Use of Neural network to predict the peak ground accelerations and pseudo spectral accelerations for Mexican Inslab and Interplate Earthquakes

Adrián Pozos-Estrada[*], Roberto Gómez and H.P. Hong

Resumen

El uso de redes neuronales artificiales es explorado para predecir aceleraciones máximas del terreno y pseudoaceleraciones para sismos de tipo intraslab e interplaca. Un total de 277 y 418 registros sísmicos de dos componentes para sismos de intraslab e interplaca, respectivamente, son usados para entrenar los modelos de las redes neuronales artificiales con alimentación hacia adelante y con un algoritmo de aprendizaje de retroalimentación. Se consideran redes neuronales artificiales con una y dos capas ocultas. Con fines de comparación, valores de aceleración máxima del terreno y pseudoaceleración predichos con los modelos de las redes neuronales son comparados con los estimados mediante relaciones de atenuación o relaciones de movimiento fuerte. La comparación indica que los valores predichos, en general, siguen la tendencia de los valores obtenidos con las relaciones de movimiento fuerte. Sin embargo, se debe llevar a cabo una verificación extensa de los modelos entrenados antes que estos puedan emplearse en análisis de peligro y riesgo sísmico ya que, en ocasiones, los valores predichos no reflejan el comportamiento observado de los registros.

Palabras clave: red neuronal artificial, sismos de subducción, aceleración máxima del terreno, pseudoaceleración, México.

Abstract

The use of Artificial Neural Networks (ANN) is explored to predict peak ground accelerations (PGA) and pseudospectral acceleration (SA) for Mexican inslab and interplate earthquakes. A total of 277 and 418 seismic records with two horizontal components for inslab and interplate earthquakes, respectively, are used to train the ANN models by using an ANN with a feed-forward architecture with a back-propagation learning algorithm. Both ANN with single and two hidden layers are considered. For comparison purposes, the PGA and SA values predicted by the trained ANN models are compared with those estimated with attenuation relations or ground motion prediction equations (GMPEs). The comparison indicates that the predicted PGA and SA values by the trained ANN models, in general, follow the trends predicted by the GMPEs. However, an extensive verification of the trained models must be conducted before they can be used for seismic hazard and risk analysis since, on occasion, the PGA and SA values predicted by the trained ANN models depart from the behaviour observed from the actual records.

Key words: artificial neural network, subduction earthquakes, peak ground acceleration, pseudospectral acceleration, Mexico.

A. Pozos-Estrada[*]
Instituto de Ingeniería
Universidad Nacional Autónoma de México
Ciudad Universitaria
Delegación Coyoacán, 04510
México D.F., México
[*]Corresponding author:APozosE@iingen.unam.mx

R. Gómez
Instituto de Ingeniería
Universidad Nacional Autónoma de México
Ciudad Universitaria
Delegación Coyoacán, 04510
México D.F., México
E-mail: RGomezM@iingen.unam.mx

H.P. Hong
The University of Western Ontario
Department of Civil and Environmental Engineering
London, ON N6A 5B9
Canada
E-mail: hongh@eng.uwo.ca

Introduction

Artificial neural networks (ANNs) have been used in seismic engineering due to their flexibility to deal with highly nonlinear problems (Fausett, 1994). ANNs have been used to predict ground motion measures such as the peak ground displacement (PGD), peak ground velocity (PGV), or peak ground acceleration (PGA), and spectral acceleration (SA) (Günaydin and Günaydin, 2008; Kamatchi *et al*., 2010). ANNs have also been used for generating artificial earthquakes and response spectra, and spectrum compatible accelerograms (Ghaboussi and Li, 1998; Lee and Han, 2002). More recently, Hong *et al*. (2012) showed that the prediction of the PGA and SA by using ANNs with a single hidden layer may not be robust, although it could be considered as an alternative to the commonly used attenuation relations or ground motion prediction equations (GMPEs). Estimation of PGAs for Mexican subduction earthquakes using the ANNs has been explored by García *et al*. (2007). However, the application of ANNs to predict SA for Mexican earthquakes has not been reported in the literature.

The main objective of this study is to investigate the applicability of ANNs to estimate PGA and SA for ground motion records caused by Mexican subduction earthquakes. Two sets of records of Mexican subduction earthquakes obtained at firm soil sites (i.e., site class B according to the NEHRP (BSSC, 2004)) are used in training and qualifying ANNs. Training of the ANN models was carried out using a feed-forward architecture with a back-propagation learning algorithm. Only single and two hidden layers are considered to minimize potential overfitting. The parameters considered in the input layer are: moment magnitude (M_w), closest distance to the fault (R_c) and focal depth (H), while the logarithm of the ground motion measures is used to represent the outcome from the output layer. The predicted PGA and SA values are compared with those estimated from GMPEs to assess the adequacy of the trained ANN models.

ANN Modeling

Description of the ANN modeling

The ANN modeling involves the selection of the number of neurons in the input as well as the hidden and output layers. In this study, the number of neurons in the input layer is considered to be 3, representing M_w, R_c and H as defined earlier. Single and two hidden layers (denoted by 1HL and 2HL, respectively) with multiple hidden neurons are used to approximate the mapping between the input and output layers. The output layer consists of a single neuron that represents the logarithm of the ground motion measures (PGA or SA) for a considered earthquake type and natural vibration period.

An illustration of an ANN model with multiple hidden layers and neurons is depicted in Figure 1, where neurons are weighted and transformed into output values. By considering two hidden layers, the mathematical expression of the output neuron in the output layer, y_{output}, is given by,
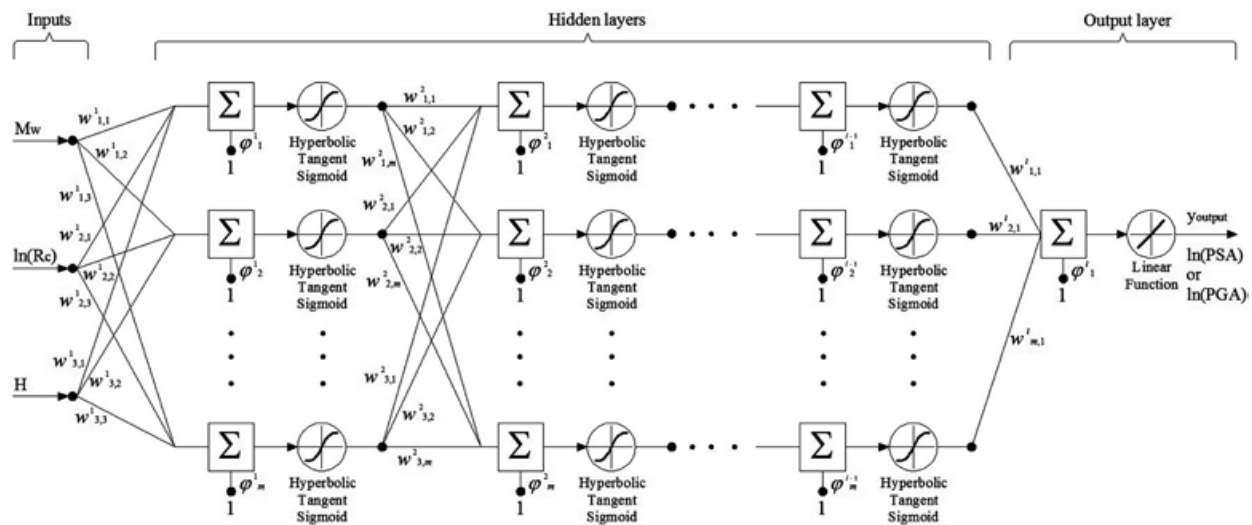

Figure 1. Sketch of an ANN model with multiple hidden layers and neurons.

$$y_{output} = f_3\left(\sum_{k=1}^{m}[w_3]_{k,1}\, f_2\left(\sum_{j=1}^{m}\left([w_2]_{j,k}\, f_1\right.\right.\right.$$

$$\left(\sum_{i=1}^{n}\left([w_1]_{i,j}\, x_i + (\varphi_1)_j\right)\right)$$

$$\left.\left.\left.+ (\varphi_2)_k\right)\right) + (\varphi_3)_1\right), \qquad (1)$$

where n is the total number of neurons in the input layer; x_i is the i-th neuron in the input layer; $[w_1]_{i,j}$, $[w_2]_{j,k}$ and $[w_3]_{k,1}$ are the weights that optimize the mapping between the input and the first hidden layer, between the first and second hidden layer and between the second hidden layer and output layer, respectively; $(\varphi_1)_j$, $(\varphi_2)_k$ and $(\varphi_3)_1$ are the biases associated with the hidden and output layers; $f_1(\ )$, $f_2(\ )$ and $f_3(\ )$ are activation (or transfer) functions between the input and the first hidden layer, the first and the second hidden layer and between the second hidden layer and the output layer, respectively; and m is the total number of neurons in the hidden layers.

Two types of activation functions, namely, the tan-sigmoid function and the linear function are commonly used. These functions are expressed as,

$$f(x) = \left(e^x - e^{-x}\right)/\left(e^x + e^{-x}\right), \qquad (2a)$$

and

$$f(x) = x, \qquad (2b)$$

The former is often used as the transfer function between the input and hidden layers, while the latter is used as the transfer function between the hidden layers and the output layer. Following García *et al.* (2007), in the present study, Eq. (2a) is used as the transfer function between the input and hidden layer(s) and Eq. (2b) is used as the transfer function between the hidden layer(s) and the output layer.

Training ANN

The training of an ANN consists in the minimization of a predefined error function, in terms of observed and predicted output values, by varying the weights and biases. One of the algorithms used to train the ANN is the back-propagation (Fausset, 1994), where the error is propagated backward by adjusting the weights from the output to the input layer. The training can be summarized as follows:

1. Provide the ANN model with sample inputs and known outputs;

2. Evaluate an error function in terms of the difference between the predicted and observed output;

3. Minimize the error function by adjusting the weights and biases of all the layers from the output to the input layer.

For the numerical analysis to be presented in this study, the error function was defined as the mean square error (MSE). The minimization of the MSE (Step 3)) was carried out using the Levenberg-Marquardt algorithm (Marquardt, 1963; Press *et al.*, 1992) that is incorporated into the back-propagation algorithm and implemented in Matlab (Hagan and Menhaj, 1994).

Strong ground motion database and GMPEs

The strong ground motion database employed to develop the ANNs model consists of 695 strong ground motion records, each one with two horizontal components at firm soil sites (class B according to NEHNP –BSSC, 2004) complied by García *et al.* (2005, 2009). There are 277 inslab records and 418 interplate records for the events shown in Figure 2 and listed in Table 1. There are 16 intermediate-depth normal-faulting inslab events with M_w within 5.2 to 7.4 and 40 interplate events with M_w ranging from 5.0 to 8.0. The distribution of M_w and H with respect to R_c is presented in Figure 3. A baseline correction and a high-pass filter with cut-off frequencies of 0.05 Hz for events with $M_w > 6.5$ and 0.1 Hz for the rest events were applied to all the records. The selection criteria of the records can be found in García *et al.* (2005, 2009). The same strong ground motion database was also employed by Hong *et al.* (2009) to develop GMPEs based on the geometric mean (i.e., for a random orientation). As these GMPEs will be used to compare with those predicted by the ANN model, the adopted functional forms and the obtained regression coefficients in Hong *et al.* (2009) are summarized below.

