Nonlinear Control System Using Universal Learning Network with RBF and RasVal

Shao, Ning
Department of Electrical and Electronic Systems Engineering, Kyushu University : Graduate Student

Hirasawa, Kotaro
Department of Electrical and Electronic Systems Engineering, Kyushu University

Ohbayashi, Masanao
Department of Electrical Engineering and Computer Science, Faculty of Engineering, Kyushu University

Murata, Junichi
Department of Electrical and Electronic Systems Engineering, Kyushu University

他

https://doi.org/10.15017/1522391

出版情報：九州大学大学院システム情報科学紀要. 2 (2), pp.197-212, 1997-09-26. Graduate School of Information Science and Electrical Engineering, Kyushu University

九州大学大学院
システム情報科学研究科報告
第2巻 第2号 平成9年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.2, September 1997

Nonlinear Control System Using Universal Learning Network with RBF and RasVal

Ning SHAO* , Kotaro HIRASAWA**, Masanao OHBAYASHI***,
Junichi MURATA**, Kazuyuki TOGO† and Jinglu HU††

(Received June 23, 1997)

Abstract: In this paper, we present a new control method firstly for nonlinear systems using Universal Learning Network(ULN) with radial basis function(RBF). ULN can model and control the large scale complicated systems such as industrial plants, economic, social and life phenomena. The basic idea of ULN is that large scale complicated control systems can be modeled by the network which consists of nonlinearly operated nodes and branches which may have arbitrary time delays including zero or minus ones. Second, a new learning algorithm is applied to the design of the optimal network controller of a nonlinear control system. The optimization method is named RasVal, which is a kind of random searching, and it can search for a global minimum systematically and effectively in a single framework which is not a combination of different methods. The searching for a global minimum is carried out based on the probability density functions of searching, which can be modified using information on success or failure of the past searching in order to execute intensified and diversified searching.
Simulation studies were carried out in the following four cases to compare the learning speed and performance: (1). comparing Radial Basis Function(RBF) with Sigmoid Function(SF) based on the gradient method. (2). comparing RBF with SF based on RasVal. (3). comparing RasVal with the gradient method for the RBF controller. (4). comparing RasVal with the gradient method for the SF controller.
By applying RasVal and the gradient method to a nonlinear crane control system, it has been proved that the simulation results of ULN with RBF based on the gradient method are superior in performance to those of neural networks, and it has also been shown that the RBF control has better performance for the generalization capability than the neural network control based on the gradient method. On the other hand, it has been shown that the neural network control based on RasVal has better performance than the RBF control.
At the same time, it has been shown that the RasVal is superior in performance to the commonly used back propagation learning algorithm whether the RBF controller or the NN controller is used. On the other hand, the generalization capability of a Radial Basis Function controller using RasVal was studied and it is shown that a new method is effective to overcome the over-fitting problem in nonlinear control systems.

Keywords: Universal learning network, Random search method, Nonlinear control, Neural network, Radial basis function, Generalization capability

1. Introduction

Nowadays, neural networks have been widely used in the control fields because of their remarkable ability to control the nonlinear systems. A neural network which has only one hidden layer can approximate exactly any nonlinear system theoretically. To achieve this, a large number of hidden units are needed. That is, the structure of the neural networks become complicated and the learning of the network becomes very time-consuming. This is the main problem to be solved for neural networks. On the other hand, the radial basis function network is taken notice with regard to the learning speed and performance. We believe that in the future the radial basis function will be widely used to control systems. But, there are some unknown problems that should be solved when the radial ba-

* Department of Electrical and Electronic Systems Engineering, Graduate Student

** Department of Electrical and Electronic Systems Engineering

*** Department of Electrical Engineering and Computer Science, Faculty of Engineering

† Department of Energy Conversion Engineering, Interdisciplinary Graduate School of Engineering Sciences, Graduate Student

†† Venture Business Laboratory

sis functions are applied to the nonlinear control systems.

A new control method has been presented for the nonlinear control systems using Universal Learning Network with the radial basis function[3]. Universal Learing Network (ULN)[1] is a new-type of network which can be used to model and control large-scale complicated systems such as economic, social and living phenomena as well as industrial plants. Universal Learning Network consists of nonlinearly operated nodes and multi-branches that may have arbitrary time delays between the nodes.

In the above system using Universal Learning Network, as learning algorithm of parameters in the controller was based on the gradient method, the problem of falling into a local minimum that leads to low efficiency of learning could not be solved. To overcome this problem, a new learning algorithm that can search for a global minimum has been presented and it was applied to build the optimal controller of a nonlinear control system. The proposed learning algorithm is called RasVal[2] which is an abbreviation of Random Search with Variable Search Length and it can search for a global minimum systematically and effectively in a single framework which is not a combination of different methods. RasVal is a kind of random search based on the probability density function of searching, which can be modified using information on the results of the past searching in order to execute the intensified and diversified searching. The features of RasVal are as follows.

(1) it does not require differential calculations as the gradient method, therefore, it takes a shorter calculation time than the gradient method.

(2) random search with the intensification and diversification is carried out in order to solve the local minimum problem.

In this paper, simulation studies were carried out in the following four cases to compare the learning speed and performance:

(1) comparing radial basis function(RBF) with sigmoid function(SF) based on gradient method.

(2) comparing radial basis function(RBF) with sigmoid function(SF) based on RasVal.

(3) comparing RasVal with gradient method for the RBF controller.

(4) comparing RasVal with gradient method for the NN controller.

By applying RasVal and the gradient method to a nonlinear crane control system, it has been proved that the simulation results of ULN with RBF based on the gradient method are superior in performance to those of neural networks, and it has also been shown that the RBF control has better performance for the generalization capability than the neural network control based on the gradient method. On the other hand, it has been shown that the neural network control based on RasVal has better performance than the RBF control.

At the same time, it has been shown that the RasVal is superior in performance to the commonly used back propagation learning algorithm whether the RBF controller or the NN controller is used.

In the simulations to study the generalization capability of RasVal, it was found that too much learning causes the over-fitting problem, that is, the control system becomes unstable at the different condition from that of learning. In the reference 5), a new method to overcome the over-fitting problem in the nonlinear control systems was proposed, where the weighting coefficients of control variables in the criterion function are increased in order to obtain the generalization capability of RasVal. From simulation results of a nonlinear crane system, it has been shown that the smaller the scale of the RBF controller is, the smaller the weighting coefficients of the control variables could be.

2. Universal Learning Network

The structure of the ULN is shown in **Fig.1.**

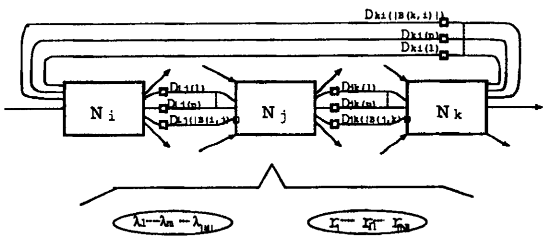


Fig.1 Structure of ULN with Multi Branches

Basic equation of ULN is represented by Eq.(1):

$$h_j(t) = O_j(\{h_i(t - D_{ij}(p))|i \in JF(j), p \in B(i,j)\}, \{r_n(t)|n \in N(j)\}, \{\lambda_m(t)|m \in M(j)\}) \quad (1)$$
$$j \in J, \quad t \in T$$

where,

$h_j(t)$: output value of node j at time t;

$\lambda_m(t)$: value of m th parameter at time t;

$r_n(t)$: value of n th external input variable at time t;

O_j: nonlinear function of node j;

$D_{ij}(p)$: time delay of p th branch from node i to

node j ;
$JF(j)$: set of numbers of nodes whose outputs are connected to node j;
$JB(j)$: set of numbers of nodes whose inputs are connected from node j;
$B(i,j)$: set of numbers of branches from node i to node j;
$N(j)$: set of numbers of external input variables that are fed into node j;
N: set of numbers of external input variables;
$M(j)$: set of numbers of parameters that are included in node j;
M: set of numbers of parameters;
J: set of numbers of nodes;
T: set of sampling times;

ULN is trained so as to minimize a criterion(evaluation) function which can be generally written as,

$$E = E\left(\{h_r(s)\}, \{\lambda_m(s)\}\right) \tag{2}$$

$$r \in J_0, m \in M_0, s \in T_0$$

where
J_0: set of numbers of nodes related with evaluation;
M_0: set of numbers of parameters related with evaluation;
T_0: set of sampling times related with evaluation.

