

# STRUCTURAL IDENTIFICATION USING NEURAL NETWORK AND KALMAN FILTER ALGORITHMS

Tadanobu SATO<sup>1</sup> and Makoto SATO<sup>2</sup>

<sup>1</sup>Member of JSCE, Dr. of Eng., Professor, Disaster Prevention Research Institute, Kyoto University (Gokasho, Uji, Kyoto 611, Japan)

<sup>2</sup>Member of JSCE, Dept. of Civil Eng., Graduate Course Kyoto University (Yoshidahonmachi, Sakyo-ku, Kyoto 606, Japan)

The dynamic characteristics of a structural system are identified. The relevant neural network characteristics of a learning algorithm are discussed in the context of system identification. Because of the self-learning nature of the neural network the dynamic characteristics identified are strongly affected by the level of noise contained in the teaching signals. A method to identify the dynamic characteristics of a structural system proof against contaminating noise in teaching signals has been developed with the aid of the Kalman filtering technique. Numerical examples to identify dynamic response characteristics of linear and nonlinear structural systems are worked out to demonstrate the stability and robustness of the proposed algorithm.

*Key Words* : identification, neural network, extended Kalman filter, Back propagation, stability

## 1. INTRODUCTION

Identification of physical models or the dynamic parameters of a structure has gained much attention recently<sup>1)</sup>. The methods developed can be categorized in two groups: frequency domain and time domain methods. The frequency domain method has been in wide use for some time and has been proved efficient in many cases. Application of these methods, however, is limited to linear structural systems. To avoid the limitation inherent in frequency domain analysis methods using time domain data have been developed<sup>2)</sup>. The identification of large systems remains difficult when conventional algorithms are used because of instability, non-uniqueness, multi-colinearity, and excessive calculation time required. To avoid these problems, it is necessary to develop algorithms by the use of parallel distributed processing units<sup>3)</sup>.

We have developed an algorithm to identify the dynamic characteristics of a structural system by

using the potential of a neural network. The processing elements are connected to each other by adjustable weights. Change in these weights changes the input/output behavior of the network. In conventional neural network algorithms, the connection weights of each link in the network are identified by the back propagation algorithm<sup>4)</sup>. Some problems have been found for this algorithm: the learning process is unstable when the teaching signal is contaminated by measurement noise, and because the connection weights are adjusted by analogy with the steepest descent method, the identified weights sometimes fall into the local minimum solution. Taking these factors into consideration, we developed a learning algorithm proof against noise in teaching signals by using the Kalman filter<sup>5)</sup>.

## 2. NEURAL NETWORK

Artificial neural network models used for

complex signal processing are based on our present understanding of biological nervous systems. These models are composed of many nonlinear computational elements that operate in parallel and that are arranged in patterns similar to biological neural nets. The neural network model used in this analysis is a multi-layer perceptron neural network shown in Fig.1. This is the simplest and most commonly used type of neural networks. Multi-layer perceptrons are feed-forward nets with one or more layers and nodes between the input and output nodes. The capabilities of the multi-layer perceptrons stem from the non-linearities used within nodes. If nodes are linear elements, a single-layer net with appropriately chosen weights can duplicate the results obtained with any multi-layer net.

$$v = \gamma(\bar{v}) \quad \bar{v} = \sum_{i=1}^n w_i u_i + b \quad (1)$$

in which  $u_i (i = 1, 2, \dots, n)$  are the inputs,  $w_i$  is the weights,  $b$  is a bias term and  $\gamma(\bullet)$  is a non-linear process function. The choice of non-linearity depends on the particular application for which the network is used. The most typical non-linearity used is the sigmoid function given by

$$\gamma(x) = \frac{1}{1 + \exp(-\alpha x)} \quad \alpha > 0 \quad (2)$$

Using these equations repeatedly we can express the output signals  $y_k (k = 1, \dots, m)$  from the final layer (layer 3 in Fig.1) as functions of the weights in each layer  $w_{ij}^1, w_{ij}^2$  and  $w_{ij}^3$ . In Fig.1 we assign a different processing function of  $\gamma(\bullet), \beta(\bullet)$  and  $\alpha(\bullet)$  for each layer. In the neural network algorithm, these connection weights between two layers are adjusted as the actual input/output relationship based on the network becomes close to a desired input/output relationship, i.e., a teaching pair of the input/output relationship. This adjusting process is called "learning". The time history of the response of a structure under earthquake excitation here is used as the teaching signal. We compose a neural network in which the input signals are the structural response values plus input acceleration to the structure at time step  $t$ , and output signals are the structural response values at time step  $(t+1)$ .