The functional form of the GMPEs for inslab earthquakes is the one given by García *et al.* (2005), which can be written as,

$$\log_{10} Y = c_1 + c_2 M_w + c_3 R - c_4 \log_{10} R + c_5 H + \varepsilon, \qquad (3)$$

where Y (cm/s^2) represents the PGA or SA values, c_i, $i = 1, \ldots, 5$, are the model parameters, M_w is the moment magnitude, $R = \sqrt{R_{cld}^2 + \Delta_0^2}$, R_{cld} (km) is the closest distance to the fault surface for events with $M_w > 6.5$, or the hypocentral distance
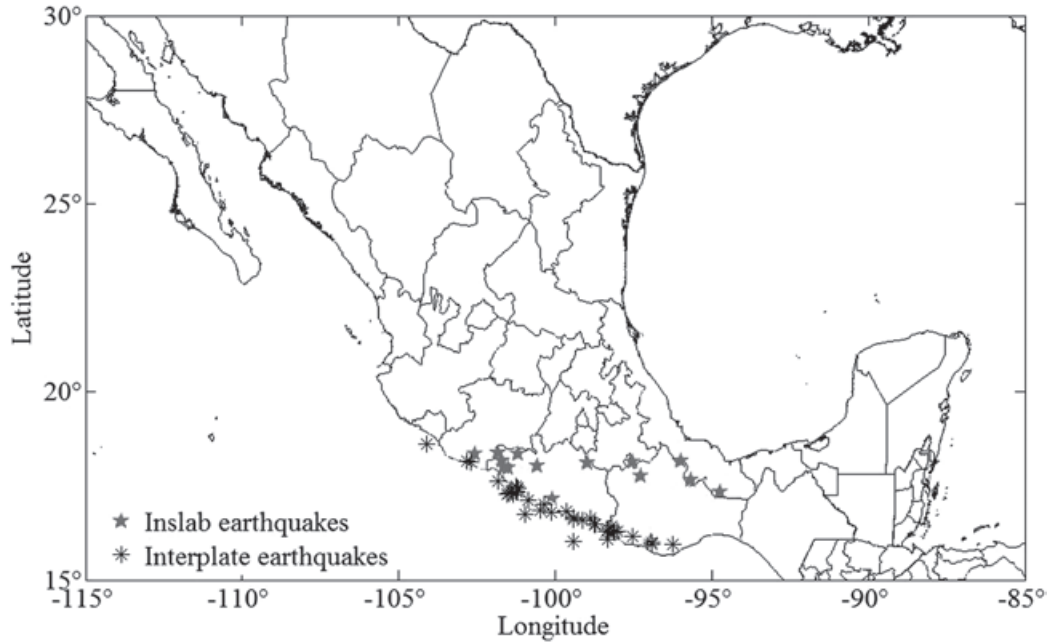
Figure 2. Location of events for the considered records.

Table 1. Inslab and interplate events used in training the ANN models.

Inslab earthquake				Interplate earthquake							
Event No.	No. of Rec.	Date (dd/mm/yy)	Mw	Event No.	No. of Rec.	Date (dd/mm/yy)	Mw	Event No.	No. of Rec.	Date (dd/mm/yy)	Mw
1	6	23/02/94	5.8	1	10	19/09/85	8	21	15	05/07/98	5.3
2	21	23/05/94	6.2	2	8	21/09/85	7.6	22	12	11/07/98	5.4
3	16	10/12/94	6.4	3	9	08/02/88	5.8	23	14	12/07/98	5.5
4	22	11/01/97	7.1	4	6	10/03/89	5.4	24	10	04/09/01	5.2
5	18	22/05/97	6.5	5	10	25/04/89	6.9	25	11	10/11/01	5.4
6	15	20/04/98	5.9	6	5	02/05/89	5.5	26	10	07/06/02	5.2
7	30	15/06/99	6.9	7	8	13/01/90	5.3	27	12	07/06/02	5.5
8	16	21/06/99	6.3	8	6	11/05/90	5.5	28	12	19/06/02	5.3
9	26	30/09/99	7.4	9	9	31/05/90	5.9	29	7	05/08/02	5.4
10	14	29/12/99	5.9	10	6	15/05/93	5.5	30	10	27/08/02	5.0
11	21	21/07/00	5.9	11	12	24/10/93	6.6	31	6	30/08/02	5.2
12	21	05/03/01	5.3	12	15	14/09/95	7.3	32	15	25/09/02	5.3
13	23	06/03/01	5.2	13	10	13/03/96	5.1	33	10	08/11/02	5.2
14	13	30/01/02	5.9	14	9	27/03/96	5.4	34	8	10/12/02	5.4
15	9	17/01/04	5.4	15	17	15/07/96	6.6	35	15	10/01/03	5.2
16	6	20/04/04	5.6	16	10	18/07/96	5.4	36	8	22/01/03	7.5
				17	12	21/01/97	5.4	37	15	01/01/04	6.0
				18	7	16/12/97	5.9	38	11	01/01/04	5.6
				19	12	09/05/98	5.2	39	8	06/02/04	5.1
				20	10	16/05/98	5.2	40	18	14/06/04	5.9

for the rest, $\Delta = 0.0075 \times 10^{0.507Mw}$ is a near-source saturation term defined by Atkinson and Boore (2003), H (km) is the focal depth, and ε is a zero mean error term with standard deviation σ, in which $\sigma = (\sigma_r^2 + \sigma_e^2)^{0.5}$, and σ_r and σ_e denote the standard deviation due to intra- and inter-event variability, respectively. If the geometric mean for Y is considered, $\sigma = (\sigma_r^2 + \sigma_e^2 + \sigma_c^2)^{0.5}$ and the standard deviation σ_c accounts for the random orientation variability.

Table 2. Coefficients for the geometric mean and the horizontal components H1 and H2 for the the records of inslab earthquakes (c_4 equals 1).

T_n (s)	c_1	c_2	c_3	c_5	σ
Geometric mean					
0.20	-0.020	0.595	-0.0036	0.0068	0.31
0.50	-0.907	0.687	-0.0024	0.0034	0.29
1.00	-1.931	0.781	-0.0016	0.0029	0.31
1.50	-2.468	0.831	-0.0014	0.0017	0.31
PGA	-0.109	0.569	-0.0039	0.0070	0.31
Horizontal component 1 (H1)					
0.20	-0.015	0.595	-0.0036	0.0065	0.31
0.50	-0.895	0.688	-0.0023	0.0028	0.29
1.00	-1.987	0.793	-0.0017	0.0029	0.29
1.50	-2.531	0.84	-0.0014	0.0019	0.28
PGA	-0.091	0.569	-0.0038	0.0065	0.31
Horizontal component 2 (H2)					
0.20	-0.034	0.596	-0.0037	0.0071	0.29
0.50	-0.913	0.683	-0.0024	0.004	0.27
1.00	-1.886	0.768	-0.0015	0.003	0.30
1.50	-2.441	0.825	-0.0014	0.0018	0.30
PGA	-0.13	0.568	-0.0039	0.0076	0.29

For the records of interplate earthquakes, the functional form of the GMPE is the one employed by García (2006), which is expressed as,

$$\log_{10} Y = c_1 + c_2 M_w + c_3 R - c_4 \log_{10}\left(R + c_5 10^{c_6 M_w}\right) + c_7 H + \varepsilon ,\tag{4}$$

where $c_i, i = 1, \ldots, 7$, are the model parameters, R (km) is the closest distance to the fault surface for events with $M_w > 6.0$, or the hypocentral distance for the rest, and Y, M_w, H and ε were defined previously. Note that in the above equation c_4 is considered to be given by the following equation (García 2006),

$$c_4 = 1.82 - 0.16 M_w .\tag{5}$$

Using the adopted GMPEs, the records for the events detailed in Table 1, and the regression analysis algorithm given by Joyner and Boore (1993), Hong *et al*. (2009) obtained the model coefficients for a range of natural vibration periods based on the geometric mean. For an easy reference, the model coefficients for a few selected values of the natural vibration period, T_n, are presented in Tables 2 and 3. Moreover, for comparison purposes, the regression analysis in this study is carried out by considering either the first horizontal component (H1), or the second horizontal component (H2). The obtained model

Table 3. Coefficients for the geometric mean and the horizontal components H1 and H2 for the records of interplate earthquakes (c_4 is defined in Equation 5).

T_n (s)	c_1	c_2	c_3	c_5	c_6	c_7	σ
Geometric mean							
0.20	2.609	0.144	-0.0034	0.009	0.475	-0.00410	0.39
0.50	1.542	0.238	-0.0015	0.003	0.515	-0.00300	0.40
1.00	0.734	0.301	-0.0005	0.002	0.509	-0.00500	0.41
1.50	0.214	0.336	-0.0002	0.002	0.495	-0.00490	0.40
PGA	2.545	0.108	-0.0037	0.0075	0.474	-0.00240	0.37
Horizontal component 1 (H1)							
0.20	2.658	0.129	-0.0036	0.009	0.475	-0.00105	0.40
0.50	1.653	0.211	-0.0017	0.003	0.515	-0.00001	0.40
1.00	0.862	0.265	-0.0004	0.002	0.509	-0.00283	0.40
1.50	0.343	0.298	-0.0002	0.002	0.495	-0.00195	0.40
PGA	2.608	0.088	-0.0038	0.0075	0.474	0.00073	0.40
Horizontal component 2 (H2)							
0.20	2.639	0.146	-0.0036	0.009	0.475	-0.00405	0.36
0.50	1.571	0.247	-0.0018	0.003	0.515	-0.00364	0.38
1.00	0.716	0.321	-0.0010	0.002	0.509	-0.00458	0.32
1.50	0.182	0.357	-0.0007	0.002	0.495	-0.00427	0.33
PGA	2.500	0.123	-0.0038	0.0075	0.474	-0.00330	0.34
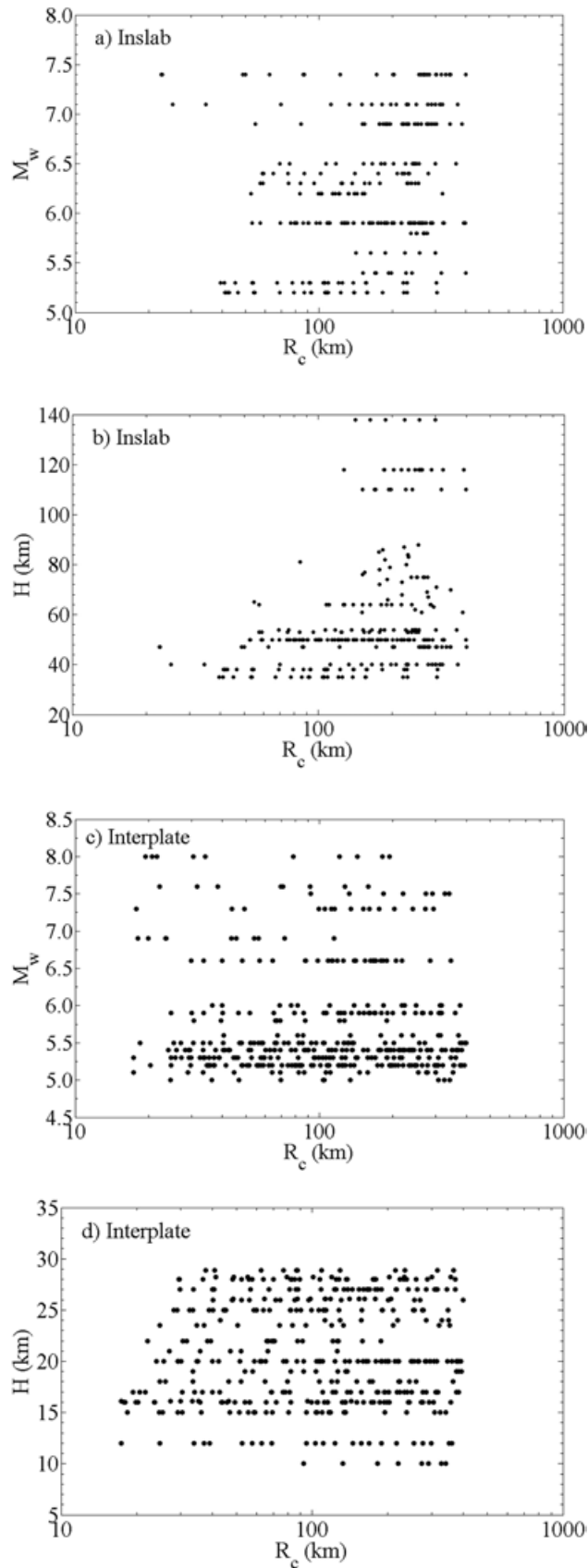
Figure 3. Distribution of M_w and H with respect to R_c: (a) and (b) for inslab events; (c) and (d) for interplate events.

coefficients are also shown in Tables 2 and 3. As the H1 and H2 components represent the seismic excitation for a random orientation (with respect to the source), the developed model coefficients shown in Tables 2 and 3 will be used to predict PGA or SA for a random orientation.