The important features of ULN are such that the function of the nodes can take any nonlinear function and that the nodes can be connected arbitrarily. So the structure of ULN is a general one in the sense that ULN with sigmoid functions and one sampling time delays corresponds to the recurrent neural network.

Therefore, ULNs form a superset of all kinds of neural network paradigms with supervised learning capability.

3. ULN with RBF and NN

3.1 Learning algorithm of ULN with RBF

ULN with RBF can be expressed as follows.

$$h_j(t) = \sum_{m\in L(j)} [f_{jm}(x_{jm})] + b_j \tag{3}$$

$$f_{jm}(x_{jm}) = k_{jm} exp(x_{jm}) \tag{4}$$

$$x_{jm} = -\frac{1}{2} \sum_{i\in JF(j)} \sum_{p\in B(i,j)} \left(\frac{h_i(t-D_{ij}(p)) - h^i_{jm}(p)}{\sigma^i_{jm}(p)}\right)^2 \tag{5}$$

where,
$L(j)$: set of numbers of nonlinear functions of node j ;
k_{jm},$h^i_{jm}(p)$,$\sigma^i_{jm}(p)$,b_j: parameters for node j.

Learning algorithms of ULN with RBF based on the gradient method is shown below.

$$k_{jm} \leftarrow k_{jm} - \gamma \left\{ \sum_{t\prime\in T} \left[\frac{\partial h_j(t\prime)}{\partial k_{jm}} \delta(j,t\prime)\right] + \frac{\partial E}{\partial k_{jm}} \right\} \tag{6}$$

$$h^i_{jm}(p) \leftarrow h^i_{jm}(p) - \gamma \left\{ \sum_{t\prime\in T} \left[\frac{\partial h_j(t\prime)}{\partial h^i_{jm}(p)} \delta(j,t\prime)\right] + \frac{\partial E}{\partial h^i_{jm}(p)} \right\} \tag{7}$$

$$\sigma^i_{jm}(p) \leftarrow \sigma^i_{jm}(p) - \gamma \left\{ \sum_{t\prime\in T} \left[\frac{\partial h_j(t\prime)}{\partial \sigma^i_{jm}(p)} \delta(j,t\prime)\right] + \frac{\partial E}{\partial \sigma^i_{jm}(p)} \right\} \tag{8}$$

$$b_j \leftarrow b_j - \gamma \left\{ \sum_{t\prime\in T} \left[\frac{\partial h_j(t\prime)}{\partial b_j} \delta(j,t\prime)\right] + \frac{\partial E}{\partial b_j} \right\} \tag{9}$$

$$\delta(j,t) = \sum_{k\in JB(j)} \sum_{p\in B(j,k)} \frac{\partial h_k(t+D_{jk}(p))}{\partial h_j(t)} \delta(k, t+D_{jk}(p)) + \frac{\partial E}{\partial h_j(t)} \tag{10}$$

where, E: criterion function.

3.2 Learning algorithms of ULN with SF

For ULN with sigmoid functions Eq.(1) can be specifically rewritten as follows,

$$h_j(t) = \frac{1-e^{-x}}{1+e^{-x}} \tag{11}$$

$$x = \sum_{i\in JF(j)} \sum_{p\in B(i,j)} [w_{ij}(p) h_i(t-D_{ij}(p))] + b_j \tag{12}$$

where,
b_j: the threshold of node j.
$w_{ij}(p)$: the weight associated with the branch from node i to j.

Learning algorithms of ULN with SF based on the gradient method are as follows.

$$w_{ij}(p) \leftarrow w_{ij}(p) - \gamma \left\{ \sum_{t\prime\in T} \left[\frac{\partial h_j(t\prime)}{\partial w_{ij}(p)} \delta(j,t\prime)\right] + \frac{\partial E}{\partial w_{ij}(p)} \right\} \tag{13}$$

4. Random Search Method with Variable Search Length — RasVal

When learning algorithm of parameters in the controller is based on the gradient method, the problem of falling into a local minimum that leads to low efficiency of learning can not be solved. In this section, a new learning algorithm named RasVal is presented and it is applied to build the optimal controller of a nonlinear control system. RasVal can search for a global minimum systematically and effectively in a single framework which is not a combination of different methods. RasVal is a kind of random search based on the probability density function of searching, which can be modified using informations on the results of the past searching in order to execute intensified and diversified searching. The features of RasVal are such that it does not require differential calculation as gradient method, therefore, it takes a shorter calculation time than the gradient method, and random search with intensification and diversification is carried out in order to solve the local minimum problem.

Calculation procedure of RasVal is as follows.

$$if \quad E(\lambda + x) < E(\lambda) \Longrightarrow \lambda \longleftarrow \lambda + x; \tag{14}$$

(Searching is a success)

$$if \quad E(\lambda + x) \geq E(\lambda) \Longrightarrow \lambda \longleftarrow \lambda. \tag{15}$$

(Searching is a failure)

where,
E : criterion function;
$\lambda = [\lambda_1 ... \lambda_m ... \lambda_{|M|}]^T$: parameter vector;
$x = [x_1 ... x_m ... x_{|M|}]^T$: parameter search vector.
The probability density function $f(x_m)$ of searching x_m(see **Fig.2**) is represented by Eq.(16) and (17):

$$f(x_m) = \begin{cases} p_m \beta e^{\beta x_m}, & x_m \leq 0 \\ q_m \beta e^{-\beta x_m}, & x_m > 0 \end{cases} \tag{16}$$

$$p_m + q_m = 1 \tag{17}$$

Therefore x_m can be calculated as follows:

$$if \quad 0 \leq z \leq p_m \quad \Longrightarrow x_m = \frac{1}{\beta} \ln(\frac{z}{p_m}) \tag{18}$$

$$if \quad p_m < z \leq 1.0 \Longrightarrow x_m = -\frac{1}{\beta} \ln(\frac{1-z}{q_m}) \tag{19}$$

where, z : random numbers in [0,1].
Parameters β, p_m, q_m of $f(x_m)$ which are related to searching range and direction are modified based on the information of success or failure of the past searching as follows.

$$\beta = \bar{\beta} e^{-\eta n} + \underline{\beta} \tag{20}$$

In the case of negative direction searching :

$$p_m \longleftarrow \alpha p_m + (1 - \alpha) \cdot SF \tag{21}$$

In the case of positive direction searching :

$$q_m \longleftarrow \alpha q_m + (1 - \alpha) \cdot SF \tag{22}$$

In the case of failure,

$$n \longleftarrow n + 1 \tag{23}$$

In the case of success,

$$n \longleftarrow n, \qquad n = 0; \tag{24}$$

$$n \longleftarrow n - 1, \quad 0 < n \leq n_0 \tag{25}$$

$$n \longleftarrow n_0, \qquad n > n_0 \tag{26}$$

where,
$SF = 1.0$, in the case of success;
$SF = 0.0$, in the case of failure;
α : exponential filter coefficient;
$\bar{\beta} + \underline{\beta}$: upper limit of β;
$\underline{\beta}$: lower limit of β;
η : coefficient.

