Journal of Sound and Vibration (2002) **249**(5), 867–883
doi:10.1006/jsvi.2001.3915, available online at http://www.idealibrary.com on IDEAL®

SELECTION OF TRAINING SAMPLES FOR MODEL UPDATING USING NEURAL NETWORKS

C. C. CHANG, T. Y. P. CHANG AND Y. G. XU

Department of Civil Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, Peoples Republic of China

AND

W. M. TO

Department of Mechanical Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, Peoples Republic of China

(*Received* 9 *August* 2000, *and in final form* 11 *May* 2001)

One unique feature of neural networks is that they have to be trained to function. In developing an iterative neural network technique for model updating of structures, it has been shown that the number of training samples required increases exponentially as the number of parameters to be updated increases. Training the neural network using these samples becomes a time-consuming task. In this study, we investigate the use of orthogonal arrays for the sample selection. A comparison between this orthogonal arrays method and four other methods is illustrated by two numerical examples. One is the update of the felxural rigidities of a simply supported beam and the other is the update of the material properties and the boundary conditions of a circular plate. The results indicate that the orthogonal arrays method can significantly reduce the number of training samples without affecting too much the accuracy of the neural network prediction.

1. INTRODUCTION

With the development of the artificial intelligence techniques, the neural network (NN) methods have been widely accepted in the structural engineering area recently. Vanluchene and Sun [1] discussed the use of an NN model with a back-propagation (BP) learning algorithm in structural engineering applications including pattern recognition, concrete beam designs and rectangular plate analyses. Masri *et al.* [2] demonstrated that the NN model is a powerful tool for the identification of systems typically encountered in the structural dynamics field. The behaviors of concrete in the state of plane stress under monotonic biaxial loading and compressive uniaxial cycle loading are modelled by Ghaboussi *et al.* [3] using a BP NN model. The use of frequency domain data and NN's to update structural parameters has been studied by, among others, Atalla and Inman [4] and Levin and Lieven [5]. Stephens and VanLuchene [6] used NN to assess the safety condition of civil engineering structures following strong-motion earthquakes. Elkordy *et al.* [7] proposed a diagnostic system that utilized NN's for identifying the damage associated with changes in structural signatures. A modified counterpropagation NN was used to develop the inverse mapping between a vector of the stiffness of individual structural elements and the vector of the global static displacements under a testing load [8]. Rhim and Lee [9] examined the feasibility of using NN in conjunction with system identification techniques to detect the existence and to identify the characteristics of damage in composite structures. An application of NN in the damage detection of steel bridge structures was studied by Pandey and Barai [10]. An NN-based approach was presented for the detection of changes in the characteristics of structure-unknown systems [11]. Zhao *et al.* [12] proposed a counterpropagation NN to locate structural damage for a beam, a frame and support movements of a beam in its axial direction. Wu *et al.* [13] used the self-organization and learning capabilities of NN's in structural damage detection. In a comprehensive literature review on damage identification and health monitoring of structural and mechanical systems, Doebling *et al.* [14] also summarized the NN-based damage detection techniques developed up till 1996.

In an analogy to the brain, NN models are made up of interconnected processing elements called neurons which respond in parallel to a set of input signals given to each. An NN model consists of three main parts: (1) neurons; (2) weighted interconnections between neurons; and (3) activation functions that act on the set of input signals at neurons to produce output signals. Training is essential to most of these NN models. Training of an NN model is referred to as the determination of weights in the model using some training algorithms. Training samples are usually needed for training an NN model. In fact, whether an NN model after training can approximate the functional mapping between the inputs and outputs depends very much on the training samples. However, unlike the training algorithms that have been addressed in many literatures, the selection of training samples is less considered.

In general, the training samples should cover all possible combinations and ranges of input variation in order to ensure that the NN model trained using these samples can accurately represent the behavior of the system to be simulated. Assuming that there are k input parameters and each parameter has s possible variations, then the total number of training samples required to guarantee the sample completeness using the full factorial design method [15] would be s^k. A large amount of training samples inevitably require intensive computational efforts and slow down the convergent speed of the training process.

Using NN methods for model updating and damage detection, Atalla and Inman [4] suggested that a random generation (within the assumed boundaries of variation) of the model parameters yields the best results. The training samples should reflect the probability distribution of the model parameters being updated if this information is known. Levin and Lieven [5] proposed a two-part scheme for the selection of training data. The first part consisted of setting each parameter in turn to one of the s possible values while setting all the other parameters to their respective design values. The second part contained a given number of training samples resulting from adjusting a random selection of the parameters by a random amount. While applying NN models for structural optimization analyses, Rogers [16] investigated four sample selection methods that include the hypercube method, the Programming System for Structural Synthesis (PROSSS) method, the random method and the linear method. He concluded that the hypercube method appeared to give the best approximation of the design space especially when it was used twice.

In this study, we investigate the use of orthogonal arrays (OA) as an alternative for sample selection. A comparison between this OA method and four other methods is illustrated by model updating of a simply supported beam and a circular plate.

2. AN ITERATIVE NEURAL NETWORK FOR MODEL UPDATING AND DAMAGE DETECTION

Chang *et al.* [17, 18] proposed an iterative NN process for the model updating and damage detection of structures. This process as shown in Figure 1 begins by feeding the

