

Analysis of electromechanical modes using an artificial neural network

Y.-Y. Hsu
C.-R. Chen
C.-C. Su

Indexing terms: Damping, Electromechanical nodes, Neural networks, Power-system oscillations

Abstract: An approach based on artificial neural networks is proposed for the analysis of electromechanical modes in Taiwan power system. To evaluate the dynamic performance of a power system in system operation and planning, the dominant eigenvalues for the worst-damped electromechanical mode must be computed. A multilayer feedforward artificial neural network is developed. It is well known that eigenvalues are complicated functions of many system variables such as bus loads, line flows, generation schedule, bus voltages etc. An important procedure in neural network design is to select those features which most affect system eigenvalues. A clustering artificial neural network is thus designed for feature selection. To demonstrate the effectiveness of the approach, results from eigenvalue analyses of Taiwan power system are reported.

List of symbols

V_i = voltage magnitude for bus i
 δ_i = voltage angle for bus i
 P_{Gi} = real-power generation of bus i
 Q_{Gi} = reactive-power generation of bus i
 P_{Di} = real power load demand of bus i
 Q_{Di} = reactive-power load demand of bus i
 P_i = injected real power of bus i ($= P_{Gi} - P_{Di}$)
 Q_i = injected reactive power of bus i ($= Q_{Gi} - Q_{Di}$)
 $P_{Li,j}$ = real-power flow over the line from bus i to bus j
 $Q_{Li,j}$ = reactive-power flow over the line from bus i to bus j

1 Introduction

Analysis of electromechanical modes is important for a longitudinal power system such as Taiwan power system which had experienced low frequency oscillations [1]. In long-term system planning, it is essential to install adequate new generating units and transmission lines such

that, in addition to other important requirements, the system will have enough damping of the low frequency oscillations. In system operation, the generating units must be properly dispatched so that all system modes (especially the electromechanical modes in our system) are effectively damped.

To evaluate the damping characteristics of the oscillation modes, both time-domain simulations and frequency-domain eigenvalue analyses have been widely employed [2]. In both approaches, numerous system variables such as bus loads, bus voltages, generation patterns etc. which affect system-damping characteristics must be considered, making these approaches very time consuming for a practical power system. Since eigenvalue analysis must be repeated when there is a change in bus voltages or generation schedules, it is essential, especially for system operators, to devise an approach which can be employed to perform eigenvalue analysis very efficiently.

In this paper, artificial neural networks (ANN) [3-9] are developed for eigenvalue analysis of Taiwan power system. Artificial neural networks are made up of some neurons (nodes) connected together via links. Information is stored as connection weights of these links. By distributing the knowledge over the neurons and conducting parallel processing on information, artificial neural networks are expected to be capable of solving complicated problems in a very efficient manner.

In the present paper, the multilayer feedforward ANN [7] is employed to evaluate system eigenvalues. The output of the ANN is the real part of the eigenvalues for the worst-damped electromechanical modes, as this is a good measure of system stability. Since the eigenvalues are complicated functions of numerous system variables such as bus loads, bus voltages, line flows, generation patterns etc., it seems natural to use these system variables as the inputs to the ANN so that the neural network will yield an accurate stability measure. However, it would be very difficult to train the neural network when this great number of system variables was directly applied to the neural network as the number of input neurons and connection weights is proportional to the number of input variables. It is thus essential to reduce the number of inputs to the ANN. The principle is to retain only one key variable in a group of variables which have similar characteristics. The process of selecting these key variables is usually referred as feature selec-

© IEE, 1994

Paper 9872C (P9), first received 22nd April and in revised form 13th September 1993

Y.-Y. Hsu is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

C.-R. Chen is with the Department of Electrical Engineering, Taipei Institute of Technology, Taipei, Taiwan, Republic of China

C.-C. Su is with the Department of Electrical Engineering, Chung Cheng Institute of Technology, Tao-Yuan, Taiwan, Republic of China

Financial support was given to this work by the National Science Council, Republic of China, under contract NSC 80-0404-E002-07.

tion. A clustering ANN [8–10] is developed in this paper to select the features. Using this clustering ANN for feature selection, the inputs to the multilayer feedforward ANN are reduced from 542 to 14.

To demonstrate the effectiveness of the proposed neural-network approach, eigenvalue analyses of Taiwan power system, which consists of 180 buses, 30 generating units and 296 lines, are performed. It is concluded from the computational results that accurate stability measures can be derived by the ANN in a very efficient manner. The average percentage error is 2.22% for the patterns outside the training set.

