagger h(T_k,t)}{\partial \lambda_1(t_0)}$ can be represented by Eq.(4).

$$\frac{\partial^\dagger h(T_k,t)}{\partial \lambda_1(t_0)} = \sum_{j\in J}\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\frac{\partial^\dagger h(T_j,t-D_{jk})}{\partial \lambda_1(t_0)}\right) + \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)} \quad (4)$$

where

J : set of suffixes for nodes whose outputs are connected to T_k node.

Putting $P_1(T_k,t,\lambda_1(t_0)) = \frac{\partial^\dagger h(T_k,t)}{\partial \lambda_1(t_0)}$, iterative equation of P_1 by forward propagation can be obtained from Eq.(4).

$$P_1(T_k,t,\lambda_1(t_0)) = \sum_{j\in J}\left[\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}P_1(T_j,t-D_{jk},\lambda_1(t_0))\right] + \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)} \quad (k\in R),\ (t\in T), \quad (5)$$

Since $h(T_j,t_0-1)$ does not depend on $\lambda_1(t_0)$, initial value of Eq.(5) is as follows.

$$P_1(T_j,t_0-1,\lambda_1(t_0)) = 0 \quad (j\in R). \quad (6)$$

3.2 Second Order Derivative

Second order derivative of E with respect to parameter variables $\lambda_1(t_0)$, $\lambda_2(t_0)$ can be obtained by differentiating Eq.(3) with respect to $\lambda_2(t_0)$,

$$\frac{\partial^{\dagger 2} E}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)} = \sum_{r\in R_o}\sum_{s\in S_o}\left[\frac{\partial^\dagger\left(\frac{\partial E}{\partial h(T_r,s)}\right)}{\partial \lambda_2(t_0)}\frac{\partial^\dagger h(T_r,s)}{\partial \lambda_1(t_0)} + \frac{\partial E}{\partial h(T_r,s)}\frac{\partial^{\dagger 2} h(T_r,s)}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}\right] + \frac{\partial^\dagger\left(\frac{\partial E}{\partial \lambda_1(t_0)}\right)}{\partial \lambda_2(t_0)} \quad (7)$$

$\frac{\partial^{\dagger 2} h(T_r,s)}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}$ in Eq.(7) can be represented by Eq.(8) by differentiating Eq.(4) with respect to $\lambda_2(t_0)$,

$$\frac{\partial^{\dagger 2} h(T_k,t)}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)} = \sum_{j\in J}\left[\frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{\partial \lambda_2(t_0)}\frac{\partial^\dagger h(T_j,t-D_{jk})}{\partial \lambda_1(t_0)} + \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\frac{\partial^{\dagger 2} h(T_j,t-D_{jk})}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}\right] + \frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{\partial \lambda_2(t_0)} \quad (8)$$

Putting $P_1(T_k,t,\lambda_1(t_0)) = \frac{\partial^\dagger h(T_k,t)}{\partial \lambda_1(t_0)}$,and $P_2(T_k,t,\lambda_1(t_0),\lambda_2(t_0)) = \frac{\partial^{\dagger 2} h(T_k,t)}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}$, as in the case of first order derivative, iterative equation of P_2 by forward propagation can be obtained from Eq.(8),

$$P_2(T_k,t,\lambda_1(t_0),\lambda_2(t_0)) = \sum_{j\in J}\left[\frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{\partial \lambda_2(t_0)}P_1(T_j,t-D_{jk},\lambda_1(t_0)) + \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}P_2(T_j,t-D_{jk},\lambda_1(t_0),\lambda_2(t_0))\right] + \frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{\partial \lambda_2(t_0)} \quad (k\in R),\ (t\in T). \quad (9)$$

$$P_2(T_j,t_0-1,\lambda_1(t_0),\lambda_2(t_0)) = 0 \quad (j\in R). \quad (10)$$

$\frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{\partial \lambda_2(t_0)}$, $\frac{\partial^\dagger\left(\frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}\right)}{\partial \lambda_2(t_0)}$ in Eq.(9) can be calculated by the computation of first order derivative, putting $E = \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}$, $E = \frac{\partial h(T_k,t)}{\partial \lambda_1(t_0)}$ respectively.

Substituting $\frac{\partial^\dagger h(T_k,t)}{\partial \lambda_1(t_0)}$, $\frac{\partial^{\dagger 2} h(T_k,t)}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}$ obtained from Eq.(5),(6) and Eq.(9),(10) respectively into Eq.(7), $\frac{\partial^{\dagger 2} E}{\partial \lambda_1(t_0)\partial \lambda_2(t_0)}$ can be calculated.