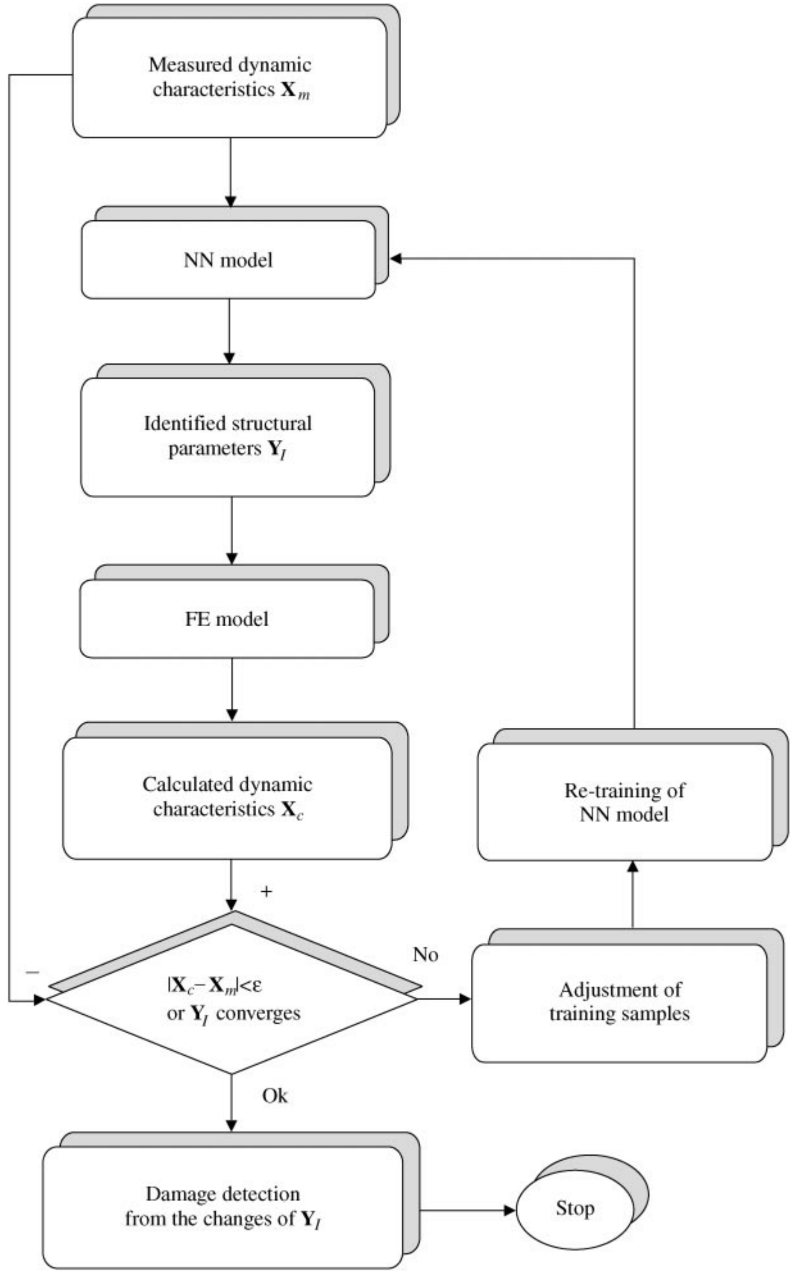

Figure 1. Iterative NN process for identification of structural parameters.

measured dynamic characteristics $\mathbf{X}_m$ into an NN model which is trained beforehand. The outputs of the NN model are the identified structural parameters $\mathbf{Y}_I$. These identified structural parameters are then fed into the finite element (FE) model to produce a set of calculated dynamic characteristics $\mathbf{X}_c$. A comparison between the calculated dynamic characteristics $\mathbf{X}_c$ and the measured dynamic characteristics $\mathbf{X}_m$ is made. If these two sets of parameters differ significantly, then the NN model will be retrained on-line using adjusted training samples that contain $\mathbf{X}_c$ and $\mathbf{Y}_I$. The retrained NN model is then used to identify

TABLE 1

Orthogonal array $OA(4, 3, 2, 2)$

Test	Factor			Response (results)
	A	B	C	
1	0	0	0	R_{000}
2	0	1	1	R_{011}
3	1	0	1	R_{101}
4	1	1	0	R_{110}

the structural parameters again by feeding in the measured dynamic characteristics $\mathbf{X}_m$. This identification and on-line retraining procedure is repeated until the difference between $\mathbf{X}_c$ and $\mathbf{X}_m$ becomes insignificantly small or until $\mathbf{Y}_I$ converges. At the end of the iteration the final identified parameters are guaranteed to produce the dynamic characteristics that are very close to the measured ones. When compared to the original design, these structural parameters can be used to infer the location and the extent of damage in the structure.

The most time-consuming portion of the procedure is the initial and the re-training of the NN model. This training process involves the calculation of the weight matrix inside the NN model using some training samples. These training samples consist of sets of inputs (natural frequencies and changes of mode shape curvatures obtained through FE analysis) and outputs (structural parameters). To effectively train the NN model, the training samples should contain all possible variations of structural parameters which are then used to produce corresponding dynamic characteristics for the structure via FE analysis. If there are k structural parameters selected for updating and each of these parameters comes with s levels of variation, then the number of parametric combinations to cover all possible variations for these parameters is s^k. While more training samples could result in a better trained NN model, the computational time inevitably increases as the number of training samples increases. It is necessary to select the samples wisely so that more effective training of the NN model can be achieved.

3. ORTHOGONAL ARRAYS (OA)

Orthogonal arrays (OAs) are originally proposed in the quality engineering for laying out the experiment to evaluate several factors (or parameters) using a minimal number of tests. The aim of the OA method is to provide a systematic way of studying the effects of the individual factor on the outcome as well as how these factors interact. These factors come with several levels of parametric variation and may have interaction effects which means that two or more factors together produce a result different from their separate effects [19]. In the following, the notation $OA(N, k, s, t)$ is used to represent an OA that has N number of experimental runs, k factors with s levels each and a strength of t [20]. The strength represents the number of columns where all the possibilities can be seen equal number of times.

As an example, Table 1 shows the orthogonal array $OA(4, 3, 2, 2)$ that outlines four experimental runs for three 2-level factors (A, B and C) with strength 2. The response or the results of the experiments are also attached in the last column of the table. The levels of factors are indicated by 0 (for low level) and 1 (for high level). This OA has four rows and three columns (excluding the response column). Each row represents a test setup with specified factor levels. It can be seen that each column (factor) contains two level 0 and two level 1 conditions. Note that any two columns in this OA have the same level combinations

(0, 0), (0, 1), (1, 0) and (1, 1). Thus, the three columns in this OA are orthogonal to each other. This orthogonality provides a fully balanced experimental arrangement which is comprehensive in terms of test results and efficient in terms of the number of tests required. For instance, after performing these four experiments, the response for the low level factor A, RA_0, and the response for the high level of factor C, RC_1, can be found, respectively, as

$$RA_0 = (R_{000} + R_{011})/2, \qquad RC_1 = (R_{011} + R_{101})/2. \tag{1, 2}$$

These simple calculations illustrate that the response of each level of every factor can be obtained from these four experimental results. Figure 1(a) shows these four experimental runs in the design space where each solid dot represents a set of design parameters.

