An Artificial Neural-Net Based Technique for Power System Dynamic Stability with the Kohonen Model

Hiroyuki Mori Yoshihito Tamaru Senji Tsuzuki
Department of Electrical Engineering
Meiji University
Tama-ku, Kawasaki 214
Japan

Abstract - This paper presents an artificial neural-net based method for evaluating on-line power system dynamic stability. Using the matrix transformation of the S-matrix method, the absolute value of the most critical eigenvalue in z-plane may be regarded as a power system dynamic stability index. This paper utilizes the artificial neural-net of Kohonen to estimate the index so that computational efforts are reduced and numerical instability problems are avoided. The Kohonen model is based on the self-organization feature mapping (SOFM) technique that transforms input patterns into neurons on the two-dimensional grid. The advantages of the model are as follows: 1) the algorithm does not require the teacher's signals; 2) the algorithm is not so complicated; 3) the resulting mapping is visually easy to understand the input pattern. In this paper, power system conditions are assigned to the output neurons on the two-dimensional grid with the SOFM technique. Two methods are presented to calculate the estimate index so that an output neuron calls the index corresponding to an input pattern. A comparison of the linear and nonlinear decreasing function employed at the learning process is made. The effectiveness of the proposed method is demonstrated in a sample system.

Keywords: dynamic stability, S-matrix method , artificial neural network, competitive learning, self-organization feature mapping

I. INTRODUCTION

Power system dynamic stability has been studied so that power system operators are able to maintain sound operation [1]-[10]. Requirements for dynamic stability techniques are both the computational efficiency and high accuracy. Eigenvalue analysis is one of the conventional methods in dealing with the dynamic stability problems. The stability is evaluated by calculating the eigenvalue of the system matrix in the linearized dynamic equation. As the conventional method, the QR method has been widely used due to the robustness and the computational efficiency. However, the method has a drawback that it requires a lot of computational efforts as the size of power systems becomes larger. In other words, the method is not so practical in a sense that it has to evaluate all the eigenvalues although the power system operators are interested in the most critical eigenvalue. As a result, different improvements are made to overcome the problem of the QR method. Uchida developed the S-matrix method that focuses on evaluating only the most critical eigenvalue [3]. Obata, Takeda and Suzuki proposed a method for estimating the eigenvalue with the operational transfer matrix [4]. Byerly, Bennon and Sherman developed the computer program for large power systems , AESOPS (Analysis of Essentially Spontaneous Oscillations in Power System) [5]. Martins implemented the implicit inverse iteration algorithm at AESOPS [6]. Wong, et al., developed a sparse-oriented method called PEALS (Program for Eigenvalue Analysis of Large Power Systems) [7]. Uchida and Nagao presented the refined Lanczos method to solve the S-matrix method efficiently [8]. Also, Semlyen and Wang proposed a two-stages method for sparse-oriented reduced order eigenvalue analysis which is named STEPS (Sequential Two-stage Eigenvalue Analysis for Power System) [9]. Furthermore, they contributed to developing the sparsity-oriented simultaneous iteration and modified Arnoldi methods [10].

Papers presented at the Seventeenth PICA
Conference at the Hyatt Regency Baltimore
Hotel, Baltimore, Maryland, May 7 - 10, 1991
Sponsored by the IEEE Power Engineering Society

Neural computing is one of the promising technologies in all the engineering fields. A number of artificial neural networks (ANNs) are developed so far. The basic techniques can be divided into three groups given in Table 1; ANN1 corresponds a multi-layer perceptron that can be constructed by the backpropagation algorithm [29]; ANN2 is the self-organization feature mapping (SOFM) developed by Kohonen [30]; ANN3 indicates the massively interconnected neural networks such as the Hopfield Net or the Boltzmann Machine. ANN1 is suitable for pattern recognition, system identification and predicting, etc. ANN2 can be used in classification, combinatorial problems, etc. ANN3 can solve the combinatorial, classification problems, etc. As far as power systems are concerned, Sobajic and Pao contributed to developing an ANN-based method for power system problems. They applied an artificial neural network to dynamic security assessment [13]. Since then, power system problems have been studied with ANNs [14-26]. The conventional applications are divided into two categories, i.e., ANN1 and ANN3 in Table 1. The former is utilized in security assessment [13], [25], alarm processing [14], harmonic load identification [16], fault diagnosis [17], [21], voltage control [19], harmonic source monitoring [20], harmonics prediction [22], [26], load forecasting [24]. On the other hand, the latter is applied to topological observability [18], [23]. ANN2 have never been applied to any power system problems although most of the ANN applications utilizes either ANN1 or ANN2. In this paper, the application of ANN2 is considered to classify the system conditions and estimate the dynamic power stability index.

This paper presents an artificial neural-net based technique for power system dynamic stability. The method is based on estimating the dynamic stability index that corresponds to the most critical eigenvalue of the S-matrix method. ANN-based pattern recognition is carried out with the self-organization feature mapping (SOFM) developed by Kohonen [30]. The differences between the proposed method and the conventional methods with ANNs [15], [25] are given as follows:

1) The proposed method uses unsupervised learning of the Kohonen model while the conventional methods utilize supervised learning of the backpropagation. In other words, the proposed method does not require teacher's signals.

Table 1 Classification of Artificial Neural Networks

	Feedforward	Feedback
Supervised	ANN1	
Unsupervised	ANN2	ANN3

0885-8950/92$03.00©1992 IEEE

2) The proposed method discusses the estimation of the dynamic power system index while the conventional methods deal with identification of the stability region at an individual node representing P-Q plane (P: active power at a node, Q: reactive power at a node). Thus, the proposed method focuses on relationship between the system conditions of all the nodes and the index.

The proposed method is applied to a sample system and the efficiency is examined. Also, the influence of the decreasing function used in updating weights of the SOFM algorithm is investigated. The contents of this paper are organized as follows: Section II reviews the S-matrix method; Section III outlines the self-organization feature mapping; Section IV deals with the application of the Kohonen model to power system dynamic stability; Section V describes the simulation results.

II. REVIEW OF S-MATRIX METHOD

This section reviews the S-matrix method that makes use of the maximum eigenvalue in absolute value in z-plane. In 1980, Uchida developed an efficient method for evaluating the power system dynamic stability [3]. The method focused on calculating the most critical eigenvalue in z-plane rather than all the eigenvalue in s-plane.