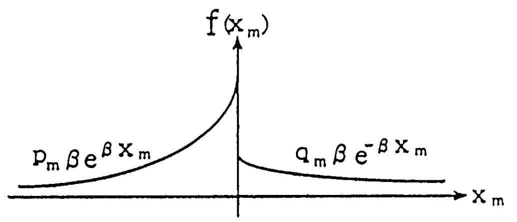


Fig.2 Probability Density Function of Searching x_m

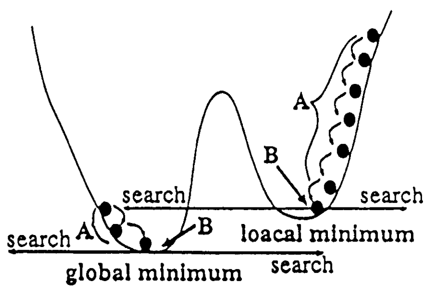


Fig.3 Explanation of Intensification and Diversification

From **Fig.3** and Eq.(14) $\sim$ (26), intensification and diversification of the search can be realized; when there is quite a possibility of finding good solutions around the current one(as A in **Fig.3**), intensified search for the vicinity of the current solution is carried out; on the other hand, when there is no possibility of finding good solutions(as B in **Fig.3**), diversified search is executed in order to find good solutions in the region far from the current solution.

5. Simulations

5.1 Nonlinear Crane Control System

A nonlinear crane system(**Fig.4**) is studied in order to compare the performance of the controller with RBF and NN and to compare the performance of the gradient learning and RasVal learning. The aim of the control is to bring the trolley to the target position, and to winch the load to the target height at the same time while making the angle of the load as small as possible.

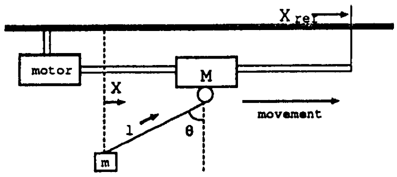


Fig.4 Nonlinear Crane System

The equations of the crane system are represented by the following:

$$\ddot{x} = \frac{mg}{M}\theta - \frac{D+G}{M}\dot{x} + \frac{G}{M}u_d \tag{27}$$

$$\ddot{\theta} = \frac{M+m}{lM}g\theta - \frac{D+G}{lM}\dot{x} + \frac{G}{lM}u_d \tag{28}$$

$$\ddot{l} = \frac{C+G_m}{m}\dot{l} + \frac{G_m}{m}u_m \tag{29}$$

where,

M: mass of the trolley;

m: mass of the load; l: height of the load;

θ: angle of the load;

x: location of the trolley;

C, D: coefficients of the friction.

u_d, u_m: input voltage to the system from the controller to the crane system.

Assuming the following,

$$h_1(t) = x(t) \quad h_2(t) = \dot{x}(t)$$
$$h_3(t) = \theta(t) \quad h_4(t) = \dot{\theta}(t)$$
$$h_5(t) = l(t) \quad h_6(t) = \dot{l}(t)$$

then, the above equations can be expressed by the discrete time form.

$$h_1(t) = a_{11}h_1(t-1) + a_{21}h_2(t-1) \tag{30}$$

$$h_2(t) = a_{22}h_2(t-1) + a_{32}h_3(t-1) + b_1u_d(t) \tag{31}$$

$$h_3(t) = a_{33}h_3(t-1) + a_{43}h_4(t-1) \tag{32}$$

$$h_4(t) = a_{24}\frac{h_2(t-1)}{h_5(t-1)} + a_{34}\frac{h_3(t-1)}{h_5(t-1)} + a_{44}h_4(t-1) + \frac{b_1}{h_5(t-1)}u_d(t) \tag{33}$$

$$h_5(t) = a_{55}h_5(t-1) + a_{65}h_6(t-1) \tag{34}$$

$$h_6(t) = a_{66}h_6(t-1) + b_2u_m(t) \tag{35}$$

where

a_{ij}: coefficients representing the system paraments.

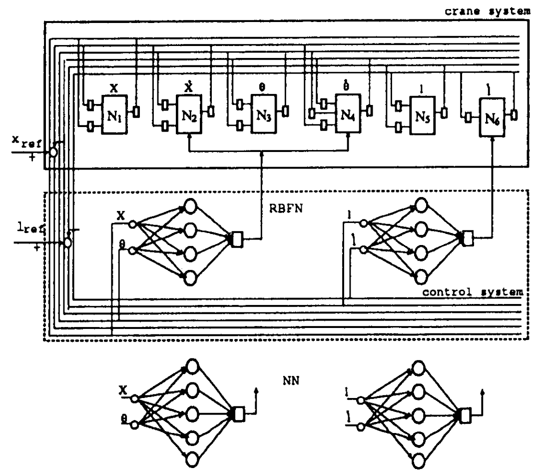


Fig.5 Structure of Nonlinear Crane Control System

The structure of the nonlinear crane control system is shown in **Fig.5.** The nonlinear crane control system has two parts. The upper part is a crane system which includes 6 nodes(real line frame) and the lower part is a controller(dotted line frame). The controller is constructed by the radial basis function network or the neural network. The arbitrary time delay is assumed to be one sampling time. The ULN with RBF controller has two RBF controllers and the ULN with SF controller has also two NN controllers. Each RBF controller has $2(|L(j)=2|)$ or $4(|L(j)=4|)$ nonlinear functions, and the NN controller has the three layered structure. In the controller, the left controller has two inputs(x,θ) and the right controller has two inputs(l, $\dot{l}$). Boch the crane system and RBF(or NN) controller are constituted by ULN.

In the simulations, control time is 40 seconds, the criterion function can be expressed as follows.

$$E = \frac{1}{2}\sum_{t=0}^{T}[Q_1(l_{ref} - l(t))^2 + Q_2(x_{ref} - x(t))^2 + Q_3\theta^2(t) + Q_4\dot{\theta}^2(t) + Q_5u_m^2(t) + Q_6u_d^2(t)] + \frac{1}{2}(Q_7\dot{x}^2(t_f) + Q_8\dot{l}^2(t_f)) \tag{36}$$

where,
l_{ref}, x_{ref} : target value of l, x;
t_f: final sampling time;
Q_i: coefficient of criterion function.

5.2 Comparison of Radial Basis Function(RBF) with Sigmoid Function(SF) Using Gradient Method

Simulation conditions were shown in **Table 1.**

Table 1 Simulation Conditions

mass of the trolley M	40.0kg
mass of the load m	2.0kg
coefficients of the friction D	300.0kg/sec
sampling numbers	2000
learning time	80000times
control times	40 s
random number range for parameters	[0,1]
coefficient of criterion function	Q1=1.0
	Q2=1.0
	Q3=1.0
	Q4=5.0
	Q5=0.001
	Q6=0.001
	Q7=1.0
	Q8=1.0
learning coefficient	0.00002

To compare the performance of the ULN with RBF and NN using the gradient method, simulations were carried out for caseA, caseB, caseC, caseD, and caseE. The reference values of x and l (x_{ref}, l_{ref}) are shown in **Table 2**.

By changing the initial parameters of RBF and SF, simulations were carried out 5 times for each case. Average E were shown in **Table 3**.

CaseA and B are for studying the learning performance and caseC, D and E are for testing the generalization capability. In caseA and caseB, the goal of the task is to bring x from 0.0m to 0.2m in the first 20 seconds and from 0.2m to 0.4m in the last 20 seconds; l from 2.0m to 1.7m in the first 20 seconds and from 1.7m to 2.0m in the last 20 seconds; and to let θ be as small as possible.

Table 2 Reference Values of x and l (Comparison of RBF and SF based on Gradient Method)

	Number of parameter		t=0	reference value			
				t=1...20		t=20...40	
				L	G	L	G
caseA	22	l_{ref}	2.0	1.7		2.0	
		x_{ref}	0.0	0.2		0.4	
caseB	42	l_{ref}	2.0	1.7		2.0	
		x_{ref}	0.0	0.2		0.4	
caseC	42	l_{ref}	2.0	1.7	1.7	1.7	2.0
		x_{ref}	0.0	0.2	0.2	0.2	0.4
caseD	42	l_{ref}	2.0	1.7	2.3	1.7	2.0
		x_{ref}	0.0	0.2	-0.2	0.2	0.0
caseE	42	l_{ref}	2.0	1.7	2.6	1.7	2.0
		x_{ref}	0.0	0.2	-0.4	0.2	0.0

L: for learning;
G: for testing generalization capability.