The existence of an OA for specified values of N (experimental runs), k (factors), s (levels) and t (strengths) is a fundamental and challenging question for researchers. Given that an OA exists, Galois fields appear to be a power tool for the construction of OAs. Hedayat *et al.* [20] summarized several tables of OAs for practical usages: (1) tables showing the smallest possible index (and hence the smallest number of experimental runs) in 2-, 3- and 4-level OAs with at most 32 factors and strength between 2 and 10; (2) a table of both mixed- and fixed-level OAs of strength 2 with up to 100 experimental runs; (3) a table outlining the connection between OAs and other combinatorial structures. Sloane [21] further put together a library of over 200 OAs including two levels, three levels and mixed-level of strengths 2 and 3.

A four-step procedure for the determination of an appropriate OA is suggested by Besterfield *et al.* [19]: (1) define the number of factors and their levels; (2) determine the degrees of freedom; (3) select an orthogonal array; and (4) consider any interactions. The degree of freedom determined the minimum number of experimental runs. For a test condition that involves k factors each with s levels, the degree of freedom is $k(s-1)+1$. An appropriate OA is the one whose number of experimental runs N is equal to or slightly greater than the degrees of freedom.

4. OTHER SELECTION METHODS

To validate the efficiency of the OA method, four other sample selection methods are also used for comparison. They are the full factorial (FF), the hypercube (HC) [22], the linear (LI) [16], and the random (RA) method [4, 16].

The samples selected using the FF method are basically the corner points of the domain spanned by the design space as shown in Figure 2(b) for the case of three 2-level parameters. In a full factorial experiment, all possible combinations of the factors are investigated. In general, the total number of experimental runs for k s-level parameters is s^k. Apparently, the sample number would increase drastically once k or s becomes large.

The HC method, or the face-centered central composite design [22], is built around the midpoint of the design space. The original HC method chooses points at each corner (2^k points for k factors), the midpoint for each face ($2k$ points), and the midpoint of the design space (the initial design). The total number of experimental runs is $2^k + 2k + 1$. This method was originally proposed for designing experiments involving k 3-level factors. For the NN applications, Rogers [16] suggested to drop the 2^k corner points from the original HC method as a first attempt to reduce the number of experimental runs. The total number of runs for this modified HC method is $2k + 1$. Figure 2(c) shows the sample selection for three 2-level factors using the modified HC method.

The LI method selects points between the lower bound (0, 0, 0) and the upper bound, (1, 1, 1) with an equal linear increment [16]. The increment is determined by the required number of samples. The RA method generates a given number of samples by random

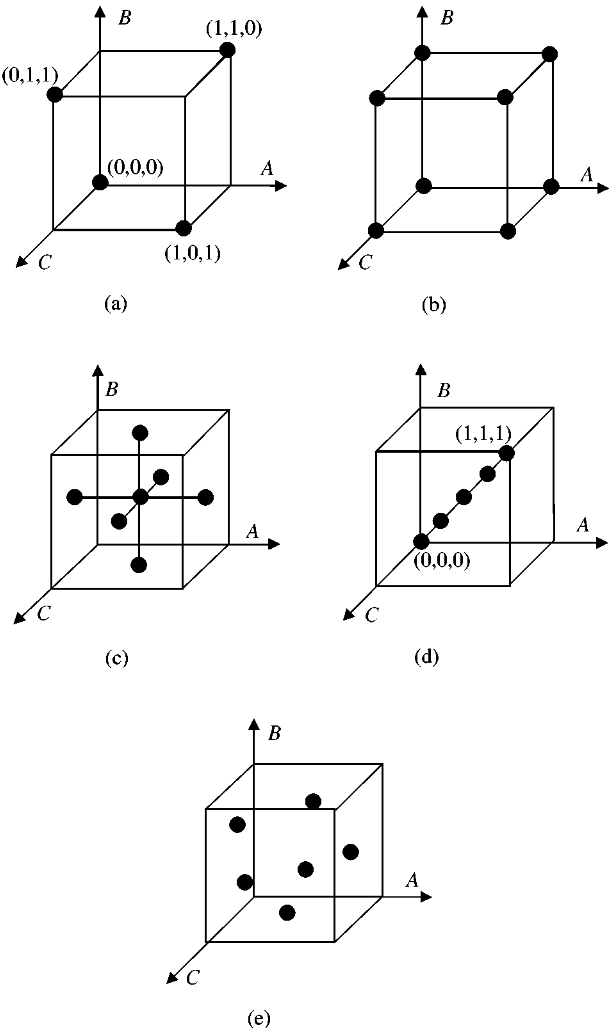

Figure 2. Methods for training sample selection: (a) orthogonal array; (b) full factorial; (c) hypercube; (d) linear; (e) random.

number generators that generate parametric values within the variation bounds [16]. The generated samples should reflect the probability distribution of the parameters to be considered. Atalla and Inman [4] reported that a random generation of samples yields the best result for the NN training.

5. NUMERICAL EXAMPLES

5.1. UPDATE OF THE FLEXURAL RIGIDITIES OF A SIMPLY SUPPORTED BEAM

The effect of different sample selection methods on the training of two NN models is first illustrated using a simply supported T beam as shown in Figure 3. The beam comes with a length of 120 cm and is modelled using four 2-node four-degree-of-freedom beam elements. The design material and geometrical parameters are: elastic module $E = 21$ GPa, mass density $\rho = 2{\cdot}5 \times 10^3$ kg/m^3, the Poisson ratio $\nu = 0{\cdot}17$, cross-sectional area $A = 1{\cdot}58 \times 10^{-2}$ m^2, and bending moment of inertia $I = 1{\cdot}26 \times 10^{-4}$ m^4. The first four natural frequencies are computed to be 123, 491, 591 and 1044 Hz respectively.

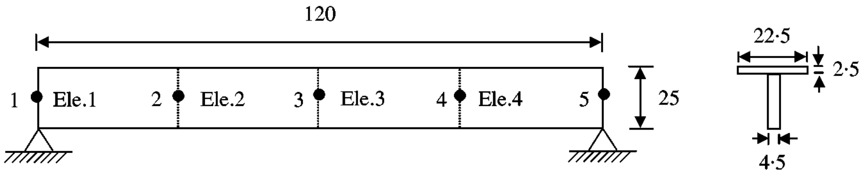

Figure 3. A simply supported T beam (unit: cm).