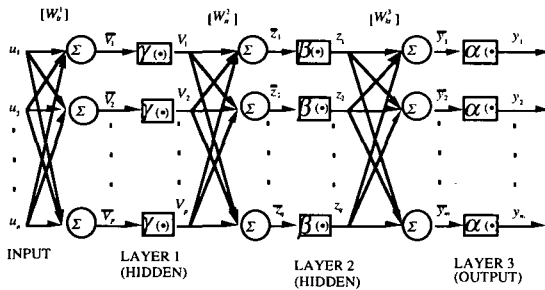


Fig.1 Perceptron Neural Network Model

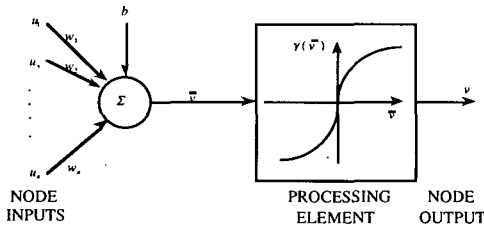


Fig.2 Typical Processing Element (Node)

Input signals flowing in this neural network through a node are shown in Fig.2. The input for a processing element in the node is the summation of weighted output node signals from the last layer which is passed through a non-linear function.

### 3. BACK PROPAGATION ALGORITHM

In the conventional back propagation algorithm, the connection weights are adjusted by minimizing the performance criterion  $E$  defined by Eq.3; i.e., the mean square error between the actual output of a multi-layer feed-forward perceptron and the desired output. Because a

perceptron and the desired output. Because a component of the actual output vector from the network is  $y_i$ , as shown in Fig.1, a component of the desired output vector can be defined by  $y_{id}$ . The performance criterion E then is written

$$E = \frac{1}{2} \sum_{k=1}^m e_k^2 \quad e_k = y_k - y_{kd} \quad (3)$$

in which  $e_k$  is the  $k$  th component of the output error. Now recall that  $y_k$  is the function of the weights in each layer  $w_{ij}^1, w_{ij}^2$  and  $w_{ij}^3$ . A typical weight  $w_{ij}$  (which could belong to any layer) is adjusted from its old value  $w_{ij}^{(old)}$  to its new value  $w_{ij}^{(new)}$  as follows<sup>4)</sup>:

$$w_{ij}^{(new)} = w_{ij}^{(old)} - \varepsilon \cdot \frac{\partial E}{\partial w_{ij}^{(old)}} \quad (4)$$

As seen in Eq.(4) the weights are adjusted in the negative gradient direction. The value of  $\varepsilon$  is termed the 'learning rate'. The derivative of the error criterion E with respect to the weight of each layer is given by

$$\begin{aligned} \nabla_3 E &= \left\{ \frac{\partial E}{\partial w_{ij}^3} \right\} = \begin{pmatrix} \delta_1^3 z_1 & \cdots & \delta_1^3 z_q \\ \vdots & \ddots & \vdots \\ \delta_m^3 z_1 & \cdots & \delta_m^3 z_q \end{pmatrix} \\ \delta_i^3 &= e_i \alpha'(\bar{y}_i) \\ \nabla_2 E &= \left\{ \frac{\partial E}{\partial w_{ij}^2} \right\} = \begin{pmatrix} \delta_1^2 v_1 & \cdots & \delta_1^2 v_p \\ \vdots & \ddots & \vdots \\ \delta_q^2 v_1 & \cdots & \delta_q^2 v_p \end{pmatrix} \\ \delta_i^2 &= \beta'(\bar{z}_i) \cdot \sum_{k=1}^m w_{ki}^3 \delta_k^3 \\ \nabla_1 E &= \left\{ \frac{\partial E}{\partial w_{ij}^1} \right\} = \begin{pmatrix} \delta_1^1 u_1 & \cdots & \delta_1^1 u_n \\ \vdots & \ddots & \vdots \\ \delta_p^1 u_1 & \cdots & \delta_p^1 u_n \end{pmatrix} \\ \delta_i^1 &= \gamma'(\bar{v}_i) \sum_{k=1}^q w_{ki}^2 \delta_k^2 \end{aligned} \quad (5)$$

in which  $\bar{y}_i, \bar{z}_i$  and  $\bar{v}_i$  are

$$\begin{aligned} \bar{y}_i &= \sum_{s=1}^q w_{is}^3 z_s & \bar{z}_i &= \sum_{s=1}^p w_{is}^2 v_s & \bar{v}_i &= \sum_{s=1}^n w_{is}^1 u_s \\ y_i &= \alpha(\bar{y}_i) & z_i &= \beta(\bar{z}_i) & v_i &= \gamma(\bar{v}_i) \end{aligned} \quad (6)$$

#### 4. LEARNING ALGORITHM USING THE EXTENDED KALMAN FILTER

A learning algorithm based on the extended Kalman filtering technique is derived. Because the component of the output from the neural network  $y_k$  is a nonlinear function of connection weights  $w_{ij}^1, w_{ij}^2$  and  $w_{ij}^3$ , the linearization of  $y_k$  with respect to the most likely values of the weights  $\hat{w}_{ij}^1, \hat{w}_{ij}^2$  and  $\hat{w}_{ij}^3$  becomes

$$\begin{aligned} \{y_k\} &= \{\alpha(\bar{y}_k)\} + [A_3] \{w^3\} - [A_3] \{\hat{w}^3\} + [A_2] \{w^2\} \\ &\quad - [A_2] \{\hat{w}^2\} + [A_1] \{w^1\} - [A_1] \{\hat{w}^1\} \end{aligned} \quad (7)$$