Comparison of the results shown in the tables indicates that the estimated model coefficients based on either only H1 or H2 components differ only slightly from those based on both components (i.e., geometric mean), as expected. Also, the average of the model coefficients obtained from H1 alone and from H2 alone is almost identical to those obtained based on the geometric mean. This fact simply confirms the robustness of the algorithm given by Joyner and Boore (1993) for developing the GMPEs. It also indicates that the number of records used for the purpose of regression analysis is adequate.

Prediction of strong ground motion measures using ANN

Effect of the number of hidden layers and neurons on the training ANN and on predicted values

The selection of the number of hidden layers and neurons is of importance in developing or training an ANN. This selection depends on the nature of the problem to be investigated, and a trial and error process is often followed to determine the adopted structure of the ANN model (Shahin *et al*., 2004). In selecting the ANN model, potential overfitting due to the use of an excessive number of hidden layers and/or neurons needs to be avoided, as this will lead to a lack of learning and the inability to predict the outcomes for scenarios not used in training. To avoid the possible overfitting, several preliminary ANN models were tested by considering combinations of the single and two hidden layers and up to 50 hidden neurons in each hidden layer. Furthermore, a total of 80% randomly selected records were used to train the model, while the remaining 20% of records were used for validating the trained model.

To investigate the effect of the record selection on the trained ANN model, a total of 300 trials were carried out. For each trial, a new set of PGA and SA values from 80% randomly selected records were used. For the analysis, only one component (H1 or H2) from each record is considered; both 1HL and 2HL models are employed for the training. As the results obtained for the H1 and H2 components from the records exhibit similar trend and the results for inslab and for interplate records are similar as well, we only illustrate the estimated MSE in Figure 4 for the H1 component from the inslab records.
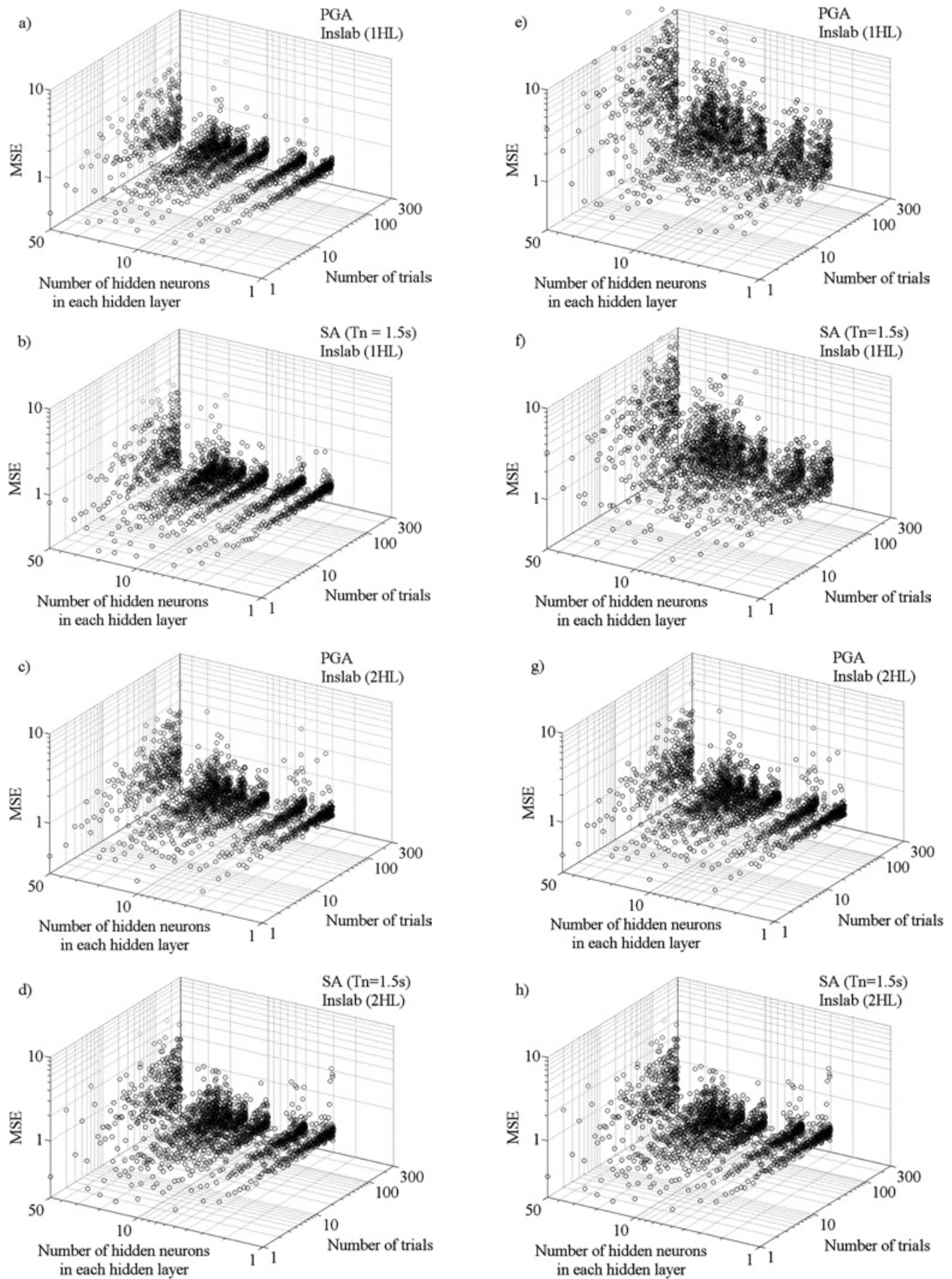
Figure 4. Variation of MSE with the number of trials and the number of neurons per hidden layer for PGA and SA (Tn = 1.5s) for inslab earthquakes: a) to d) training process; e) to h) validation process.

Figure 4, which presents the results for PGA and SA at T_n = 1.5 s, shows scatter in the MSE. As expected, the scatter (but not necessarily the standard deviation) of the MSE increases as the number of trials increases. The average MSE estimated based on 300 trials, which is considered to be sufficient large, is illustrated in Figure 5 for the training stage. The results presented in this figure indicate that the lowest average MSE is obtained for n_n around 10, and that the average MSE is relatively consistent for the number of hidden neurons, n_n, up to 25. As the average MSE for n_n = 50 is much larger than that for n_n < 50, the use of n_n > 25 is not recommended.

By using the remaining 20% records to test the trained ANN models, the average of MSE for 300 trials was also calculated and presented in Figure 6. Comparison of Figures 5 and 6 indicates that average values of MSE for the trained ANN models shown in Figure 6 are slightly greater than those presented in Figure 5. This can be explained by noting that the trained ANN models are tested with input parameters that are different than those used during the training process. Figure 6 also shows that the use of the model with n_n = 50 leads to the greatest average MSE among the considered n_n values.

During the analysis, it was observed that the optimum number of neurons and hidden layers – those leading to the lowest MSE for the trained model – depend on the selected records. In all cases, the optimum number of neurons is within 3 to 20; in about 50% of time the 1HL model outperforms the 2HL model, and vice versa. To further inspect the differences of using the 1HL and 2HL ANN models, the mean of the ratio of the MSE of the trained 1HL model to that of the trained 2HL model shown in Figure 4 was calculated. The values are presented in Table 4. The table indicates again that there is no clear preference among the 1HL and 2HL models, although the 1HL model for inslab earthquakes may be considered to perform better than the 2HL model. Based on these observations, the use of 10 neurons and the ANN model with 1HL and with 2HL will be considered in the next section.

*Comparison of predictions using trained ANN*

The training of the ANN models with 1HL and with 2HL was carried out by considering 10 neurons in each layer. For the analysis, the use of all H1 components and all H2 components are considered. As the results based on H1 or H2 components are almost the same, only the results for H1 are presented. Also, analysis was carried out by using only the geometric mean since this quantity is commonly used to develop GMPEs. For this case, the obtained weights and biases for the trained models are presented in the Appendix.

A comparison of the predicted PGA and SA by using the trained models to those obtained from the actual records is shown in Figures 7 and 8 for the H1 components and the geometric mean, respectively. It can be observed from the figures that there is a good agreement between the predicted and observed values, and that the correlation coefficient, ρ, is greater than 0.77 in all cases. The trained ANN models for inslab

Table 4. Comparison of the mean of the ratio of the MSE of ANN models with 1HL to that with 2HL by considering the H1 components.