As for the two NN models, one is termed as NN1 and has one hidden layer of 11 nodes and the other is termed as NN2 and has two hidden layers of 16 and 7 nodes each. The inputs to the NN models **X** consist of the first four natural frequencies f_1–f_4 plus the three co-ordinate modal assurance criterion (COMAC) valuces at the three internal node points,

$$\mathbf{X} = [f_1\ f_2\ f_3\ f_4\ COMAC(2)\ COMAC(3)\ COMAC(4)]^{\mathrm{T}}. \qquad (3)$$

The flexural rigidities EI's of the four elements are selected as the structural parameters **Y** output from the NN models, i.e.,

$$\mathbf{Y} = [EI_1\ \ EI_2\ \ EI_3\ \ EI_4]^{\mathrm{T}}. \qquad (4)$$

The modified back-propagation learning algorithm [16] is used to train the NN models.

It is assumed that there are three levels of variation for each of the four rigidities which correspond to 125% (level 1), 100% (level 2), and 75% (level 3) of their initial design values. The total numbers of training samples generated using the FF, the OA and the HC methods are 81, 9 and 9 respectively. The sample numbers of the LI and RA methods are both set at 9 for the purpose of comparison. The training samples for these five design methods are outlined in Table 2. The first four natural frequencies and the three nodal COMAC values are first calculated using four beam elements for each of the 81 samples. The training process is stopped when the number of iterations exceeds 200 000 or the root mean square (r.m.s.) value becomes smaller than 0·1%.

Figures 4 and 5 show the convergence of the r.m.s. values of the difference between the actual and the predicted EI's during the training of the NN1 and NN2 models using these five sets of samples respectively. It can be seen that the r.m.s. values all converge as the number of iterations increases for both models. The fastest convergence is seen for the LI method which is followed by the OA method. The FF method appears to give the slowest convergence rate perhaps due to the large amount of training samples involved. Table 3 shows the training results of the NN1 and NN2 models. It can be seen that the FF method requires the longest time to converge and the fastest convergence is seen for the LI method. For this particular example, the training time of the OA method is only about one-tenth of that of the FF method.

Next, nine test samples (see the last column of Table 2) are randomly selected from the 81 training samples produced by the FF method to validate the NN models trained using the five methods mentioned above. Except one test sample (TS-6) that coincided with one of the samples of the HC method, all other test samples are not used in the training for the OA, HC, LI and RA methods. Figures 6 and 7 show the r.m.s. values of the differences between the actual and the predicted EI's for the NN1 and NN2 models respectively. It can be seen from both the figures that the NN models trained using the FF method produce the lowest r.m.s. errors for all nine samples. This is to be expected since these samples are included in the FF training set. The r.m.s. errors for the OA method are quite constant for these test samples and on the average appear to be lower than those for the other three methods (HC, LI and RA). The r.m.s. errors for the HC and RA methods appear to be higher than those for the other three methods on the average. To further verify the accuracy of the NN

TABLE 2

Training samples for five design methods

	Sample design									
No.	EI_1	EI_2	EI_3	EI_4	FF	OA	HC	LI	RA	TS†
1	1	1	1	1	∨			×		
2	1	1	1	2	∨				◆	
3	1	1	1	3	∨					
4	1	1	2	1	∨					
5	1	1	2	2	∨					
6	1	1	2	3	∨					1
7	1	1	3	1	∨					
8	1	1	3	2	∨	O				
9	1	1	3	3	∨					
10	1	2	1	1	∨					
11	1	2	1	2	∨					
12	1	2	1	3	∨				◆	
13	1	2	2	1	∨	O				
14	1	2	2	2	∨		●			
15	1	2	2	3	∨					
16	1	2	3	1	∨					
17	1	2	3	2	∨				◆	
18	1	2	3	3	∨					
19	1	3	1	1	∨					
20	1	3	1	2	∨					
21	1	3	1	3	∨	O			◆	
22	1	3	2	1	∨					
23	1	3	2	2	∨				◆	
24	1	3	2	3	∨					2
25	1	3	3	1	∨					3
26	1	3	3	2	∨					
27	1	3	3	3	∨					
28	2	1	1	1	∨	O				
29	2	1	1	2	∨					
30	2	1	1	3	∨					
31	2	1	2	1	∨					
32	2	1	2	2	∨		●			
33	2	1	2	3	∨					
34	2	1	3	1	∨					
35	2	1	3	2	∨					4
36	2	1	3	3	∨					
37	2	2	1	1	∨					
38	2	2	1	2	∨		●			
39	2	2	1	3	∨					5
40	2	2	2	1	∨		●			
41	2	2	2	2	∨		●	×		
42	2	2	2	3	∨		●			6
43	2	2	3	1	∨					
44	2	2	3	2	∨		●			
45	2	2	3	3	∨	O				
46	2	3	1	1	∨					
47	2	3	1	2	∨					
48	2	3	1	3	∨					7
49	2	3	2	1	∨					
50	2	3	2	2	∨	O	●			
51	2	3	2	3	∨					
52	2	3	3	1	∨					

TABLE 2

Continued

No.	Sample design				FF	OA	HC	LI	RA	TS†
	EI_1	EI_2	EI_3	EI_4						
53	2	3	3	2	$\vee$					
54	2	3	3	3	$\vee$					8
55	3	1	1	1	$\vee$					
56	3	1	1	2	$\vee$					
57	3	1	1	3	$\vee$					
58	3	1	2	1	$\vee$					
59	3	1	2	2	$\vee$					
60	3	1	2	3	$\vee$	O			◆	
61	3	1	3	1	$\vee$					
62	3	1	3	2	$\vee$					
63	3	1	3	3	$\vee$					
64	3	2	1	1	$\vee$					
65	3	2	1	2	$\vee$	O				
66	3	2	1	3	$\vee$					
67	3	2	2	1	$\vee$					
68	3	2	2	2	$\vee$		●			
69	3	2	2	3	$\vee$					
70	3	2	3	1	$\vee$					
71	3	2	3	2	$\vee$				◆	
72	3	2	3	3	$\vee$					
73	3	3	1	1	$\vee$				◆	
74	3	3	1	2	$\vee$					
75	3	3	1	3	$\vee$					
76	3	3	2	1	$\vee$					
77	3	3	2	2	$\vee$					
78	3	3	2	3	$\vee$					
79	3	3	3	1	$\vee$	O			◆	
80	3	3	3	2	$\vee$					9
81	3	3	3	3	$\vee$			$\times$		
82	$3+\Delta$‡	$3+\Delta$	$3+\Delta$	$3+\Delta$				$\times$		
83	$3+2\Delta$	$3+2\Delta$	$3+2\Delta$	$3+2\Delta$				$\times$		
84	$3+3\Delta$	$3+3\Delta$	$3+3\Delta$	$3+3\Delta$				$\times$		
85	$3+5\Delta$	$3+5\Delta$	$3+5\Delta$	$3+5\Delta$				$\times$		
86	$3+6\Delta$	$3+6\Delta$	$3+6\Delta$	$3+6\Delta$				$\times$		
87	$3+7\Delta$	$3+7\Delta$	$3+7\Delta$	$3+7\Delta$				$\times$		

