Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), *12*, 243–258. Printed in the USA.
Copyright © 1998 Cambridge University Press 0890-0604/98 $12.50

Modelling earthquake accelerograms using neural networks and linear predictors

IAN FLOOD,[1] NICOLAS GAGARIN,[2] AND NABIL KARTAM[3]

[1]M.E. Rinker, Sr. School of Building Construction, University of Florida, Gainesville, FL 32611, U.S.A.
[2]Office of Research and Development, Turner-Fairbank Highway Research Center, 6300 Georgetown Pike, McLean, VA 22101, U.S.A.
[3]Department of Civil Engineering, University of Kuwait, Safat, Kuwait.

(RECEIVED August 13, 1997; ACCEPTED December 16, 1997)

Abstract

The paper describes and evaluates three methods of modelling earthquake accelerograms using artificial neural networks and linear predictors. These are a radial-Gaussian neural networking system, a linear predictor, and a hybrid of these two. Two methods of using these models to predict ground acceleration are adopted. The first is based on a direct prediction of ground acceleration at some point in time in the future from a series of ground accelerations that occurred earlier in the earthquake event. The second is a recursive approach whereby a model predicts a sequence of future ground accelerations by feeding back its predictions to its inputs. The performances of the models are tested using the 1985 Mexico earthquake and aftershock. The linear predictor and hybrid models perform best at direct prediction while none of the models perform particularly well at recursive prediction. The paper concludes with an outline of some areas for future work.

Keywords: Earthquake Accelerograms, Hybrid Models, Linear Predictors, Radial-Gaussian Neural Networks, Recursive Neural Networks

1. INTRODUCTION

The strong ground motion from earthquakes are a potential cause of human injury and property damage. In the past 15 years, studies have focused on developing methods of predicting such forces by understanding certain fundamental characteristics of these earthquakes, including the source of the seismic waves, the medium through which the waves propagate, the local geology of the site, and the structures located at the site (EERI, 1988). Some site-specific models of strong ground motion are developed with a set of recordings when available (Hanks & McGuire, 1981). For other sites, predictive models referred to as attenuation relations, are used to hypothesize for analysis (Campbell, 1985). In the attenuation relations, the independent variables must be easily estimated from available seismotectonic characteristics of the region. Peak ground acceleration is the most commonly used dependent variable.

The acceleration time history describes the ground motion with all its features, and can be used for elastic and inelastic analyses of any structure. In practice, design engineers have used response spectra with standard shapes, which are related to peak ground acceleration. The peak displacement, velocity, and acceleration are determined for a specific natural frequency and damping ratio of the structure of interest.

Ground motions are recorded on three orthogonal components. Figures 1 and 2 show as an example the acceleration records of the 1985 Mexico earthquake and aftershock, respectively, recorded on the l9th and 21st of September at Papanoa station. These data were made available by the Institute of Geophysics and Planetary Physics of the University of California, San Diego, and the Instituto de Ingeniera at the Universidad Nacional Autonoma de Mexico, Mexico City. The ability to provide quick and accurate predictions of such ground accelerations at specific points in time in the future during an earthquake is important to many prob-

Reprint requests to: Ian Flood, Director, Fire Research Center, M.E. Rinker, Sr. School of Building Construction, University of Florida, Gainesville, FL 32611, U.S.A.; E-mail: flood@nervm.nerdc.ufl.edu

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

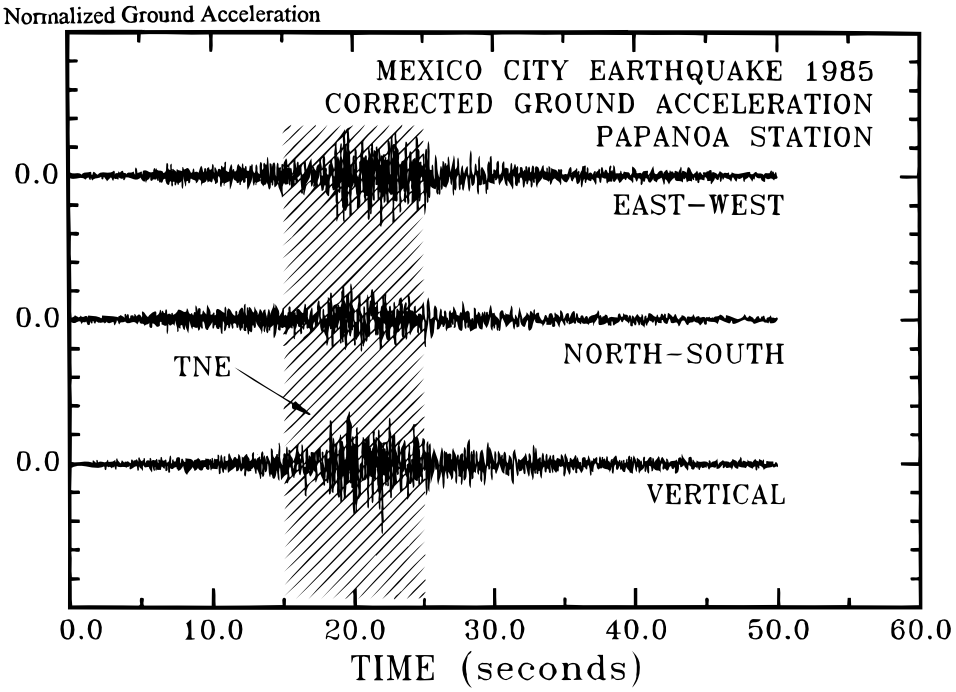


Fig. 1. Three plots showing, respectively, the east–west, north–south, and vertical components of ground acceleration measured during an earthquake event. The plots represent normalized ground acceleration versus time from the start of the earthquake. The shaded bar indicates the portion of the readings used to develop the various models (data set E1).

lems, including the success of active structural systems. These are structures that incorporate some mechanical means of responding to loading, providing greater structural stability, and possibly lower construction costs than conventional nonactive systems (when the approach is established). Examples include active bracing systems (Riley et al., 1992) and friction controllable sliding isolation systems (Nagarajaiah et al.,1992).

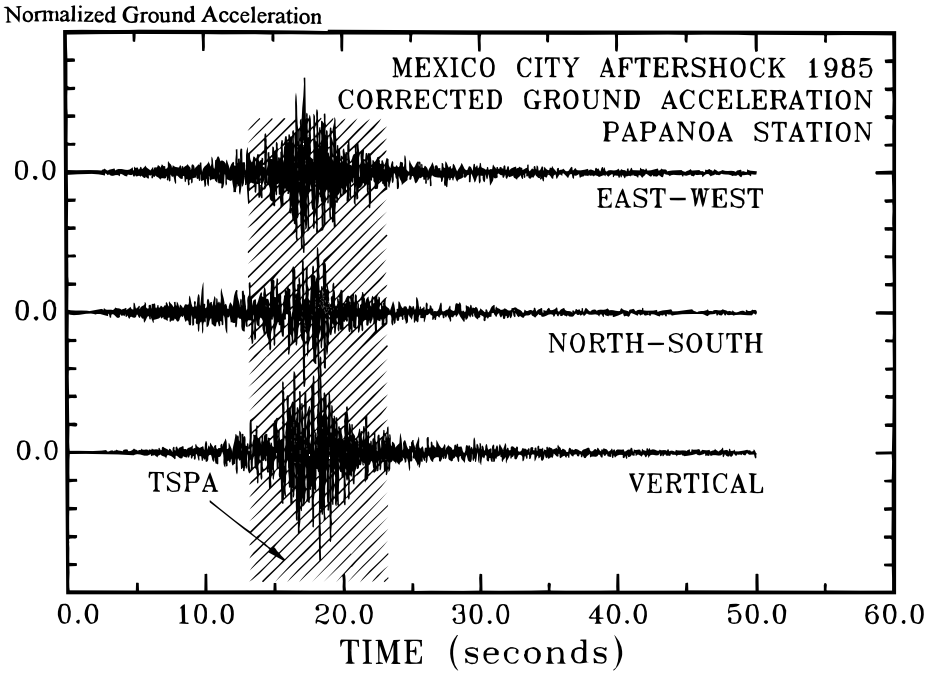


Fig. 2. Three plots showing, respectively, the east–west, north–south, and vertical components of ground acceleration measured during an aftershock event. The plots represent normalized ground acceleration versus time from the start of the aftershock. The shaded bar indicates the high amplitude portion of the readings used as the main basis for validating the models (data set A1).

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

Most methods of predicting ground accelerations are based on linear predictor models; see for example Safak (1988). However, recent interest has been shown in the application of artificial neural network technology to earthquake analysis. Wong et al. (1992), for example, have used this method for predicting the spatial distribution of Modified Mercalli Intensity values. In the present study, the authors consider artificial neural networks as an alternative to linear predictors to model ground acceleration records. Specifically, a radial-Gaussian neural network architecture and complementary training scheme are adopted, having demonstrated excellent learning capabilities in other engineering applications (Gagarin et al., 1992). A linear predictor model is developed and used as a benchmark for evaluating the performance of the neural network. Also, in an attempt to exploit the potential advantages of both the neural network and linear predictor, a hybrid of the two systems is developed and evaluated.