Earthquake Type	Number of neurons	PGA	SA (T_n=0.2s)	SA (T_n=0.5s)	SA (T_n=1.0s)	SA (T_n=1.5s)
Inslab	3	0.88	0.86	0.88	0.88	0.91
	5	1.12	1.19	1.19	1.19	1.15
	10	1.00	1.00	1.06	1.10	1.09
	15	0.98	1.06	1.11	1.05	1.08
	20	0.86	0.93	0.95	0.91	0.93
	25	0.86	0.83	0.77	0.84	0.87
	50	0.54	0.50	0.48	0.50	0.45
Interplate	3	0.91	0.93	0.98	0.95	0.93
	5	1.15	1.11	1.07	1.11	1.11
	10	1.02	1.03	1.09	1.08	1.05
	15	1.05	1.08	0.98	1.03	1.11
	20	0.98	1.00	1.04	1.02	1.10
	25	1.01	0.99	0.97	0.88	0.94
	50	0.47	0.48	0.54	0.47	0.49
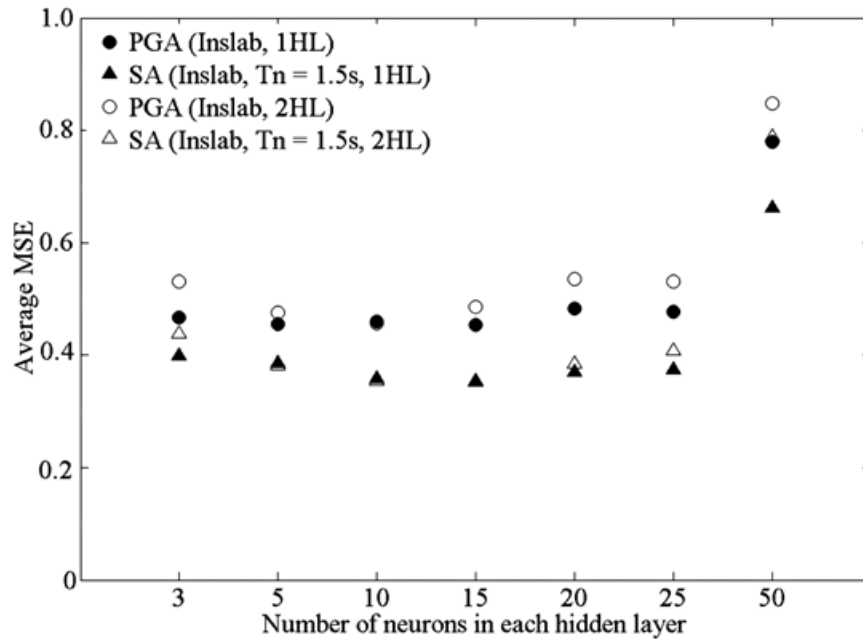
Figure 5. Average mean square error for the trained ANN models by using the samples employed for training.
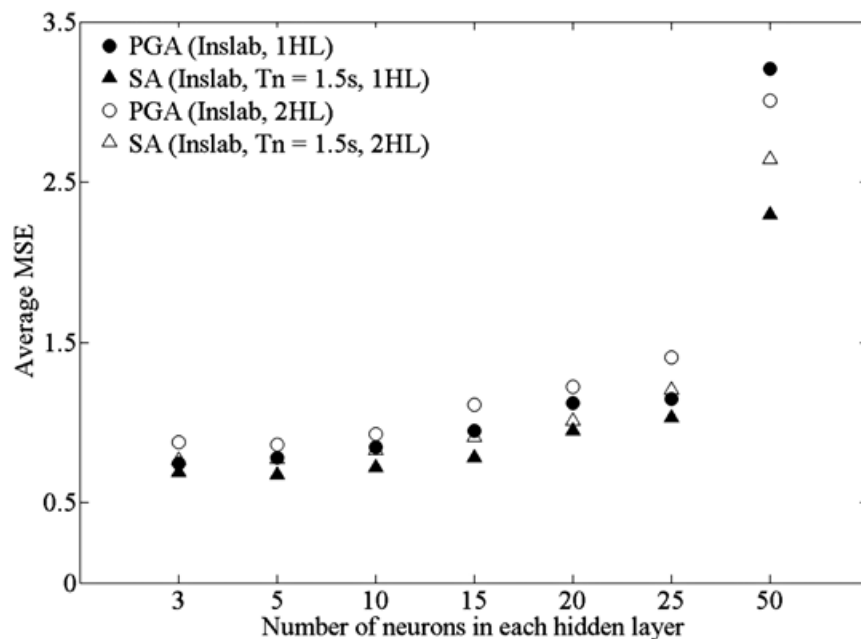
Figure 6. Average mean square error for the trained ANN models by using the samples that are not employed for training.

earthquakes provide better estimates than those for interplate earthquakes if H1 is considered; if the geometric mean is considered, both ANN models provide similar estimates. The scatter shown in the figures appears to be independent of the logarithm of PGA or logarithm of SA. A more detailed statistical investigation considering that the residual – η defined as the difference between the logarithmic of the actual PGA (or SA) and the logarithmic of the predicted PGA (or SA) – as a function of the predicted PGA (or SA) is beyond the scope of this study.

To provide a probabilistic characterization of the residual, η is shown in Figure 9 for a few selected cases presented in Figures 7 and 8. Inspection of the plots and use of a Kolmogorov-Smirnov test (Benjamin and Cornell, 1970) indicate that η can be modeled as a normal variable. The mean and the standard deviation of η for the cases presented in Figure 9 are summarized in Table 5, where the statistics of η, shown for the geometric mean case, were calculated by taking into account that the trained model will be used to predict ground motion measures for a random orientation, rather than the geometric mean (see Section 3).
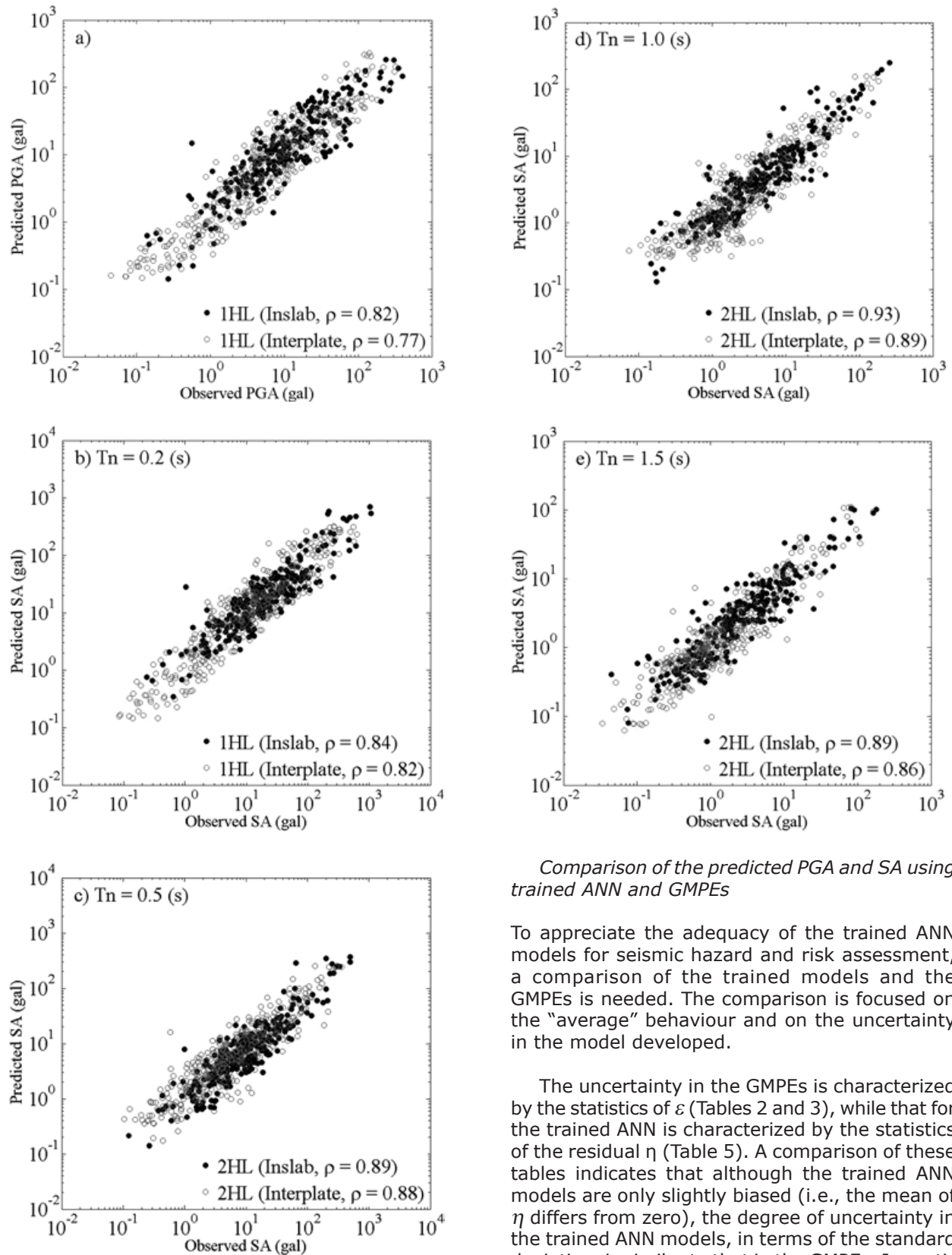
Figure 7. Comparison of predicted and observed PGA and SA H1 component values for inslab and interplate earthquakes (ρ in the plots denotes correlation coefficient).

Comparison of the predicted PGA and SA using trained ANN and GMPEs

To appreciate the adequacy of the trained ANN models for seismic hazard and risk assessment, a comparison of the trained models and the GMPEs is needed. The comparison is focused on the "average" behaviour and on the uncertainty in the model developed.

The uncertainty in the GMPEs is characterized by the statistics of ε (Tables 2 and 3), while that for the trained ANN is characterized by the statistics of the residual η (Table 5). A comparison of these tables indicates that although the trained ANN models are only slightly biased (i.e., the mean of η differs from zero), the degree of uncertainty in the trained ANN models, in terms of the standard deviation, is similar to that in the GMPEs. In particular, the statistics of η and ε are very consistent for the case when the geometric mean was used to develop the GMPEs and to train the ANN models. Since η and ε shown in Eqs. (3) and (4) represent
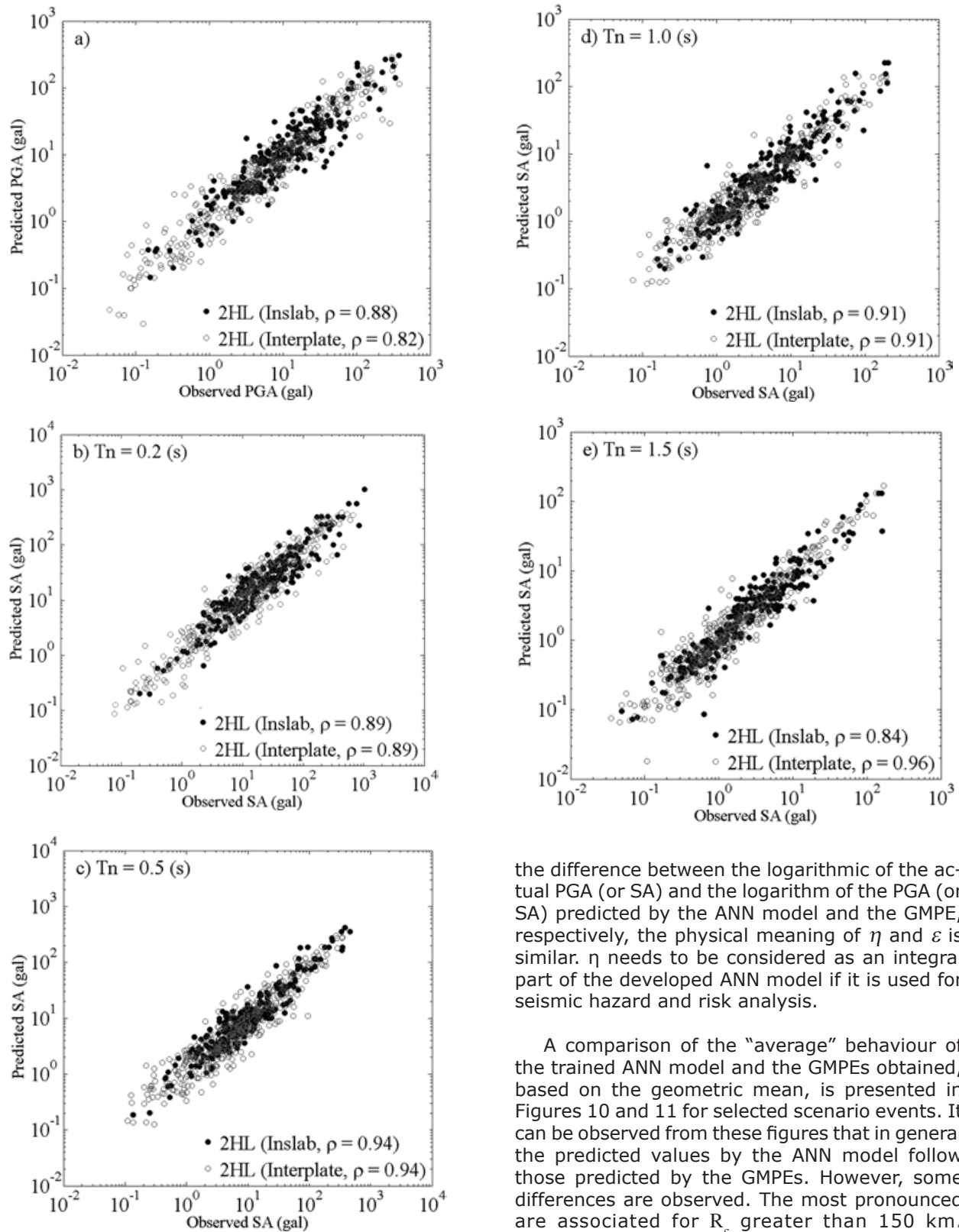
Figure 8. Comparison of predicted and observed geometric mean of PGA and SA for inslab and interplate earthquakes (ρ in the plots denotes correlation coefficient).

the difference between the logarithmic of the actual PGA (or SA) and the logarithm of the PGA (or SA) predicted by the ANN model and the GMPE, respectively, the physical meaning of η and ε is similar. η needs to be considered as an integral part of the developed ANN model if it is used for seismic hazard and risk analysis.