†TS: test samples. ‡Δ = (level 1 − level 3)/8.

models, four additional samples which are not in the 81 training samples of the FF method are arbitrarily generated. They are: sample A: (EI_1, EI_2, EI_3, EI_4) = (116%, 112%, 91%, 79%) of the design value; sample B: (113%, 84%, 86%, 106%); sample C: (105%, 83%, 91%, 79%); and sample D: (87%, 81%, 77%, 111%). The r.m.s. of the differences between the actual and the predicted *EI*'s for NN1 and NN2 models are plotted in Figures 8 and 9 respectively. Again, it can be seen from both the figures that the NN models trained using the FF method produce the lowest r.m.s. values for all four samples. Among the four methods that are trained using nine samples only (OA, HC, LI and RA), the NN models trained using the samples generated by the OA method produce the lowest r.m.s. values. And just like in Figures 6 and 7, the r.m.s. values for the HC and RA methods appear to be higher than those for the other three methods.

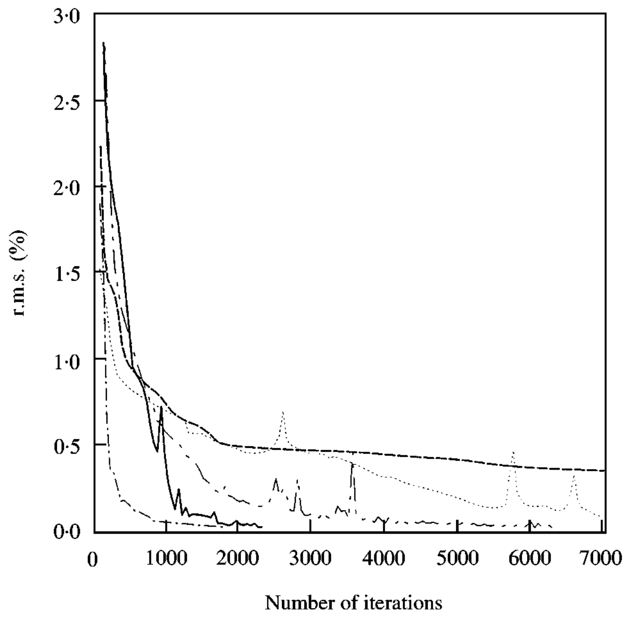

Figure 4. Convergence of the root mean square (r.m.s.) values of the difference between the actual and the predicted *EI's* during the training of the NN1 model: ------, FF; ——, OA; ······, HC; ------, LI; —--—, RA.

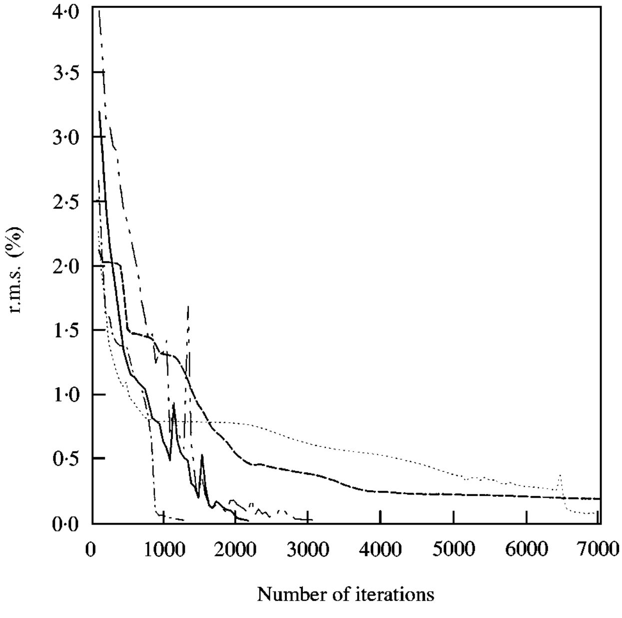

Figure 5. Convergence of the root mean square (r.m.s.) values of the difference between the actual and the predicted *EI*'s during the training of the NN2 model: ------, FF; ——, OA; ······, HC; ------, LI; —--—, RA.

5.2. UPDATE OF THE MATERIAL PROPERTIES AND THE BOUNDARY CONDITIONS OF A CIRCULAR PLATE

Figure 10 shows a circular plate with two different thickness values, where t_1 and t_2 are the plate thicknesses for the regions $a/2 \leqslant r \leqslant a$ and $0 \leqslant r \leqslant a/2$ respectively. The elastic

TABLE 3

Training results of the NN1 and NN2 models

Method	r.m.s. (%)	No. of iterations	Time (min)
	NN1 model		
FF	0·1	99 962	41
OA	0·08	2259	3
HC	0·06	9244	7
LI	0·06	1754	2
RA	0·07	6268	5
	NN2 model		
FF	0·08	26 715	36
OA	0·07	2148	4
HC	0·07	8652	12
LI	0·06	1250	2
RA	0·07	3002	4

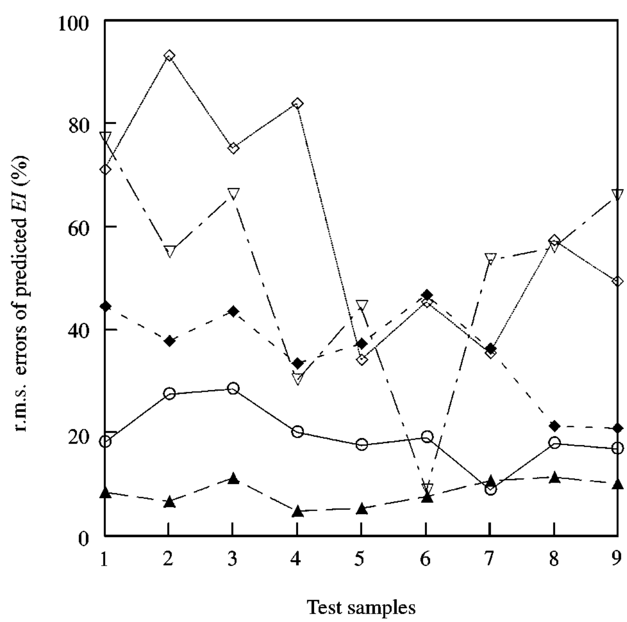

Figure 6. r.m.s. values for the nine test samples using the NN1 model: —▲—, FF; —○—, OA; —▽—, HC; --◆--, LI; --◇--, RA.