in which

$$\begin{aligned} \{w^3\} &= \left\{ \{w_{1j}^3\}^T \{w_{2j}^3\}^T \cdots \{w_{mj}^3\}^T \right\}^T \quad (j=1,2,\dots,q) \\ \{w^2\} &= \left\{ \{w_{1j}^2\}^T \{w_{2j}^2\}^T \cdots \{w_{pj}^2\}^T \right\}^T \quad (j=1,2,\dots,p) \\ \{w^1\} &= \left\{ \{w_{1j}^1\}^T \{w_{2j}^1\}^T \cdots \{w_{nj}^1\}^T \right\}^T \quad (j=1,2,\dots,n) \end{aligned} \quad (8)$$

and  $[A_3], [A_2]$  and  $[A_1]$  respectively, are subdivided into the row vector components  $A_3^{ks}, A_2^{ks}$  and  $A_1^{kl}$  which are

$$\begin{aligned} A_3^{kk} &= \{\alpha'(\bar{y}_k) z_1 \quad \alpha'(\bar{y}_k) z_2 \quad \cdots \quad \alpha'(\bar{y}_k) z_q\} \\ \text{and } A_3^{ks} &= \{0, 0, \dots, 0\} \quad \text{for } k \neq s \end{aligned}$$

$$\begin{aligned} A_2^{ks} &= \{F_{ks} v_1 \quad F_{ks} v_2 \quad \cdots \quad F_{ks} v_p\} \\ \text{in which } F_{ks} &= \alpha'(\bar{y}_k) \hat{w}_{ks}^3 \beta'(\bar{z}_s) \end{aligned} \quad (9)$$

$$A_1^{kl} = \{G_{kl} u_1 \quad G_{kl} u_2 \quad \cdots \quad G_{kl} u_n\}$$

in which  $G_{kl} = \alpha'(\bar{y}_k) \sum_{s=1}^q \hat{w}_{ks}^3 \beta'(\bar{z}_s) \sum_{l=1}^p \hat{w}_{sl}^2 \gamma'(\bar{v}_l)$

To apply the extended Kalman filtering technique, the following vectors and matrix are defined.

$$w_t = \left\{ \left\{ w^3 \right\}^T \left\{ w^2 \right\}^T \left\{ w^1 \right\}^T \right\}^T \quad (10)$$

$$H_t = \left[ \left[ A_3 \right] \quad \left[ A_2 \right] \quad \left[ A_1 \right] \right]$$

Using Eq.(10) we can rewrite Eq.(7) as

$$y_t = h_t + H_t w_t - H_t \hat{w}_t + v_t \quad (11)$$

in which  $H_t$  is the observation matrix and  $v_t$  the noise vector with the covariance matrix  $R_t$ .

If the vector  $y_t$  is replaced by the observed value, Eq.(11) yields to the observation equation in the Kalman filtering algorithm. If the weight vector  $w_t$  is assumed to be unchanged with time, the system transfer equation for weight vector  $w_t$  is given by

$$w_{t+1} = I w_t \quad (12)$$

Note that the transition matrix is a unit matrix and that the system noise vector is equal to 0 in this algorithm and the time step changes from  $t$  to  $(t+1)$ , therefore the algorithm of the extended Kalman filtering for the weight vector  $w_t$  is formulated as

A Initial conditions for the weight vector, the error covariance matrix and  $R_t$  ;

$$\hat{w}_{0|0} = w_0 \quad P_{0|0} = \Sigma_0 \quad R_0 = r_0 \quad (13)$$

B Kalman gain;

$$K_t = P_{t|t-1} H_t^T \left[ H_t P_{t|t-1} H_t^T + R_t \right]^{-1} \quad (14)$$

C Predictor equation;

$$\hat{w}_{t|t-1} = \hat{w}_{t-1|t-1} \quad P_{t|t-1} = P_{t-1|t-1} \quad (15)$$

D Filter equation;

$$\hat{w}_{t|t} = \hat{w}_{t|t-1} + K_t \left[ y_t - h_t \left( \hat{w}_{t|t-1} \right) \right] \quad (16)$$

E Estimation error covariance matrix;

$$P_{t|t} = P_{t|t-1} - P_{t|t-1} H_t^T \left[ H_t P_{t|t-1} H_t^T + R_t \right]^{-1} H_t P_{t|t-1} \quad (17)$$

Through A to E the weight vector is updated stepwise when the observed structure response is given sequentially.

## 5. APPLICATION TO A LINEAR STRUCTURE

To demonstrate the application of the algorithm, we modeled a two-story building (Fig.3). The parameters for this model are given in Table 1.

The accelerogram at El Centro (NS, 1940) modified for maximum acceleration to 50 cm/sec<sup>2</sup> was used to simulate the dynamic response of the structure.