A comparison of the "average" behaviour of the trained ANN model and the GMPEs obtained, based on the geometric mean, is presented in Figures 10 and 11 for selected scenario events. It can be observed from these figures that in general the predicted values by the ANN model follow those predicted by the GMPEs. However, some differences are observed. The most pronounced are associated for R_c greater than 150 km. Moreover, in some cases, the predicted values by the trained ANN models may not necessarily reflect reality. For example, the results shown in Figure 10c indicate that SA can increase with distance beyond R_c = 200 km for M_w = 5.9 and H = 50 km, which is unrealistic. Since this drawback
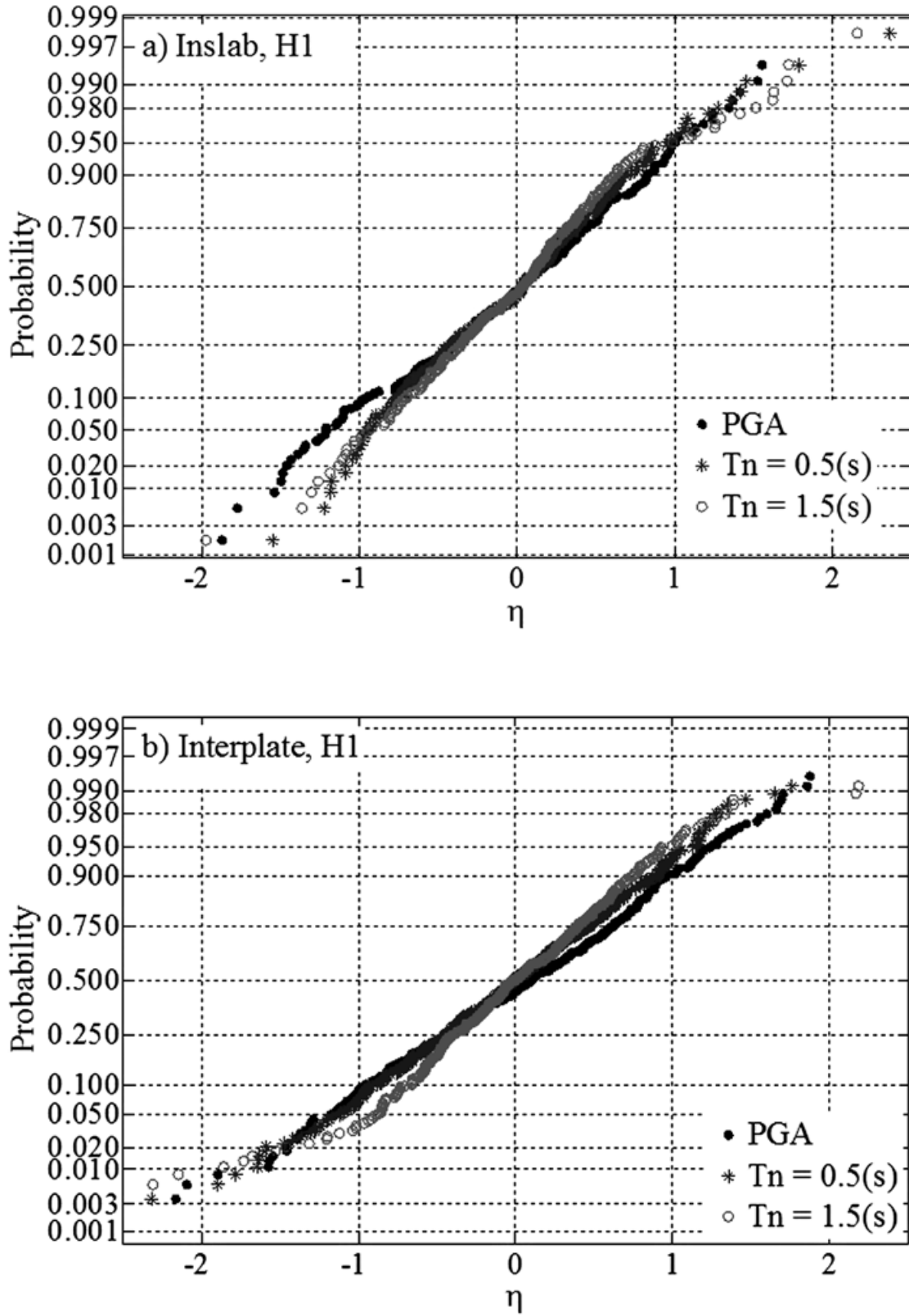
Figure 9. Normal probability plots of the residual η.
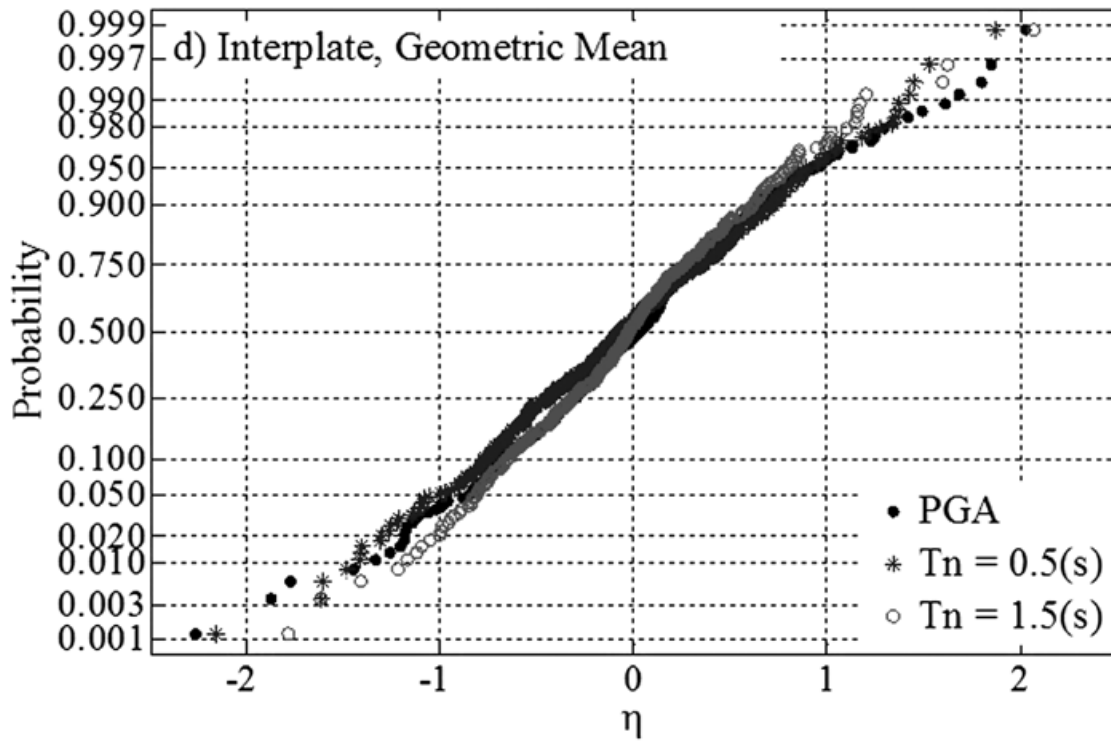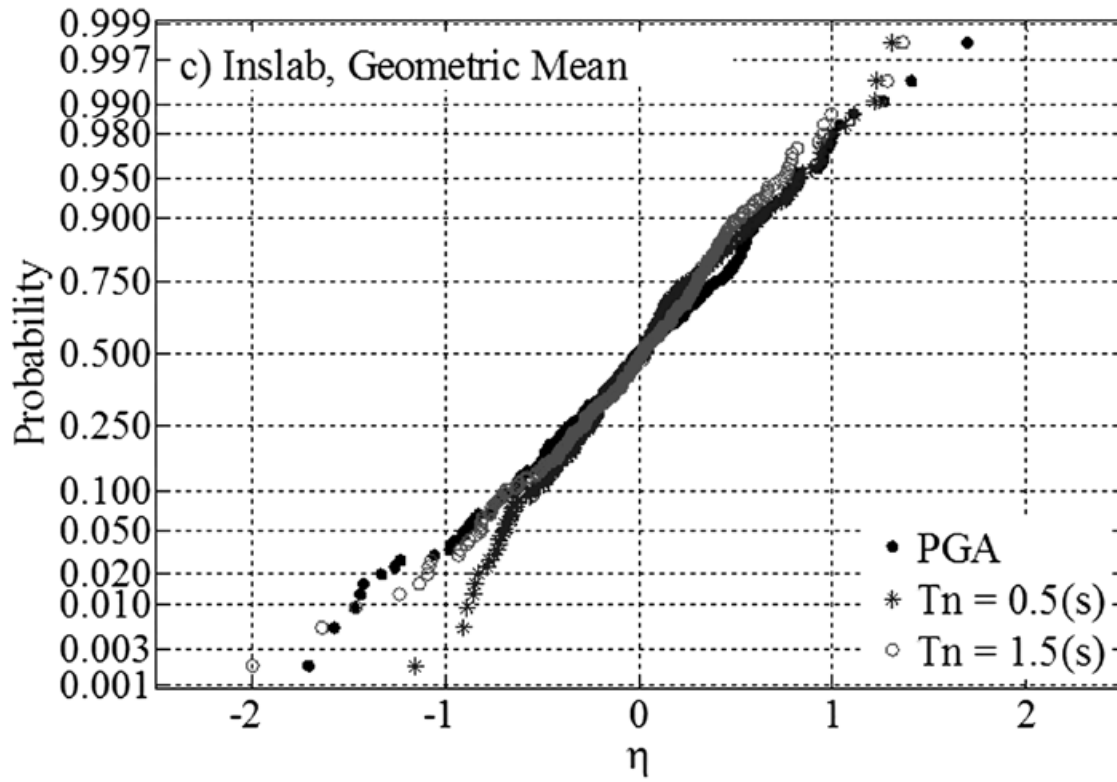
Figure 9. Continued.

Table 5. Statistics of η for cases shown in Figure 10.