Now, consider a power system under small disturbances. The linearized system state equation can be written as

$$\dot{\underline{w}} = \underline{A}_s \underline{w} \qquad (1)$$

where, $\underline{w}$: state variable vector

$\underline{A}_s$: system matrix

The S-matrix method employs the transformation of the left half-plane into the unit circle. Transforming system matrix $\underline{A}_s$ into the following matrix:

$$\underline{A}_z = (\underline{A}_s + h\,\underline{I})(\underline{A}_s - h\,\underline{I})^{-1} \qquad (2)$$

where, $\underline{A}_z$: transformed system matrix

$\underline{I}$: unit matrix

h: positive real number

Eqn. (2) indicates a mapping transformation from s-plane to z-plane as shown in Fig. 1, where the difference between the stable region in s-plane and in z-plane is given in hatched area. Fig. 1 (a) denotes the stable region in s-plane while Fig. 1(b) illustrates that in z-plane. The advantage in z-plane is that the system stability depends upon whether the eigenvalue exist within the unit circle. As a result, power system dynamics is evaluated by the absolute value of most critical eigenvalue of matrix $\underline{A}_z$.

In the S-matrix method, the power system dynamic stability can be judged by the absolute value of the most critical eigenvalue such as

$$\mu = |\lambda_z^M| \qquad (3)$$

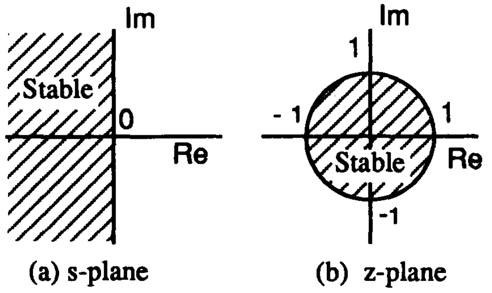
where, $|\bullet|$: absolute value of $\bullet$

λ_z^M : most critical eigenvalue of matrix $\underline{A}_z$

Namely, we have

$$\left.\begin{array}{lll} \mu & < 1: & \text{(stable)} \\ \mu & = 1: & \text{(critical)} \\ \mu & > 1: & \text{(unstable)} \end{array}\right\} \qquad (4)$$

Since the method makes uses of mapping of the eigenvalue from s-plane to z-plane, the most critical eigenvalue in s-plane is equivalent to the eigenvalue with the largest absolute value in z-plane. The power system dynamic stability can be judged by examining if the eigenvalue with the largest absolute value exists within the unit circle. Uchida proposed the power method to calculate the critical eigenvalue [3]. The algorithm is very simple, but the method often has poor convergence characteristics as well as low accuracy. In 1987, Uchida and Nagao developed the refined Lanczos method for evaluating the eigenvalue [8]. It is well known that the Lanczos does not provide rigorous results due to the round-off errors or the numerical instability. Although several improvements of the Lanczos method are made, they are based on the experience rather than theoretically guaranteed backgrounds [12]. Also, they depend on available data. Thus, it seems that the mapping of the eigenvalue is theoretically attractive in the S-matrix method, but the efficient method is not obtainable for evaluating only the most critical eigenvalue in z-plane.



(a) s-plane (b) z-plane

Fig. 1 Stable Region of Eigenvalues in s-plane and z-plane

This paper proposes a pattern recognition technique for evaluating dynamic stability. Using the SOFM algorithm , the two-dimensional neural network is constructed to estimate index μ.

III. SELF-ORGANIZATION FEATURE MAPPING (SOFM)

3.1 Competitive Learning

This paragraph briefly describes the concept of competitive learning that is related to the self-organization feature mapping (SOFM) developed by Kohonen [30]. The SOFM technique is classified into comparative learning that is one of the unsupervised learning techniques. Depending on the use of the teacher's signal, the learning algorithms are divided into supervised and unsupervised. The competitive learning is based on the rule that a neuron is fired while the others are suppressed not to be fired [29]. The ranges of fired neurons varies as the learning process proceeds. The neural network has the competitive relationship in a sense that once a neuron responds to a specific pattern, other neurons never reply to it. In other words, the input pattern that fires a certain neuron makes the neighboring neurons unfired. Neurons with distance d from the fired neuron have a weight defined as ω(d) which is called the Mexican hat function (see Fig. 2). This function enables us to carry out competitive learning so that the neighboring neurons are fired while other neurons are suppressed not to be fired. The output neuron is determined by reducing the selectivity of the neighboring neurons for an input pattern. Let us consider a neural network with a feedback loop as shown in Fig. 3. The output of neuron i can be expressed as

$$O_i(k) = f(I_{1i}(k) + I_{2i}(k) - \theta_i) \qquad (5)$$

where, $f(\bullet)$: nonlinear function

$I_{1i}(k)$: *input at neuron i from input units*

$I_{2i}(k)$: input at neuron i from output feedback

θ_i: threshold value

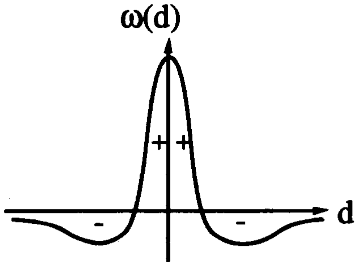


Fig. 2 Mexican Hat Fuction

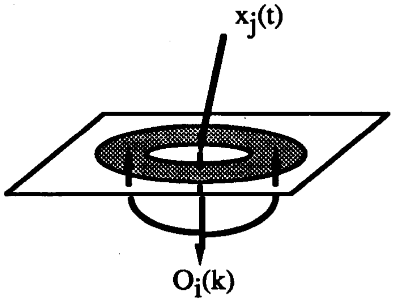


Fig. 3 Competitive Learning

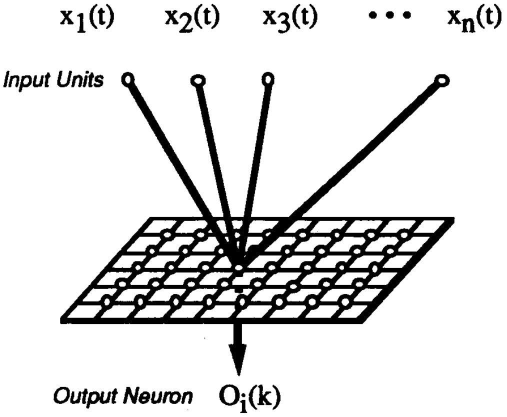


Fig. 4 Self-Orgnaization Feature Mapping

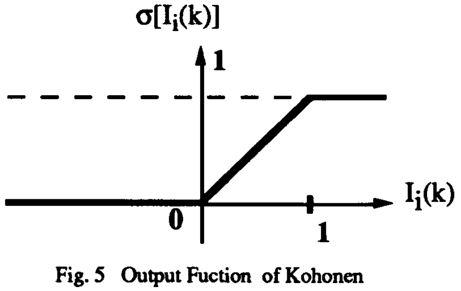


Fig. 5 Output Fuction of Kohonen

k: learning step

It should be noted that Eqn. (5) has a complicated form due to the nonlinearity. As a result, the learning process is difficult to handle.