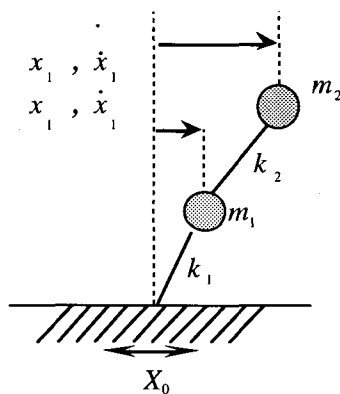


Fig.3 Linear Structure Model

Table 1 Model Parameters

Mass (kgfs <sup>2</sup> /cm)	2 F	0.12553
	1 F	0.12553
Stiffness (kgf/cm)	1F-2F	24.5
	1F-Base	24.5
Damping (%)	h <sub>1</sub>	0.863
	h <sub>2</sub>	2.260

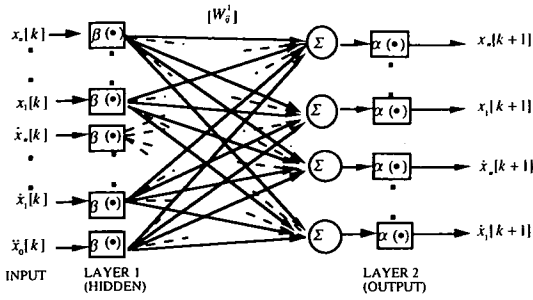


Fig.4 2-Layered Neural Network

The time interval to integrate the equation of motion is 0.01 sec. The neural network used in this section has two layers (Fig.4). The input values are the displacement and the velocity response of the model structure plus ground acceleration at time  $t$ . The output from the net is the structural response at time  $(t + 1)$ . The weight  $w_{ij}^1$  becomes a  $(5 \times 4)$  matrix.

To make the noise-contaminated teaching signal we generated pink noise with a frequency range of 0 to 25Hz by modifying its maximum amplitude to be equal to 5% of the maximum response amplitude. This pink noise is added to the simulated structural response. Identified time histories for four components of the weight matrix,  $w_{13}$ ,  $w_{14}$ ,  $w_{23}$  and  $w_{24}$ , are shown in Fig.5.

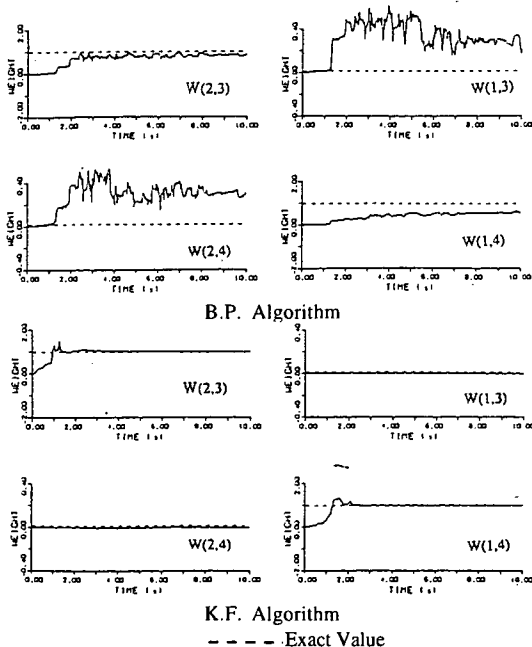


Fig.5 Time History of Weight

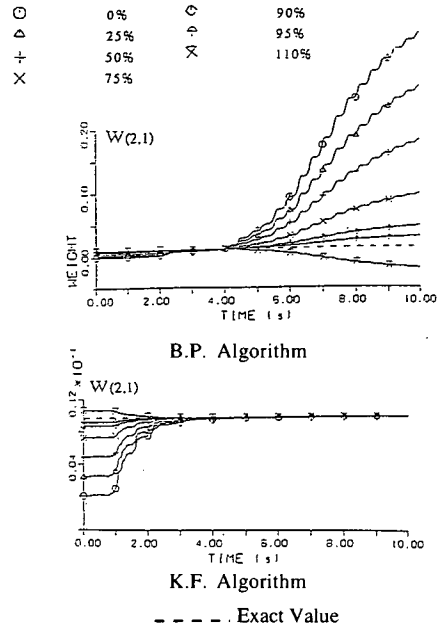


Fig.6 Time History of Weight (effect of initial values)

Solid lines are the identified results and broken lines the exact values of the weights. Results obtained by the back propagation algorithm are strongly affected by the noise, whereas the Kalman filtering algorithm removes the noise effect thereby providing stable learning results.

The effect of initial conditions on the learning process of the weight  $w_{21}$  is shown in Fig.6. We use a teaching signal non-contaminated with noise and seven different initial values for the weight matrix varying from 0 to 110% of its exact value. The convergent values based on the back propagation algorithm are spread over a wide range whereas the Kalman filtering algorithm shows a very good convergence. Because the duration of the teaching signal is 10 seconds a temporally convergent value for the weight matrix under a certain initial condition is obtained at the end of 10 seconds of learning. Assuming this value to be the initial value, we repeat the 10 second learning process until all the components of the weight matrix show steady change within the 10 second learning period. We define each 10 second learning process as global learning.