Earthquake Type		H1		Geometric Mean	
		mean	Std. Dev.	mean	Std. Dev.
Inslab	PGA	1.5E-01	0.68	9.2E-03	0.35
	SA (Tn=0.5s)	-3.0E-01	0.58	-7.3E-03	0.32
	SA (Tn=1.5s)	4.9E-02	0.57	5.0E-04	0.33
Interplate	PGA	-1.4E-01	0.67	7.3E-03	0.38
	SA (Tn=0.5s)	-5.5E-03	0.79	-1.2E-02	0.43
	SA (Tn=1.5s)	-6.6E-02	0.72	7.0E-03	0.40
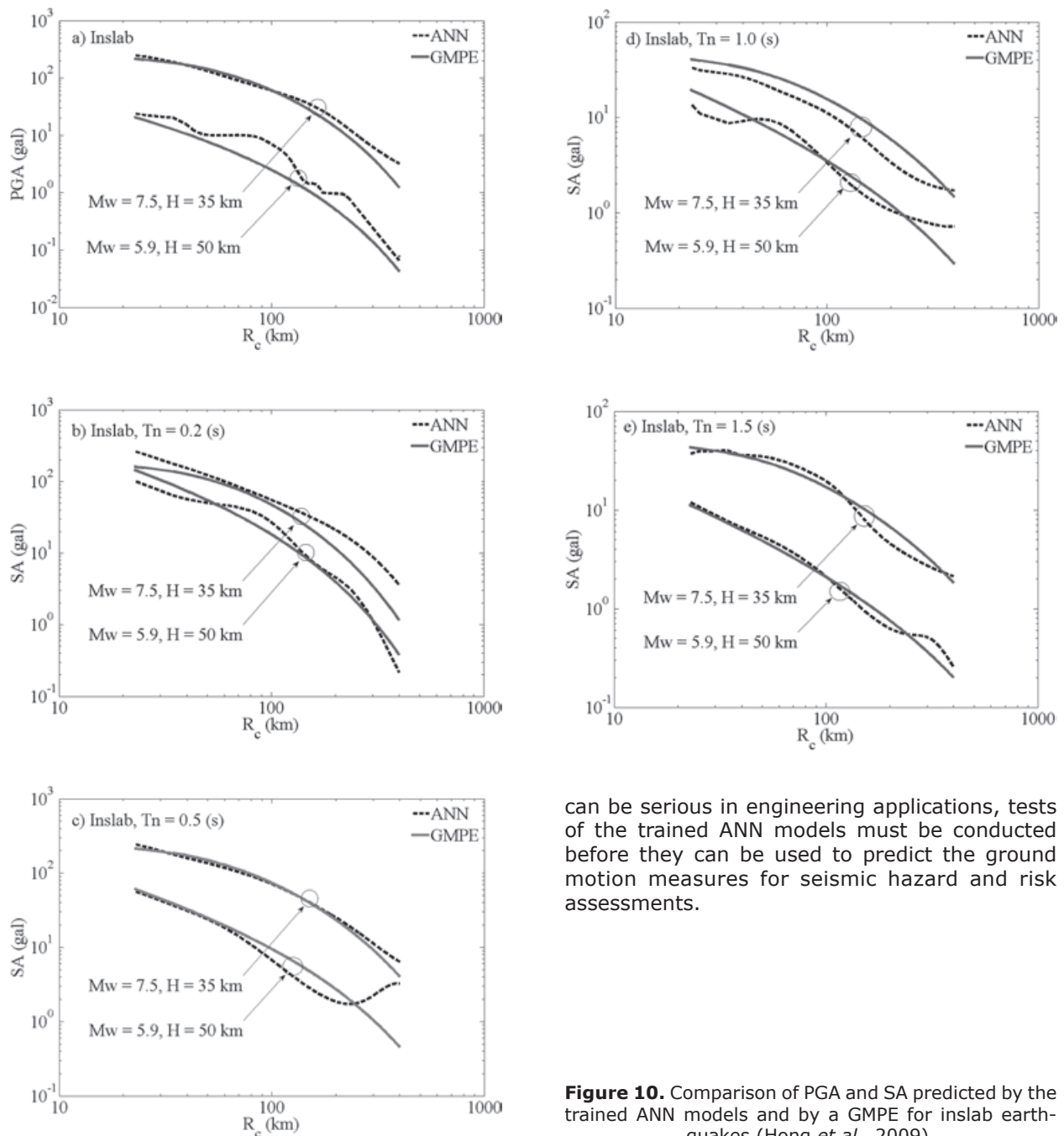
can be serious in engineering applications, tests of the trained ANN models must be conducted before they can be used to predict the ground motion measures for seismic hazard and risk assessments.

Figure 10. Comparison of PGA and SA predicted by the trained ANN models and by a GMPE for inslab earthquakes (Hong *et al.*, 2009).
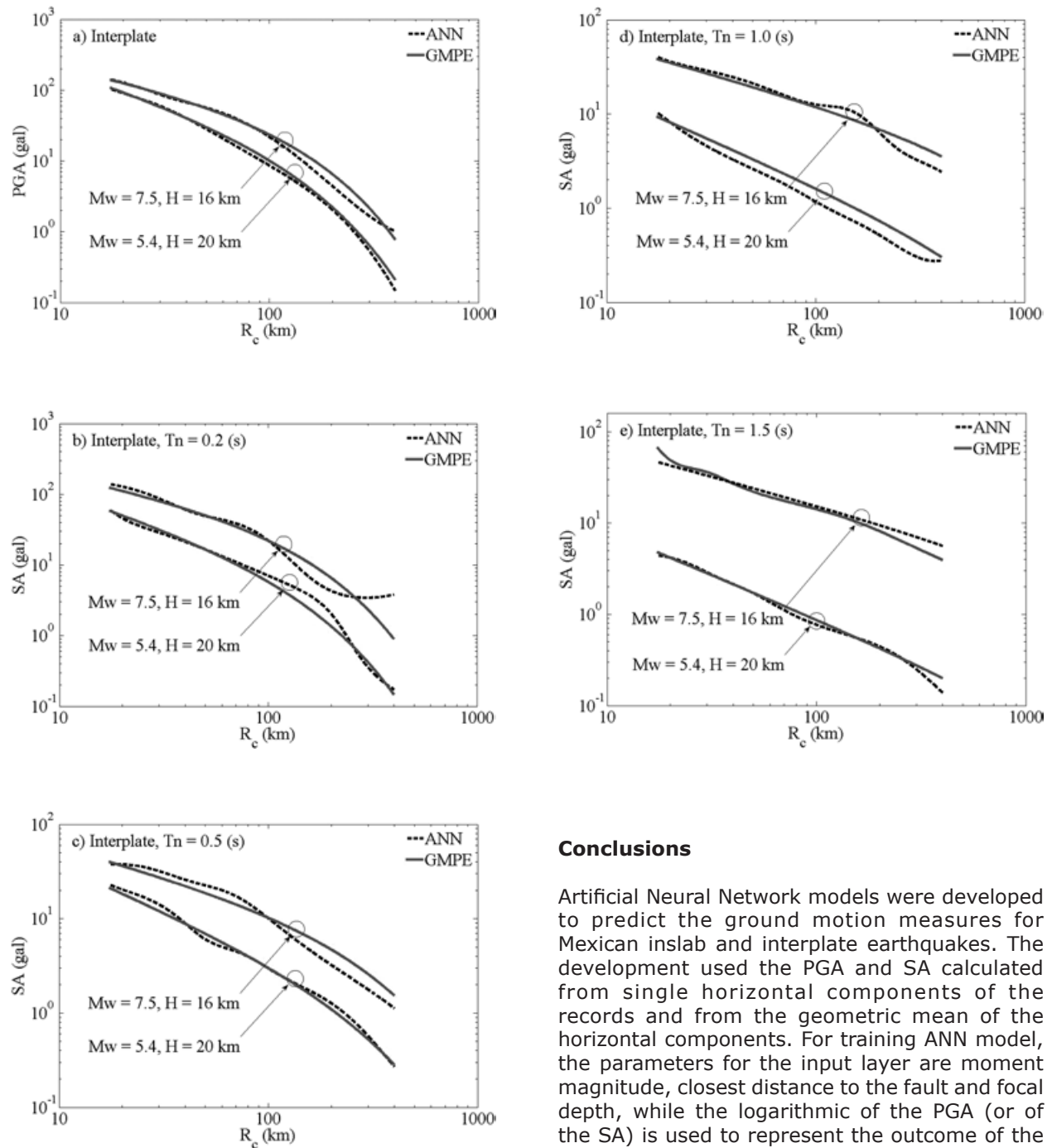
Figure 11. Comparison of PGA and SA predicted by the trained ANN models and by a GMPE for interplate earthquakes (Hong *et al*., 2009).

Conclusions

Artificial Neural Network models were developed to predict the ground motion measures for Mexican inslab and interplate earthquakes. The development used the PGA and SA calculated from single horizontal components of the records and from the geometric mean of the horizontal components. For training ANN model, the parameters for the input layer are moment magnitude, closest distance to the fault and focal depth, while the logarithmic of the PGA (or of the SA) is used to represent the outcome of the model. The main observations that can be drawn from the analysis results are:

1. The performance of the trained ANN model by using a single hidden layer is similar to that by using two hidden layers. The most appropriate number of neurons per hidden layer seems to be within 3 to 20.

2. The use of a single horizontal component or the geometric mean of the two horizontal components leads to similar trained ANN models, implying that the number of considered records is adequate.

3. The statistics of the residuals associated with the trained ANN models are similar to those associated with the GMPEs. The ground motion measures predicted by the trained ANN models follow those predicted by the GMPEs. This indicates that the ANN models may be a good alternative to GMPEs in some applications.

4. In some cases, the SA predicted by the trained ANN models increases as the R_c increases. This does not reflect the general behaviour observed from actual records. Therefore, extensive verification of the trained ANN models should be carried out before the models can be used for seismic engineering applications.

Acknowledgments

Bibliography

Atkinson G.M., Boore D.M., 2003, Empirical ground-motion relations for subduction-zone earthquakes and their application to Cascadia and other regions. *Bull. Seism. Soc. Am*., 93, 1703-1729.

BSSC, 2004, NEHRP Recommended provisions for seismic regulations for new buildings and other structures (FEMA 450), 2003 Edition.

Benjamin J.R., Cornell C.A., 1970, Probability, Statistics and Decision for Civil Engineers. McGraw-Hill, New York.

Fausset L.V., 1994, Fundamentals of neural networks: Architecture, algorithms, and applications. Prentice-Hall, Englewood Cliffs, N.J., 461 pp.

García D., Singh S.K., Herraiz M., Ordaz M., Pacheco J.F., 2005, Inslab earthquakes of Central Mexico: Peak ground-motion parameters and response spectra, *Bull. Seism. Soc. Am.*, 95, 2272-2282.

García D., 2006, Estimación de parámetros del movimiento fuerte del suelo para terremotos interplaca e intraslab en México Central, PhD Tesis, Universidad Complutense de Madrid, Madrid, España. (In Spanish).

García D., Singh S.K., Herraiz M., Ordaz M., Pacheco J.F., 2009, Erratum to Inslab Earthquakes of Central Mexico: Peak ground-motion parameters and response spectra, *Bull. Seism. Soc. Am.*, 99, 2607-2609.

García S.R., Romo M.P., Mayoral J.M., 2007, Estimation of peak ground accelerations for Mexican subduction zone earthquakes using neural networks. *Geofisica Internacional*, 46, 1, 51–63.

Ghaboussi J., Li C.C.J., 1998, New method of generating spectrum compatible accelerograms using neural networks. *Earthhquake Engineering and Structural Dynamics*, 27, 4, 377–396.

Günaydn K., Günaydn A., 2008, Peak ground acceleration prediction by artificial neural networks for northwestern Turkey. *Mathematical Problems in Engineering*, Art. No.: 919420.