2 Study system

Fig. 1 is a single-line diagram of Taiwan power system. There are 180 buses, 296 transmission lines and 30 generators in the study system. The generators are geographically located at three different areas on the island of Taiwan with a distance of 400 km from the north to the south. The three areas are the norther (N), the central (C),

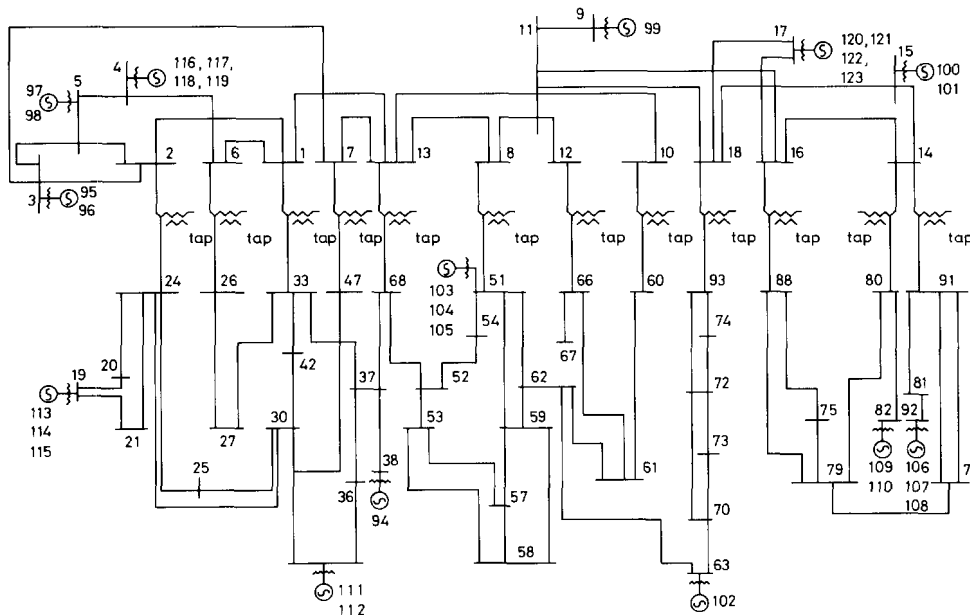


Fig. 1 Single-line diagram of Taiwan power system

and the southern (S). Heavy interarea line flows may cause low-frequency oscillations if there is insufficient damping [1]. It is thus desirable to evaluate small-signal stability measures in system operation. Table 1 lists the buses in the study system. Note that nuclear units and some hydro-electric units are dispatched as base load units while thermal units are used for load-following.

Table 1: Summary of system buses

| Bus type | Bus number |
|---------------------------------|------------|
| Swing bus | 105 |
| PV buses (base-load units) | 94–104 |
| PV buses (load-following units) | 106–123 |
| PQ buses (345 kV, north area) | 1–7 |
| PQ buses (345 kV, centre area) | 8–13 |
| PQ buses (345 kV, south area) | 14–18 |
| PQ buses (161 kV) | 19–93 |
| PQ buses (69 kV) | 124–180 |

For the purpose of eigenvalue analysis, we need the system variables as listed in Table 2. Note that other variables such as voltage-angle separation between two buses can also be included in Table 2. However, these variables are not considered in the present work because

Table 2: System variables required for eigenvalue analysis

| Categories | Descriptions |
|---------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 1 Bus-voltage magnitudes and angles (329 variables) | V_i ($i = 1-93, 124-180$) δ_i ($i = 1-104, 106-180$) |
| 2 Injected real power and reactive power (163 variables) | P_i ($i = 105-180$) Q_i ($i = 94-180$) |
| 3 Real and reactive power flows over 345 kV lines (46 variables) | P_L, Q_L |
| 4 Total real and reactive power generations (four variables) | $\sum_{i=1}^{180} P_{G_i}$ $\sum_{i=1}^{180} Q_{G_i}$ $\sum_{i=1}^{180} P_{D_i}$ $\sum_{i=1}^{180} Q_{D_i}$ |

they are functions (e.g. difference or sum of two variables) of the variables in Table 2 and these functions can be accomplished by the neurons in the hidden and output layers of the neural network.

Note that the total number of system variables is 542, system variables being denoted by S_j ($j = 1, 2, \dots, 542$). Certain constant parameters such as generator reactances, time constants, exciter constants and line reactances do not change with system-loading conditions. They are therefore not selected as potential candidates for neural-network inputs. In fact, these constant parameters can be dealt with using the bias terms of neurons in neural-network implementations. However, in conventional time-domain and frequency-domain approaches, both these constant parameters and the system variables are required for digital simulations or eigenvalue computations.

In Section 3, an artificial-neural-network approach

will be introduced to evaluate system-stability measures without the laborious task of digital simulation or eigenvalue computations.

3 Artificial-neural-network approach

As shown in Fig. 2, the design of ANN involves four major steps: training-set creation, feature selection, training and testing.