Table 3 Average Values of Criterion Function (Comparison of RBF and SF based on Gradient Method)

	Function	E
caseA	RBF	7.75
	SF	9.80
caseB	RBF	6.73
	SF	8.93
caseC	RBF	3.94
	SF	5.18
caseD	RBF	3.94
	SF	5.18
caseE	RBF	3.94
	SF	5.18

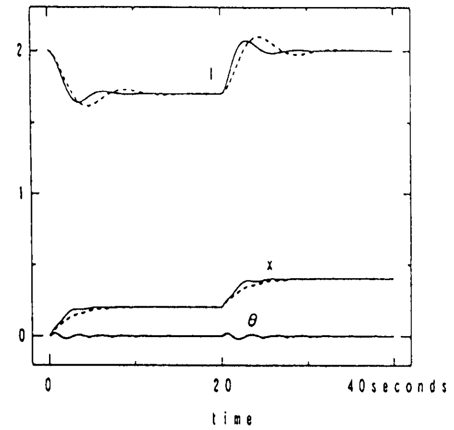


Fig.6 Simulation Results(caseC)

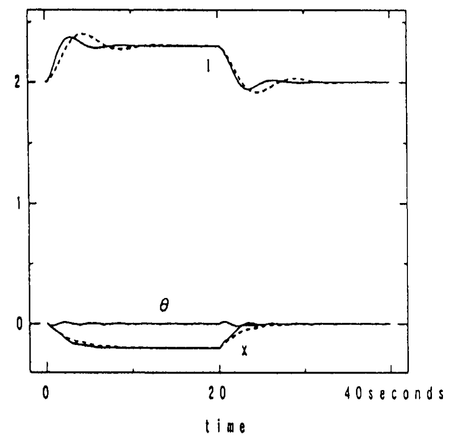


Fig.7 Simulation Results(caseD)

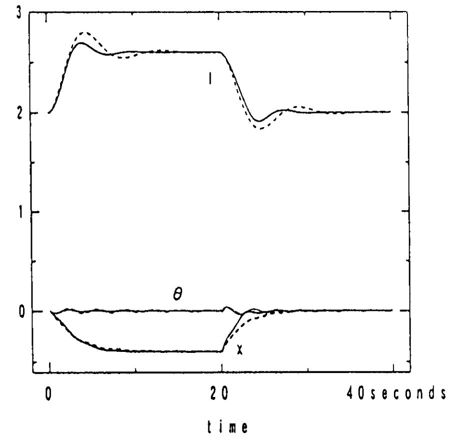


Fig.8 Simulation Results(caseE)

Parameters learning was carried out in order to bring x from 0.0m to 0.2m and l from 2.0m to 1.7m in the 40 seconds, on the other hand, the kinetic dynamics for testing the generalization capability was calculated changing the reference input x_{ref} from 0.0m to 0.2m(caseC), -0.2m(caseD), -0.4m(caseE), l_{ref} from 2.0m to 1.7m(caseC), 2.3m(caseD), 2.6m(caseE) in the first 20 seconds; and return back to the original value in the last 20 seconds. **Fig.6**, **Fig.7**, and **Fig.8** show the kinetic dynamics of simulations corresponding to caseC, caseD, and caseE. The real lines were obtained using RBF and the dotted lines were obtained using SF.

In the above simulations, a new control methodology was presented, which discribes the RBF controller and the NN controller in an unified manner using Universal Learning Network. The simulation results have proved that the ULN with RBF has better performance than that of the system using ULN with SF, regardless of the number of parameters(caseA, caseB), and it is also shown that the RBF control has better performance for the generalization capability than the neural network control(caseC, caseD, caseE).

5.3 Comparison of Radial Basis Function(RBF) with Sigmoid Function(SF) based on RasVal

In this subsection, we will compare RBF with SF based on RasVal.

By the same way as the preceding simulation, a nonlinear crane system(**Fig.4**) was studied in order to compare the performance of RBF and SF based on RasVal. The structure of Nonlinear Crane Control System is shown in **Fig.5**. The controller is constructed by the radial basis function network or sigmoid function network. The arbitrary time delay is assumed to be 1.0 sampling time.

Simulation conditions for RasVal are shown in **Table 4**.

Table 4 Simulation Conditions for RasVal Learning

exponential filter coefficient α	0.45
upper limit of $\bar{\beta}+\underline{\beta}$	500
lower limit of $\underline{\beta}$	200
coefficient η	0.0001

The reference values of x and l (x_{ref}, l_{ref}) are shown in **Table 5**.

By changing the initial parameters of RBF controller and SF controller randomly, simulations were carried out 5 times for caseF, caseG, caseH, caseI and caseJ, average values of criterion function of E are also shown in **Table 6**. CaseF is for studying the learning performance and caseG, H, I and J are for testing the generalization capability.

Table 5 Reference Values of x and l (Comparison of RBF and SF based on RasVal)

	Number of parameter		t=0	reference value			
				t=1...20		t=20...40	
				L	G	L	G
caseF	42	l_{ref}	2.0	1.7		2.0	
		x_{ref}	0.0	0.2		0.4	
caseG	42	l_{ref}	2.0	1.7	1.7	1.7	2.0
		x_{ref}	0.0	0.2	0.2	0.2	0.4
caseH	42	l_{ref}	2.0	1.7	2.3	1.7	2.0
		x_{ref}	0.0	0.2	-0.2	0.2	0.0
caseI	42	l_{ref}	2.0	1.7	2.6	1.7	2.0
		x_{ref}	0.0	0.2	-0.4	0.2	0.0
caseJ	42	l_{ref}	2.0	1.7	2.3	1.7	2.0
		x_{ref}	0.0	0.2	0.2	0.2	0.4

L: for learning;
G: for testing generalization capability.

Table 6 Average Values of Criterion Function (Comparison of RBF and SF based on RasVal)

	Function	E
caseF	RBF	4.96
	SF	4.76
caseG	RBF	3.12
	SF	2.34
caseH	RBF	3.12
	SF	2.34
caseI	RBF	3.12
	SF	2.34
caseJ	RBF	3.12
	SF	2.34

Fig.9 and **Fig.10** show the kinetic dynamics of simulations corresponding to caseF and caseJ. The real lines were obtained using RBF and the dotted lines were obtained using SF.