2. NEURAL NETWORKS AND THE RGIN SYSTEM

The radial-Gaussian with Incremental Learning Network system (RGIN) (Flood, 1991) was adopted for this study since it has demonstrated excellent learning capabilities in complex engineering problems (Gagarin et al., 1992). In particular, unlike many other neural networking systems, it does not get stuck in local error-minima during training, it learns rapidly, and it determines automatically how many hidden neurons to incorporate in the network to achieve a given level of accuracy. The details of the RGIN system, as relevant to this paper, are presented in the following.

2.1. Mapping functions

RGIN networks comprise three layers of neurons (each neuron is indicated by a circle) connected in a feedforward manner, as illustrated in Figure 3. These networks, in their simplest form, provide a mapping function from a vector of inputs (representing the problem to be solved) to a vector of outputs (representing the neural network's solution to the problem). They operate by passing information through the network in a forward direction only. As the information passes through the network, it is subject to simple processing both along the links and within the neurons.

Table 1. *Effect of varying number of inputs to RGIN*[a]

No. of Inputs	Theil's Test	Correlation Factor	Average Error
1	—	Unable to train satisfactorily	—
2	0.145	0.922	0.01264
3	0.141	0.932	0.01229
4	0.138	0.965	0.01167
5	0.106	0.960	0.00666
6	0.101	0.966	0.00617
7	0.098	0.961	0.00579
8	0.099	0.961	0.00575
9	0.099	0.961	0.00571
10	0.101	0.956	0.00594

[a]Using 100 hidden neurons, predicting one point in time ahead, and testing on complete earthquake data.

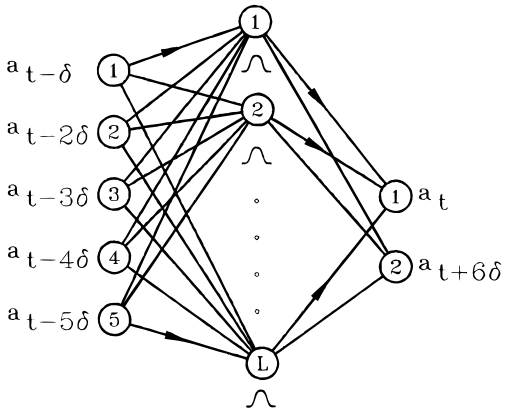


Fig. 3. RGIN ground acceleration predictor.

Consider, for example, the use of this type of network to predict ground acceleration at some point in time in the future from the ground accelerations that occurred at earlier points in time during an earthquake event. The network may be developed to perform this function using ground acceleration data measured at a specific site and resulting from movement at a particular section of a geological fault.

In Figure 3, there are five input neurons, each of which receives a different value representing the ground acceleration at a given point in time, say $(t-\delta)$, $(t-2\delta)$, $(t-3\delta)$, $(t-4\delta)$, and $(t-5\delta)$, respectively, where t is the current time and δ is a small time increment (maybe 10 ms). Given this information, the network would generate values at the output neuron representing an estimate of ground acceleration at times t and $t+6\delta$. The RGIN could be trained to make predictions at other times in the future if required. The ground acceleration values, $a_{t-i\delta}$, presented to the input neurons are passed along the links to the neurons in the hidden layer. Each of these input links has an associated offset $o_{i,j}$ which is subtracted from values passed along the link, following which the residual value is squared. All such values arriving at a hidden neuron are then summed. Formally, these operations can be summarized as follows:

$$\tau_j = \sum_{i=1}^{N} (a_{t-i\delta} - o_{i,j})^2, \tag{1}$$

where τ_j is the summed input to the jth hidden neuron, $a_{\tau-i\delta}$ is a measured ground acceleration presented to the ith input neuron, and $o_{i,j}$ is the offset on the connection between the ith input neuron and the jth hidden neuron.

The summed input at each hidden neuron is then put through a Gaussian activation function:

$$h_j = e^{-s_j \tau_j}, \tag{2}$$

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

where h_j is the level of activation generated at the jth hidden neuron, and s_j is a squeezing parameter associated with that hidden neuron.

The resultant activations generated at the hidden neurons are then transmitted across the links connecting with the output neurons. Each of these output links has an associated weight which is used to multiply values passed along the link. All such weighted values arriving at the output neuron are then summed to produce an output from the network. These operations can be summarized as follows:

$$y_n = \sum_{j=1}^{L} w_{j,n} h_j, \tag{3}$$

where y_n is the value generated at the nth output neuron, and $w_{j,n}$ is the weight on the connection between the jth hidden neuron and the nth output neuron.

This basic mode of operation is the same for all RGIN networks, irrespective of the number of input neurons, hidden neurons, and output neurons they comprise. The specific function performed by a network, however, is dependent on the values of the offsets on its input links, the squeezing parameters at its hidden neurons, and the weights on its outputs links. A primary problem is to determine an appropriate set of values for these parameters that results in a network that performs the required mapping function. These values are derived through a process of training, whereby the network is presented with a set of examples of known input vector to output vector mappings which it attempts to learn. An error-gradient descent technique is adopted for this purpose, where error is measured as the sum of the absolute differences between the actual output from the network and that targeted, over all training patterns:

$$E = \left(\sum_{p=1}^{P} \sum_{n=1}^{M} |y_{p,n} - r_{p,n}| \right) \Big/ p, \tag{4}$$

where $r_{p,n}$ is the targeted output for the pth training pattern at the nth output.

A more detailed description of the structure of these networks and the complementary training algorithm is provided in Flood (1991).

2.2. Simulation using a recursive neural network architecture

The use of a neural network, as described above, is limited in the sense that it can only be used to make predictions at steps in time into the future that were planned for during training. In the example illustrated in Figure 3, it would not be possible to predict ground acceleration at say time $(t + 3\delta)$ without retraining the network.

This limitation could be alleviated by modifying the network to operate recursively. Figure 4 is an example of a neural network that works in this manner. In this instance, the network receives ground acceleration data at five input

Table 2a. *Average error for RGIN*

Data Set	No. of Hidden Neurons	Time Steps in Future				
		1	5	10	50	100
E1						
	1	0.02971	0.06943	0.07018	0.06890	0.07152
	5	0.02119	0.06370	0.06433	0.06206	0.06394
	10	0.01774	0.05902	0.05847	0.05679	0.05906
	50	0.00969	0.03864	0.03832	0.03818	0.03813
	100	0.0610	0.02636	0.02611	0.02748	0.02623
E2						
	1	0.00429	0.00804	0.00860	0.00869	0.01220
	5	0.00431	0.00805	0.00859	0.00885	0.01307
	10	0.00415	0.00806	0.00859	0.00945	0.01361
	50	0.00372	0.00921	0.00975	0.01015	0.01662
	100	0.00377	0.00948	0.01000	0.01014	0.01761
A1						
	1	0.01760	0.02596	0.02690	0.02617	0.03001
	5	0.01354	0.02608	0.02712	0.02634	0.03010
	10	0.01339	0.02608	0.02553	0.02649	0.03004
	50	0.01273	0.02708	0.02590	0.02537	0.03032
	100	0.01276	0.02689	0.02591	0.02559	0.03059
A2						
	1	0.06634	0.09981	0.10233	0.09987	0.10059
	5	0.04807	0.10038	0.10332	0.10065	0.10097
	10	0.04745	0.10038	0.09616	0.10112	0.10062
	50	0.04402	0.10258	0.09642	0.09431	0.09769
	100	0.04405	0.10120	0.09569	0.09510	0.09671

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

Table 2b. *Correlation factor for RGIN*

Data Set	No. of Hidden Neurons	Time Steps in Future 1	5	10	50	100
E1						
	1	0.78	0.37	0.36	0.38	0.34
	5	0.84	0.45	0.44	0.46	0.44
	10	0.87	0.51	0.51	0.52	0.50
	50	0.93	0.69	0.69	0.70	0.70
	100	0.96	0.80	0.80	0.79	0.80
E2						
	1	0.86	0.69	0.67	0.67	0.64
	5	0.86	0.69	0.67	0.66	0.61
	10	0.87	0.69	0.67	0.66	0.61
	50	0.89	0.68	0.66	0.65	0.59
	100	0.90	0.68	0.66	0.65	0.59
A1						
	1	0.85	0.77	0.76	0.76	0.75
	5	0.88	0.77	0.75	0.76	0.75
	10	0.89	0.76	0.77	0.75	0.75
	50	0.90	0.75	0.76	0.77	0.74
	100	0.90	0.75	0.76	0.76	0.74
A2						
	1	0.66	0.42	0.39	0.41	0.37
	5	0.75	0.41	0.38	0.39	0.36
	10	0.76	0.41	0.43	0.37	0.36
	50	0.78	0.35	0.40	0.41	0.36
	100	0.78	0.35	0.40	0.40	0.37

neurons, in the manner described above. The output generated by the network is limited to a single prediction of the ground acceleration at time t. This predicted value is then fed back to input neuron x_1 (the one receiving data representing ground acceleration at time $t - \delta$). Simultaneously with this, the previous value presented to x_1 is fed across to x_2 [the one receiving ground accelerations at time $(t - 2\delta)$], the value previously presented to x_2 is fed across to x_3, and so on across all input neurons. Note that the value previously input to x_5 is discarded. The network then processes this new set of input data and makes a prediction of the ground acceleration at what is in effect time $(t + \delta)$. This entire procedure can be repeated as many times as desired, each occasion generating a prediction at an additional increment, δ, of time in the future. For this network, after five iterations through this process, all input to the network is data it has generated itself at an earlier step.