Hagan M.T., Menhaj M., 1994, Training feed-forward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks 5, 6, 989-993.

Hong H.P., Pozos-Estrada A., Gómez R., 2009, Orientation effect on ground motion measure for Mexican subduction earthquakes, *Earthquake Engineering and Engineering Vibration*, 8, 1, 1-16.

Hong H.P., Liu T.J., Lee C.S., 2012, Observation on the application of artificial neural network for predicting ground motion measures. *Earthq Sci*., 25, 161–175.

Joyner W.B., Boore D.M., 1993, Methods for regression analysis of strong-motion data. *Bull. Seism. Soc. Am*., 83, 469–487.

Kamatchi P., Rajasankar J., Ramana G.V., Nagpal A.K., 2010, A neural network based methodology to predict site-specific spectral acceleration values. *Earthq. Eng. & Eng. Vib*., 9, 4, 459–472.

Lee S.C., Han S.W., 2002, Neural-network-based models for generating artificial earthquakes and response spectra. *Computers and Structures*, 80, 1627–1638.

Marquardt D.W., 1963, An algorithm for least squares estimation of non-linear parameters. *Journal of the Society for Industrial and Applied Mathematics,* 11, 2, 431-41.

Press W.H., Teukolsky S.A., Vetterling A.W.T., Flannery B.P., 1992, Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, New York.

Shahin M., Maier H.R., Jaksa M.B., 2004, Data division for developing neural networks applied to geotechnical engineering. *Journal of Computing in Civil Engineering*, 18, 2, 105–114.

Appendix

Weights and biases for the trained models.

The coefficients of the trained models needed for their application are given in the following tables (next page).

Table A1. Trained ANN based models based on the geometric mean for inslab earthquakes.

Column structure: Ground motion measure | Weights $[w_1]_{i,j}$ | Biases $[w_2]_{j,k}$ | $[w_3]_{k,l}^{\mathsf{T}}$ | $(\varphi_1)_j$ | $(\varphi_2)_k$ | $(\varphi_3)_l$

PGA — $(\varphi_3)_l = [-0.906]$

$[w_1]$ i=1	i=2	i=3	$[w_2]_{j,k}$ (k = 1 … 10)										$[w_3]^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$
-2.821	-0.442	-0.326	-0.112	-0.867	0.572	0.620	0.283	0.919	0.792	-0.849	0.711	0.278	-0.189	3.127	1.616
1.550	0.298	-1.117	0.297	0.445	0.669	0.410	0.114	0.233	0.574	0.552	-0.086	0.853	-1.126	-3.449	-1.249
-0.403	1.062	-2.946	0.196	-0.449	-0.720	-0.513	-0.545	1.273	-1.493	-0.155	-0.530	0.199	-0.542	1.883	-0.664
1.393	1.396	-3.542	-0.534	0.622	-0.601	-0.773	-0.120	0.095	0.041	0.115	1.209	-0.811	-0.342	-0.255	0.404
-0.321	-4.174	-1.435	0.414	-0.334	1.165	-0.643	0.135	0.005	0.363	0.784	-0.164	0.710	0.118	-0.072	-0.023
-2.260	1.522	-2.953	-0.155	0.047	-0.122	0.726	1.037	0.133	-0.775	-0.347	-0.517	-0.766	0.197	-0.792	-0.239
-1.717	1.224	-1.047	-0.645	0.207	1.085	-1.037	0.083	-0.628	-0.080	0.536	1.290	-0.310	-1.009	-1.157	-0.421
1.661	-2.410	1.250	0.271	-1.032	-0.398	-1.218	-0.722	1.412	0.024	-0.584	-0.561	0.220	-0.193	2.001	0.955
-1.502	1.250	2.445	-0.612	-0.534	0.038	-0.634	-1.105	0.132	1.154	0.034	-0.083	-1.051	0.145	-2.461	-1.871
1.450	0.044	-2.334	-0.778	-0.121	0.105	0.710	0.433	1.376	0.152	0.050	0.278	0.082	-0.510	3.317	-1.856

SA($T_n = 0.2$s) — $(\varphi_3)_l = [-0.199]$

$[w_1]$ i=1	i=2	i=3	$[w_2]_{j,k}$ (k = 1 … 10)										$[w_3]^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$
-1.224	-1.508	0.791	1.272	0.610	-0.819	1.773	-2.535	-0.731	-0.917	-0.996	2.389	0.103	-0.061	2.794	-2.175
1.951	-1.116	-2.350	0.236	1.020	-0.231	-0.724	-1.369	3.030	-1.324	-0.322	-0.830	0.785	0.085	-1.198	-1.302
4.020	-2.741	1.574	0.463	0.362	-0.056	-1.236	-1.865	0.106	-0.484	-1.132	-1.934	0.755	-0.591	-1.397	0.714
-2.343	-1.005	0.470	-1.295	0.058	0.053	0.474	1.656	-0.033	-0.398	0.119	-0.025	-0.180	-0.510	-0.365	0.112
3.942	2.336	-1.447	-0.251	-2.146	0.704	0.288	2.652	0.022	-1.594	1.170	2.310	0.425	-0.001	0.147	0.332
-3.192	2.519	1.096	-1.065	0.577	-0.202	1.599	-0.456	-0.566	-1.019	-1.898	-0.221	1.603	0.859	-0.254	0.641
0.433	-1.072	1.655	-0.616	0.653	0.453	-0.645	-0.373	-0.649	-0.457	-0.812	-0.220	-0.317	0.493	-1.531	0.871
-2.459	-0.142	-2.169	-0.043	-0.732	-1.272	0.659	1.252	0.283	-1.198	-1.200	1.408	0.676	-0.156	-3.569	-0.871
2.804	1.662	0.401	-0.472	-0.556	-2.625	0.135	0.843	-0.558	-2.408	-0.108	-1.570	0.210	-0.158	2.171	-0.777
1.559	2.303	-1.665	-0.244	0.549	0.812	-0.326	-0.053	-0.414	0.005	-0.640	1.300	0.303	0.295	2.996	-1.797

SA($T_n = 0.5$s) — $(\varphi_3)_l = [\,0.075]$

$[w_1]$ i=1	i=2	i=3	$[w_2]_{j,k}$ (k = 1 … 10)										$[w_3]^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$
3.467	-3.972	0.384	-0.206	-1.082	2.408	-0.953	0.304	1.036	0.258	-1.810	3.089	1.195	0.222	-1.079	2.428
-1.492	0.465	3.979	0.953	-0.245	-1.262	0.652	-2.087	1.062	0.563	1.313	0.942	1.750	-0.117	-2.069	-1.890
1.757	-5.789	0.310	2.095	-0.391	-0.762	0.744	-1.352	-0.357	1.739	-1.721	1.621	0.456	0.432	-1.501	0.877
-1.875	-0.108	0.604	0.261	0.106	-0.337	-0.307	-1.157	0.071	0.258	0.439	2.123	1.449	-0.039	1.690	0.898
2.382	-0.296	0.087	-1.796	0.253	-1.052	-0.588	-1.763	1.693	0.867	-0.581	-2.418	0.063	-0.176	2.796	-0.313
5.506	1.845	-1.531	1.328	-1.285	1.094	0.564	-2.257	-1.572	-0.470	-1.763	-0.554	-1.288	-0.189	3.670	-0.795
-1.144	0.552	-1.498	1.455	0.011	0.253	-1.988	1.423	-0.080	1.055	-0.173	-0.767	-0.251	0.196	-0.639	0.368
-2.236	1.740	-0.007	1.531	-0.363	0.742	0.393	2.778	-1.354	0.636	1.310	0.074	-1.887	-0.127	-1.938	1.688
0.768	-1.121	2.577	-0.932	-1.853	0.528	1.457	1.450	-4.191	-0.193	2.582	1.314	3.679	-0.079	0.477	2.703
1.864	-2.560	-1.610	-1.452	-0.830	-1.892	1.849	-2.418	-0.971	-1.559	-1.284	-1.208	0.168	0.176	-3.354	-2.165

SA($T_n = 1.0$s) — $(\varphi_3)_l = [-0.029]$

$[w_1]$ i=1	i=2	i=3	$[w_2]_{j,k}$ (k = 1 … 10)										$[w_3]^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$
-0.072	1.100	-3.185	0.192	-0.368	0.793	0.222	-0.391	-0.113	-0.846	-0.298	0.922	-0.225	0.186	-3.056	-1.958
1.722	-2.105	1.244	0.199	0.096	-0.648	-0.114	-0.647	0.508	0.420	-0.828	1.301	-1.145	0.832	-1.994	-1.507
0.463	-2.734	-1.250	0.351	-0.260	-0.344	0.900	0.701	-0.977	-0.188	-0.875	0.422	0.772	0.647	-1.652	0.767
-0.575	-2.223	-1.059	1.375	0.481	0.359	1.050	0.291	1.264	0.628	-0.138	0.878	-0.341	-0.838	1.738	-0.732
2.771	1.275	1.707	0.801	-0.592	-0.884	0.827	-0.832	-0.148	0.378	-0.659	-0.239	0.189	-0.274	-0.569	-0.336
-1.460	-1.329	-1.897	0.128	0.619	0.224	1.281	-0.580	-0.660	0.984	0.388	0.210	0.092	0.589	-0.424	-0.345
-0.922	-1.371	-2.621	1.279	-0.374	-1.256	-0.235	0.431	-0.354	0.068	0.419	-0.105	-0.052	-0.594	-0.781	0.719
0.863	1.517	2.682	0.358	-0.386	-0.373	-0.504	0.467	0.455	-0.099	-0.835	-0.597	-0.499	-0.501	1.675	1.273
0.678	-0.974	-2.366	0.583	0.375	-0.518	-0.721	-0.030	0.314	-0.983	0.977	0.229	-0.237	0.548	-2.271	1.369
-1.375	-2.302	-1.508	0.494	-0.378	-0.821	1.427	0.277	0.756	-0.041	0.292	-0.758	0.933	0.816	-2.991	1.698

SA($T_n = 1.5$s) — $(\varphi_3)_l = [\,0.184]$

$[w_1]$ i=1	i=2	i=3	$[w_2]_{j,k}$ (k = 1 … 10)										$[w_3]^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$
-0.733	-1.391	-2.628	-1.047	0.382	0.187	0.414	-0.091	0.164	-0.176	0.514	-1.575	-0.534	-1.394	3.025	1.626
1.561	1.020	2.597	0.690	0.700	0.625	0.042	-0.614	1.350	-0.263	0.791	-0.523	-0.629	0.371	-2.042	-1.396
1.118	1.785	1.810	-0.701	-0.393	1.271	-0.471	-0.860	1.176	0.642	0.397	-0.153	0.022	-0.938	-2.013	0.641
-0.859	-0.971	0.828	0.114	-0.237	-0.295	-0.044	0.376	-1.404	0.058	-0.486	0.828	0.933	-1.268	-0.945	-0.656
-2.939	1.481	2.526	-0.735	-0.719	0.457	0.411	-0.403	-1.238	-0.450	-1.205	0.612	0.578	0.844	0.820	0.375
-2.563	-2.328	1.668	-0.809	-0.708	-0.158	0.468	-0.889	-0.147	0.776	0.108	0.765	0.975	1.492	-0.304	-0.060
2.670	3.101	-0.386	-0.750	-0.870	0.006	-0.101	-0.803	1.064	0.304	-0.632	0.369	-0.457	-0.958	0.060	-0.427
-1.206	2.729	0.984	0.724	0.438	0.350	-1.811	-1.264	2.281	-0.083	-1.224	-0.832	-0.867	0.577	-2.518	0.973
2.467	2.431	-0.890	-0.412	1.044	1.244	0.008	0.244	1.134	1.245	-1.587	0.607	-0.219	1.300	2.370	1.403
1.280	1.701	-1.879	0.189	-0.255	0.759	0.182	-1.099	0.762	0.467	0.037	-0.553	0.597	-0.078	3.204	1.650

Table A2. Trained ANN based prediction models based on the geometric mean for interplate earthquakes.