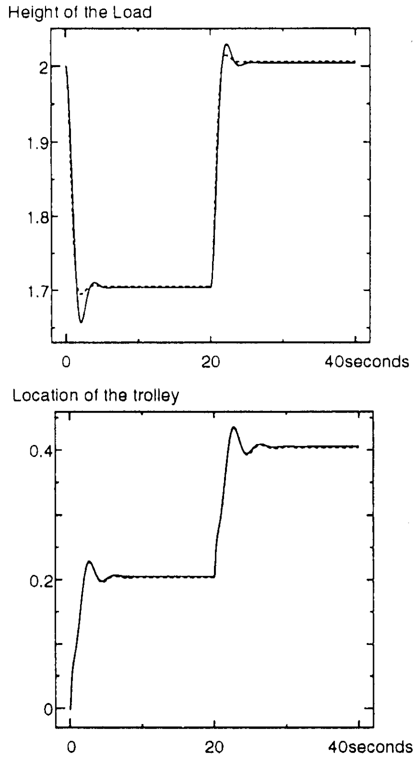


Fig.9 Simulation Results(caseF)

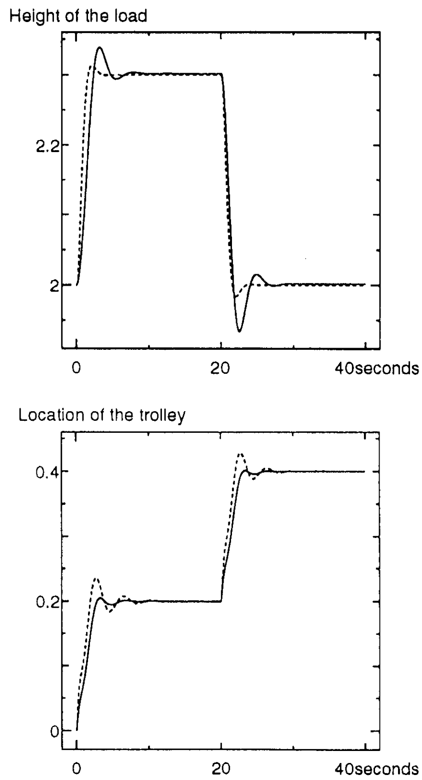


Fig.10 Simulation Results(caseJ)

From the simulation results, it is shown that the ULN with sigmoid function has better performance than that of the system using ULN with RBF(caseJ), and it is also shown that the NN control has better performance for the generalization capability than the RBF control(caseG, caseH, caseI and caseJ).

Why did we get different conclusions from 5.2 and 5.3? We think that in the case of the gradient method, sigmoid function falls into the local minimia problem easily, so the RBF was better than the SF in performance; While in the case of RasVal, the RasVal can solve the local minimia problem, so the SF was better than RBF in performance.

5.4 Comparison of RasVal with Gradient Method for the RBF Controller

As the preceding subsections, a nonlinear crane system(**Fig.4**) was studied in order to compare the performance of RasVal with the gradient method.

In the learning, control time is 40 seconds, the goal of the task is to bring x from 0.0m to 0.2m in the first 20 seconds and from 0.2m to 0.4m in the last 20 seconds; l from 2.0m to 1.7m in the first 20 seconds and from 1.7m to 2.0m in the last 20 seconds; and to let θ be as small as possible. Therefore the criterion function can be expressed by Eq.(36).

Table 7 Simulation Conditions for Gradient and RasVal Learning

gradient method		
learning coefficient		0.00001
RasVal		
exponential filter coefficient	α	0.45
upper limit of	$\bar{\beta}+\underline{\beta}$	500
lower limit of	$\bar{\beta}$	200
coefficient	η	0.0001

Simulation conditions for gradient and RasVal learning are shown in **Table 7**.

The Target values are shown in **Table 8**.

By changing the initial parameters of RBF controller randomly, simulations were carried out 5 times in caseK, caseL, and caseM, CaseM is for testing the generalization capability, and the values E are also shown in **Table 9**.

Table 8 Reference Values of x and l (Comparison of RasVal and Gradient Method for RBF Controller)

	Number of parameter		t=0	reference value t=1...20 L	t=1...20 G	t=20...40 L	t=20...40 G
caseK	22	l_{ref}	2.0	1.7		2.0	
		x_{ref}	0.0	0.2		0.4	
caseL	42	l_{ref}	2.0	1.7		2.0	
		x_{ref}	0.0	0.2		0.4	
caseM	42	l_{ref}	2.0	1.7	1.7	1.7	2.0
		x_{ref}	0.0	0.2	0.2	0.2	0.4

L: for learning;
G: for testing generalization capability.

Table 9 Average Values of Criterion Function (Comparison of RasVal and Gradient Method for RBF Controller)

	Function	E
caseK	RasVal	7.29
	Gradient	10.83
caseL	RasVal	6.98
	Gradient	9.71
caseM	RasVal	4.80
	Gradient	5.77

Fig.11 and **Fig.12** show the kinetic dynamics of simulations corresponding to caseL and caseM respectively. The real lines were obtained using RasVal; the dotted lines were obtained using gradient method.

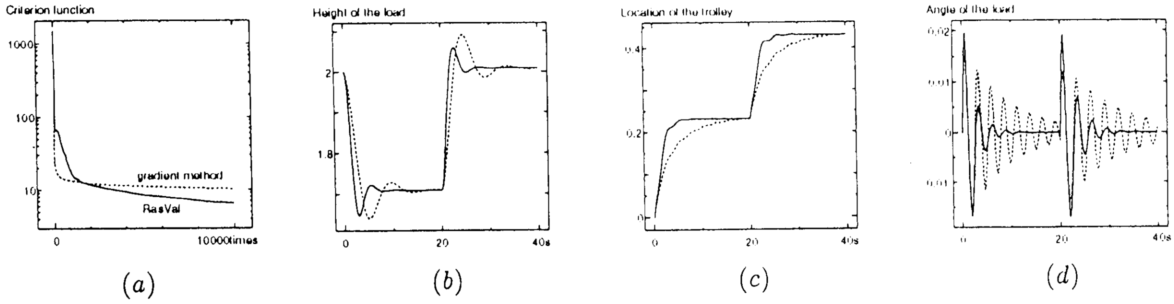


Fig.11 Simulation Results for Studying the Learning Perforance(caseL)

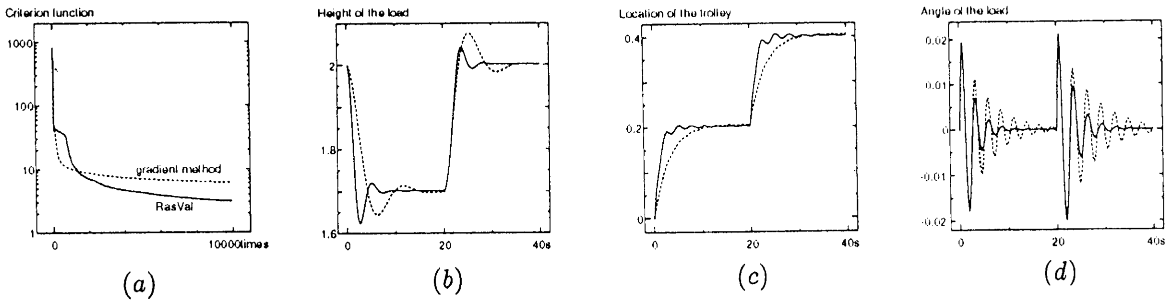


Fig.12 Simulation Results for Testing the generalization Capability(caseM)

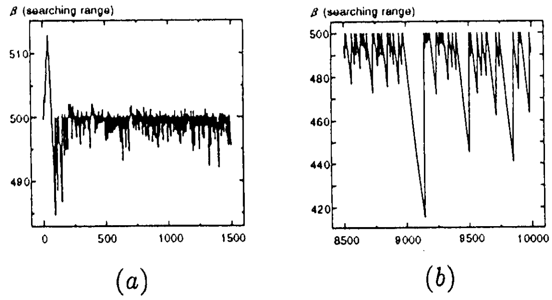


Fig.13 Searching Range of RasVal

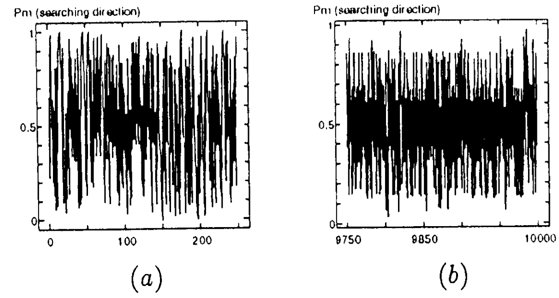


Fig.14 Searching Probability of RasVal

From simulation results, it is shown that the learning speed, learning performance and the generalization capability of RasVal are better than those of gradient method. **Fig.13** and **Fig.14** show the results of searching range(β) and searching probability of negative direction(p_m) at the beginning of and at the end of searching respectively. From **Fig.13**, **Fig.14**, it is shown that intensification and diversification of searching can be realized by RasVal.