3.2 Topology-Preserving Mapping

Kohonen proposed a simplified rule for carrying out the competitive learning. This method is regarded as a vector quantizer since it is based on the weight modification in the vector clustering. As shown in Fig. 4, the neural network can be expressed as a two-dimensional grid that consists of output neurons. There exists a set of input units that connect with all the output neurons. Fig. 4 shows a case where a input pattern fires an output neuron. Now let us define the input signal of dimension n as $\underline{x}(t)$.

$$\underline{x}(t) = (x_1(t), x_2(t), ..., x_n(t))^T \qquad (6)$$

where t: input pattern number ($t = 1, 2, 3, ..., t_{max}$)
(t_{max}: number of input patterns)

n: number of input units

The input of output neuron i can be expressed as

$$I_i(k) = \sum_j x_j(t)\, w_{ij}(k) \qquad (7)$$

or

$$= \underline{x}(t)^T \underline{w}_i(k) \qquad (8)$$

where, $w_{ij}(k)$: weight between neurons i and j

$\underline{w}_i(k)$: weight vector of neuron i such as

$$\underline{w}_i(k) = (w_{i1}(k), w_{i2}(k), ..., w_{in}(k))^T \qquad (9)$$

Weight vector $\underline{w}_i(k)$ in Eqn. (8) is called the reference vector in clustering. For convenience the magnitude of vectors $\underline{x}(t)$ and $\underline{w}_i(k)$ are normalized as follows:

$$\| \underline{x}(t) \| = 1 \qquad (10)$$

$$\| \underline{w}_i(k) \| = 1 \qquad (11)$$

where, $\| \bullet \|$: magnitude of vector •

The output of output neuron i is given

$$O_i(k) = \sigma(I_i(k)) \qquad (12)$$

where, σ(•): nonlinear function (see Fig. 5) such as

$$\left.\begin{array}{lll} \sigma(I_i(k)) & = 1 & \text{if } 1 < I_i(k) \\ \sigma(I_i(k)) & = I_i(k) & \text{if } 0 \le I_i(k) \le 1 \\ \sigma(I_i(k)) & = 0 & \text{if } I_i(k) < 0 \end{array}\right\} \qquad (13)$$

Once the neural network is trained, the input pattern fires the output neuron closest to the input pattern in terms of a criterion. The method creates the network that allocates the features of input patterns to the nodes. In particular, what is of interest in the SOFM algorithm is that the nodes with similar characteristics are close to each other on the grid. Therefore, it is easy to extract some characteristics of input patterns on the two-dimensional grid. This allows us to visually understand the characteristics.

3.3 Updating Weights

This paragraph describes a scheme of updating weights between output neurons in the SOFM technique that is based on the Hebbian learning rule [27]. Hebb proposed an unsupervised rule that the weights connected to fired neurons are increased. The weights are changed by

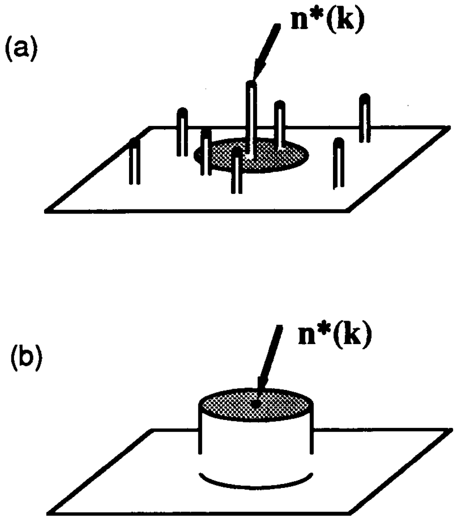


Fig. 6 Occurance of Bubble at SOFM

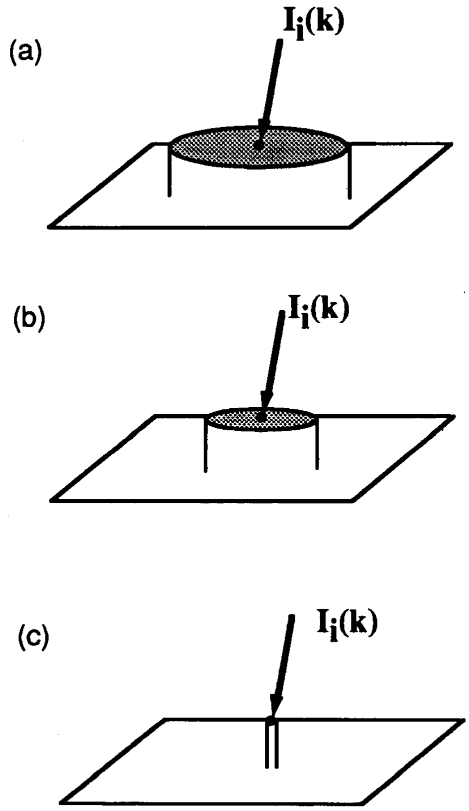


Fig. 7 Determination of Output Neuron Using Bubble

$$\Delta w_{ij}(k) = \eta\, I_i(k) x_j(k) \qquad (14)$$

where $\Delta w_{ij}(k)$: correction of weight between neurons i and j at learning step k

η: learning rate

However, the Hebbian learning has instability that the weights are excessively increased once the neurons are fired. von der Marlsburg developed a competitive learning rule that the sum of weights between neurons was constant [28]. Afterward, some competitive rules were proposed by several researchers. Kohonen employed the following equation:

$$\Delta w_{ij}(k) = \alpha(k)\, (I_i(k) x_j(k) - O_i(k)\, w_{ij}(k)) \qquad (15)$$

where, $\alpha(k)$: decreasing function with learning step k

$$0 < \alpha(k) < 1 \qquad (16)$$

The first and second terms in Eqn. (15) correspond to the Hebbian learning rule and the output feedback, respectively.