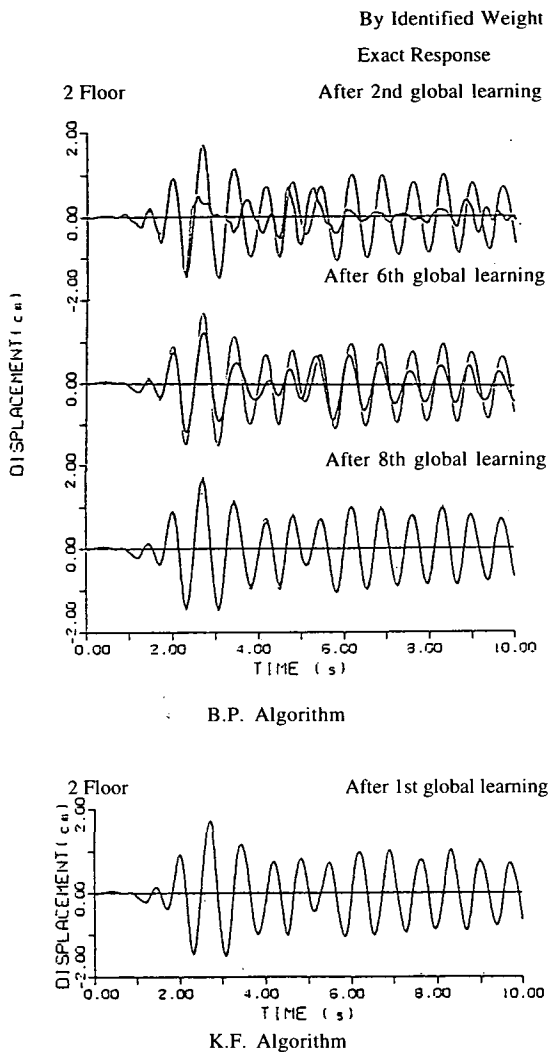


Fig.7 Time History of Response

The simulated second floor displacement using a temporary convergent weight matrix is shown in Fig.7. The thin line is the exact response, the thick one is the simulated one. For simulation, the neural network with a fixed weight matrix is used. Starting from the given initial conditions of structural response and the external excitation time history, the structural response one time step march is obtained through the network. On the assumption this as the input to the network, we can calculate the entire time history of structural response. Because the weight matrix converges to the exact value at the end of the first global learning when the Kalman filtering algorithm is

used, the simulated response coincides with the exact response. The simulated response obtained using the back propagation algorithm, however, requires several global learnings (in this case 8 times) to achieve good agreement between the simulated and exact responses because even though after 8 global learnings the components of the weight matrix converge to very different values from the exact ones. This means that the results obtained with the back propagation algorithm fall into the local minimum solution of the weight matrix.

An eigen value of the weight matrix of this network is an exponential of the corresponding eigen value of the dynamic equation of the structural system. Undamped natural frequencies and the damping coefficients of the model structure calculated from the identified weight matrix are given in Table 2. We assign a very low damping coefficient for the first mode to check the efficiency of the proposed learning algorithm. Exact values of the dynamic characteristics of the structure model are given in the upper part of this table. The dynamic characteristics identified by the B.P. and K.F. algorithm are compared for the cases of teaching signals with and without noise. When the teaching signal is not contaminated with noise, the identified results are better than those obtained using the noise-contaminated signal.

Table 2 Identified Parameters

		$\omega$ (rad/s)		h					
Exact	1st	8.63	0.00863						
	2nd	22.6	0.0226						
Initial weight	B.P. Algorithm				K.F. Algorithm				
	Without Noise		With Noise		Without Noise		With Noise		
	$\omega$ (rad/s)	h	$\omega$ (rad/s)	h	$\omega$ (rad/s)	h	$\omega$ (rad/s)	h	
0%	1st	—	—	—	—	8.61	0.0120	7.98	0.1530
	2nd	—	—	—	—	19.2	0.0615	21.0	0.1050
50%	1st	8.65	0.0310	7.14	0.0839	8.64	0.0081	8.04	0.1300
	2nd	49.0	0.8460	39.7	0.9250	20.9	0.0166	22.7	0.0694
80%	1st	8.41	0.0076	7.24	0.0928	8.64	0.0097	8.09	0.1020
	2nd	30.3	0.4630	24.0	0.3640	21.9	0.0073	23.7	0.0469
90%	1st	8.43	0.0145	7.85	0.1480	8.64	0.0093	8.11	0.1060
	2nd	26.0	0.2640	21.7	0.0203	22.3	0.0153	24.1	0.0407
110%	1st	9.10	0.0029	8.90	0.1360	8.63	0.0068	8.13	0.1140
	2nd	20.2	0.2630	21.7	0.5180	22.9	0.0296	24.7	0.0275

Because the B.P. algorithm does not guarantee convergence of the weight matrix to the exact value, the identified dynamic parameters are strongly affected by the initial condition. The results obtained using the K.F. algorithm are very robust not only for initial condition but for contaminated noise in the teaching signal.

## 6. APPLICATION TO A MODEL BUILDING

To demonstrate the application of the algorithm to real observed data, we identified the dynamic parameters, natural frequency and damping constants, of an eight-story model building shown in Fig.8. The specification of this model is shown in Table 3.