Table 2c. *Theil's test for RGIN*

Data Set	No. of Hidden Neurons	Time Steps in Future 1	5	10	50	100
E1	1	0.16	0.25	0.25	0.25	0.26
	5	0.13	0.24	0.24	0.24	0.24
	10	0.12	0.23	0.23	0.22	0.23
	50	0.09	0.18	0.18	0.18	0.18
	100	0.07	0.15	0.14	0.15	0.14
E2	1	0.18	0.24	0.25	0.25	0.28
	5	0.18	0.24	0.25	0.25	0.28
	10	0.17	0.24	0.25	0.25	0.29
	50	0.16	0.25	0.25	0.26	0.30
	100	0.16	0.25	0.26	0.26	0.31
A1	1	0.24	0.30	0.30	0.30	0.31
	5	0.20	0.30	0.30	0.30	0.31
	10	0.20	0.30	0.29	0.30	0.31
	50	0.19	0.30	0.29	0.29	0.30
	100	0.19	0.30	0.29	0.29	0.30
A2	1	0.24	0.30	0.31	0.30	0.30
	5	0.19	0.31	0.31	0.30	0.30
	10	0.19	0.30	0.29	0.30	0.30
	50	0.18	0.30	0.29	0.29	0.29
	100	0.18	0.30	0.29	0.29	0.28

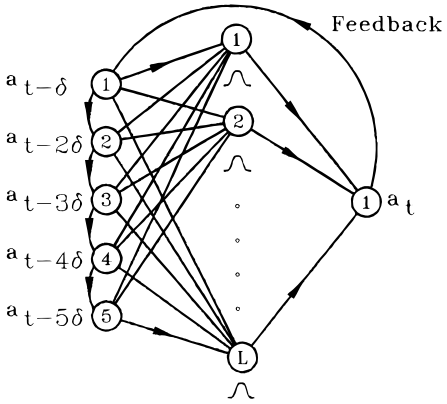


Fig. 4. RGIN used recursively.

Using this approach, it is possible to simulate the ground accelerations generated by an earthquake at a specific location, as far into the future as necessary. Nevertheless, for any prediction it would be expected that the network would introduce some error. Using the network in a recursive manner as such, these errors might compound rapidly until the predictions it generates are, within a few iterations, completely inaccurate. This is most likely where the function being simulated demonstrates chaotic and/or stochastic behavior, as is the case for ground accelerations during an earthquake. The question is how far into the future could a neural network, operating recursively as such, make accurate ground acceleration predictions.

3. LINEAR PREDICTION MODEL

Linear predictor models (LPCs) have been used in seismic analysis in the past few decades (Safak, 1988). Makhoul (1975) describes LPCs as follows:

> "Linear prediction is an autocorrelation-domain analysis. Therefore, it can be approached from either the time or frequency domain. The least squares error criterion in the time domain translates into a spectral matching criterion in the frequency domain."

Applications reported in the literature have concentrated on predicting ground accelerations one time step in the future

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

from a series of past measurements. In this study, however, consideration is given to predicting more than one time step ahead. This generalization of the application of LPCs is outlined in the following paragraphs.

The generalized approach is based on a time domain formulation of the all-pole method, also known as the autocorrelation method described by Makhoul (1975). Figure 5 is a flowchart of the linear prediction method based on the least-squares minimum criteria using Burg's approach (Burg, 1967, 1968). This involves the assembly and inversion of a Toeplitz matrix to determine the linear prediction coefficients. The Yule–Walker formulation is an alternative to Burg's. It uses a specialized procedure, known as the Levinson algorithm, to reach efficiently a solution for the linear prediction coefficients. A complete description of both methods is available in Marple (1987).

The basic hypothesis of the all-pole method is that a signal, a_t, (an acceleration signal) sampled at a constant rate, is predicted by taking a linear combination of past outputs:

$$a_{t+n\delta} = -\sum_{k=1}^{N} \alpha_k a_{t-k\delta} + Gu_t, \tag{5}$$

where α_k are the linear prediction coefficients, G is the gain, and u_t is an input in the underlying process.

Let $\hat{a}_t$ be an approximation of a_t defined as follows:

$$\hat{a}_{t+n\delta} = -\sum_{k=1}^{N} \alpha_k a_{t-k\delta}. \tag{6}$$

The error, e_t, is the difference between the actual and the predicted values. The total error, E, is defined as

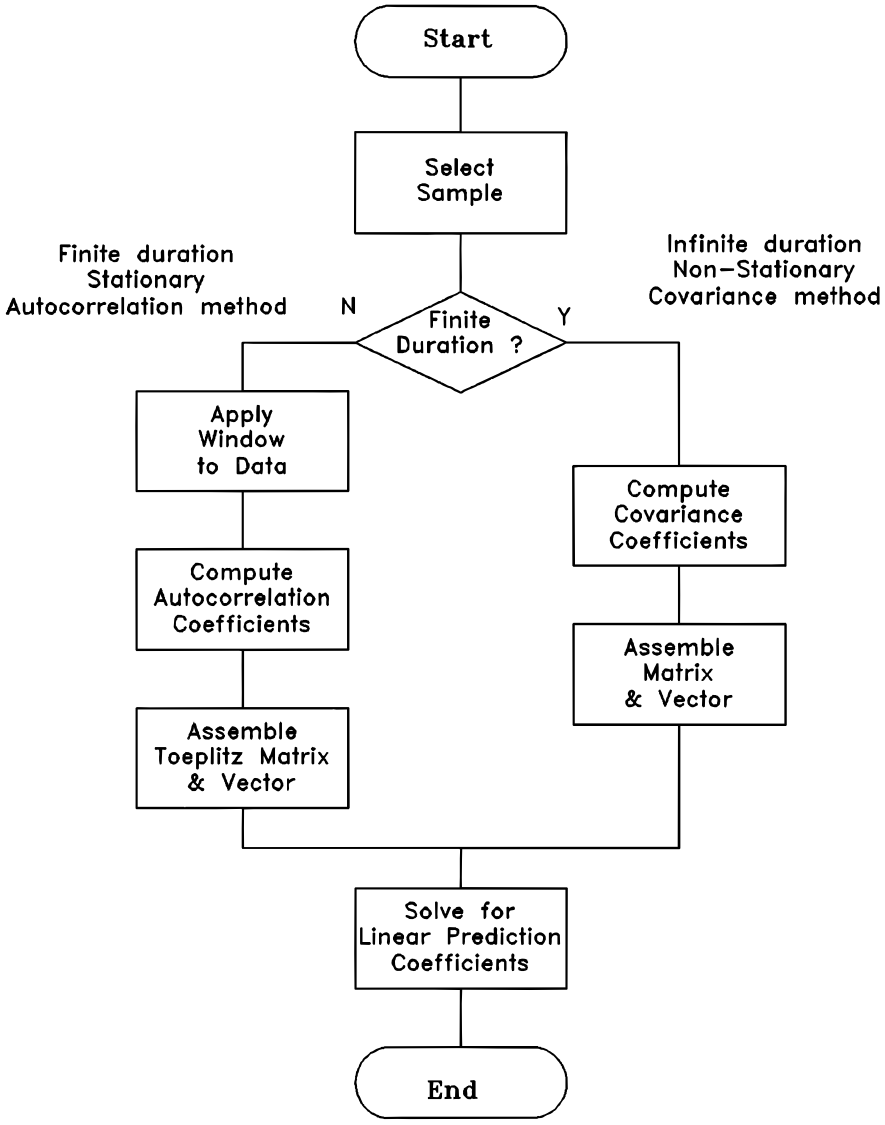


Fig. 5. The all-poles model.