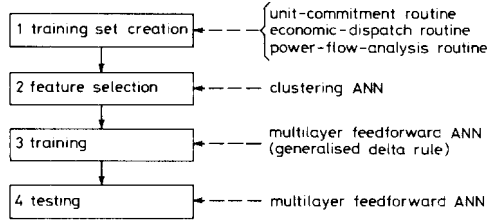


Fig. 2 Neural-network design procedures

The first step in neural-network design is to create the training patterns in the training set. To do this, the historical operating records of Taiwan power system are compiled. The collected data sets consist of bus loads, bus voltages, line flows and generation patterns which characterise system operating conditions. The compiled operating point is referred to as the base operating point. To describe the system more adequately, we need other operating points in addition to these compiled base operating points. This is achieved by randomly varying each bus load in the base operating point over a wide range of $\pm 20\%$. Some derived operating points can be obtained by executing unit-commitment, economic-dispatch and power-flow-analysis routines for each perturbed loading condition. We use this method to generate 200 operating points, among which 120 points are randomly chosen as the training patterns. The remaining 80 points will be used as testing patterns. Note that the generation of these patterns is achieved by limited variation of the base operating point. Therefore, these patterns are somewhat related. Each operating point (including base operating points and derived operating points) is described by a

$$\mathbf{X}' = \begin{bmatrix} x'_{11} & x'_{12} & \cdots & x'_{1N} \\ x'_{21} & x'_{22} & \cdots & x'_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{M1} & x'_{M2} & \cdots & x'_{MN} \end{bmatrix} \begin{array}{l} \rightarrow \text{operating point 1} \\ \rightarrow \text{operating point 2} \\ \vdots \\ \rightarrow \text{operating point } M \end{array}$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \text{variable 1} & \text{variable 2} & \text{variable } N \\ (S_1) & (S_2) & (S_N) \end{array} \quad (3)$$

state vector \mathbf{X}' which comprises the 542 system variables as listed in Table 2. These 542 variables S_j ($j = 1, 2, \dots, 542$) form the set of potential candidates for the inputs to the multilayer feedforward ANN. For the convenience of later discussions, we use the row vector $\mathbf{X}'_i = [x'_{i1}, x'_{i2}, \dots, x'_{i,542}]$ to represent the state vector for the i th operating point. Note that x'_{ij} is the value for system variable S_j under i th operating condition.

To evaluate the damping characteristic for each operating point, the eigenvalues for the electromechanical modes are computed using the AESOPS program [2]. The real part of the eigenvalues for the worst-damped modes σ is a good measure of small-signal stability. Therefore, we select σ as the output of the multilayer

feedforward ANN. As a result, each training pattern t'_i in the training set for the multilayer feedforward ANN is described by the input (\mathbf{X}'_i)-output (σ_i) pair as follows

$$t'_i = (\mathbf{X}'_i, \sigma_i) \quad (1)$$

where σ_i is the real part of the eigenvalues for the worst-damped modes under the operating point described by \mathbf{X}'_i .

If this set of training patterns is directly applied to train the multilayer feedforward ANN using the generalised delta rule [6], one may face the problem of local minima and slow convergence. This is caused by the huge number of neurons and unknown connection weights which are proportional to the number of inputs (542 in the present case). Thus it is desirable to reduce the number of inputs to the ANN so that the resultant state vector $\mathbf{X}_i = [x_{i1}, x_{i2}, \dots, x_{iG}]$ will include only those G system variables (S_1, S_2, \dots, S_G) which are essential for stability-measure evaluation by the ANN. With these reduced state vectors \mathbf{X}_i and training patterns

$$t_i = (\mathbf{X}_i, \sigma) \quad (2)$$

at hand, it is expected that the multilayer feedforward ANN can be efficiently trained by using the generalised delta rule in the training phase. Finally, in the testing phase, the trained multilayer feedforward ANN must yield accurate stability measures for operating points both in and outside the training set. In Section 4, a clustering ANN will be presented to reduce system variables and to select key features for stability evaluation.

4 Clustering ANN for feature selection

The basic principle for a clustering ANN [8] is to group the total N ($N = 542$) system variables (S_1, S_2, \dots, S_N) into G clusters such that the variables in a cluster have similar characteristics. We then pick out one representative variable in a cluster as the feature for this cluster. This will reduce the number of system variables from N to G . For the purpose of feature selection using the clustering ANN, $M = 120$ training patterns ($t'_i = (\mathbf{X}'_i, \sigma_i)$, $i = 1, 2, \dots, 120$) are created.

With the state vectors $\mathbf{X}'_i = [x'_{i1}, x'_{i2}, \dots, x'_{iN}]$ ($i = 1, 2, \dots, M$) at hand, form the matrix \mathbf{X}'

It is observed that row i of matrix \mathbf{X}' contains the values for N ($= 542$) system variables (S_1, S_2, \dots, S_N) at operating point i while column j of matrix \mathbf{X}' consists of the S_j variables in the M ($= 120$) training patterns. Define the column vector $\mathbf{Y}_j = [x'_{1j}, x'_{2j}, \dots, x'_{Mj}]^T = [y_{1j}, y_{2j}, \dots, y_{Mj}]^T$. Then the 542 system variables S_j ($j = 1, 2, \dots, 542$) can be clustered based on these vectors \mathbf{Y}_j . Those system variables with similar vectors \mathbf{Y}_j will be grouped in a cluster. As shown in Fig. 3, two vectors \mathbf{Y}_1 and \mathbf{Y}_2 which are similar to each other will have a small angle between them. We can therefore put a group of these similar vectors in a cluster. To do this, define the cosine value of the angle θ_{jk} between two vectors \mathbf{Y}_j and \mathbf{Y}_k as

$$\cos \theta_{jk} = (\mathbf{Y}_j \mathbf{Y}_k) / (|\mathbf{Y}_j| |\mathbf{Y}_k|) \quad (4)$$