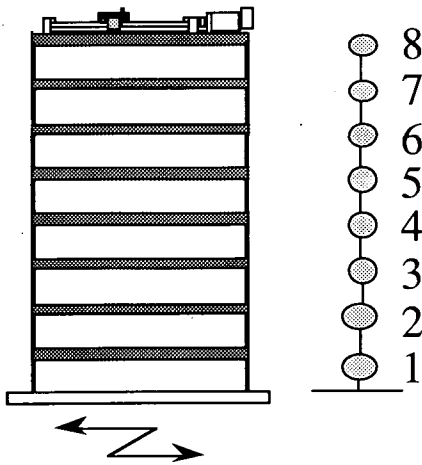


Fig.8 Model Structure

Table 3 Model Parameters

Node No.	Mass (kgf)	Stiffness (kgf/cm)
8	150.6	245.0
7	123.1	245.0
6	123.1	245.0
5	123.1	245.0
4	123.1	245.0
3	123.1	245.0
2	123.1	245.0
1	123.1	245.0

Data of dynamic responses of the model building were obtained through experiments conducted for seismic response control analyses using a active mass driver system<sup>6)</sup>. The model structure is a steel frame model of a shear building. The experiments were done on a 4m by 4m shaking table which can simulate various excitations, including seismic ones. The input seismic motion is the accelerogram at El Centro (NS,1940) adjusted to a peak acceleration of 50 cm/sec<sup>2</sup> for experimental analyses. The sampling time interval of model structure response is 0.01 sec. The neural network used in this section has two layers shown in Fig.4. The input values are the displacement and velocity responses of all floors, and the ground acceleration at time  $t$ . The outputs from the net are the structural responses at time  $(t + 1)$ . The  $w_{ij}^1$  is a  $(16 \times 17)$  matrix.

Taking actual problems into consideration, we assume following conditions to set initial values of the weight matrix and the estimation error covariance matrix: [1] Because it is easy to get exact values of mass, the values given in Table 3 are used as initial values. [2] Since it is hard to estimate the values of stiffness exactly, the initial values of stiffness are assumed as  $\alpha$  times of the exact values given in Table 3 (If initial estimation is exact, the  $\alpha$  is equal to 1.0). [3] The damping coefficients are not able to be estimated, the initial values of modal damping constants are assumed to be 0.

Using the initial weight matrix calculated under these assumptions, the estimation error covariance matrix is assumed in proportion to a square of the initial weight with a proportional coefficient  $p$ . Carrying out the identification process only for initial 300 steps (3.0sec) using four values of  $p$  (0.001, 0.01, 0.1, and 1.0), the  $p$  value which minimize the summation of the performance criterion  $E$  (Eq. (3)) during 300 steps is selected as the optimal value.

The identified dynamic parameters (natural frequencies and dampings of eight modes) from the observed time histories of the model structure response for several initial conditions are shown in

**Table 4.** The optimal value of  $p$  is also shown at the bottom row. In this table, the case A is the result obtained by the proposed algorithm, the case B is a result obtained from a sinusoidal excitation test, and the case C is the exact values calculated using the values in **Table 3** under the assumption of stiffness proportional damping.

From **Table 4**, when the value of  $\alpha$  is within the range of 0.9 - 1.1 which means the error on initial stiffness to be within 10% of the exact values, the identified results are very close to the exact values by setting the appropriate value of  $p$ . As the discrepancy between the initial values and the exact values becomes large, we have to set the large value of  $p$  to get reliable identified values. Even in case that the initial estimation error exceeds 20%, the accuracy of identified parameters in lower modes is guaranteed.

**Table 4** Identified Parameters

Parameters	Mode	A Neural Network						B	C	
		$\alpha=0.80$	$\alpha=0.90$	$\alpha=0.95$	$\alpha=1.00$	$\alpha=1.05$	$\alpha=1.10$	$\alpha=1.20$	Response Curve	Calculated
f (Hz)	1	1.21	1.25	1.24	1.26	1.25	1.27	1.30	1.25	1.26
	2	3.42	3.56	3.67	3.81	3.83	3.94	3.91	3.83	3.76
	3	5.53	5.82	5.99	6.15	6.28	6.45	6.77	6.26	6.15
	4	7.45	7.92	8.14	8.45	8.55	8.76	8.66	8.51	8.35
	5	9.18	9.77	10.03	10.58	10.54	10.80	10.60	10.50	10.29
	6	10.59	11.29	11.59	11.89	12.17	12.48	12.73	12.20	11.89
	7	11.65	12.42	12.75	13.08	13.38	13.72	13.74	—	13.08
	8	12.28	13.42	13.46	13.81	14.13	14.48	13.98	—	13.81
h (%)	1	0.40	0.30	0.30	0.38	0.32	0.37	0.29	0.35	0.38
	2	0.20	0.30	0.30	0.36	0.50	0.59	0.37	0.56	0.36
	3	0.20	0.30	0.30	0.33	0.42	0.10	0.28	0.56	0.33
	4	0.00	0.40	0.30	0.28	0.67	0.09	0.68	0.58	0.28
	5	0.20	0.60	0.40	0.23	0.32	0.10	0.51	0.54	0.23
	6	0.40	0.80	0.30	0.17	0.06	0.20	0.30	0.52	0.17
	7	0.60	0.70	1.00	0.04	0.10	0.13	0.00	—	0.03
	8	0.40	0.80	0.60	0.10	0.20	0.40	0.38	—	0.10
P		1.000	0.100	0.010	0.001	0.010	0.100	1.000		