Kohonen introduced the concept of "bubbles" as to how to select neurons to be fired. The response of neurons are expressed with "bubbles" in the self-organization process. As shown in Fig. 6 (a), there are several output neurons that respond to the input pattern. Among the output neurons, the neuron with the largest output becomes the center of the bubble. Let us define it as n*(k). The input and output of neurons in Eqn. (15) are determined by the Euclidean norm, i.e. the distance between neurons such as

$$r_i = \| n_i - n^*(k) \| \qquad (17)$$

where, $\| \bullet \|$: Euclidean norm

n_i : neuron i

If the following equation is satisfied,

$$r_i \leq \gamma(k) \qquad (18)$$

where, $\gamma(k)$: radius from $n^*(k)$ which is a function of learning step k

then we have

$$I_i(k) = 1, \quad O_i(k) = 1 \qquad (19)$$

i.e., Eqn. (15) becomes

$$\Delta w_{ij}(k) = \alpha(k)\, (x_j(t) - w_{ij}(k)) \qquad (20)$$

Otherwise,

$$I_i(k) = 0, \quad O_i(k) = 0 \qquad (21)$$

From Eqn. (21), Eqn. (15) becomes

$$\Delta w_{ij}(k) = 0 \qquad (22)$$

Neurons around n*(k) satisfying Eqn. (18) are called "topological neighborhood". Fig. 6(b) shows the bubble or the topological neighbourhood. As the self-organization proceeds, the radius of the bubble decreases due to $\gamma(k)$ in Eqn. (18) and eventually the neuron to be fired for an input pattern is determined. Fig. 7 (a) - (c) illustrates the transition of the bubble. Fig. 7 (a) shows the initial stage while Fig. 7(c) designates the final stage.

The algorithm of SOFM can be summarized as follows:

Step1: Set learning step k =0 and inner loop count t = 0. Then prepare the initial value of weights $w_{ij}(0)$.

Step 2: Set k = k +1 .

Step 3: Set $t = t+1$ and give input signal $x(t)$.

Step 4: Calculate neuron $n^*(k)$ that have the closest distance, $d^*(k)$, to input pattern $x(t)$, where $d^*(k)$ is given

$$d^*(k) = \min_i \{d_i(k)\}$$

$$= \min_i \{ \sum_j (x_j(t) - w_{ij}(k))^2 \} \qquad (23)$$

Step 5: Find topological neighbourhood around $n^*(k)$ using Eqn. (18).

Step 6: Update the weights for $n^*(k)$ and topological neighborhood.

$$w_{ij}(k+1) = w_{ij}(k) + \alpha(k)(x_j(t) - w_{ij}(k)) \qquad (24)$$

Step 7: If $k = k_{max}$, then stop. Otherwise go to Step 8.

Step 8: If $t = t_{max}$, then set $t = 0$ and go to Step 2. Otherwise, go to Step 2.

IV. APPLICATION OF THE KOHONEN MODEL TO POWER SYSTEM DYNAMIC STABILITY

This section proposes a method for calculating the dynamic stability index with the self-organization features mapping mentioned in the previous section.

Consider two nodal quantities at each node as the input vector

$$\underline{x}(t) = (V_1(t), P_2(t), Q_2(t), ..., P_m(t), Q_m(t))^T \qquad (25)$$

where, P_p: active power at node p

Q_p: reactive power at node p

V_p: magnitude of voltage at node p

m: number of nodes

Eqn. (25) corresponds to the specified value at node in the load flow calculation. It is assumed in Eqn. (25) that node 1 is the slack node and the voltage magnitude is given for node 1. However, the expression of the power system conditions has a drawback if we have the following linear relationship:

$$\underline{x}(t_1) = c\,\underline{x}(t_2) \qquad (c: \text{constant}, t_1 \neq t_2) \qquad (26)$$

Since the normalization of the input vector is carried out with Eqn. (10), the original input vectors satisfying Eqn. (26) make no difference. To overcome the problem, the unity is added to the original data. Thus, Eqn. (25) can be rewritten as

$$\underline{x}(k) = (V_1(k), P_2(k), Q_2(k), ..., P_m(k), Q_m(k), 1)^T \qquad (27)$$

On the other hand, each output neuron in the two dimensional grid corresponds to one category of the power system conditions.

In this paper, the power system dynamic index is estimated by calling an output neuron where the estimated index is assigned. After an output neuron on the grid responds to a input pattern, the output calls the estimated index given in Eqn. (3). It is assumed that the index is prepared by the off-line calculation. Now, define the estimate as μ_e. In this paper, the following two methods are proposed to evaluate μ_e.

Method A

Let us assume that more than one learning data is assigned to each output neuron. The estimated index is calculated by the average value of the index corresponding to power system conditions classified into the same output neuron. Namely

$$\mu^i_e = \frac{1}{N_i} \sum_k \mu^i_k \qquad (28)$$

where, μ^i_e: estimated index at output neuron i

N_i: number of input patterns classified into output neuron i

μ^i_k: index of input pattern k classified into output neuron i

The algorithm of Method A can be described as

Step 1: Prepare for learning patterns of SOFM and index μ for each learning pattern.

Step 2: Carry out SOFM.

Step 3: Calculate μ^i_e at output neuron i on the grid using Eqn. (28). Store it at each output neuron.

Step 4: Give an unknown input pattern to the constructed network. Then, evaluate output neuron i* that is closest to the input pattern.

Step 5: Call μ^{i*}_e at output neuron i*.

Method B

The other method is based on the fact that the weight vector or reference vector represents the inherent input pattern corresponding to the output neuron. In other words, the features of the output neuron is expressed in the weights between the output neuron and the input units. The estimated index is obtained using the weight vectors after finishing the learning process to recalculate index μ.

Similarly, the algorithm of Method B can be written as

Step 1: Prepare for learning patterns of SOFM and index μ for each learning pattern.

Step 2: Carry out SOFM.

Step 3: Recalculate μ^i with the weight vector at each output neuron i.

Step 4: Give an unknown input pattern to the constructed network. Then, evaluate output neuron i* that is closest to the input pattern.

Step 5: Call μ^{i*}_e at output neuron i*.