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

Table 3a. *Average error for LPC*

Data Set	No. of Inputs	Time Steps in Future				
		1	5	10	50	100
E1						
	2	0.02230	0.08839	0.08773	0.09377	0.09582
	5	0.01551	0.0732	0.06877	0.07614	0.08011
	10	0.01478	0.06187	0.05806	0.06363	0.07265
	15	0.01432	0.05626	0.05689	0.06124	0.07115
	20	0.01394	0.05578	0.05595	0.06045	0.07013
E2						
	2	0.00250	0.00727	0.00738	0.00716	0.00708
	5	0.00214	0.00585	0.00616	0.00585	0.00600
	10	0.00207	0.00539	0.00513	0.00522	0.00544
	15	0.00203	0.00480	0.00486	0.00497	0.00522
	20	0.00198	0.00467	0.00470	0.00478	0.00503
A1						
	2	0.00909	0.01548	0.01671	0.01811	0.02047
	5	0.00737	0.01293	0.01358	0.01503	0.01779
	10	0.00724	0.01187	0.01251	0.01389	0.01652
	15	0.00717	0.01166	0.01208	0.01385	0.01635
	20	0.00706	0.01152	0.01193	0.01353	0.01628
A2						
	2	0.04731	0.08233	0.08999	0.09512	0.10806
	5	0.03619	0.06928	0.07161	0.07855	0.09482
	10	0.03665	0.06221	0.0666	0.07322	0.08712
	15	0.03521	0.06247	0.06472	0.07321	0.08622
	20	0.03559	0.06134	0.06331	0.07199	0.08659

Table 3b. *Correlation factor for LPC*

Data Set	No. of Inputs	Time Steps in Future				
		1	5	10	50	100
E1						
	2	0.86	0.34	0.36	0.40	0.47
	5	0.89	0.40	0.45	0.48	0.53
	10	0.90	0.49	0.54	0.56	0.57
	15	0.90	0.54	0.55	0.58	0.58
	20	0.91	0.55	0.57	0.59	0.59
E2						
	2	0.90	0.68	0.67	0.61	0.58
	5	0.91	0.74	0.72	0.67	0.63
	10	0.92	0.75	0.76	0.70	0.65
	15	0.92	0.77	0.76	0.71	0.66
	20	0.91	0.77	0.76	0.72	0.67
A1						
	2	0.90	0.81	0.80	0.78	0.76
	5	0.92	0.85	0.84	0.82	0.79
	10	0.92	0.86	0.85	0.84	0.81
	15	0.92	0.86	0.86	0.83	0.81
	20	0.92	0.86	0.86	0.84	0.81
A2						
	2	0.90	0.81	0.80	0.78	0.75
	5	0.92	0.84	0.84	0.82	0.78
	10	0.93	0.86	0.85	0.84	0.80
	15	0.93	0.86	0.86	0.83	0.80
	20	0.93	0.86	0.86	0.84	0.80

Table 3c. *Theil's test for LPC*

Data Set	No. of Inputs	Time Steps in Future				
		1	5	10	50	100
E1						
	2	0.13	0.28	0.28	0.30	0.31
	5	0.11	0.25	0.24	0.26	0.27
	10	0.11	0.23	0.22	0.24	0.26
	15	0.11	0.22	0.22	0.23	0.26
	20	0.11	0.22	0.22	0.23	0.26
E2						
	2	0.15	0.27	0.27	0.29	0.31
	5	0.14	0.24	0.24	0.26	0.27
	10	0.13	0.22	0.22	0.24	0.26
	15	0.13	0.22	0.22	0.23	0.25
	20	0.13	0.21	0.22	0.23	0.25
A1						
	2	0.19	0.27	0.28	0.29	0.31
	5	0.17	0.24	0.24	0.26	0.29
	10	0.17	0.23	0.23	0.25	0.27
	15	0.17	0.22	0.23	0.25	0.27
	20	0.16	0.22	0.23	0.24	0.27
A2						
	2	0.19	0.26	0.28	0.29	0.31
	5	0.16	0.24	0.24	0.26	0.29
	10	0.16	0.23	0.23	0.24	0.27
	15	0.16	0.22	0.23	0.24	0.27
	20	0.16	0.22	0.22	0.24	0.27

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

$$E = \sum_t e_t^2 = \sum_t \left(a_{t+n\delta} + \sum_{k=1}^{N} \alpha_k a_{t-k\delta} \right)^2 \tag{7}$$

The linear prediction coefficients, α_k, are determined by minimizing the total error, E. For this purpose, the partial derivatives of E with respect to α_k are set to zero, for $1 \le k \le N$. Equation (7), for $1 \le i \le N$, becomes

$$\sum_{k=1}^{N} \alpha_k \sum_t a_{t-k\delta} a_{t-i\delta} = -\sum_t a_{t+n\delta} a_{t-i\delta}. \tag{8}$$

Equation (8) represents a set of N linear equations with N unknowns, the linear prediction coefficients. The minimum total error, E_N, is obtained from Eqs. (7) and (8):

$$E_N = \sum_t a_{t+n\delta}^2 + \sum_{k=1}^{N} \alpha_k \sum_t a_{t+n\delta} a_{t-k\delta}. \tag{9}$$

If the total error is minimized over an infinite segment of data, the autocorrelation function, $R(i)$, is substituted in Eqs. (8) and (9)

$$R(i) = \sum_{t=-\infty}^{\infty} a_t a_{t+i\delta} = \sum_{t=-\infty}^{\infty} a_t a_{t-i\delta}$$

$$\sum_{k=1}^{N} \alpha_k R(i-k) = -R(i+n), \qquad \forall 1 \le i \le N$$

$$E_N = R(n) + \sum_{k=1}^{N} \alpha_k R(k). \tag{10}$$

A window is applied to the signal so that all values before and after the sample of data are equal to zero. The autocorrelation function is then equal to

$$R(i) = \sum_{t=0}^{(N-1-i)\delta} a_t a_{t+i\delta}. \tag{11}$$

The generalization to predicting multiple steps ahead does not violate requirements of the frequency domain formulation. Figure 6 is a flowchart of spectral matching using linear prediction coefficients. The signal spectrum, $P(\omega)$, of signal a_t is equal to the square magnitude of its z-transform:

$$P(\omega) = |S(e^{i\omega})|^2 \tag{12}$$

The z-transform of the error in prediction expressed in Eq. (7) is equal to

$$E(z) = \left(1 + \sum_{k=1}^{N} \alpha_k z^{-k} \right) S(z) = A(z) S(z), \tag{13}$$

where $A(z)$ is the inverse filter. Note that $z = e^{i\omega}$.

Dividing both sides of Eq. (13) by $A(z)$ and substituting the newly defined expression for $S(z)$ into Eq. (12), the power spectrum is equal to the ratio of the squared magnitudes of $E(z)$ and $A(z)$:

$$P(\omega) = \frac{|E(e^{i\omega})|^2}{|A(e^{i\omega})|^2}. \tag{14}$$

The power spectrum is approximated using the all-poles model:

$$\hat{P}(e^{i\omega}) = |H(e^{i\omega})|^2 = \frac{G^2}{\left| 1 + \sum_{k=1}^{P} a_k e^{-ik\omega} \right|^2} = \frac{G^2}{|A(e^{i\omega})|^2}. \tag{15}$$

The error term, $E(z)$, may be expressed in terms of the approximation and exact power spectrums by manipulation of Eqs. (14) and (15):

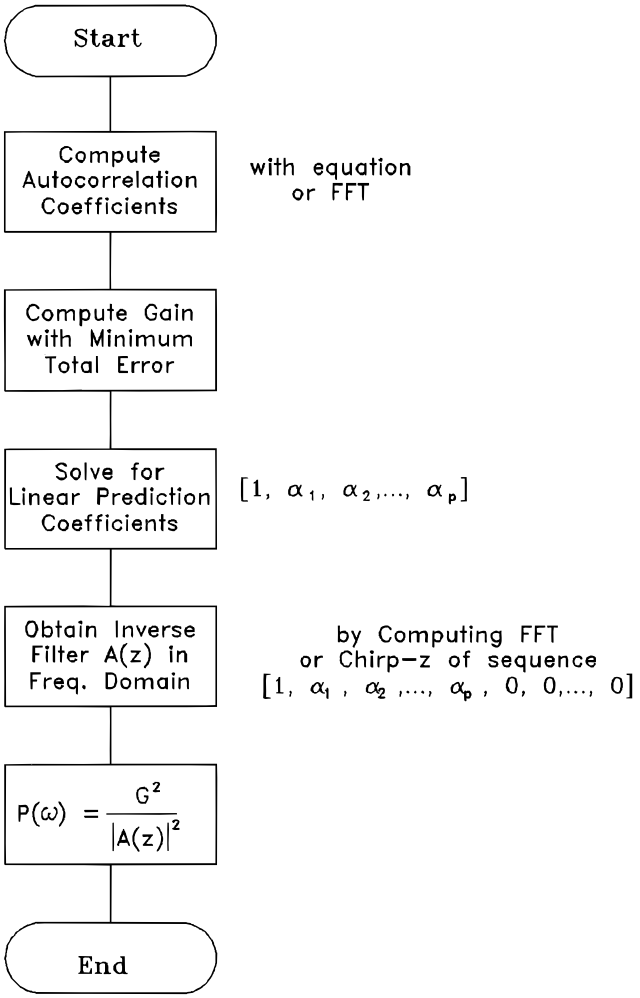


Fig. 6. Spectral matching with all-poles model.

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

$$|E(e^{i\omega})|^2 = G^2 \frac{P(\omega)}{\hat{P}(\omega)}. \tag{16}$$

Assuming a deterministic signal a_t, the total error is obtained by applying Parseval's theorem, and substituting Eq. (16):

$$E = \sum_t e_t^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{i\omega})|^2 d\omega$$

$$= \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega. \tag{17}$$

Minimizing the total error, as described above, is equivalent to the minimization of the integrated ratio of the signal spectrum and its approximation.