module, the Poisson ratio and the mass density of the plate are denoted as E, ν, ρ respectively. Also, the boundary supports are modelled using rotational and translational springs with constants of Φ and K respectively. For the sake of simplicity, the following characteristic parameters are used in this example:

$$\bar{\omega}_m = \omega_m a^2 \sqrt{\rho t_1/\bar{E}}, \quad \bar{K} = Ka^3/\bar{E}, \quad \bar{\Phi} = \Phi a^3/\bar{E},$$

$$\bar{E} = Et_1^3/12(1-\nu^2), \quad \bar{t} = t_2/t_1. \tag{5}$$

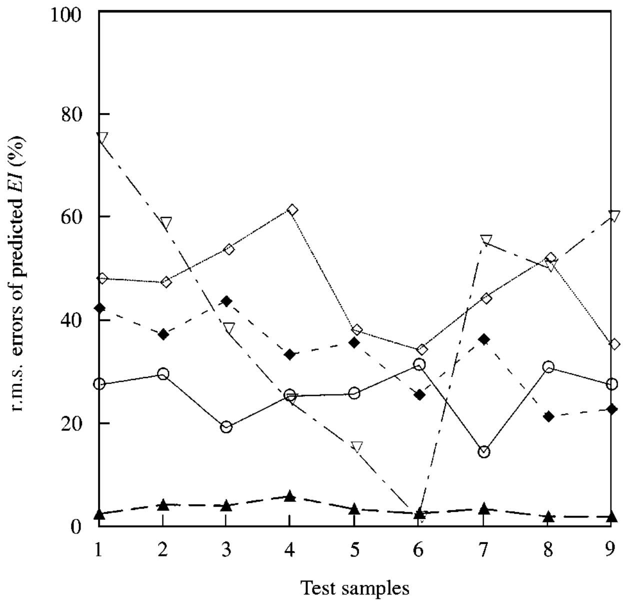

Figure 7. r.m.s. values for the nine test samples using the NN2 model: —▲—, FF; —○—, OA; —▽—, HC; --◆--, LI; --◇--, RA.

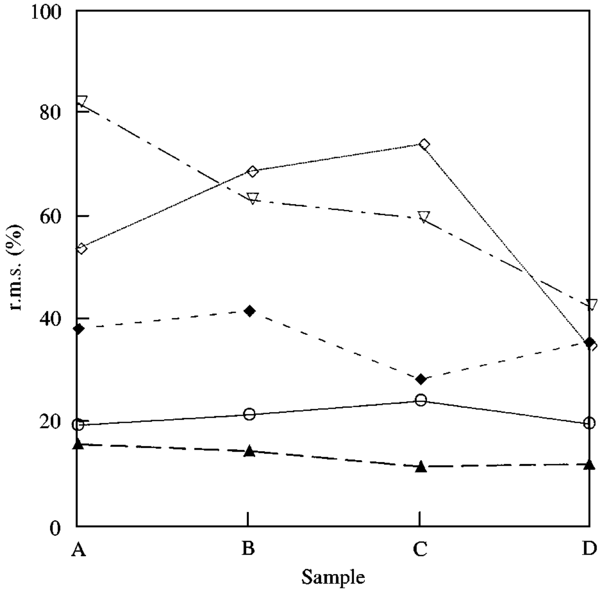

Figure 8. r.m.s. values for the four additional samples using the NN1 model: —▲—, FF; —○—, OA; —▽—, HC; --◆--, LI; --◇--, RA.

The design characteristic parameters are: $\bar{K} = 16$, $\bar{\Phi} = 4$, $\bar{E} = 19{\cdot}2$ and $\bar{t} = 1{\cdot}5$. The corresponding first six design characteristic frequencies $\bar{\omega}_m$ $(m = 1, \ldots, 6)$ are calculated to be 4·6, 17·1, 50·8, 113·3, 196·4 and 302·7, respectively, using the differential quadrature method [23].

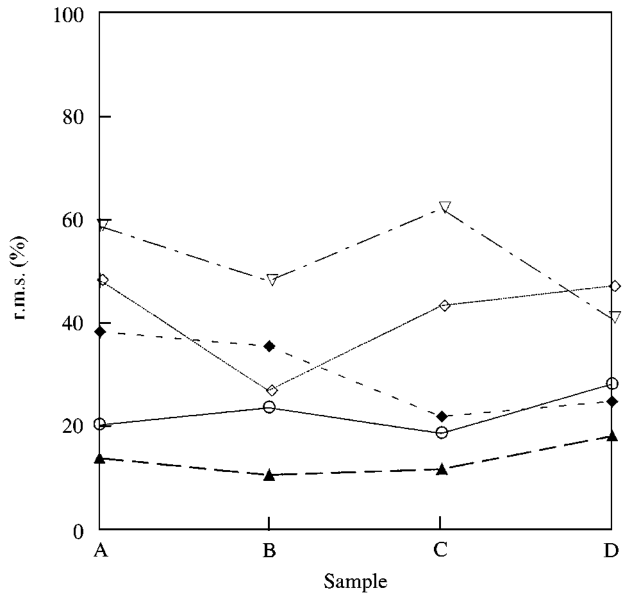

Figure 9. r.m.s. values for the four additional samples using the NN2 model: —▲—, FF; —○—, OA; —▽—, HC; --◆--, LI; --◇--, RA.