V. SIMULATION

5.1 Simulation Conditions

To demonstrate the effectiveness of the proposed methods, they are tested in a nine bus system as shown in Fig. 8. The following conditions are employed:

(a) As the learning data in this simulation, 5000 power system conditions are used to construct the two-dimensional grid. Namely, we have $t_{max} = 5000$. The specified values of the load flow calculation are determined by the uniform random number with the lower and upper bounds given in Table 2, where P, Q, V denote nodal active power, reactive power and voltage magnitudes, respectively. It should be noted that bus 1 is the slack node and that buses 4, 7 an 9 have no bounds due to floating nodes. The active and reactive power at nodes 4, 7 and 9 of floating nodes are omitted from the input vector. The number of input units is equal to twelve. Thus, from Eqn. (27), the following input vector is used as the input pattern:

$$\underline{x}(k) = (V_1(k), P_2(k), Q_2(k), ..., P_8(k), Q_8(k), 1)^T \quad (29)$$

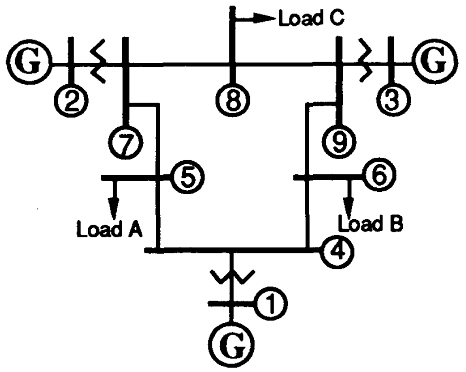


Fig. 8 Nine-node System

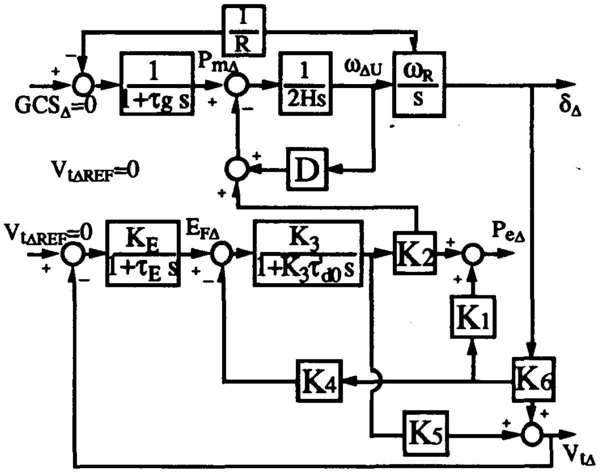


Fig. 9 AVR and Governor Models

Table 2 Lower and Upper Bounds of Node Specified Values

Node	P[p.u.]		Q[p.u.]		V[p.u.]	
No	Lower	Upper	Lower	Upper	Lower	Upper
1	—	—	—	—	0.9	1.1
2	0.5	1.5	—	—	0.9	1.1
3	0.5	1.5	—	—	0.9	1.1
4	0.0	0.0	0.0	0.0	—	—
5	-1.5	- 0.5	-2.0	0.0	—	—
6	-1.5	- 0.5	-2.0	0.0	—	—
7	0.0	0.0	0.0	0.0	—	—
8	-1.5	- 0.5	-2.0	0.0	—	—
9	0.0	0.0	0.0	0.0	—	—

(b) Using the system conditions obtained by the load flow calculation, the dynamic stability index μ is calculated for each learning pattern with the QR method. From Eqn. (2), index μ has the following relationship with the most critical eigenvalue of matrix $\underline{A}_s$:

$$\mu = (\lambda_s^M + h)/(\lambda_s^M - h) \qquad (30)$$

where, λ_s^M is the most critical eigenvalue of matrix $\underline{A}_s$ in Eqn. (1) and parameter h is set to 0.005 in this simulation.

(c) The three generators in Fig. 8 are expressed by the d-q two axis model [11]. The AVR and governor models of each generator are given in Fig. 9. The state vector expressing generator k is

$$\underline{w}_k = (E_{qk}, E_{dk}, \omega_k, \delta_k, T_{mk}, E_{FDk})^T \qquad (31)$$

where, $E_{qk}(E_{dk})$: d-axis (q-axis) node voltage

ω_k: angular velocity

δ_k: angle

P_{mk}: turbine output

E_{FDk}: field voltage

The loads are expressed with the equivalent impedance. Thus, the size of the system matrix is 17 ×17 because of the three machine system.

(d) The two-dimensional grid consists of 30 x 30 output neurons. In other words, 5000 power system conditions are classified into 900 patterns.

(e) In the SOFM algorithm, the initial value of the weights, $w_{ij}(0)$, is given by the uniform random number that satisfies the bounds in Table 2. That is because the weight vector or reference vector actually represents the classified power system condition in SOFM, where, from Eqn. (11), the weight vector is normalized to the unity. It is assumed that the decreasing functions $\alpha(k)$ and $\gamma(k)$ given in Eqn. (15) and (18), respectively, takes the following linear and nonlinear models:

Linear model

$$\alpha_L(k) = 0.7\,(1 - \frac{k}{k_{max}}) \qquad (32)$$

$$\gamma_L(k) = 10\,(1 - \frac{k}{k_{max}}) \qquad (33)$$

Nonlinear model

$$\alpha_N(k) = 0.7 \exp(-3 \times k/k_{max}) \qquad (34)$$

$$\gamma_N(k) = 10 \exp(-3 \times k/k_{max}) \qquad (35)$$

where, k_{max} is the number of learning steps that is employed for the learning process. The following ten kinds of k_{max} is used:

$$k_{max} = t_{max} \times n_k = 5000 \times n_k \qquad (36)$$

(where, n_k = 1, 2, 3,..., 10)

Eqns. (36) implies that ten kinds of the networks are constructed in this simulation for each method.

Since Methods A and B have the two decreasing functions, networks constructed with four SOFM algorithms are constructed. Thus, we have the following cases:

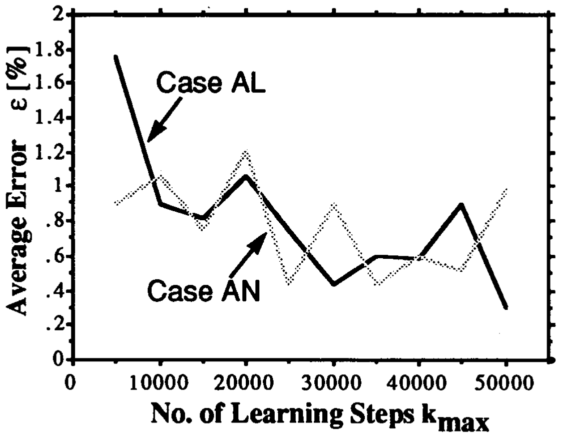


Fig. 10 No. of Learning Steps vs. Average Error in Method A

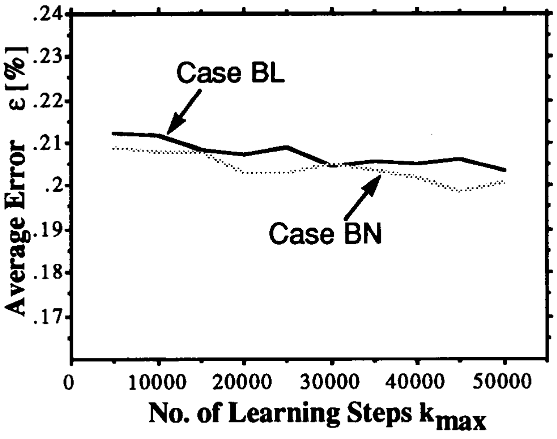


Fig. 11 No. of Learning Steps vs. Average Error in Method B

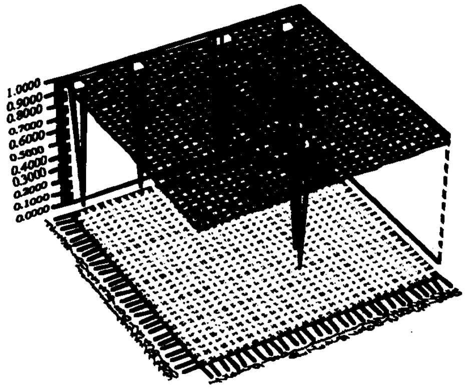


Fig. 12 Self-Organization Feature Mapping in Case AL (where, k_{max} =50000)

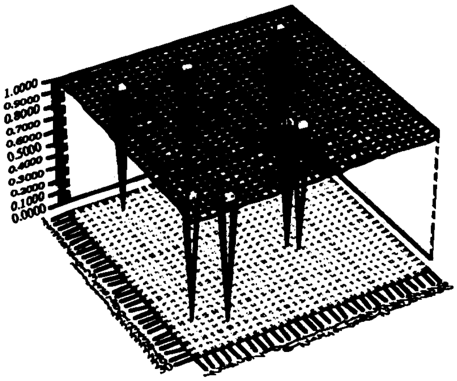


Fig. 13 Self-Organization Feature Mapping in Case AN (where, k_{max} =50000)

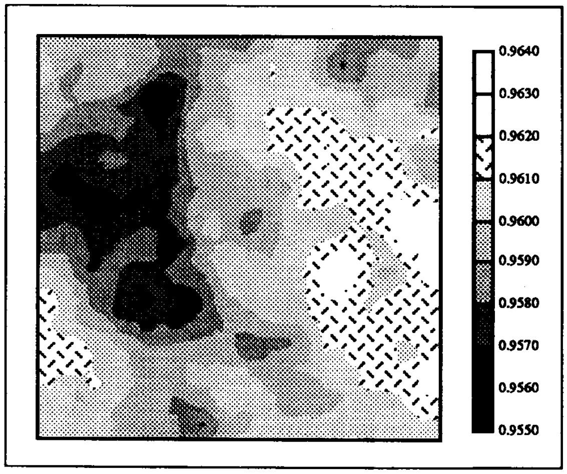


Fig. 14 Self-Organization Feature Mapping in Case BL (where, k_{max} =50000)

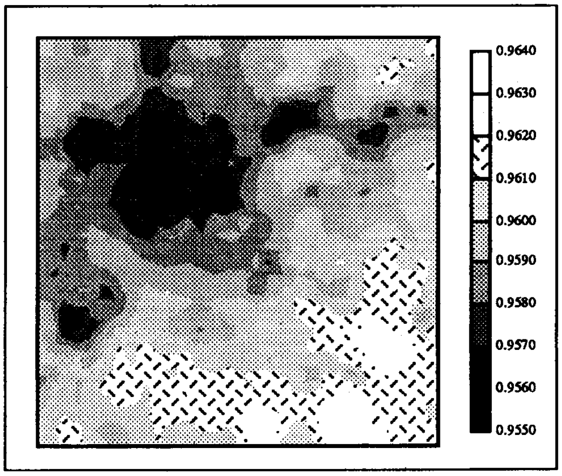


Fig. 15 Self-Organization Feature Mapping in Case BN (where, k_{max} =50000)

Case AL: Method A with the linear decreasing functions given in Eqns. (32) and (33)

Case AN: Method A with the nonlinear decreasing functions given in Eqns. (34) and (35)

Case BL: Method B with the linear decreasing functions given in Eqns. (32) and (33)

Case BN: Method B with the nonlinear decreasing functions given in Eqns. (34) and (35)

(f) In order to test the efficiency of the networks, 1300 power system conditions are used as the test data. The conditions are determined by the different uniform random number from the learning data. As the index to measure the accuracy of the networks constructed with SOFM, the following average error is employed:

$$\varepsilon = \frac{1}{M} \sum_{k} \frac{|\mu^k_e - \mu^k_t|}{\mu^k_t} \times 100 \quad [\%] \qquad (37)$$

where, M: number of test patterns

μ^k_e: estimate of the index for input pattern k

μ^k_t: true value of the index for input pattern k

The true value of the index, μ^k_t, in Eqn. (37) is the absolute value of the most critical eigenvalue given in Eqn. (3). In this paper, μ^k_t is calculated with the QR method.

5.2 Results

Table 3 shows the simulation results of four cases. In the table, the average error given by Eqn. (37) is shown in percent. It can be observed from the table that Case BN has higher accuracy of average error 0.2030 [%] than other cases and that Case BL is the second best case. Thus, Method B is superior to Method A in terms of the accuracy of the estimates. Figs. 10 and 11 show the relationship between the number of the learning step (k_{max}) and the average error ε in Methods A and B, respectively. In Fig. 10, it is shown that increasing the number of the learning steps does not imply enhance the accuracy. That is because, as mentioned later, Method A gives output neurons where the estimated index is not assigned. Comparing Fig. 10 with Fig. 11, the simulation results shows that ε in Cases BL and BN decreases almost monotonously that that in Cases AL and AN. The nonlinear decreasing function contributed to enhancing the accuracy in Method B while it deteriorated the accuracy in Method A.