4 Robust Control Method

4.1 Criterion for suppressing perturbations of node outputs by change of the external input

E is a usual criterion function, E_H is a new term which takes charge of suppressing perturbations of node outputs of the system caused by the change of the external input at time t_1. Then a new criterion function L is defined as follows:

$$L = E + E_H, \quad (11)$$

$$E_H = \sum_{s\in S_L}\sum_{r\in R_s} CH_r\left(\sum_{m'\in I_p}\frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)}\Delta r_{m'}(t_1)\right)^2, \quad (12)$$

R_s : set of suffixes for nodes related with suppression,
I_p : set of suffixes for external inputs related with perturbation,
S_L : set of sampling times related with suppression,
$r_{m'}(t)$: m'th external input at time t
$CH_r > 0$: coefficient for T_r node.

$\frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)}\Delta r_{m'}(t_1)$ means the perturbation of T_r node output caused by the change of the external input $r_{m'}(t)$ at t_1. Eq.(12) is the sum of those squared.

4.2 Learning Algorithm

The aim of the optimization learning using U.L.N. is to search for the parameters which make the above criterion function L minimal. (From now on the parameter variables are considered to be time invariant.)

The parameter variables in order to minimize Eq.(11) can be calculated by a gradient method.

$$\lambda_m \quad \leftarrow \quad \lambda_m - \gamma \frac{\partial^\dagger L}{\partial \lambda_m} \tag{13}$$

$$where \quad \frac{\partial^\dagger L}{\partial \lambda_m} = \frac{\partial^\dagger E}{\partial \lambda_m} + \frac{\partial^\dagger E_H}{\partial \lambda_m},$$

$\gamma > 0$: coefficient.

Now,computation of $\frac{\partial^\dagger E}{\partial \lambda_m}$ and $\frac{\partial^\dagger E_H}{\partial \lambda_m}$ can be carried out by making use of the first and the second order derivative in chapter 3.

[A] Computation of $\frac{\partial^\dagger E}{\partial \lambda_m}$

Putting $\lambda_1(t_0) = \lambda_m$, $\frac{\partial^\dagger E}{\partial \lambda_m}$ can be calculated using Eq.(3),(5)

[B] Computation of $\frac{\partial^\dagger E_H}{\partial \lambda_m}$

First order derivative of E_H with respect to λ_m can be obtained by differentiating Eq.(12) with respect to λ_m,

$$\frac{\partial^\dagger E_H}{\partial \lambda_m} = 2 \sum_{s \in S_L} \sum_{r \in R_s} CH_r \Big[\Big(\sum_{m' \in I_p} \frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)} \Delta r_{m'}(t_1)\Big) \times \Big(\sum_{m' \in I_p} \frac{\partial^{\dagger 2} h(T_r,s)}{\partial r_{m'}(t_1) \partial \lambda_m} \Delta r_{m'}(t_1)\Big)\Big] \tag{14}$$

Now, $\frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)}, \frac{\partial^{\dagger 2} h(T_r,s)}{\partial r_{m'}(t_1)\partial \lambda_m}$ are necessary for the computation of $\frac{\partial^\dagger E_H}{\partial \lambda_m}$.

< Computation of $\frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)}$ >

Putting $E=h(T_r,s)$, $\lambda_1(t_0)=r_{m'}(t_1)$ and making use of the first order derivative in chapter 3, Eq.(15),(16) can be obtained,

$$\frac{\partial^\dagger h(T_r,s)}{\partial r_{m'}(t_1)} = P_1(T_r, s, r_{m'}(t_1)). \tag{15}$$

$$P_1(T_k,t,r_{m'}(t_1)) = \sum_{j \in J} \Big[\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})} P_1(T_j, t-D_{jk}, r_{m'}(t_1))\Big] + \frac{\partial h(T_k,t)}{\partial r_{m'}(t_1)}. \tag{16}$$

< Computation of $\frac{\partial^{\dagger 2} h(T_r,s)}{\partial r_{m'}(t_1)\partial \lambda_m}$ >

Putting $E=h(T_r,s)$, $\lambda_1(t_0)=r_{m'}(t_1)$, $\lambda_2(t_0) = \lambda_m$ and making use of the second order derivative in chapter 3, Eq.(17),(18) can be obtained,

$$\frac{\partial^{\dagger 2} h(T_r,s)}{\partial r_{m'}(t_1)\partial \lambda_m} = P_2(T_r, s, r_{m'}(t_1), \lambda_m). \tag{17}$$

$$P_2(T_k,t,r_{m'}(t_1),\lambda_m) = \sum_{j \in J} \Big[\frac{\partial^\dagger \Big(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\Big)}{\partial \lambda_m} P_1(T_j, t-D_{jk}, r_{m'}(t_1)) + \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})} P_2(T_j, t-D_{jk}, r_{m'}(t_1), \lambda_m)\Big] + \frac{\partial^\dagger \Big(\frac{\partial h(T_k,t)}{\partial r_{m'}(t_1)}\Big)}{\partial \lambda_m} \tag{18}$$

Both the coefficient of P_1 in Eq.(18), $\frac{\partial^\dagger \left(\frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}\right)}{\partial \lambda_m}$, and $\frac{\partial^\dagger \left(\frac{\partial h(T_k,t)}{\partial r_{m'}(t_1)}\right)}{\partial \lambda_m}$ can be calculated by computing the first order derivative of $E = \frac{\partial h(T_k,t)}{\partial h(T_j,t-D_{jk})}$ and $E = \frac{\partial h(T_k,t)}{\partial r_{m'}(t_1)}$ respectively using Eq.(3),(5).