This cosine value can be used to evaluate the degree of similarity between two vectors. If $\cos(\theta_{jk})$ is greater than a specified threshold ρ_t , the two vectors Y_j and Y_k are

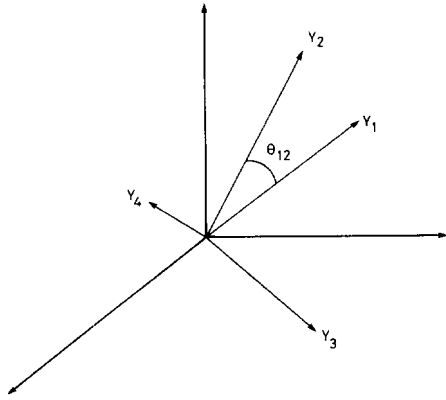


Fig. 3 Vectors Y shown in 3-dimensional space

regarded as two similar vectors and are put in the same cluster. Details of the clustering algorithm are described as follows.

Step 1: Let system variable S_1 belong to cluster 1 and let the cluster vector for cluster 1, $C_1 = [c_{11}, c_{21}, \dots, c_{M1}]^T$, be equal to column vector Y_1 , i.e.

$$c_{i1} = y_{i1} \quad i = 1, 2, \dots, M \quad (5)$$

Set initial count $j = 0$, $G = 1$.

Step 2: Increase j by one.

Step 3: Compute the cosine value D_g between vectors Y_j and C_g ($g = 1, 2, \dots, G$).

$$D_g = \left(\sum_{i=1}^M y_{ij} c_{ig} \right) / \left(\sum_{i=1}^M y_{ij}^2 \sum_{i=1}^M c_{ig}^2 \right)^{1/2} \quad g = 1, 2, \dots, G \quad (6)$$

Let

$$D_{gm} = \max_g (D_g) \quad (7)$$

If $D_{gm} < \rho_t$ (a specified threshold), the present vector Y_j is far from any existing cluster vectors and we have to create a new cluster for system variable S_j . Go to step 5. If $D_{gm} > \rho_t$, the present vector Y_j is close to cluster vector C_{gm} of cluster g_m and system variable S_j should be assigned to cluster g_m . Proceed to step 4.

Step 4: If vector Y_j has not been presented before, let system variable S_j be grouped in cluster g_m and update the cluster vector $C_{gm} = [c_{1gm}, c_{2gm}, \dots, c_{Mgm}]^T$ as follows

$$c_{igm} = y_{ij} + c_{igm} k_{gm} \quad i = 1, 2, \dots, M \quad (8)$$

where k_{gm} is the number of system variables in cluster g_m . Go to step 6. If vector Y_j has been presented before and variable S_j has been grouped in cluster g_m go to step 6.

If vector Y_j has been presented before and variable S_j has been grouped in a different cluster $g_{m'}$, move system variable S_j to cluster g_m and execute the update formulae

$$c_{igm} = y_{ij} + c_{igm} k_{gm} \quad i = 1, 2, \dots, M \quad (9)$$

$$c_{igm'} = -y_{ij} + c_{igm'} k_{gm'} \quad i = 1, 2, \dots, M \quad (10)$$

where $k_{gm'}$ is the number of system variables in cluster $g_{m'}$. In this case, this a move in cluster elements. Go to step 6.

Step 5: Create a new cluster g_n for system variable S_j with the cluster vector $C_{gn} = [c_{1gn}, c_{2gn}, \dots, c_{Mgn}]^T$ where

$$c_{ign} = y_{ij} \quad i = 1, 2, \dots, M \quad (11)$$

Increase G by one and go to step 6.

Step 6: If $j = N$, proceed to step 7; otherwise, go to step 2.

Step 7: If there is any more in cluster elements in the preceding N iterations, reset j to zero and proceed to step 2. Otherwise, go to step 8.

Step 8: For each cluster g , find a system variable S_g whose column vector Y_g is closest to the cluster vector C_g , i.e.

$$\left(\sum_{i=1}^M y_{ig} c_{ig} \right) / \left(\sum_{i=1}^M y_{ig}^2 \sum_{i=1}^M c_{ig}^2 \right)^{1/2} \geq \left(\sum_{i=1}^M y_{ij} c_{ig} \right) / \left(\sum_{i=1}^M y_{ij}^2 \sum_{i=1}^M c_{ig}^2 \right)^{1/2} \quad (12)$$

for any S_j in cluster g . Let system variable S_g be the feature variable for cluster g . Print out the system variables and feature variable for each cluster and stop.