Ground motion measure	Weights $[w_1]_{i,j}$	Biases $[w_2]_{j,k}$	$[w_3]_{k,1}^{\mathsf{T}}$	$(\varphi_1)_j$	$(\varphi_2)_k$	$(\varphi_3)_1$

PGA

$[w_1]_{i,j}$:
```
 1.791  -0.869  -2.141
 0.365   1.294  -2.107
 4.105   4.119  -1.505
 2.162   3.641   0.440
 0.404  -2.190  -1.070
-0.559   0.745  -1.400
-4.120   0.402   2.188
 6.272  -4.973   1.308
 1.631   1.394   0.943
-0.986   2.693  -0.063
```

$[w_2]_{j,k}$:
```
 0.094   1.226   1.961   0.514  -1.157  -0.186   0.045   0.184  -0.419   1.498
 0.280   1.264   0.123   0.035  -0.134  -1.150   1.045   0.256  -0.407   0.792
-0.291   0.373  -0.372   0.701   0.946  -0.213   0.908  -0.137  -0.928   1.499
 0.587  -0.377  -1.206   1.643   2.231  -1.531   0.819  -1.217   0.456  -0.463
-0.107   0.105   0.142  -1.171   1.519  -1.700   1.666  -1.779   1.991  -0.795
 0.573  -0.045  -0.487   1.221   0.754  -0.583  -0.394   0.278  -0.347   0.653
-1.336  -0.120  -1.190   1.629   0.409  -0.562   2.130   0.668   2.194  -0.748
 0.485   0.382  -1.370  -1.015  -1.673   1.279   0.495   4.568   1.189   0.946
 0.847   0.869  -1.635  -0.626  -2.092   0.774  -0.460  -0.089  -0.003   0.531
-0.382  -0.184  -0.378  -0.032   0.318   1.078  -0.717  -0.428   1.293   0.278
```

$[w_3]_{k,1}^{\mathsf{T}}$: `-0.425  0.236  -0.346  0.104  -0.106  0.428  -0.106  0.011  -0.180  -0.681`

$(\varphi_1)_j$: `-1.840  -2.731  -2.027  -2.539  1.652  -0.079  -3.293  3.875  0.891  -3.448`

$(\varphi_2)_k$: `1.350  -2.033  1.645  -0.498  0.449  0.110  -0.210  -1.539  1.352  2.543`

$(\varphi_3)_1$: `[ 0.827 ]`

SA ($T_n = 0.2$s)

$[w_1]_{i,j}$:
```
 1.488  -1.858  -0.906
-2.964   2.267   0.711
 2.516  -1.195  -1.058
-3.650   0.815  -2.497
 0.422   1.359  -0.917
-2.388  -0.526   0.159
 0.881  -0.630   0.064
-1.414  -0.609   2.831
 1.535   2.322  -2.874
-1.599   2.782  -0.566
```

$[w_2]_{j,k}$:
```
 1.066  -0.661  -0.541  -1.642   2.877  -0.178   3.677  -0.336  -0.622  -1.551
 1.554  -0.903  -0.312   1.177  -0.893   0.487  -0.674   2.919   2.611  -1.528
 0.533  -0.484   1.177   1.338  -0.577   0.644  -0.964   0.523  -0.365   0.523
 1.958   1.327   1.338   0.536  -0.688  -1.000  -3.058   1.259  -0.222  -0.899
 1.723   0.152  -0.933  -0.007   1.019  -0.071   3.421   0.794  -0.927   1.201
-1.722   1.002  -1.037   0.394   1.241  -0.376   2.911  -0.449   0.941  -1.324
-0.860   0.244  -0.625  -0.565  -1.250   2.973  -0.311  -0.320   0.028   0.534
 1.599  -1.381   2.239  -0.651  -1.386   0.594   0.884  -0.108  -0.229  -0.992
-1.318   0.250   2.095  -2.196   1.318   0.052   0.843  -0.307   0.532  -0.746
-1.756   0.061  -0.757  -0.145   0.149  -1.133  -3.209   0.532  -0.307  -0.746
```

$[w_3]_{k,1}^{\mathsf{T}}$: `1.456  -0.145  0.493  0.181  -0.281  1.397  0.183  1.098  -1.361  -0.330`

$(\varphi_1)_j$: `-3.599  2.751  -1.521  0.007  -0.211  -1.048  0.648  -1.048  4.065  0.688`

$(\varphi_2)_k$: `1.535  -1.139  -1.166  -0.048  -0.971  1.827  1.244  -2.517  0.500  -0.198`

$(\varphi_3)_1$: `[ 0.252 ]`

SA ($T_n = 0.5$s)

$[w_1]_{i,j}$:
```
-2.590  -1.334   0.359
 1.683  -2.001  -1.997
-1.286   1.843   2.949
 1.054   3.509  -0.110
 0.251  -1.177   2.782
 1.339  -2.444  -2.715
-1.525   1.977   2.194
 2.903  -0.077  -2.836
-0.896   2.287   1.382
 2.242  -0.719   1.493
```

$[w_2]_{j,k}$:
```
 1.120   0.372  -1.141   0.501   0.843  -1.303   0.636   0.418  -1.094  -0.196
 0.496   0.371   0.076  -1.258  -0.725   1.290   0.601   1.407   1.132  -0.126
-0.548   0.163  -1.305   0.309  -1.289   0.949   0.605   0.776  -0.677  -0.986
 0.736   0.163  -0.691  -0.065  -1.163  -0.978  -1.012   0.173   0.411   1.463
-0.920   0.464   0.616   0.509   0.653   0.261  -1.092  -0.174  -0.155  -0.096
-2.365  -0.199   0.267  -1.724   0.173  -1.069   0.828   0.598   0.227   0.870
-0.110  -1.286  -0.035   0.533  -0.359   0.038   1.562   0.377  -0.907  -1.083
-0.423  -0.074   1.366   0.524   0.171  -1.359  -0.125  -0.296   0.072   0.225
-1.110   0.253  -0.935   0.249  -0.145   0.663   0.681  -0.719  -0.231   0.641
 0.599  -0.533   0.851  -0.217   0.680   0.949  -0.551  -1.376   0.439   0.297
```

$[w_3]_{k,1}^{\mathsf{T}}$: `0.589  0.051  0.173  0.449  0.412  0.149  -0.261  -0.004  0.025  0.514`

$(\varphi_1)_j$: `3.022  -2.070  1.421  -1.754  -0.312  -0.126  -1.038  1.866  -2.223  1.980`

$(\varphi_2)_k$: `-1.905  -1.259  1.144  -0.157  -0.365  0.152  -0.048  -0.957  -2.027  1.924`

$(\varphi_3)_1$: `[ 0.172 ]`

SA ($T_n = 1.0$s)

$[w_1]_{i,j}$:
```
 3.655   3.506   0.137
 3.371   4.949  -0.431
 0.868  -4.826  -2.522
 1.241  -0.876  -0.261
 0.386   4.467  -0.583
 0.394  -0.229   0.209
-4.093   1.291   3.585
 0.274  -2.868  -2.607
 2.700   3.344  -2.763
 3.372   0.236   1.217
```

$[w_2]_{j,k}$:
```
-1.482   0.862   2.061   0.453  -5.179  -2.552   2.095   0.475   1.271  -0.840  -1.432
 0.736   3.335   0.525   0.249  -0.781  -0.700   1.464  -0.039   2.426   1.417   2.211
-0.768   1.377   1.045  -0.519  -0.798  -0.135  -1.514   0.294  -0.464  -0.332  -0.539
-0.511   2.096  -3.683  -0.582  -0.383  -3.655  -2.650   1.188  -1.748   0.123  -0.715
 2.310  -1.332  -0.879   0.189  -0.742  -0.438  -1.749  -0.688   0.338   0.830  -0.454
-2.893   3.685  -4.077   1.285   0.616  -1.509  -1.884   0.679  -0.818   0.392
 0.445  -0.913   1.673   1.227   1.618  -3.604  -1.164  -2.019  -1.931  -1.765
 0.125   1.110   1.111  -0.145   0.528  -2.481   1.084   1.141   1.115  -1.714  -0.264
-0.821  -0.842  -0.006  -1.047   1.954  -2.364  -3.406   2.119  -1.170  -0.351   0.276
-2.626   0.883   2.018   1.120   0.837   0.032  -0.490  -1.535  -1.808  -0.646  -1.686
```

$[w_3]_{k,1}^{\mathsf{T}}$: `0.081  0.137  -0.424  -0.124  -0.184  0.090  0.081  0.163  -0.046  0.100`

$(\varphi_1)_j$: `-2.063  -2.433  -1.592  -0.598  -0.609  -0.146  -3.907  -1.872  4.147  3.378`

$(\varphi_2)_k$: `-3.125  0.684  0.170  1.057  -1.979  -1.578  -0.036  -1.445  -0.363  -2.380`

$(\varphi_3)_1$: `[ 0.180 ]`

SA ($T_n = 1.5$s)

$[w_1]_{i,j}$:
```
-0.904   3.866  -0.944
-0.922   0.699  -3.588
-1.699  -1.149  -3.006
-0.761   0.553   1.416
-0.599   1.575  -1.682
-1.283   1.987  -3.340
-1.442   3.198   1.463
-4.101  -0.683  -2.028
-3.111  -1.363   0.235
-2.997   0.019  -0.479
```

$[w_2]_{j,k}$:
```
 0.878   0.722  -0.211   0.453   1.480   0.445   0.172   3.350  -0.080   1.836
 1.251  -1.228   0.201   0.249   1.070   0.815   0.045  -2.285   0.083
-0.832  -2.388   1.032  -0.519   1.370  -0.252   0.797  -0.344  -0.496  -0.127
-0.318  -1.479   1.002  -0.582  -0.067  -0.274  -0.803   0.342  -0.289   0.347
 1.725   0.762   0.220   0.189   0.412   0.006  -1.199  -1.543   1.949   0.735
-0.864  -0.904  -0.104   1.285  -0.037   0.558   0.864   0.418  -0.640
 0.722  -1.866  -1.975   1.227   1.285  -0.106  -1.876  -0.612  -0.832   0.555
-0.115  -0.104   0.057  -0.145   0.864  -1.047   0.197   0.701  -0.237   0.513
-0.139  -1.564  -1.658   1.120  -0.906   0.128   0.573  -0.701   1.756   1.271
 1.792  -0.343   1.133   0.260   0.074   0.098  -0.495  -0.330   0.544  -0.744
```

$[w_3]_{k,1}^{\mathsf{T}}$: `-0.438  -0.390  -0.979  -1.764  -0.692  -1.486  -0.578  1.943  -0.898  0.998`

$(\varphi_1)_j$: `2.840  2.391  1.803  -1.779  -0.757  0.213  -1.583  -3.481  -3.157  -2.701`

$(\varphi_2)_k$: `-1.631  1.596  0.876  0.142  -0.209  0.128  1.118  -0.808  2.944  1.711`

$(\varphi_3)_1$: `[ -0.650 ]`