At the beginning of searching which corresponds to the left part of **Fig.13** and **Fig.14**, searching range is small and searching probability is evenly distributed around 1.0 or 0.0, which means the intensification of search (corresponds to A in **Fig.3**); on the other hand, at the end of searching which corresponds to the right part of **Fig.13** and **Fig.14**, searching range is large and searching probability is mainly distributed around 0.5, which means the diversification of search(corresponds to B in **Fig.3**).

In this subsection, it has been proved that a nonlinear crane control system using RasVal has better performance than that of the system using the gradient method. It has also been shown that the RasVal has better performance for generalization capability than the gradient method.

5.5 Comparison of RasVal with Gradient Method for the NN Controller

In the preceding subsection, we discussed the ULN with RBF trained by RasVal, and it was shown that the RasVal was superior in performance to the gradient method. In this subsection, we discuss the ULN with sigmoid functions trained by RasVal. By applying the proposed method to the design of nonlinear crane control system, we obtain the same conclusion, that is, RasVal is superior in performance to the back propagation learning algorithm irrespective of the functions in the ULN nodes.

For ULN with sigmoid functions Eq.(1) can be specifically rewritten as Eq.(11) and Eg.(12).

A nonlinear crane system(**Fig.4**) was studied in order to compare the performance of RasVal with that of the gradient method. The structure of the ULN nonlinear crane control system are shown in **Fig.5.** The controller is constructed by the neural network. The time delay is assumed to be one sampling time.

Simulations were carried out to compare the performance of RasVal and that of the gradient method. The criterion function employed here is Eq.(36),

Simulation conditions for gradient and RasVal learning are the same as **Table 7.**

Two different structures are considered for NN controllers for studying the learning performance; in CaseN-1,2, the left NN in **Fig.3** has three hidden nodes and the right controller has two, whereas, in CaseO-1,2, each has five nodes. Also two sets of target values of the trolley position and the load height are used. These are summarized in **Table 10**.

By changing the initial settings of the parameters of NN controllers randomly, simulations were carried out 5 times. The average of the criterion function E is shown in **Table 11**, the behaviors of $l(t)$, $x(t)$, and $\theta(t)$ are shown in **Fig.15**, **Fig.16**, **Fig.17** and **Fig.18**. The real lines were obtained using RasVal; the dotted lines show the results of the gradient method.

The trained controllers are then employed in the new control problems with different target values from those considered in the learning. The new target values are listed in **Table 12**. The controller in CaseN-1 is applied to CaseN-3, the one from CaseN-2 is used in CaseN-4, and so on.

The criterion function values shown in **Table 13** and the behaviors of the crane in **Fig.19-23** show that the RasVal learning again gives the better results.

This means that with RasVal we can well minimize the criterion function and that the resulting controllers perform well.

From these, we can confirm that RasVal learning provides better performance.

In this section, a new optimization method named RasVal is compared its performance with the gradient method in traing of neural network based ULN control systems. It has been proved that a nonlinear crane control system trained by RasVal has better performance than that of the system obtained by the gradient method, and this indicates, together with our previous results, that RasVal gives better performance than the gradient methods in optimizing ULN control systems irrespective of the specific functions contained in ULN.

Table 10 Reference Values of x and l (Learning Ability)

	Number of parameter		t=0	reference value t=0,...20	reference value t=20....40	
case N-1	22	l_{ref}	2.0	1.7	2.0	Fig.15
		x_{ref}	0.0	0.2	0.4	
case N-2	22	l_{ref}	2.0	2.3	2.0	Fig.16
		x_{ref}	0.0	0.2	0.4	
case O-1	42	l_{ref}	2.0	1.7	2.0	Fig.17
		x_{ref}	0.0	0.2	0.4	
case O-2	42	l_{ref}	2.0	2.3	2.0	Fig.18
		x_{ref}	0.0	0.2	0.4	

Table 11 Average Values of Criterion Function (Learning Ability)

	algorithm	E
caseN-1	RasVal	4.75835
	Gradient	9.92467
caseN-2	RasVal	4.71445
	Gradient	10.11639
caseO-1	RasVal	4.74484
	Gradient	8.71698
caseO-2	RasVal	4.70168
	Gradient	8.65597

Table 12 Reference Values of x and l (Generalization Capability)

	Number of parameter		t=0	reference value t=0...20 L	t=0...20 G	t=20...40 L	t=20...40 G	
case N-3	22	l_{ref}	2.0	1.7	1.7	1.7	2.0	Fig.19
		x_{ref}	0.0	0.2	0.2	0.2	0.4	
case N-4	22	l_{ref}	2.0	1.7	2.3	1.7	2.0	Fig.20
		x_{ref}	0.0	0.2	0.2	0.2	0.4	
case O-3	42	l_{ref}	2.0	1.7	1.7	1.7	2.0	Fig.21
		x_{ref}	0.0	0.2	0.2	0.2	0.4	
case O-4	42	l_{ref}	2.0	1.7	2.3	1.7	2.0	Fig.22
		x_{ref}	0.0	0.2	0.2	0.2	0.4	

L: for learning;

G: for testing generalization capability

Table 13 Average Values of Criterion Function (Generalization Capability)

	algorithm	E
caseN-3	RasVal	2.68128
	Gradient	5.88958
caseN-4	RasVal	2.68128
	Gradient	5.88958
caseO-3	RasVal	2.32378
	Gradient	4.67268
caseO-4	RasVal	2.32378
	Gradient	4.67268

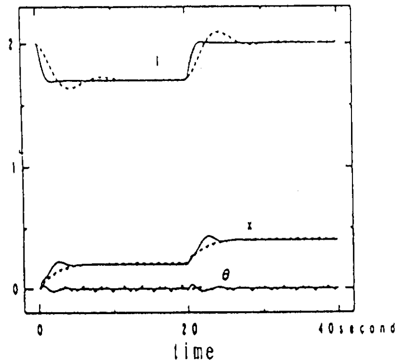


Fig.15 Simulation Results(caseN-1)

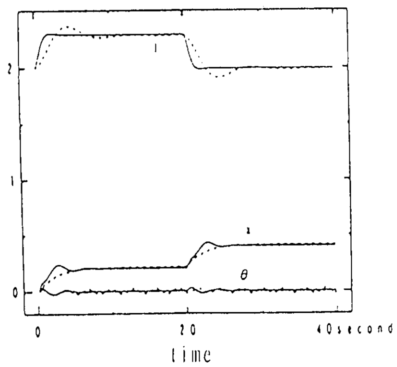


Fig.16 Simulation Results(caseN-2)

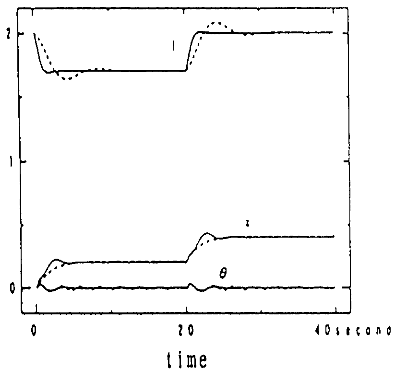


Fig.17 Simulation Results(caseO-1)

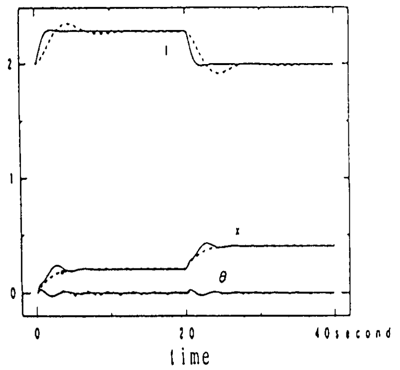


Fig.18 Simulation Results(caseO-2)