Also, Table 3 gives the number of the output neurons where the estimated index is not given. Since Method A calculates the estimated index with Eqn. (28), there is a possibility that the estimated index is not assigned to some neurons if any learning data is not classified into them. Figs. 12 and 13 show the self-organization features mapping in Cases AL and AN at k_{max}= 50000, respectively. For convenience, they are expressed in three-dimension to understand the output neurons where the estimated index is not assigned. The x-axis and y-axis show the two-dimensional grid of the output neurons. Also, the z-axis indicates the estimated index. From the figures, it can be observed that there are seven holes in both Figs. 12 and 13. Since the holes correspond to the output neurons where the estimated index is not assigned, μ_e at such neurons is set to zero in the figures. On the other hand, Method B can evaluates the estimated index with the reference vector regardless of the relationship between output neurons and the number of learning patterns belonging to them. As a result, Method B provided more reliable solutions in a sense that Method B does not yield the output neurons where the estimated index is not assigned.

Table 3 Average Error for Test Data and Number of Output Neurons Where the Estimated Index Is Not Given

No. of Learning	CASE			
Steps (k_{max})	AL	AN	BL	BN
5000	1.7532 (38)	0.9004 (13)	0.2124 (0)	0.2095 (0)
10000	0.7442 (12)	1.0628 (8)	0.2091 (0)	0.2076 (0)
15000	0.8226 (8)	0.7445 (8)	0.2082 (0)	0.2078 (0)
20000	1.0587 (12)	1.2122 (11)	0.2075 (0)	0.2030 (0)
25000	0.7442 (12)	0.4347 (3)	0.2091 (0)	0.2030 (0)
30000	0.4361 (6)	0.9015 (6)	0.2046(0)	0.2052 (0)
35000	0.5954 (10)	0.4329 (11)	0.2055(0)	0.2035 (0)
40000	0.5913 (6)	0.5946 (5)	0.2049 (0)	0.2017 (0)
45000	0.9030 (10)	0.5180 (6)	0.2060 (0)	0.1983 (0)
50000	0.2928 (7)	0.9780 (7)	0.2037 (0)	0.2009(0)

Note: (•) indicates the number of the output neurons where the estimated index is not assigned.

Figs. 14 and 15 give the simulation results in Cases BL and BN at k_{max}= 50000, respectively. They show the contour of the estimated index which corresponds to the self-organization feature mapping (30×30) in two dimension. As the region colour becomes darker in the figures, the power system approaches securer conditions. However, it should be noted that from the standpoint of neuron topology, the mapping of the neurons in Fig. 14 is different from that in Fig. 15 because of the different learning process. In other words, the characteristics of neurons in Fig. 14 do not correspond to those in Fig. 15 in terms of the topology. That is the reason why Fig. 14 has the different mapping from Fig. 15.

From Table 3 and Figs. 10-15, it can be found that Method B provided the better results than Method A. Investigating the computation efficiency of Method B, the CPU time of the learning process in Cases BL and BN took 753 [s] and 599 [s] at k_{max}= 50000. That is because the decreasing function in Case BN decreases faster than that in Case BL. The calculation was performed on Fujitu FACOM 380S. On the other hand, it took about 9.7 [ms] to process one test pattern and estimate index μ. Thus, the proposed method is well suitable for on-line applications.

Once the mapping is created, the proposed method is helpful for power system operation and control. Since the mapping is expressed in two dimension, it is easy to visually understand both secure and insecure conditions close to the operational conditions. At the same time, tracking of power system conditions on the mapping enables us to conjecture which conditions a power system is heading for. Therefore this method allows power system operators to monitor power system conditions efficiently.

VI. CONCLUSIONS

The results of this paper can be summarized as follows:

(1) This paper has presented a method for estimating a dynamic stability index with an artificial neural network. As the learning algorithm, the self-organization feature mapping of Kohonen was utilized to construct the network. This algorithm has an advantage that it does not requires the teacher's signals.

(2) Concerning the estimation of the index, two methods (Methods A and B) have been proposed. It can been seen that Method B is superior to Method A in terms of the accuracy and the reliability. The simulation results have shown that Method B provided the good results with the average error of about 0.2[%].

(3) A comparison has been made between the linear and nonlinear decreasing functions for updating weights between output neurons. It can be seen that the nonlinear decreasing function provided more rigorous results in Method B. Thus, it is suggested that the nonlinear decreasing function be employed in updating the weights in Method B.

(4) The self-organization feature mapping allows us to understand the transition of the stability margin on the two-dimensional

grid. From a standpoint of the computational efficiency, this method is well suitable for on-line applications. Based on tracking the system conditions on the grid, it is possible to understand how the power system to be studied performs. Thus, the proposed method is promising in a sense that the mapping technique helps power system operators to deal with security assessment.

REFERENCES

[1] M.A. Laughton, "Matrix Analysis of Dynamic Stability in Synchronous Multimachine Systems", Proc. IEE, Vol. 113, No. 2, pp. 325-336, Feb. 1966.

[2] P. Kunder and P.L. Dandeno, "Practical Application of Eigenvalue Techniques in the Analysis of Power System Stability Problems", Proc. PSCC V, pp. 325-336, Cambridge, U.K., Sept. 1975.

[3] N. Uchida, "Dynamic Stability Studies of Power Systems by a New Eigenanalysis Method", Trans. of IEE of Japan, Vol. 100-B, pp. 389-396, July 1980 (in Japanese).

[4] Y. Obata, S. Takeda and H. Suzuki, "An Efficient Eigenvalue Estimation Technique for Multi-Machine Power System Dynamic Stability Analysis", IEEE Trans. on Power App. and Syst., Vol. 100, pp. 259-263, Jan. 1981.

[5] R. T. Byerly, R.J. Bennon, and D.E. Sherman, "Eigenvalue Analysis of Synchronizing Power Flow Oscillations in Large Electric Power Systems", IEEE Trans. on Power App. and Syst., Vol. 101, pp. 235-243, Jan. 1982.

[6] N. Martins, "Efficient Eigenvalue and Frequency Response Methods Applied to Power System Small-Signal Stability Studies", IEEE Trans. on Power Syst., Vol. 1, pp. 217-266, Jan. 1986.

[7] D.Y. Wong, G.J. Rogers, B. Porretta and P. Kunder, "Eigenvalue Analysis of Very Large Power Systems", IEEE Trans. on Power Syst., Vol. 3, No. 2, pp. 472-480, May 1988.

[8] N. Uchida and T. Nagao, "A New Eigen-Analysis Method of Steady-State Stability Studies for Large Power Systems: S Matrix Method", IEEE Trans. on Power Syst., Vol. 3, No. 2, pp. 706-714, May 1987.

[9] A. Semlyen and L. Wang, "Sequential Computation of the Complete Eigensystem for the Study Zone in Small Signal Stability Analysis of Large Power Systems", IEEE Trans. on Power Syst., Vol. 3, No. 2, pp. 715-725, May 1988.