5 Numerical Example

5.1 Controlled System

The controlled system is a nonlinear crane system. A position of the crane stand, an angle between the rope and vertical line and a position of the load are represented by x, θ, l respectively. Then the nonlinear crane system is described as follows:

$$\begin{aligned} \frac{d^2x}{dt^2} &= -\frac{mg}{M}\theta - \frac{D+G}{M}\frac{dx}{dt} + \frac{G}{M}u_1, \\ \frac{d^2\theta}{dt^2} &= -\frac{M+m}{lM}g\theta - \frac{D+G}{lM}\frac{dx}{dt} + \frac{G}{lM}u_1, \\ \frac{d^2l}{dt^2} &= -\frac{C+G_m}{m}\frac{dl}{dt} + \frac{G_m}{m}u_2, \end{aligned} \tag{19}$$

where, u_1, u_2 are input voltage to a motor for moving the crane stand and to a motor for rolling up the load respectively,and C, G, G_m, D, M, m are appropriate system parameters.

Putting

$$h(T_1,t) = x(t), \quad h(T_2,t) = \dot{x}(t), \quad h(T_3,t) = \theta(t),$$
$$h(T_4,t) = \dot{\theta}(t), \quad h(T_5,t) = l(t), \quad h(T_6,t) = \dot{l}(t),$$

Eq.(19) can be transformed into discrete type equations as follows:

$$\begin{aligned} h(T_1,t) &= a_{11}h(T_1,\hat{t}) + a_{21}h(T_2,\hat{t}), \\ h(T_2,t) &= a_{22}h(T_2,\hat{t}) + a_{32}h(T_3,\hat{t}) + b_1u_1(\hat{t}), \\ h(T_3,t) &= a_{33}h(T_3,\hat{t}) + a_{43}h(T_4,\hat{t}), \\ h(T_4,t) &= a_{24}\frac{h(T_2,\hat{t})}{h(T_5,\hat{t})} + a_{34}\frac{h(T_3,\hat{t})}{h(T_5,\hat{t})} \\ &\quad + a_{44}h(T_4,\hat{t})\frac{b_1}{h(T_5,\hat{t})}u_1(\hat{t}), \\ h(T_5,t) &= a_{55}h(T_5,\hat{t}) + a_{65}h(T_6,\hat{t}), \\ h(T_6,t) &= a_{66}h(T_6,\hat{t}) + b_2u_2(\hat{t}). \\ & \quad where\ \hat{t} = t-1. \end{aligned} \tag{20}$$

A control model of the nonlinear crane system using U.L.N. is shown in **Fig.2**. Each control input u_1, u_2 is constructed by two control nodes respectively ,one is the node with linear function,the other is the node with tanh function. (All branches have one sampling time delay.)

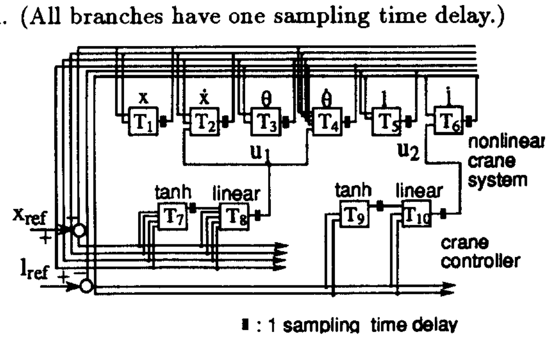

Fig.2 Control model of a nonlinear crane system using Universal Learning Network

5.2 Criterion Function

M=40[kg], D=300[kg/sec], G=700[N/V], m=2[kg],g=9.8 [m/sec^2],G_m=0.98[N/V], C=0.42[kg/sec] are used, and reference of moving the crane stand(x_{ref}) is 1[m], reference of rolling up the load(l_{ref}) is 0.5[m], assuming initial positions of the crane stand and the load to be 0[m], 1[m] respectively.

Therefore, when the parameter variables are tuned through learning, initial values of node outputs of the system are set up as follows:

$$h(T_i,0) = \begin{cases} 1.0 & i=5 \\ 0.0 & otherwise \end{cases} \qquad (21)$$

In numerical example, a step change of reference input at initial time is assumed, namely $r_{m'}(t_1) = l_{ref}$.