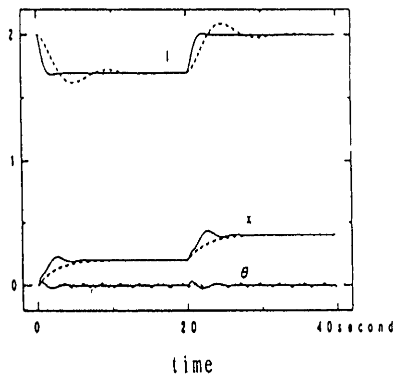


Fig.19 Simulation Results(caseN-3)

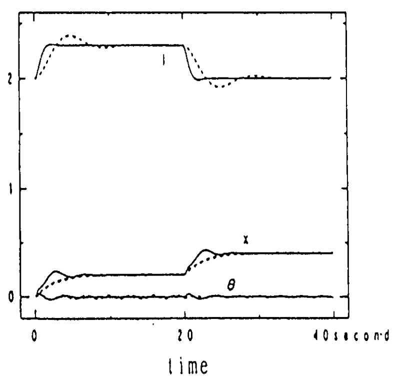


Fig.20 Simulation Results(caseN-4)

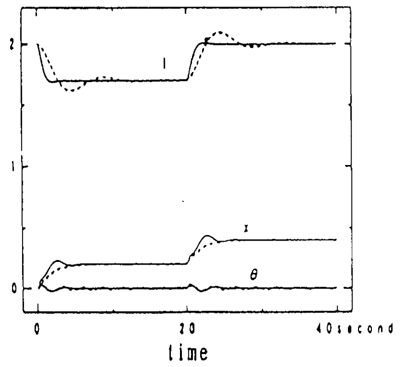


Fig.21 Simulation Results(caseO-3)

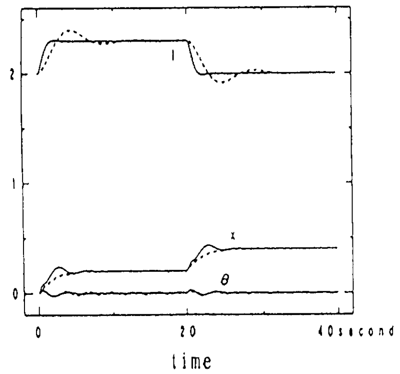


Fig.22 Simulation Results(caseO-4)

6. A New Method to Improve Generalization Capability of Nonlinear Crane Controller Systems Using RasVal

Generally, the words "generalization capability" mean the ability of assuring of that the system works well even in the different environment from that at learning stage. It is commonly said that the generalization capability will be improved by learning a great number of cases with different environments and also by reducing the scale of networks for learning. Recently, some papers on the enhancement of the generalization capability have been reported by using the second order derivatives in Universal Learning Network[6)]. These methods are based on the idea that a criterion function relating to the improvement of system rebustness is added to the usual criterion function in order to enhance the generalization capability.

In this section, a new method for the enhancement of the generalization capability in the nonlinear control systems is presented, where control signals to a plant to be controlled are suppressed by increasing the weighting coefficients related to the control signals in the criterion function. Simulations of a nonlinear crane system are carried out in order to study the above new method.

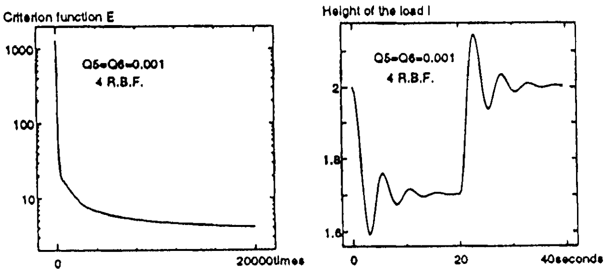
Simulations were carried out to study the generalization capability of the proposed method. In the simulations, control time is 40 seconds.

The generalization capability was investigated as follows. While learning of parameters was carried out so as to bring x from 0.0m to 0.2m and l from 2.0m to 1.7m in the 40 seconds, kinetic dynamics for investigating the generalization capability was calculated by changing the reference input from $x =$ 0.0m to 0.2m, from $l = 2.0$m to 1.7m in the first 20 seconds; and from $x = 0.2$m to 0.4m, from $l = 1.7$m to 2.0m in the last 20 seconds. By changing the initial parameters of the RBF controller randomly, simulations were carried out 5 times.

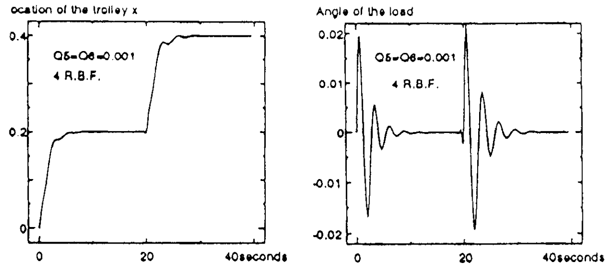
Simulation conditions were shown in **Table 14.**

Table 14 Simulation Conditions

mass of the trolley M	40.0kg
mass of the load m	2.0kg
coefficients of the friction D	300.0kg/sec
sampling numbers	2000
learning time	20000times --80000times
control times	40 s
random number range for parameters	(0,1)
coefficient of criterion function	Q1=1.0 Q2=1.0 Q3=1.0 Q4=5.0 Q5=0.001 --1.0 Q6=0.001 --1.0 Q7=1.0 Q8=1.0
RasVal	
exponential filter coefficient α	0.45
upper limit of $\bar{\beta}+\underline{\beta}$	500
lower limit of $\underline{\beta}$	200
coefficient η	0.0001

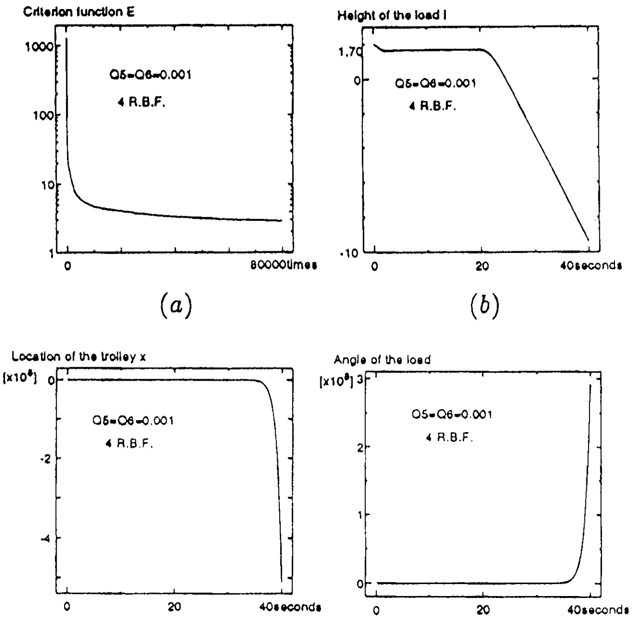


(a) (b)



(c) (d)

Fig.23 Simulation Results(Learning times:20000)



(a) (b)

(c) (d)

Fig.24 Simulation Results(Learning times:80000)

Fig.23 and **Fig.24** show the average learning curives and $l(t), x(t), \theta(t)$ of the nonlinear crane system which were obtained for the study of the generalization capability, on the condition that L(j)=4 (four RBF functions), $Q_5 = Q_6 = 0.001$ and learning was carried out 20000 times and 80000 times respectively.

From **Fig.24** it is shown that when the learning was continued until 80000 times, dynamics of the system becomes unstable because of the overfitting of the learning. Curved surfaces of control signals u_d, u_m which are the function of (x, θ) and $(l, \dot{l})$ respectively are shown in **Fig.25** and **Fig.26**. It is understood that curved surfaces of **Fig.26** obtained when the learning is carried out too much can not calculate the appropriate control signal u_m around $l_{ref} - l = 0.3$. Therefore, suppressing of the control signals was tried by increasing the weighting coefficients Q_5, Q_6 related to u_m, u_d in the criterion function.