[10] L. Wang and A. Semlyen, "Application of Sparse Eigenvalue Techniques to the Small Signal Stability Analysis of Large Power Systems", IEEE Trans. on Power Syst., Vol. 5, No. 2, pp. 715-725, May 1990.

[11] P.M. Anderson and A.A. Fouad, "Power System Control and Stability", The Iowa State University Press, 1977.

[12] B.N. Parlett, "The Symmetric Eigenvalue Problem", Prentice-Hall, Engelwood-Cliffs, 1980.

[13] D.J. Sobajic and Y.-H. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems", IEEE Trans. on Power Systems, Vol. 4, No. 1, pp. 220-224, Feb. 1989.

[14] E.H.P. Chan, "Application of Neural-Network Computing in Intelligent Alarm Processing ", IEEE Proc. of 1989 PICA, pp. 246-251, Seattle, WA, May 1989.

[15] M. Aggoune, M.A. El-Sharkawi, D.C. Park, M.J. Damborg, and R.J. Marks II, "Preliminary Results on Using Artificial Neural Networks for Security Assessment", IEEE Proc. of 1989 PICA, pp. 252-258, Seattle, WA, May 1989.

[16] H. Mori, H. Uematsu, S. Tsuzuki, T. Sakurai, Y. Kojima, K. Suzuki, "Identification of Harmonic Loads in Power Systems Using an Artificial Neural Networks", Proc. of Second Symposium on Expert Systems Application to Power Systems", pp. 371-378, Seattle, WA, July 1989.

[17] H. Tanaka, S. Matsuda, H. Ogi, Y. Izui, H. Taoka, and T. Sakaguchi., "Design and Evaluation of Neural Network for Fault Diagnosis", Proc. of Second Symposium on Expert Systems Application to Power Systems", pp. 378-384, Seattle, WA, July 1989.

[18] H. Mori and S. Tsuzuki, "Power System Topological Observability Analysis Using a Neural Network Model", Proc. of Second Symposium on Expert Systems Application to Power Systems", pp. 385-391, Seattle, WA, July 1989.

[19] N. Iwan Santoso and O.T. Tan, "Neural-Net Based Real-Time Control of Capacitors Installed on Distribution Systems", IEEE Trans. on Power Delivery, Vol. 5, No. 1, pp. 266-272, Jan. 1990.

[20] R.K. Hartana and G.G. Richards, "Harmonic Source Monitoring and Identification Using Neural Networks", IEEE PES 1990 Winter Meeting, Paper No. 90 WM 238-6 PWRS, Atlanta, GA, Feb. 1990.

[21] S. Ebron, D. Lubkeman and M. White, "A Neural Network Approach to the Detection of Incipient Faults on Power Distribution Feeders", IEEE Trans. on Power Delivery, Vol. 5, No. 2, pp. 905-914, April 1990.

[22] H. Mori and S. Tsuzuki, "Comparison between Backpropagation and Revised GMDH Techniques for Predicting Voltage Harmonics", IEEE Proc. of 1990 ISCAS, pp. 1102-1105, New Orleans, LA, May 1990.

[23] H. Mori and S. Tsuzuki, "Determination of Power System Topological Observability Using the Boltzmann Machine", IEEE Proc. of 1990 ISCAS, pp. 2938-2941, New Orleans, LA, May 1990.

[24] D.C. Park, M.A. El-Sharkawi, R.J. Marks, M.E. Aggoune, L.E. Atlas and M.J. Damborg, "Electric Load Forecasting Using an Artificial Neural Network", IEEE PES 1990 Summer Meeting, Paper No. 90 SM 377-2 PWRS, Minneapolis, MN, July 1990.

[25] C.-R. Chen, and Y.-Y. Hsu, "Synchronous Machine Steady-State Stability Analysis Using an an Artificial Neural Network", IEEE PES 1990 Summer Meeting, Paper No. 90 SM 430-9 EC, Minneapolis, MN, July 1990.

[26] H. Mori, K. Itou, H. Uematsu, and S. Tsuzuki, "An Artificial Neural Net Based Method for Predicting Power System Voltage Harmonics", IEEE PES 1990 International Power Meeting, Paper No. 90 IC 583-5 PWRD, New Delhi, India, Oct. 1990.

[27] D.O. Hebb, "The Organization of Behavior", Wiley, 1949.

[28] C. von der Marlsburg, "Self-organization of Orientation Sensitive Cells in the Striate Convex", Kybernetik, Vol. 14, pp. 85-100, 1973.

[29] D.E. Rumelhart and J.L. McClland (Eds.), "Parallel Distributed Processing", MIT Press, 1986.

[30] T. Kohonen, "Self-Organization and Associative Memory", Springer Verlag, 1988.

BIOGRAPHIES

Hiroyuki Mori (S'82, M'85) was born in Tokyo, Japan on November 10, 1954. He received the B.S., M.S. and Ph.D. degrees all in electrical engineering from Waseda University, Tokyo, Japan in 1979, 1981, and 1985, respectively. From 1984 to 1985 he was a Research Assistant at Waseda University. In 1985 he joined the faculty in Electrical Engineering at Meiji University, Kawasaki, Japan. He is currently an Associate Professor. His research interests include harmonics, voltage instability, load flows, state estimation, system identification and neural network applications. Dr. Mori is a member of ACM, AAAI, INNS, and IEE of Japan.

Yoshihito Tamaru (Non-member) was born in Asahikawa, Hokkaido, Japan on October 8, 1965. He received the B.S. and M.S. degrees in electrical engineering from Meiji University, Kawasaki, Japan in 1989 and 1991, respectively. In 1991 he joined TEPCO (Tokyo Electric Power Co.), Tokyo, Japan. His research interests are in the areas of neural network applications. Mr. Tamaru is a member of IEE of Japan.

Senji Tsuzuki (M'89) was born in Japan on January 3, 1930. He received the B.S., M.S., and Ph.D. degrees all in electrical engineering from Waseda University, Tokyo, Japan in 1954, 1956, and 1968, respectively. From 1956 to 1977, he worked for CRIEPI (Central Research Institute of Electric Power Industry), Tokyo, Japan, where he was engaged in research and development of advanced power system engineering applications on automatic generation control and security control. Since 1977 he has been a Professor of Department of Electrical Engineering at Meiji University. His areas of interest are harmonics, automatic frequency control, economic load dispatching, and security control problems. Dr. Tsuzuki is a member of IEE of Japan.