A third NN model, termed as NN3, is used to update the material properties and the boundary conditions of this circular plate. The NN3 has one hidden layer of 11 nodes and its inputs consist of the first six characteristic frequencies $\bar{\omega}_m$ ($m = 1, \ldots, 6$).

$$\mathbf{X} = [\bar{\omega}_1 \ \bar{\omega}_2 \ \bar{\omega}_3 \ \bar{\omega}_4 \ \bar{\omega}_5 \ \bar{\omega}_6]^{\mathrm{T}}. \tag{6}$$

The design parameters $\bar{K}$, $\bar{\Phi}$, $\bar{E}$ and $\bar{t}$ are selected as the output $\mathbf{Y}$ of the NN3 model,

$$\mathbf{Y} = [\bar{K} \ \ \bar{\Phi} \ \ \bar{E}\,\bar{t}]^{\mathrm{T}}. \tag{7}$$

As in the previous example, it is also assumed that there are three levels of variation for each of the four design parameters which correspond to 125% (level 1), 100% (level 2), and 75% (level 3) of their initial design values. The total number of training samples generated using the FF method is again $3^4 = 81$. The configurations of these 81 samples are identical to those shown in Table 2.

The total number of training samples for this 4-factor 3-level condition using the OA method is again nine. For this example, the orthogonal array $OA(9, 4, 3, 2)$ by Sloane [21] as shown in Table 4 is used. The configurations of these nine OA training samples appear to be different from those in Table 1. It should be noted at this point that the orthogonal array is normally non-unique for any given condition since the sequence of factors and levels can be arbitrarily assigned.

The training samples for the HC, LI and RA methods are all set to be 9. The training samples for the RA method are arbitrarily selected from the 81 sample configurations for the FF method in Table 2 and are corresponded to samples 3, 10, 15, 20, 23, 35, 52, 58 and 70. The training samples for the HC and LI methods on the other hand are identical to those selected in the previous example as shown in Table 2. Table 5 shows the training results in terms of the number of iterations and the computing time for these five methods. These training results are obtained using the same learning algorithm under the same convergence criterion as the previous example. Again, it is seen that, due to the large

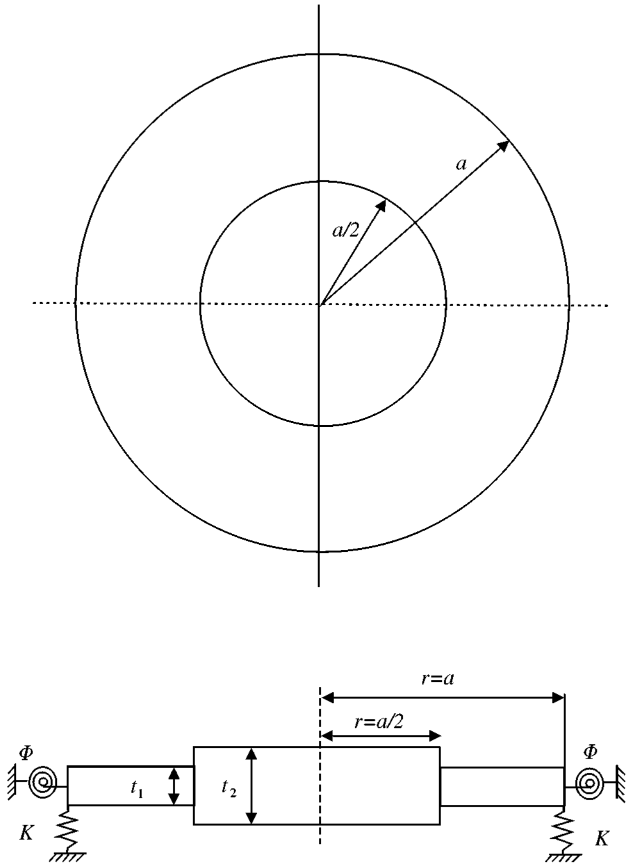

Figure 10. A circular plate with the elastic supports.

TABLE 4

Training samples generated from the OA method

No.	$OA(9, 4, 3, 2)$ Factor 1	Factor 2	Factor 3	Factor 4	Configuration of samples $\bar{K}$	$\bar{\Phi}$	$\bar{E}$	$\bar{t}$
1	1	1	1	1	20	5	24	1·875
2	1	2	2	3	20	4	19·2	1·125
3	1	3	3	2	20	3	14·4	1·5
4	2	1	2	2	16	5	19·2	1·5
5	2	2	3	1	16	4	14·4	1·875
6	2	3	1	3	16	3	24	1·125
7	3	1	3	3	12	5	14·4	1·125
8	3	2	1	2	12	4	24	1·5
9	3	3	2	1	12	3	19·2	1·875

number of training samples, the computing time for the FF method is significantly longer than for the other four methods that use only nine training samples. Among these four methods, the LI method converges faster than the other three.

TABLE 5

Training results for the NN3 model

Method	r.m.s. (%)	No. of iterations	Time (min)
FF	0·1	119 058	45
OA	0·07	2864	4
HC	0·06	10 085	8
LI	0·08	1913	2
RA	0·07	6838	5

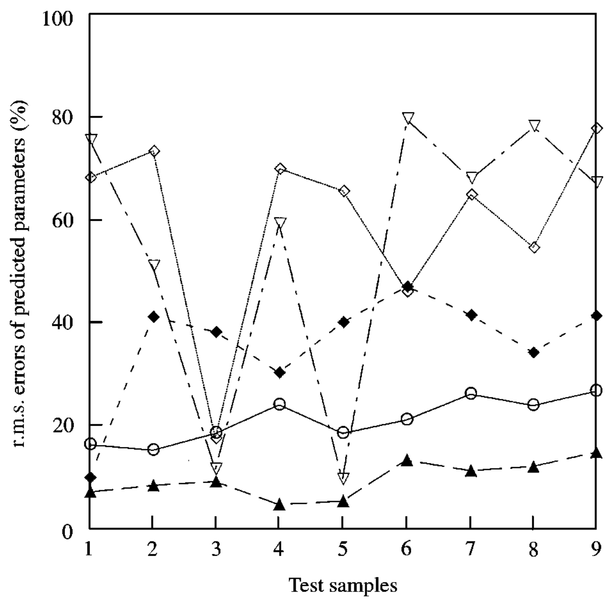

Figure 11. r.m.s. values for the nine test samples using the NN3 model: —▲—, FF; —○—, OA; —▽—, HC; --◆--, LI; --◇--, RA.