5 Multilayer feedforward ANN for eigenvalue analysis

Using the clustering ANN as described in Section 4, the 542 system variables can be grouped into G clusters. By selecting one representative feature S_g for each cluster g , a total of G ($G = 14$ for our study system) features can be derived. These G feature variables S_g ($g = 1, 2, \dots, 14$) will be employed as the inputs to the multilayer feedforward ANN as shown in Fig. 4 [6] to conduct eigenvalue analysis.

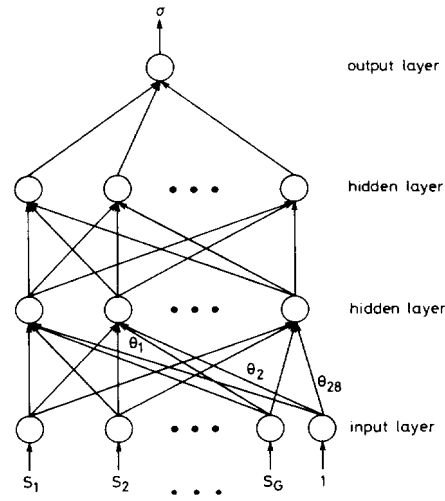


Fig. 4 Multilayer feedforward neural network

The nodes in the input layer receive input signals from the outside world and directly pass the signals to the nodes in the next layer. The node in the output layer provides the desired real part of the eigenvalues for the worst-damped modes (σ). In addition to the input layer and the output layer, we need one or more hidden layers. The nodes in the hidden layers take signals from the input layer and send their outputs to the nodes in the next layer when computations within the nodes have

been completed. In this work, two hidden layers with 28 hidden units at each layer are used. It is noted that we need at least two hidden layers to cover all possible kinds of training pattern in the training set [7]. As for the number of hidden units per layer, an empirical figure of twice the number of inputs has been recommended [3]. In the present work, a neural network of 28 hidden units has been found to yield satisfactory results.

Before the neural network can be used to yield the desired eigenvalue real part, the connection weights must be determined. The process of determining the connection weights is usually referred to as the training process.

In the training process, 120 different operating conditions are considered covering a system load from 6200 MW to 9200 MW for the study system. The test patterns are also selected from the same range since the system load is within the range for the study period. However, the ANN can be applied to generate the solution for a pattern outside the range. This solution speed is similar, but the accuracy may be not as good when the system load is remote from the specified range. For each operating condition p , the values for the G features S_g ($g = 1, 2, \dots, G$), i.e., x_{p1}, x_{p2}, \dots , and x_{pG} are first recorded and the real part of the eigenvalues for the worst-damped modes σ_p is computed by using the AESOPS program. Thus the training pattern p is described by

$$t_p = \{(\text{input } p), (\text{output } p)\} \\ = \{(x_{p1}, x_{p2}, \dots, x_{pG}), (\sigma_p)\} \quad (13)$$

where x_{pg} is the value of the system variable S_g for training pattern p .

A commonly used approach to training the multilayer feedforward ANN is the generalised delta rule [6]. In which the connection weights are updated in order to minimise the sum-of-squares errors between the desired output and the actual output. Details of the update formula can be found in Reference 6.

6 Application to eigenvalue analysis of Taiwan power system

Consider the power system as shown in Fig. 1. To select the features from the 542 variables, the clustering ANN is executed using the 120 training patterns. Three different values are considered for the threshold, i.e. $\rho_t = 0.85$, $\rho_t = 0.9$ and $\rho_t = 0.95$. The results from the clustering ANN are given in Table 3.

Table 3: Results from the clustering ANN

| ρ_t | Number of cluster | Feature variable (number of system variable) for each cluster |
|----------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.85 | 10 | $\delta_{14}(15), \delta_{42}(98), P_{112}(4), P_{113}(349), Q_{L3,5}(3), Q_{L3,7}(33), Q_{L11,12}(2), P_{L11,16}(10), P_{L16,17}(27), Q_{L17,18}(1)$ |
| 0.90 | 14 | $\delta_{14}(15), \delta_{25}(118), \delta_{52}(17), P_{113}(328), Q_{115}(42), Q_{122}(4), P_{L2,3}(2), Q_{L11,12}(1), P_{L11,16}(10), Q_{L14,15}(1), P_{L14,16}(1), Q_{L14,16}(1), Q_{L15,16}(1), Q_{L17,18}(1)$ |
| 0.95 | 28 | $\delta_{19}(5), \delta_{19}(90), \delta_{28}(2), \delta_{37}(2), \delta_{52}(11), \delta_{80}(40), \delta_{107}(2), \delta_{112}(3), \delta_{123}(6), \delta_{141}(6), P_{108}(3), Q_{108}(13), P_{111}(2), P_{113}(308), P_{122}(6), Q_{122}(4), P_{L2,3}(1), Q_{L3,5}(1), Q_{L3,7}(26), Q_{L4,5}(1), Q_{L11,12}(1), P_{L11,16}(2), Q_{L14,15}(1), P_{L14,16}(1), Q_{L14,16}(1), Q_{L15,16}(1), Q_{L16,17}(2), Q_{L17,18}(1)$ |

It is observed that the number of resultant clusters increases as the threshold ρ_t is increased. This is as

expected since a greater threshold in step 3 of the clustering algorithm will tend to create more new clusters. A proper value of ρ_t must be selected such that the resultant features will yield fewest errors in the testing phase when they are applied as inputs to the multilayer feedforward ANN to evaluate system eigenvalues. The issue will be addressed below.