## 7. APPLICATION TO SIMULATE NONLINEAR STRUCTURAL RESPONSE

The efficiency of the Kalman filtering algorithm on the learning process of the neural

network is confirmed by its application to a linear structural system. To illustrate the applicability of developed algorithm to non-linear structural systems a two-story building similar to that shown in **Fig.3** is considered, but with versatile restoration characteristics given by

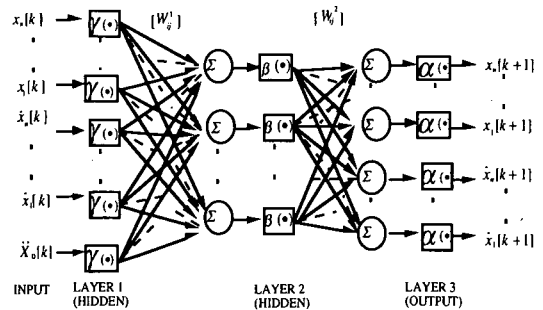
$$\dot{Z} = -\alpha|\dot{x}|Z^n - \beta\dot{x}|Z^n| + k\dot{x} \quad (18)$$

in which  $Z$  is the restoring force,  $n$  the odd number,  $\alpha$  and  $\beta$  constitutive parameters, and  $k$  the stiffness. For each story we assign the values of  $n = 1, \alpha = 1.0, \beta = 0.0$  and the stiffness given in **Fig.3**. The teaching signal is the simulated velocity and displacement of the model building excited by the modified El Centro accelerogram.

The neural network used to identify the nonlinear response pattern is shown in **Fig.9**. To give the nonlinear mapping ability, a three layered network is used in the analysis. The Processing Function of the middle layer is

$$f(x) = 2C \left\{ \frac{1}{1 + e^{-\frac{x}{T}}} - \frac{1}{2} \right\} + aC \quad (21)$$

in which the second term on the right hand side of this equation is added to assign a non-symmetric nature to the processing function. Combining the four different  $\alpha$  values of 1.0, 5.0, 10.0, 20.0 and the five different  $C$  values of 1.0, 0.5 0.0, -0.5 -1.0, we set twenty processing nodes in the middle layer. The value  $T$  is 10.0.



**Fig.9** 3-Layered Neural Network



Simulated displacement using the trained weights after 150 times global learning is shown in the upper part of Fig.10. The external excitation is the El Centro accelerogram with a modified maximum amplitude of 50 cm/sec<sup>2</sup>. The thick line, the simulated displacement, agrees well with the thin line, i.e. the exact one. The identified hysteresis shown in Fig.10 also expresses the global tendency of hysteresis.

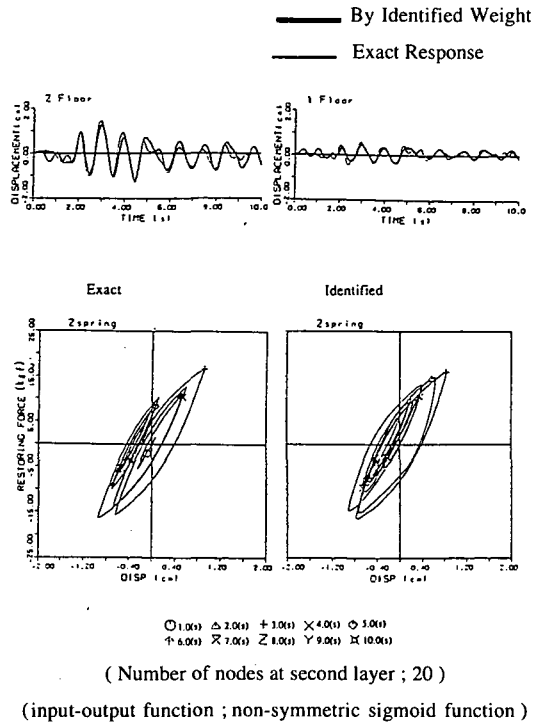


Fig.10 Time History of Resp. and Hysteresis Loop

To check the mapping ability of using a neural network fixed with trained weights, we simulated the structural response for different amplitudes of external excitation, 25 and 100 cm/sec<sup>2</sup>, and for different Taft (EW, 1952) and Hachinohe (NS, 1968) accelerograms with modified maximum amplitude of 50 cm/sec<sup>2</sup>. Results are shown in Fig.11. If the maximum external excitation amplitudes are lower than those of the teaching signals the network can output reliable structure responses.