An estimate of the power spectrum is found by computing the autocorrelation coefficients, $R(i)$, and solving for the linear prediction coefficients, α_k, using Eq. (10). The gain, G, is equal to the square root of the minimum total squared error, *Ep*, as shown in Makhoul (1975). Then, based on Eq. (15), the estimate of the power spectrum is equal to G^2 divided by the magnitude squared of the *z*-transform of the sequence $[1, \alpha_1, \alpha_2, \ldots, \alpha_N]$.

3.1. Optimal number of linear prediction coefficients

In practice, the optimal number of linear prediction coefficients is found by a process of iteration. The value of the first expression in Eq. (18), written in terms of the normalized minimum total error, V_N, is computed recursively until its value becomes less than a preset error threshold, ν, for two or three consecutive iterations.

$$1 - \frac{V_{N+1}}{V_N} < \nu$$

$$V_N = \frac{E_N}{R(0)}. \tag{18}$$

Note that the LPC can be used to provide both direct and recursive predictions of future ground accelerations in precisely the same manner described for the RGIN models.

4. HYBRID NEURAL NETWORK-LINEAR PREDICTOR SYSTEM

Artificial neural networks (ANNs) are generally not suited to solving problems that fall outside the training domain (the region of the input space that embraces the training patterns) (Flood & Kartam, 1994). This is illustrated in Figure 7a for a problem comprising one input variable. Linear models, on the other hand, can often provide valid results when extrapolating since they model trends in the data that may continue, at least some way, outside the region where the training data resides. This is illustrated in Figure 7b, which shows a linear model of the same problem presented in Figure 7a. In the case of modelling earthquake accelerograms, this suggests that the LPC could perform better than RGIN when the input ground accelerations are generally smaller, or larger, than those used for training.

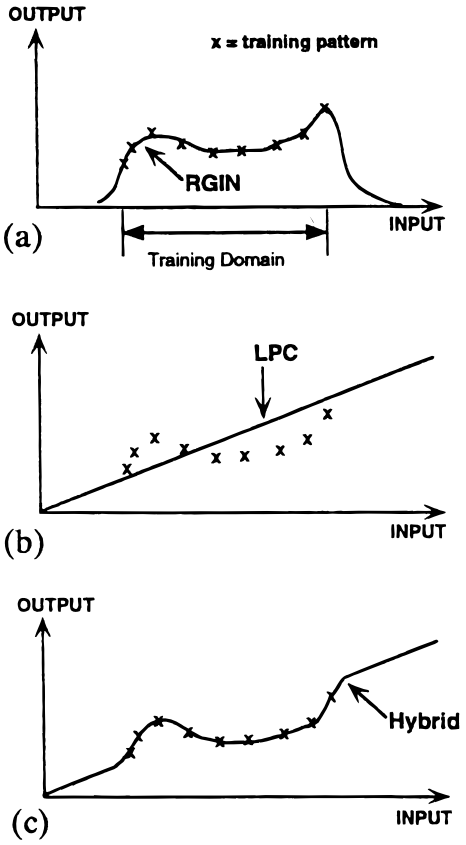


Fig. 7. RGIN, LPC, and hybrid models of data points.

RGINs, as with most types of ANNs, are nonlinear models. Consequently, if the nonlinearities in the function mapping past ground accelerations to future ground accelerations are significant, an RGIN might be expected to perform better than an LPC for points within the training domain (see Figure 7b). For many problems, this closer correspondence with the training data could correspond to a more accurate generalization of the trend of the function being modeled, though only within the training domain.

Arguably, therefore, both types of models have their own merits: LPCs provide the potential for extrapolation while RGINs provide the potential for greater accuracy at interpolation. Clearly, these advantages, if realized, are complementary and could possibly both be captured within a hybrid combining an LPC with an RGIN. One approach to developing such a hybrid would be to first set up an LPC on the training data and then to develop an RGIN model of the difference between the LPC's output and that required according to the training data. The result would be a function

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

Table 4a. *Average error for hybrid*

Data Set	No. of Hidden Neurons	Time Steps in Future 1	5	10	50	100
E1						
	1	0.01344	0.05446	0.05485	0.06024	0.07113
	5	0.01248	0.05077	0.05128	0.05568	0.06446
	10	0.01145	0.04722	0.04878	0.05227	0.05864
	50	0.00757	0.03311	0.03266	0.03690	0.04098
	100	0.00515	0.02433	0.02353	0.02715	0.02870
E2						
	1	0.00198	0.00466	0.00472	0.00499	0.00554
	5	0.00198	0.00466	0.00471	0.00509	0.00687
	10	0.00199	0.00470	0.00475	0.00568	0.01184
	50	0.00196	0.00514	0.00751	0.00872	0.01336
	100	0.00201	0.00532	0.00765	0.00904	0.01307
A1						
	1	0.00710	0.01554	0.01197	0.01344	0.01604
	5	0.00703	0.01554	0.01203	0.01345	0.01626
	10	0.00705	0.01164	0.01204	0.01361	0.01783
	50	0.00702	0.01298	0.01293	0.01547	0.01825
	100	0.00709	0.01320	0.01314	0.01568	0.01819
A2						
	1	0.03523	0.06113	0.06326	0.07076	0.08531
	5	0.03481	0.06113	0.06357	0.07080	0.08591
	10	0.03498	0.06160	0.06354	0.07140	0.08653
	50	0.03522	0.06879	0.06666	0.07811	0.08794
	100	0.03561	0.06987	0.06796	0.07894	0.08793

Table 4b. *Correlation factor for hybrid*

Data Set	No. of Neurons	Time Steps in Future 1	5	10	50	100
E1	1	0.88	0.50	0.50	0.44	0.33
	5	0.89	0.54	0.54	0.49	0.41
	10	0.89	0.58	0.57	0.52	0.44
	50	0.92	0.72	0.72	0.67	0.63
	100	0.94	0.79	0.80	0.76	0.75
E2	1	0.81	0.67	0.67	0.63	0.59
	5	0.81	0.67	0.67	0.62	0.54
	10	0.81	0.67	0.67	0.61	0.51
	50	0.81	0.66	0.60	0.60	0.52
	100	0.81	0.65	0.60	0.60	0.52
A1	1	0.40	0.31	0.30	0.28	0.24
	5	0.40	0.31	0.30	0.28	0.23
	10	0.40	0.31	0.30	0.28	0.23
	50	0.40	0.28	0.29	0.25	0.22
	100	0.40	0.28	0.29	0.25	0.22
A2	1	0.33	0.31	0.30	0.30	0.28
	5	0.33	0.31	0.30	0.29	0.28
	10	0.33	0.30	0.30	0.29	0.27
	50	0.33	0.30	0.30	0.28	0.27
	100	0.33	0.30	0.30	0.28	0.27

Table 4c. *Theil's test for hybrid*

Data Set	No. of Hidden Neurons	Time Steps in Future 1	5	10	50	100
E1	1	0.10	0.21	0.21	0.22	0.24
	5	0.09	0.20	0.20	0.21	0.23
	10	0.09	0.19	0.19	0.20	0.21
	50	0.07	0.16	0.15	0.16	0.17
	100	0.06	0.13	0.13	0.14	0.14
E2	1	0.11	0.19	0.19	0.20	0.22
	5	0.11	0.19	0.19	0.20	0.23
	10	0.12	0.19	0.19	0.21	0.26
	50	0.11	0.19	0.22	0.21	0.27
	100	0.11	0.19	0.22	0.23	0.27
A1	1	0.05	0.07	0.08	0.08	0.09
	5	0.05	0.07	0.08	0.08	0.09
	10	0.05	0.07	0.08	0.08	0.09
	50	0.05	0.08	0.08	0.09	0.10
	100	0.05	0.08	0.08	0.09	0.10
A2	1	0.07	0.10	0.10	0.11	0.13
	5	0.07	0.10	0.10	0.11	0.13
	10	0.07	0.10	0.10	0.11	0.13
	50	0.07	0.11	0.11	0.12	0.13
	100	0.07	0.11	0.11	0.12	0.13

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

that is essentially linear in form outside the training domain and distinctly nonlinear inside the training domain, such as illustrated in Figure 7c. The output from the hybrid network can be expressed as

$$a_{t+n\delta} = \sum_{i=1}^{N1} (\alpha_i a_{t-i\delta}) + \sum_{j=1}^{L} w_{j,n} e^{-s_j \sum_{i=1}^{N2} (a_{t-i\delta} - o_{i,j})^2}, \qquad (19)$$

where $N1$ is the number of inputs used by the linear component of the model and $N2$ is the number of inputs used by the nonlinear component of the model.