To give a clear picture on how the system variables are clustered in different groups, Table 4 lists the system variables in each cluster for $\rho_t = 0.9$.

Table 4: System variables in each cluster ($\rho_t = 0.90$)

| Cluster | Feature | System variables |
|---------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | P_{113} | $V_i, i = 1-93, 124-180$ $\delta_i, i = 51, 54-67, 69, 85, 95-99, 103, 104, 149-159, 162, 164, 173-175$ $P_i, i = 105, 108-110, 113-115, 124-180$ $Q_i, i = 99, 124-180$ $P_{L1,2}, P_{L1,6}, P_{L2,5}, P_{L3,5}, P_{L3,7}, P_{L4,6}, P_{L9,11}, P_{L10,11}, P_{L11,12}, P_{L14,15}, P_{L15,18}, Q_{L9,11}, Q_{L10,11}, \Sigma P_G, \Sigma P_L, \Sigma Q_L$ |
| 2 | δ_{14} | $\delta_i, i = 14, 16-18, 36, 38, 106, 107, 125, 132, 140, 141, 145, 148$ |
| 3 | δ_{25} | $Q_{L11,13}$ $\delta_i, i = 1-13, 19-35, 40-50, 70-84, 86-94, 102, 108-119, 124, 126-131, 133-135, 138, 142, 144, 146-147, 160-163, 165-172, 176-180$ $P_{L1,13}, P_{L7,13}, P_{L8,11}, P_{L8,13}, P_{L11,13}, P_{L8,11}, Q_{L8,13}, Q_{L11,16}, Q_{L11,18}$ |
| 4 | δ_{52} | $\delta_i, i = 39, 52, 53, 100, 101, 136, 137, 143$ $P_i, i = 106, 107, 120-123, Q_{L4,5}, P_{L16,17}, P_{L17,18}$ |
| 5 | Q_{115} | $P_i, i = 111, 112, 116-119,$ $Q_i, i = 94-98, 100-119,$ $P_{L4,5}, Q_{L1,2}, Q_{L1,6}, Q_{L1,13}, Q_{L2,3}, Q_{L2,5}, Q_{L3,7}, Q_{L4,6}, Q_{L7,13}, Q_{L16,17}, \Sigma Q_G$ |
| 6 | Q_{122} | $Q_i, i = 120-123$ |
| 7 | $P_{L2,3}$ | $P_{L2,3}, Q_{L3,5}$ |
| 8 | $P_{L11,16}$ | $\delta_i, i = 15, 37, 68, 120-123, 139$ $P_{L11,16}, P_{L11,18}$ |
| 9 | $P_{L14,16}$ | $P_{L14,16}$ |
| 10 | $Q_{L11,12}$ | $Q_{L11,12}$ |
| 11 | $Q_{L14,15}$ | $Q_{L14,15}$ |
| 12 | $Q_{L14,16}$ | $Q_{L14,16}$ |
| 13 | $Q_{L15,16}$ | $Q_{L15,16}$ |
| 14 | $Q_{L17,18}$ | $Q_{L17,18}$ |

To examine the physical significance of the clustering results, Table 1 lists the 345 kV buses in the three areas of Taiwan. Since the excess power in the northern area or the southern area must be transmitted to other areas through the 345 kV trunk lines, the 345 kV buses in one area can be regarded as belonging to the same coherent group. In addition, the bus-voltage angles are important indicators for system stability. Therefore, it is expected that the bus-voltage angles in one area are grouped in the cluster. This observation is confirmed by the clustering results in Table 4. The voltage angles in the north ($\delta_1, \delta_2, \dots, \delta_7$) are grouped in cluster 3. The voltage angles in the centre ($\delta_8, \delta_9, \dots, \delta_{13}$) are also grouped in cluster 3 while the voltage angles in the south ($\delta_{14}, \delta_{16}, \delta_{17}, \delta_{18}$) are grouped in cluster 2. Although there is a minor exception (δ_{15}), the correlation between the clustering results and the coherent groups can generally be established. Note that the variances of voltage magnitudes are very small (within 0.001). Therefore the variables for the voltage magnitudes are in the same cluster in Table 4. On the other hand, the variances of phase angles, real powers

and reactive powers are about 0.45, 1.3 and 0.2, respectively, these are greater than those for the voltage magnitudes. These system variables therefore belong to different clusters (the variances of eigenvalues are within 0.000 24).