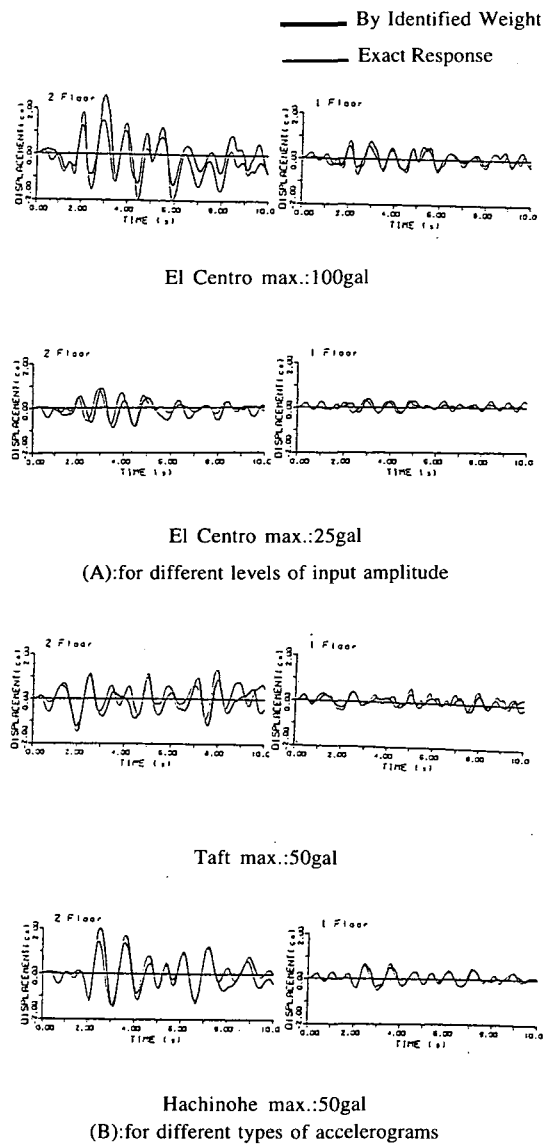


Fig.11 Time History of Structure Responses

## 8. CONCLUSION

The Kalman filtering technique was applied to identify the weight matrices of arbitrary perceptron neural networks. We investigated the efficiency of the proposed algorithm by using it to identify the dynamic structural characteristics of linear and nonlinear structural systems and compared the results with those obtained with the classical back propagation algorithm.

The proposed algorithm was very robust to noise-contaminated teaching signals for identifying the linear structural systems, whereas the classical learning algorithm often showed instability in the learning process. To check the ability to apply this algorithm to real observed data we applied the proposed algorithm to identify the dynamic structural characteristics of a eight-story linear model structure and compared the obtained results with exact values. When the initial stiffness is within  $\pm 10\%$  of exact values, we can identify not only natural frequencies but also damping constants with high accuracy. In the case that the initial estimation error is beyond  $\pm 10\%$ , we can still get good identified value for lower modes by defining the large initial estimation error covariance matrix. We could not identify the network weights for simulating the nonlinear structural system without using the neuro-Kalman algorithm.

## REFERENCES

1) H. G. Natke: *Identification of Vibration Structures*, New York, Springer, 1982.

- 2) L. Ljung: *System Identification, Theory for the User*, PRT Prentice Hall, 1987.
- 3) S. F. Masri, A. G. Chassiakos and T. K. Caughey: Structure-unknown non-linear dynamic system; identification through neural networks, *Smart Materials and Structures*, Vol.1 pp.45-56, 1992.
- 4) D.E. Rumelhart and J. L. McClelland: *Parallel Distributed Processing*, Explorations in the Microstructure of Cognition, MIT Press, 1986.
- 5) T.Sato and M.Sato: Self Learning Structural Identification Algorithm, *Proc. of Smart Structures and Materials 1995*, Vol.2443 pp.503-510, 1995.
- 6) H.Murase and S.Koyama: Application of Neural Networks to Agricultural Engineering Problem, *ASAE Paper No.91750*, pp.1-11, 1991.
- 7) H.Murase, S.Koyama, N.Honami and T.Kuwabara: Kalman Filter Neron Tarining, *Bull. Univ. of Osaka Pref. Ser.B*, Vol.43, pp.91-101, 1991.
- 8) K.Toki, T.Sato, M.Yoshikawa, M.Kurimoto and K.Inaba: Active Control of Seismic Response of Structure by aMass Driver System: First World Conference on Structural Control, Vol.3, pp.FP1-33 - FP1-42. 1994.

(Received November 29 ,1995)

## ニューラル・カルマンフィルターを用いた構造物の地震時応答特性の同定

佐藤忠信・佐藤誠

ニューラルネットワークの持つ並列処理能力、非線形写像能力に着目し、構造物の応答特性を同定する手法を展開した。ネットワーク中の重み係数の学習則として、拡張カルマンフィルターを用いた重み係数の学習法を利用した。線形な復元力特性を持つ解析モデルの同定を通して、観測の際にノイズが混入する場合にも安定した学習を行うことが可能であり、精度のよいパラメーターの同定が行えることを示し、構造物が非線形な復元力特性を有する場合にも、ネットワーク構造を工夫することで、構造物が地震時に示す応答のシミュレーションを行うことができることも明らかにした。また、実験構造物模型の振動試験により得られた地震時応答の観測データを基に模型の動特性を同定し、提案手法の実問題への適用性も検証した。