5. EXPERIMENTAL DATA

The accelerogram data, plotted in Figures 1 and 2, is digitized at 100-Hz sampling rate (time step equal 0.01 s). It was divided into four sets for the development and evaluation of the models:

1. Data set E1. This is the data used to train the RGIN and develop the coefficients for the LPC. It embraces 1000 consecutive measurements from the earthquake, from 15.01 s through to 25.00 s, as indicated by the shaded region of Figure 1. This portion of the accelerogram was chosen to ensure the models were provided with information about the behavior of the high amplitude region of the accelerogram (arguably the most critical in terms of causing damage to structures).
2. Data set E2. This is the remaining data from the earthquake accelerogram not used for developing the models (i.e., the earthquake data excluding data set E1). This data set was established to test the performances of the models on the lower amplitude regions of the event on which they were developed.
3. Data set A1. This includes the complete record of the aftershock, and was established to test if the models could predict ground accelerations for an event not used to develop the models (though measured at the same recording station).
4. Data set A2. This is a partial record of the aftershock comprising 1000 consecutive points from 13.01 s to 23.00 s during the event. It focuses on the high amplitude region of the aftershock (indicated by the shaded region of Figure 2) to test the predictive capabilities of the models during the most critical portion of the event.

To reduce the number of input variables required in the models, the experiments reported in this paper combined the three components of ground acceleration, measured at each point in time, into their vectors:

$$a_i = \sqrt{a^2_{vertical} + a^2_{east-west} + a^2_{north-south}}. \qquad (20)$$

Three indicators were computed to evaluate different characteristics of the performances of the models for each of the four data sets:

1. Average error, measuring the mean absolute difference between the predicted and actual values in a data set:

$$E = \frac{\sum_{i=1}^{P} |\hat{a}_i - a_i|}{P}. \qquad (21)$$

The average error provides an indication of how far individual predictions can be expected to vary from actual accelerations.

2. Correlation factor which varies between -1.0 and 1.0, measuring the similarity in shape between the predicted and actual plots of ground acceleration (1.0 = perfect correlation, 0.0 = no correlation, and -1.0 = perfect negative correlation):

$$\rho = \frac{P \sum_{i=1}^{P} \hat{a}_i a_i - \sum_{i=1}^{P} \hat{a}_i \sum_{i=1}^{P} a_i}{\sqrt{\left[P \sum_{i=1}^{P} \hat{a}_i^2 - \left(\sum_{i=1}^{P} \hat{a}_i\right)^2\right]\left[P \sum_{i=1}^{P} a_i^2 - \left(\sum_{i=1}^{P} a_i\right)^2\right]}}. \qquad (22)$$

The correlation coefficient provides a measure of the correspondence in the up and down trends in the predicted and actual ground acceleration plots.

3. Theil's test which varies between 0.0 and 1.0, measuring the similarity, both in magnitude and shape, between the plots of predicted and actual ground accelerations (0.0 = maximum equality or perfect forecast, and l.0 = maximum inequality):

$$U = \frac{\sqrt{\sum_{i=1}^{P} (\hat{a}_i - a_i)^2}}{\sqrt{\sum_{i=1}^{P} \hat{a}_i^2} + \sqrt{\sum_{i=1}^{P} a_i^2}}. \qquad (23)$$

Theil's test, compared to the correlation coefficient, is more sensitive to the relative magnitudes of actual and predicted values.

6. RESULTS USING DIRECT PREDICTION

The first set of experiments were designed to evaluate the ability of the three alternative models (the RGIN, LPC, and hybrid systems) to predict ground acceleration directly at a given point in time in the future from earlier ground accelerations. Before this could be done, however, there was a need to determine the optimal number of parameters to in-

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

corporate in each model (i.e., the number of input and hidden neurons for the neural network, and the number of coefficients for the linear predictor).

6.1. Number of neurons for RGIN

A parametric study was undertaken to determine the number of input and hidden neurons to include in the RGIN model. RGINs were trained to predict one time step ahead of the input data using data set E1, for a range of 1 to 10 input neurons and 1 to 100 hidden neurons. Data sets E1 and E2 were combined for testing performance using the error measure, E, given in Eq. (4). In all experiments, no discernible improvement in performance was attained in the region of 100 hidden neurons. For the 100 hidden neuron networks, the errors for different numbers of inputs were E: 2 = 0.01264, 3 = 0.01229, 4 = 0.01167, 5 = 0.00666, 6 = 0.00617, 7 = 0.00579, 8 = 0.00575, 9 = 0.00571, and 10 = 0.00594. From this it can be seen that there is little improvement in performance (measured to an accuracy of 0.0005) moving beyond five input neurons. It was decided therefore, on the basis of these experiments, to use a network comprising five input neurons and 100 hidden neurons. While there could be some improvement in the accuracy of predictions by extending beyond this size, the benefits that may accrue appeared negligible.

6.2. Number of linear prediction coefficients

The optimum number of linear predictor coefficients (one for each input) is based on a subjective selection of an error threshold as described in the section on LPCs. A parametric study was undertaken, varying the number of inputs in the LPC from 2 to 10. In each case, the LPCs were developed using data set E1 and tested on the combined data sets E1 and E2, as for the RGIN model. These experiments yielded the following error values of V [see Eq. (18)]: 2 = 0.0707, 3 = 0.2581, 4 = −0.0017, 5 = 0.0192, 6 = −0.0032, 7 = 0.0262, 8 = 0.0182, 9 = 0.0179, and 10 = 0.0059. Moving beyond seven inputs resulted in a steady reduction in the error function. Since the LPC showed a continual improvement with the addition of more inputs, an arbitrary limiting value had to be selected. A total of 20 inputs was selected since it was clear that there would be no significant change in the error (measured to an accuracy of 0.0005) moving beyond this value.

6.3. Number of input and hidden neurons for the hybrid

The proposed hybrid system for modelling ground accelerations used $N1 = 20$ previous values for the linear component and $N2 = 5$ previous values for the nonlinear component [see Eq. (19)], being the maximums adopted for the respective LPC and RGIN models. In other words, the linear component was connected to input neurons $i = 1$ to $i = 20$ while the nonlinear component of the hybrid network was connected to input neurons $i = 1$ to $i = 5$.

6.4. Comparative performances

The performances of the three models were compared in terms of their ability to predict ground accelerations at 1, 5, 10, 50, and 100 time steps in the future from the earlier data. Data set E1 was used for training/developing the models. Figure 8 compares the performances of the three types of model for the average error, correlation factor, and Theil's test, respectively, for each of the four data sets. In all cases, performance is plotted as a function of the number of time steps into the future for predictions.

As might be expected, these figures indicate that there was a general reduction in performance (decreasing correlation factor, and increasing average error and Theil's test) with an increase in the reach (number of time steps ahead) of the predictions.

The most important performance experiments were those using data set A2 since this represents, firstly, an event (the aftershock) that was not used to develop the models, and secondly, the high amplitude region of the accelerogram being the part most likely to cause damage to structures. For the average error, it can be seen that for data set A2 (indicated by the dash-double-dotted lines) the performances of the LPC and hybrid are almost indistinguishable and markedly better than that of RGIN. For the correlation factor, the LPC provided the best performance, followed by RGIN and the hybrid which both performed poorly. For Theil's test, the hybrid provided the best performance followed by the LPC and RGIN which both performed relatively poorly. In summary, according to these results, RGIN was consistently a relatively poor performer, while the LPC and hybrid systems were comparable and generally good performers.

7. RESULTS USING RECURSIVE PREDICTION

A further set of experiments were undertaken to assess the performances of the three types of model when operating recursively. The models developed in the previous section for predicting 1 time step ahead were adopted for this purpose. Each model was presented with a starting sequence of ground accelerations as input and then allowed to run recursively for 100 iterations, as described earlier. A total of seven start points were selected, from the earthquake and aftershock, to provide a range from low to high amplitude regions in the accelerograms. The start points for each experiment are provided in the keys to the Figures 9 and 10.

In each experiment, the performance of the models was measured using the correlation factor and Theil's test for 5, 10, 50, and 100 iterations. Figures 9 and 10 show the results