To test the accuracy of the trained NN3 model, nine additional test samples are generated. The first five are randomly selected from the 81 training samples of the FF method. They have the same configurations as samples 1, 15, 33, 40 and 67, as shown in Table 2. The other four samples have the same configurations as those of the four additional samples (A, B, C, and D) in the previous example. Figure 11 shows the r.m.s. values of the differences between the actual and the predicted characteristic parameters from the NN3 model. It can be seen that the NN3 model trained using the samples from the FF method has the lowest r.m.s. values which again is at the expense of extra computing time. Among the other four methods, the NN3 model trained using the OA method produces the smallest and the most uniform r.m.s. values. This example once again reaffirms the superiority of the OA method for the training sample selection. As compared to the FF method, the number of training samples for the OA method is considerably smaller, which leads to considerable saving in computing time and the accuracy of the trained NN model does not seem to be significantly deteriorated.

6. CONCLUDING REMARKS

In developing an iterative neural network technique for model updating of structures, it has been shown that the number of training samples required increases exponentially as the number of parameters to be updated increases. It is noted that the selection of training samples for NN models resembles the design of experiments which involve several factors varying with several levels. The orthogonal arrays have been developed and adopted by the experimentalists for laying out a minimal number of tests while retaining all the necessary information. In this study, we investigate the use of orthogonal arrays for the sample selection for training NN models. Four other selection methods, including the full factorial, the modified hypercube, the linear and the random methods, are used for comparison.

The comparison is made based on two numerical examples. One is the update of the flexural rigidities of a simply supported beam and the other is the update of the material properties and the boundary conditions of a circular plate. The results suggest that although the NN models trained using the samples generated by the full factorial method give the most accurate prediction, the training time is about 10 times that of the orthogonal arrays method. It is expected that the full factorial method would not be practically feasible if the number of parameters further increases. Among the four methods that use the same number of training samples, the NN models trained using the orthogonal arrays method consistently produce the most accurate prediction.

It is concluded that the use of the orthogonal arrays method can significantly reduce the number of training samples without affecting too much the accuracy of the neural network prediction.

ACKNOWLEDGMENTS

This work is supported by an Earmarked Grant from the RGC of Hong Kong and a research grant UIC99/98.EG01 funded by the NSF via the University of Illinois at Chicago.

REFERENCES

1. R. D. VANLUCHENE and R. SUN 1990 *Microcomputers in Civil Engineering* **5**, 207–215. Neural networks in structural engineering.
2. S. F. MASRI, A. G. CHASSIAKOS and T. K. GANGHEY 1993 *Journal of Applied Mechanics* **60**, 123–133. Identification of nonlinear dynamic systems using neural networks.
3. J. GHABOUSSI, J. H. GARRETT and X. WU 1991 *Journal of Engineering Mechanics, American Society of Civil Engineers* **117**, 132–153. Knowledge-based modeling of material behavior with neural networks.
4. M. J. ATALLA and D. J. INMAN 1998 *Mechanical System and Signal Processing* **12**, 135–161. On model updating using neural networks.
5. R. I. LEVIN and N. A. J. LIEVEN 1998 *Journal of Sound and Vibration* **210**, 593–607. Dynamic finite element model updating using neural networks.
6. J. E. STEPHENS and R. D. VANLUCHENE 1994 *Microcomputers in Civil Engineering* **9**, 119–128. Integrated assessment of seismic damage in structures.
7. M. F. ELKORDY, K. C. CHANG and G. C. LEE 1994 *Microcomputers in Civil Engineering* **9**, 83–96. A structural damage neural network monitoring system.
8. P. Z. SZEWCZYK and P. HAJELA 1994 *Journal of Computing in Civil Engineering, American Society of Civil Engineers* **8**, 163–178. Damage detection in structures based on feature-sensitive neural networks.
9. J. RHIM and S. W. LEE 1995 *Computational Mechanics* **16**, 437–443. Neural network approach for damage detection and identification of structures.

10. P. C. PANDEY and S. V. BARAI 1995 *Computers and Structures* **54**, 597–608. Multilayer perception in damage detection of bridge structures.
11. M. NAKAMURA, S. F. MASRI, A. G. CHASSIAKOS and T. K. CAUGHEY 1998 *Earthquake Engineering & Structural Dynamics* **27**, 997–1010. A method for non-parametric damage detection through the use of neural networks.
12. J. ZHAO, J. N. IVAN and J. T. DEWOLF 1998 *Journal of Infrastructure Systems* **4**, 93–101. Structural damage detection using artificial neural networks.
13. X. WU, J. GHABOUSSI and J. H. GARRETT JR 1992 *Computers and Structures* **42**, 649–659. Use of neural networks in detection of structural damage.
14. S. W. DOEBLING, C. R. FARRAR, M. B. PRIME and D. W. SHEVITZ 1996 *Technical Report LA*-13070-*MS*, *Los Alamos National Laboratory*, *U.S.A*. Damage identification and health monitoring of structural and mechanical systems from changes in their vibrations characteristics: a literature review.
15. G. E. P. BOX and N. R. DRAPER 1987 *Empirical Model-Building and Response Surface*. New York: John Wiley & Sons.
16. J. L. ROGERS 1994 *Journal of Computing in Civil Engineering*, *American Society of Civil Engineers* **8**, 252–265. Simulating structural analysis with neural network.
17. C. C. CHANG, T. Y. P. CHANG and Y. G. XU 2000 *Smart Materials and Structures* **9**, 59–68. Adaptive neural networks for model updating of structures.
18. C. C. CHANG, T. Y. P. CHANG Y. G. XU and M. L. WANG 2001 *Journal of Intelligent Material Systems and Structures* **11**, 32–42. Structural damage detection using an iterative neural network.
19. D. H. BESTERFIELD, C. BESTERFIELD-MICHNA, G. H. BESTERFIELD and M. BESTERFIELD-SACRE 1995 *Total Quality Management*. Englewood Cliffs, NJ: Prentice-Hall, Inc.
20. A. S. HEDAYAT, N. J. A. SLOANE and J. STUFKEN 1999 *Orthogonal Arrays—Theory and Applications*. New York: Springer, Verlag.
21. N. J. A. SLOANE 2000 Web site: *http://www.research.att.com/ ~ njas/oadir/index.html*. A library of orthogonal arrays.
22. R. L. MANSON, R. F. GUNST and J. L. HESS 1989 *Statistical Design and Analysis of Experiments*. New York: John Wiley and Sons.
23. T. Y. WU and G. R. LIU 2000 *Journal of Sound and Vibration* **233**, 195–213. The differential quadrature rule for initial-value differential equations.