The criterion E to achieve the desired dynamics of the system and E_H to achieve the suppression of the perturbation of the system caused by the change of the reference of the position of the load are defined as follows respectively,

$$\begin{aligned} E = & \frac{1}{2}\Big[\sum_{s\in S_o}\{Q_{11}(x_{ref}-h(T_1,s))^2\} + Q_{12}(h(T_2,t_f))^2 \\ & + \sum_{s\in S_o}\{Q_{13}(h(T_3,s))^2 + Q_{14}(h(T_4,s))^2\} \\ & + \sum_{s\in S_o}\{Q_{15}(l_{ref}-h(T_5,s))^2\} + Q_{16}(h(T_6,t_f))^2 \\ & + \sum_{s\in S_o}\{R_1(h(T_8,s))^2 + R_2(h(T_{10},s))^2\}\Big] \qquad (22) \end{aligned}$$

$$E_H = \sum_{s\in S_L}\sum_{r\in R_s} CHr\Big(\frac{\partial^\dagger h(T_r,s)}{\partial l_{ref}}\Delta l_{ref}\Big)^2 \qquad (23)$$

Using the values of **Table 1**, two cases have been studied, one(named $case_E$) is the case of using the criterion function E, the other(named $case_L$) is the case of using the criterion function L. A total number of learning is four thousand.

Table 1 Simulation Conditions

S_L	:	set of all sampling times
R_s	:	nodes related with x, $\dot{x}$, θ, and $\dot{\theta}$
t_f	:	final time (= 7.5[sec])
Q_{11}	:	0.5
Q_{21}	:	10.0
Q_{others}	:	1.0
$R_1 \sim R_2$	:	0.1
Δl_{ref}	:	1, (CH_r is represented in **Figure**.)

5.3 Simulation Results

Control results of both $case_E$ and $case_L$ at learning stage are shown in **Fig.3**. **Fig.4**,.**5** show the control results in case of l_{ref} = 3.0, 6.0 at control stage respectively. For example, x, θ of $case_E$ are oscillating in **Fig.4**,especially in **Fig.5**.

From these results, it is found that the bigger difference between l_{ref} at control stage and l_{ref} at learning stage is, the worse the performance of $case_E$ is , in spite of $case_L$'s better performance.

6 Conclusion

In this paper, a new robust control method is proposed for a change of the reference input and it is found that the method is very useful through simulation study.

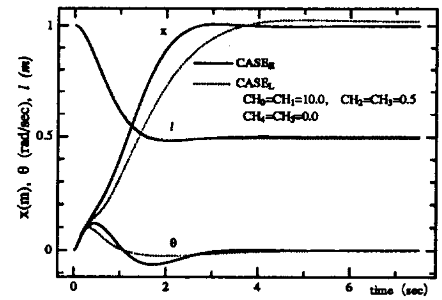

Fig.3 Control results at learning stage : l_{ref} = 0.5

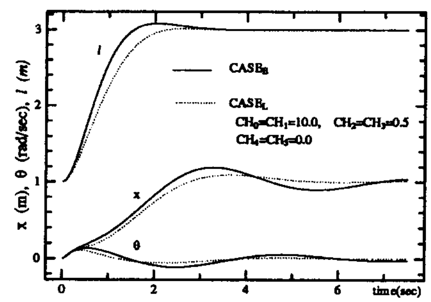

Fig.4 Control results at control stage : l_{ref} = 3.0

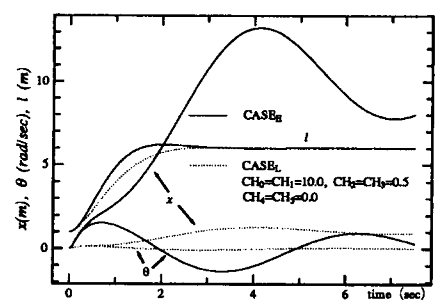

Fig.5 Control results at control stage : l_{ref} = 6.0

References

[1] K.Hirasawa,et al:**Forward Propagation Universal Learning Network** ,Proc. of 1996 IEEE ICNN, pp.353-358,1996.

[2] K.Hirasawa,et al:**Universal Learning Network and Computation of its Higher Order Derivatives**,Proc. of 1995 IEEE ICNN, pp.1273-1277,1995.

[3] M.Ohbayashi,et al:**Robust Learning Control using Universal Learning Network**, Proc. of 1996 IEEE ICNN, pp.2208-2213,1996.

[4] M.Ohbayashi,et al:**Robust Control using Second Order Derivative of Universal Learning Network** ,Proc. of 1996 IEEE SMC,1996.

[5] P.Werbos:**New Tools for Prediction and Analysis in the Behavioral Sciences**,Ph.D dissertation,Harvard university,1974.