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

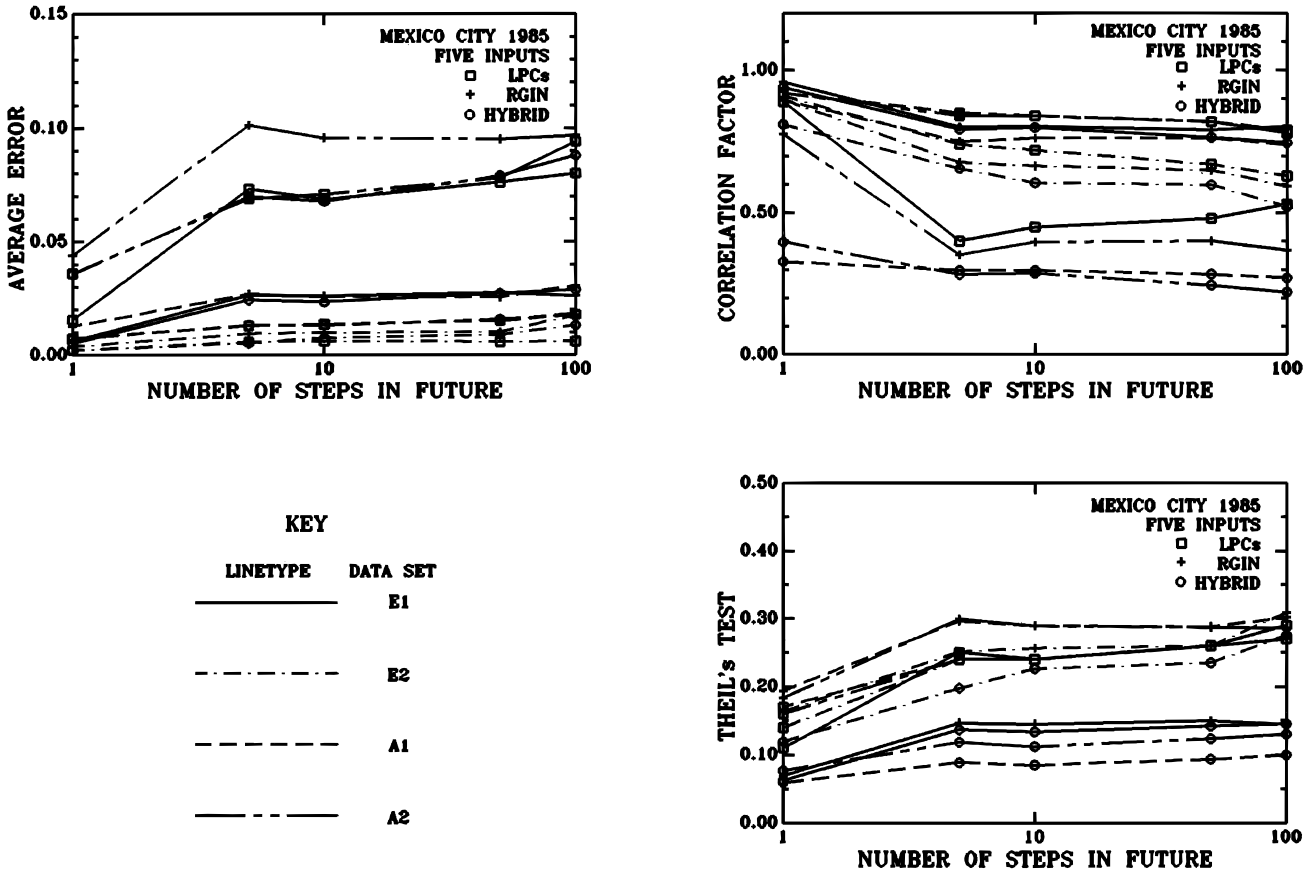


Fig. 8. Average error, correlation factor, and Theil's test for direct prediction.

of these experiments for starting points taken from the earthquake and aftershock accelerograms, respectively, using the correlation factor, and Theil's test. Inspecting the results for Theil's test, the worst performer was RGIN for both the earthquake and aftershock when starting at a point 2.01 min into ground acceleration readings. In both cases, RGIN is attempting to simulate a low amplitude region of the accelerograms and thus is operating outside its training domain. The LPC and hybrid models show reasonable and similar performance when starting at these same low amplitude regions of the accelerogram. This is in agreement with the argument outlined earlier that RGIN, unlike the LPC and hybrid systems, is not a valid predictor when asked to extrapolate. On the other hand, RGIN was the best performer when starting in the high amplitude region of the earthquake (at 22.01 s into the accelerogram readings) and demonstrated a performance that was comparable to the other models when started in the high amplitude region of the aftershock (at 18.01 s into the accelerogram readings). In these cases, RGIN was operating in a region that was largely within its training domain, and therefore, was expected to provide a more accurate model than the LPC given that the required mapping function is nonlinear in form.

According to the correlation factor (Figures 9 and 10), the performance of each model appears disappointing. The only experiment that yielded a good correlation for the first 10 iterations was that of RGIN started at a point 22.01 min into the earthquake. The reason for this is simply that, in this case, RGIN was started at a position in the accelerogram that was used for its training and in this sense was really just attempting to reproduce the training data.

Figures 11 and 12 plot the output generated by each model for the first 100 iterations, for the recursion starting points located in the high amplitude regions of the earthquake and aftershock (i.e., at 22.01 s and 18.01 s into the accelerogram readings, respectively). In each case, the simulated output is superimposed on the actual readings. Referring to Figure 11, it can be seen that the RGIN output follows the measured ground accelerations for the first 9 iterations before it starts to deviate. This was the experiment that provided a good correlation over the first 10 iterations discussed above. Apart from this, none of the simulation plots follow the measured plots particularly closely. A visual inspection of these figures suggests that the LPC is the worst at capturing the characteristic behavior of the accelerogram in terms of the range of the amplitudes of the peaks and troughs

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

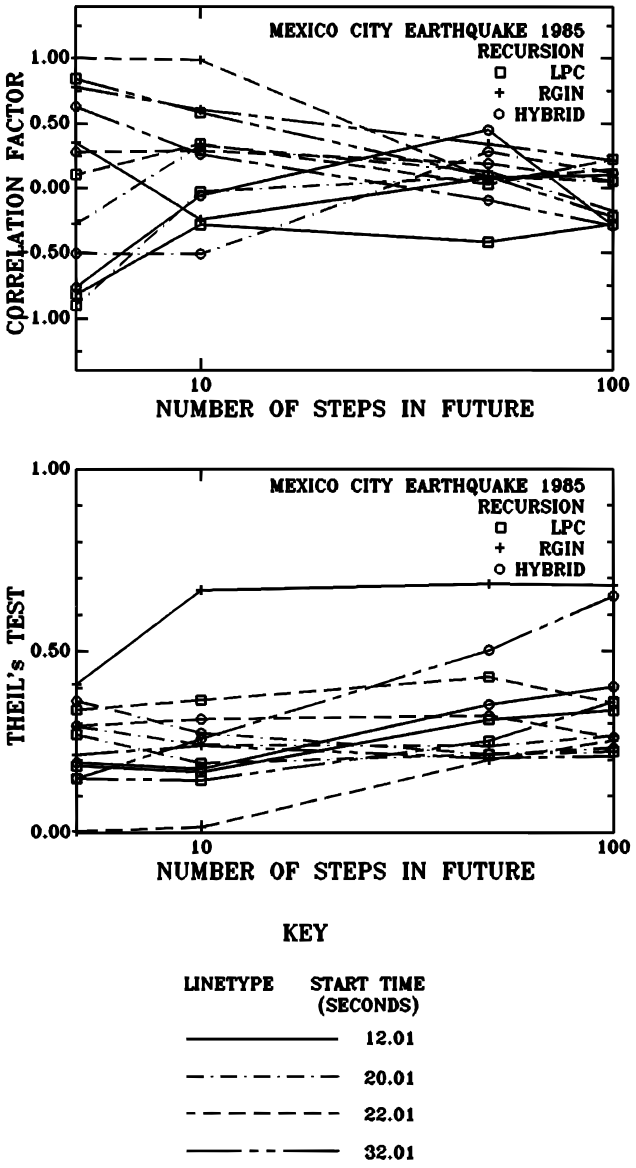


Fig. 9. Correlation factor and Theil's test for recursive prediction with earthquake data.

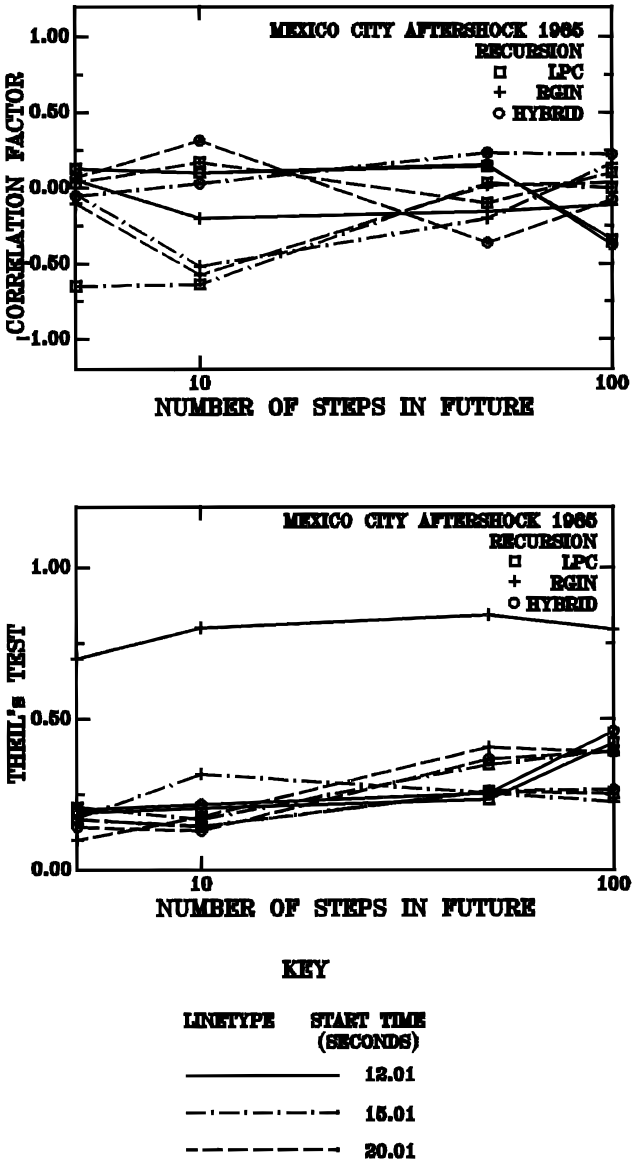


Fig. 10. Correlation factor and Theil's test for recursive prediction with aftershock data.

in the plot. The best performance in this respect is provided by the hybrid model.