Fig.27 show the curved surfaces of control signals u_d, u_m obtained by the condition that four RBF functions, $Q_5 = Q_6 = 1.0$ were used and learning was carried out 80000 times. Q_5, Q_6 in **Fig.27** are the lowest value(see **Table 15**), where stable dynamics is obtained even when the reference inputs x_{ref}, l_{ref} are changed in the middle of the control.

Table 15 Relation between L(j) and the Lowest Bound of Q_5 and Q_6

L(j)	The lowest coefficient to assure the generalization capability
1	--------
2	Q5=Q6>=0.3
3	Q5=Q6>=0.4
4	Q5=Q6>=1.0

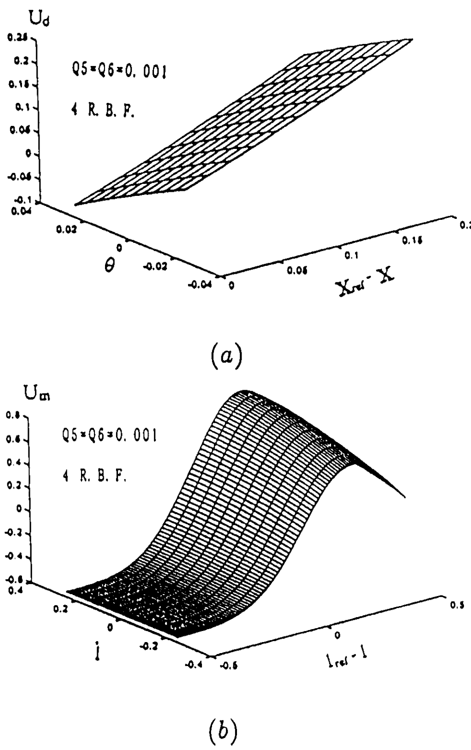


Fig.25 Curved Surfaces of Control Signals U_d and U_m (Learning Times: 20000)

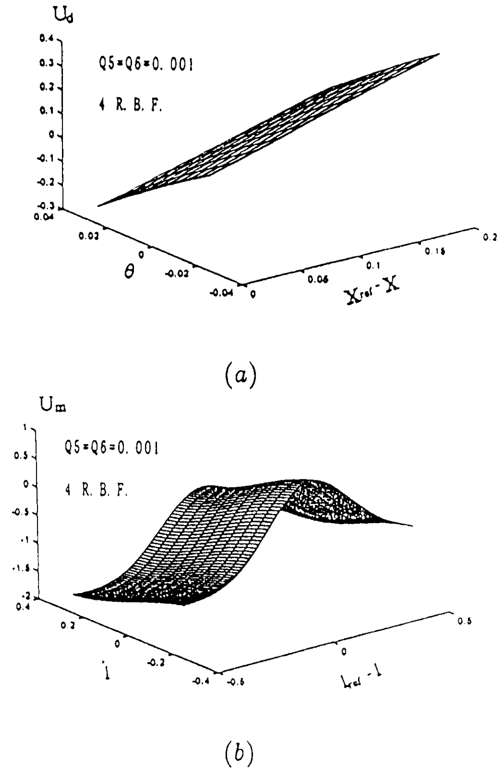


Fig.26 Curved Surfaces of Control Signals U_d and U_m (Learning Times: 80000)

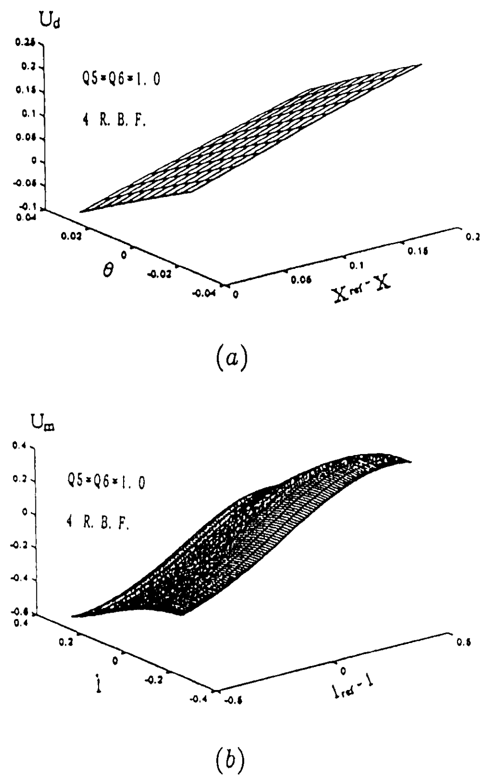


Fig.27 Curved Surfaces of Control Signals U_d and U_m ($Q_5 = Q_6 = 1.0$)

From **Fig.27** and **Table 15**, it is known that the larger the number of RBF functions is, the more needed the suppression of control signals are, and it is also known that sufficient generalization capability can not be obtained by using one RBF function.

In this section, the generalization capability of RBF controller using RasVal in Universal Learning Network was studied. From simulations of a nonlinear control system, it has been proved that the generalization capability is enhanced by suppressing the control signals, and a great numbers of RBF functions in the controller of the system deteriorate the generalization capability.

7. Conclusions

In this paper, we presented a new control method for nonlinear systems using ULN with RBF firstly. From the simulation results, it has been proved that ULN with RBF based on the gradient method is superior in performance to that of neural networks, and it has also been shown that the RBF control has better performance for the generalization capability than the neural network control. We also introduced a new learning algorithm named RasVal. By applying RasVal method to a nonlinear crane control system which can be controlled by the Universal Learning Network with the RBF and the sigmoid functions, it has also been shown that the sigmoid function control has better performance than RBF control. At the same time, it have been shown that the RasVal is superior in performance to the commonly used back propagation learning algorithm. Finally, generalization capability of a Radial Basis Function controller using RasVal in Universal Learning Network was studied and it has been shown that a new method is effective to overcome the over-fitting problem in nonlinear control systems.

References

1) K.Hirasawa,M.Ohbayashi and J.Murata: Universal Learning Network and Computation of its Higher Order Derivatives, Proc.of 1995 IEEE International conference on Neural Networks, pp.1273-1277, 1995

2) N.Shao,K.Hirasawa,M.Ohbayashi and K.Togo: Nonlinear Control System with Radial Basis Function Controller using Random Search Method of Variable Search Length,Proc.of IEEE International Conference on Neural Networks, pp. 788–793, 1997

3) N.Shao,K.Hirasawa,M.Ohbayashi: Nonlinear Control System Using Universal Learning Network with Radial Basis Function,Proc.of 2nd Asian control conference, PP. II 19-22, 1997

4) N.Shao,K.Hirasawa,M.Ohbayashi and K.Togo: Nonlinear Control System with Neural Network Controller using RasVal Learning,Proc.of IEEE SMC, 1997

5) N.Shao,K.Hirasawa,M.Ohbayashi and K.Togo: Generalization Capability of Radial Basis Function Controller using Random Search Method of Variable Search Length in Univerasl Learning Network, Research Reports on Information Science and Electrical Engineering, Vol.2, No.1, pp.31–38, March, 1997

6) M.Ohbayashi, K.Hirasawa, J.Murata and M.Harada: Robust Learning Control using Universal Learning Network, Proc. of IEEE International Conference on Neural Networks, pp.2208–2213, 1996

7) Jyh-shing Roger Jing, Chuen-tsai Sun: Neuro-Fuzzy Modeling and Control, Proc. of IEEE, Vol.83, No.3, MARCH,1995

8) S.Chen,C.F.N.Cowan, and P.M.Grant: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, IEEE Transaction on Neural Networks, Vol.2,No.2,March, 1991

9) Jone Moody,Christian J.Darken: Learning in Networks of Locally-Tuned Processing Units, Neural Computation, pp.1281-1294, 1989