After the features have been selected, they are used as the inputs to the multilayer feedforward ANN which are trained using the 120 training patterns in the training set. The learning rate η and momentum constant α used in this work are 0.5 and 0.8, respectively. The resultant ANN is tested using the 120 training patterns in the training set and 80 other patterns outside the training set. To examine neural-network training and testing results, define:

Average percentage error for the training patterns in the training set

$$E_{AVE1} = \left(\frac{1}{120} \sum_{n=1}^{120} \left| \frac{t_n - o_n}{t_n} \right| \right) \times 100\% \quad (14)$$

where t_n and o_n are the desired and computed output (σ) of the ANN, respectively, when pattern n in the training set is applied.

Maximum percentage error for the training patterns in the training set

$$E_{MAX1} = \max_{n=1}^{120} \left| \frac{t_n - o_n}{t_n} \right| \times 100\% \quad (15)$$

Average percentage error for the patterns outside the training set

$$E_{AVE2} = \left(\frac{1}{80} \sum_{n=1}^{80} \left| \frac{t_n - o_n}{t_n} \right| \right) \times 100\% \quad (16)$$

where t_n and o_n are the desired and computed output (σ) of the ANN, respectively, when pattern n outside the training set is applied.

Maximum percentage error for the patterns outside the training set

$$E_{MAX2} = \max_{n=1}^{80} \left| \frac{t_n - o_n}{t_n} \right| \times 100\% \quad (17)$$

The average percentage error E_{AVE1} and maximum percentage error E_{MAX1} are depicted in Figs. 5-7 as functions of the number of iterations in the training process

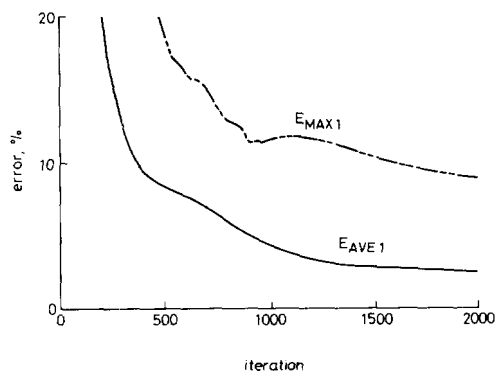


Fig. 5 Average percentage error E_{AVE1} and maximum percentage error E_{MAX1} in the training process ($\rho_t = 0.85$)

for $\rho_t = 0.85$, $\rho_t = 0.9$ and $\rho_t = 0.95$, respectively. Note that, in one iteration, the 120 training patterns in the training set are all presented once. The training times for 2000 iterations on a Sun IV-60 Workstation and the

errors from testing the resultant ANN are compared in Table 5.

Table 5: Comparison of the training for 2000 iterations and percentage errors for different threshold ρ_t

| Threshold ρ_t | Number of features (cluster) | Training time for 2000 iterations | Average error | | Maximum error | |
|--------------------|------------------------------|-----------------------------------|---------------|------------|---------------|------------|
| | | | E_{AVE1} | E_{AVE2} | E_{MAX1} | E_{MAX2} |
| 0.85 | 10 | 56.3 | 2.43 | 2.87 | 8.91 | 13 |
| 0.9 | 14 | 60 | 1.51 | 2.22 | 5.29 | 7.69 |
| 0.95 | 28 | 73.6 | 0.99 | 2.96 | 5.46 | 10.8 |

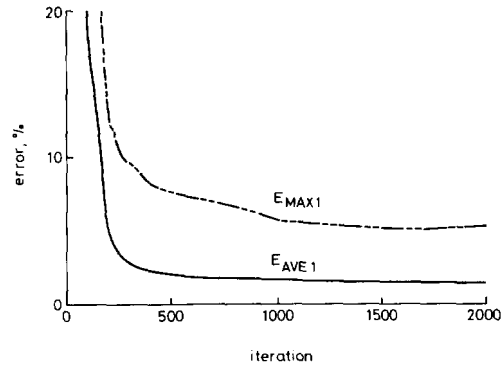


Fig. 6 Average percentage error E_{AVE1} and maximum percentage error E_{MAX1} in the training process ($\rho_t = 0.9$)

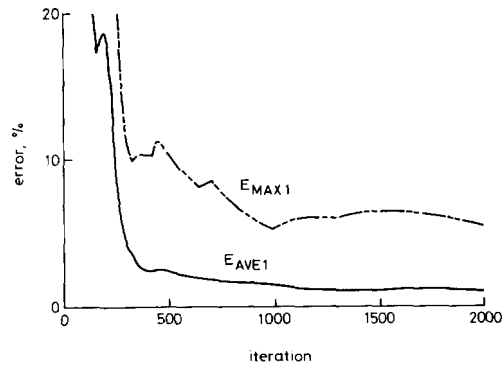


Fig. 7 Average percentage error E_{AVE1} and maximum percentage error E_{MAX1} in the training process ($\rho_t = 0.95$)

From the results in Figs. 5-7 and Table 5, the following observations are in order.

The number of connection weights in the ANN increases with increasing number of inputs (features). Therefore, the required training time will increase as the number of inputs to the ANN is increased.

Among the three thresholds used in the present work, a value of $\rho_t = 0.85$ will result in an ANN with greatest percentage errors. The average percentage error reaches 2.87% while the maximum percentage error reaches 13% for those patterns which are not in the training set. It seems that a large threshold will result in fewer features than are required fully to describe the small-signal stability of the study system.

If the threshold is increased to 0.9, the clustering ANN recommends that we use 14 inputs for the multilayer feedforward ANN. In this case, the average percentage

error E_{AVE1} and maximum percentage error E_{MAX1} for the patterns in the training set are 1.51% and 5.29%, respectively, while those for the patterns outside the training set are 2.22% and 7.69%, respectively. A considerable improvement in ANN performance over the case of $\rho_t = 0.85$ is achieved. In normal operation of Taiwan power system, the real parts of the electro-mechanical mode eigenvalues range from -0.1 to -0.2 . In this case, a error of less than 10% is acceptable from the viewpoint of stability assessment.

If the threshold is further increased to 0.95 and the number of features increased to 28, no significant improvements in ANN performance can be made. In fact, the maximum percentage errors for the present case are greater than those in the case of $\rho_t = 0.9$. A possible reason for the deterioration of performance in this case is that some minor variables are included as features of the system, causing interactions with major system features. A value of 0.9 seems to be a proper choice for the threshold in our study system. Note that the threshold value is system dependent and a proper value can only be achieved by a trial-and-error process.

Once the multilayer feedforward ANN has been trained, it can be employed to generate the desired eigenvalue real part very efficiently. The average CPU time required to obtain an answer on a Sun IV-60 Workstation is about 0.05 s. It is expected that the neural networks designed can provide on-line operational aids to system operators.

7 Conclusions

In this paper we propose to use artificial neural networks to perform eigenvalue analysis of Taiwan power system. A clustering ANN is first developed to select those features from system variables which can be used to characterise system stability. These features are then employed as the inputs to a multilayer feedforward ANN which is designed to evaluate system stability based on the input signals. The multilayer feedforward ANN has been trained using 120 training patterns in the training set and tested with other 80 patterns outside the training set. Results from eigenvalue analyses of Taiwan power system reveal that an average percentage error of 2.22% can be achieved for the patterns outside the training set.

In the present work, the training patterns used in the training phase and the test patterns used in the test phase cover a wide range of bus loads and generation profile. Since line switchings are rare events in normal operation of the Taiwan power system, the change of system topology due to such events is not considered in this paper. To take this factor into account, additional system variables which characterise system topology must be employed as the inputs to the neural networks.

For systems larger than Taiwan power system such as the US West Coast System, one may be interested in more than one mode. In this case, we need more than one output from the ANN. However, the techniques remains the same.

8 References

- 1 HSU, Y.Y., SHYUE, S.W., and SU, C.C.: 'Low frequency oscillations in longitudinal power systems: experience with dynamic stability of Taiwan power system', *IEEE Trans.*, 1987, **PWRS-2**, pp. 92-100
- 2 BYERLY, R.T., BENNON, R.J., and SHERMAN, D.E.: 'Eigenvalue analysis of synchronising power flow oscillations in large electric power systems', *IEEE Trans.*, 1982, **PAS-101**, pp. 235-243
- 3 SOBAJIC, D.J., and PAO, Y.H.: 'Artificial neural-net based dynamic security assessment for electric power systems', *IEEE Trans.*, 1989, **PWRS-4**, pp. 220-228
- 4 MORI, H.: 'An artificial neural-net based method for estimating power system dynamic stability index'. First international forum on Applications of Neural Networks to Power Systems, Seattle, USA, July 1991, pp. 127-133
- 5 MORI, H., and TAMURA, S.: 'An artificial neural net based technique for power system dynamic stability with the Kohonen Model', *IEEE Trans.*, 1992, **PWRS-7**, pp. 856-864
- 6 RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J.: 'Learning internal representations by error propagation', in RUMELHART, D.E., and McCLELLAND, J.L. (Eds.): 'Parallel distribution processing'. Vol. 1, Chap. 8
- 7 LIPPMANN, R.P.: 'An introduction to computing with neural nets', *IEEE ASSP Mag.*, 1987, pp. 4-22
- 8 PAO, Y.H.: 'Adaptive pattern recognition and neural networks' (Addison-Wesley Publishing Co., 1989), Appendix B
- 9 SOBAJIC, D.J., PAO, Y.H., and DOLCE, J.: 'Real-time security monitoring of electric power systems using parallel associative memories'. IEEE international symposium on Circuit and Systems, New Orleans, USA, May 1990, pp. 2929-2932
- 10 WEERASOORIYA, S., and EL-SHARKAWI, M.A.: 'Use of Karhunen-Loeve expansion in training neural network for static security assessment'. First international forum on Applications of Neural Networks to Power Systems, Seattle, USA, July 1991, pp. 59-64