8. CONCLUSIONS

None of the three methods of modelling adopted in this study consistently outperformed the others. However, for the direct method of predicting ground accelerations, the LPC and hybrid models performed significantly better than RGIN. While there was little to distinguish between the LPC and hybrid models in this regard, the LPC provides a more practical means of modelling since it is much simpler in form than the hybrid. The LPC models comprise just 20 coefficients which are used simply to weight and sum a set of input ground accelerations. In contrast, the hybrid includes an additional seven parameters (five offsets, a squeezing parameter, and an output weight) for every hidden neuron it incorporates, and requires the application of power functions. Consequently, even if the hybrid was implemented in parallel operating neural hardware, it would still be computationally more expensive than the LPC. This is particularly important when making predictions of ground acceleration at small steps in time in the future since the results are required early enough to allow for an effective response decision.

None of the models were capable of providing accurate predictions of ground acceleration when operating recursively. RGIN, however, showed some promise when oper-

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

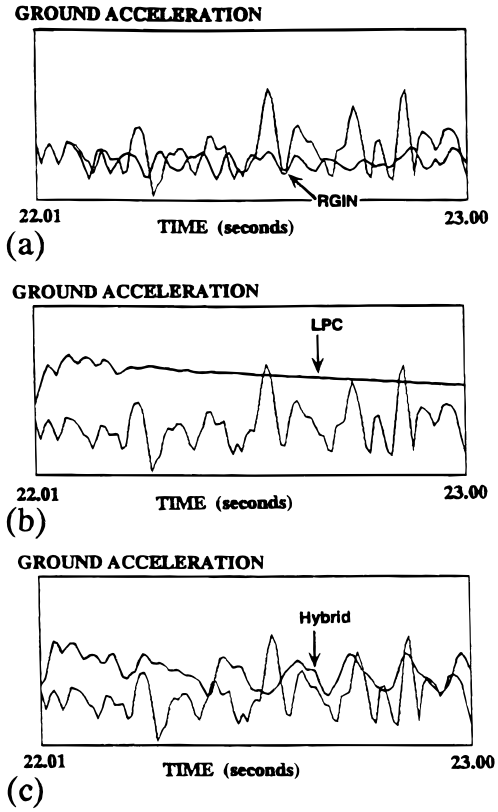


Fig. 11. Ground acceleration simulations in high amplitude region of earthquake.

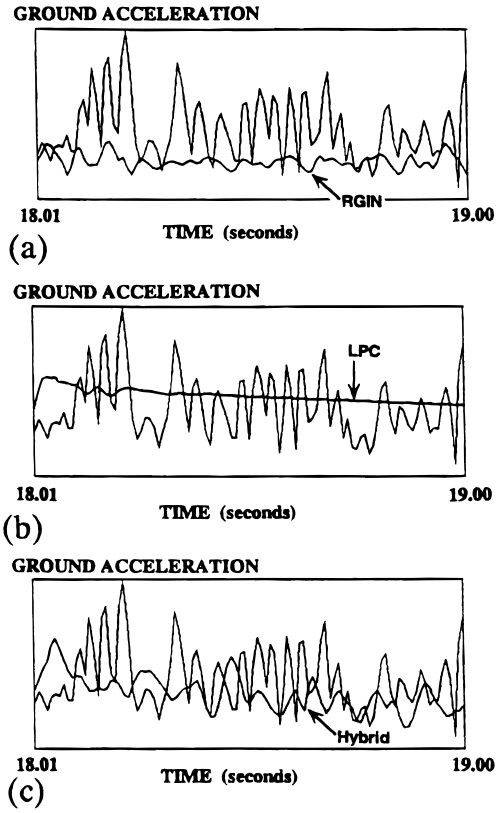


Fig. 12. Ground acceleration predictions in high amplitude region of aftershock.

ating within its training domain. This suggests that it may be possible to develop a reasonable recursive model using RGIN if the training domain is extended to include examples from low amplitude regions of the earthquake accelerogram, or even from a range of different earthquakes. Since the hybrid model appeared to capture more closely the time-wise characteristic behavior of the accelerogram, consideration should also be given to developing this model by extending the training set.

The study was limited to the application of the technique to the 1985 Mexico earthquake on which the models were developed and the aftershock. Further work is required, therefore, comparing the performances of the alternative models for a broader range of earthquake events. Finally, consideration should also be given to modelling the individual east–west, north–south, and vertical components of the ground acceleration vectors since in many problems, such as active bracing systems, these would be the values required.

Although the proposed neural approach was shown not to outperform a linear predictor-based model for this application, it is possible that it could offer a more efficient means of modelling other time-evolving functions. Potential applications in this regard might include predicting wave induced stresses on off-shore structures (such as oil rigs), and extrapolating vibrations monitored in mechanical components to provide advanced warning of failure.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and editor for their constructive and useful comments.

REFERENCES

Burg, J.P. (1967). Maximum entropy spectral analysis. *Proc. 37th Meeting of the Society of Exploration Geophysicists*, 131–148.

Burg, J.P. (1968). *A new analysis technique for time series data*. NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, 197–208.

Campbell, K.W. (1985). Strong motion attenuation relations: A ten-year perspective. *Earthquake Spectra* (EERI) *1(4)*, 759–804.

EERI, (1988). *Strong ground motion: Seismic analysis, design, and code issues*. A two-day course from EERI. The Citadel, Charleston, SC.

Flood, I. (1991). A Gaussian-based neural network architecture and complementary training algorithm. *Proc. Int. Joint Conf. on Neural Networks*, Vol. 1, pp. 171–176. Singapore.

Flood, I., & Kartam N. (1994). Neural networks in Civil Engineering: Principles and understanding. *Journal of Computing in Civil Engineering* (ASCE), *8(2)*, 131–148.

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press

Gagarin, N., Flood, I., & Albrecht, P. (1992). Weighing trucks in-motion using Gaussian-based neural networks. *Proc. Int. Joint Conf. on Neural Networks*, Vol. II, pp. 484–489. Baltimore.

Hanks, T.C., & McGuire, R.K. (1981). The character of high-frequency strong ground motion. *Bulletin of the Seismological Society of America 71(6)*, 2071–2095.

Makhoul, J. (1975). Linear prediction: A tutorial preview. *Proceedings of the IEEE 63*, 561–580.

Marple, S.L., Jr. (1987). *Digital Spectral Analysis*, Prentice Hall, Inc., NJ.

Nagarajaiah, S., Feng, M., Shinozuka, M., & Reinhorn, A.M. (1992). Analysis of bridges with friction controllable sliding isolation system. *Proc. Third NSF Workshop on Bridge Engineering in Progress*, pp. 243–246. La Jolla, California.

Riley, M.A., Nagarajaiah, S., & Reinhorn, A.M. (1992). Hybrid control of absolute motion in aseismically isolated bridges. *Proc. Third NSF Workshop on Bridge Engineering in Progress*, pp. 239–242. La Jolla, California.

Safak, E. (1988). *Analysis of recordings in structural engineering: Adaptive filtering, prediction, and control*. Open-File Report 88-647, US Dept. of the Interior Geological Survey, Menlo Park, CA.

Wong, F.S., Tung, A.T.Y., & Dong, W. (1992). Seismic hazard prediction using neural nets. *Proc. Tenth World Conf. on Earthquake Engineering*, pp. 339–343. Balkema, Rotterdam, Holland.

Ian Flood is Director of the Fire Research Center and a faculty member of the M.E. Rinker, Sr. School of Building Construction, at the University of Florida, Gainesville. Prior to this, he held academic positions in the Department of Civil Engineering at the University of Maryland, College Park, and in the School of Building and Estate Management at the National University of Singapore. He has published extensively in the areas of construction management and computer-based modelling in engineering. His current areas of research include the development of artificially intelligent methods of modelling construction processes, neural network-based approaches to the weigh-in-motion of trucks, and the modelling of uncertainty in construction costs and schedules.

Nicolas Gagarin is a graduate from the Department of Civil Engineering at the University of Maryland, College Park, where he received his Ph.D. in 1991. His graduate research while at Maryland was concerned with the weighing-in-motion of trucks as they cross steel bridges, and he has authored a number of papers in this field. He has been a postdoctoral research fellow at the Federal Highways Administrations' Office of Research and Development, McLean Virginia, and is a consultant to industry in software development and computer-based modelling.

Nabil A. Kartam is an Associate Professor of Civil Engineering at Kuwait University. His current teaching and research are centered around the design, implementation, and validation of advanced intelligent information systems in civil engineering. Special research emphasis on supporting the design-construction integration throughout the project life cycle, contracts and claims management, value engineering constructability improvement, and construction safety. Dr. Kartam has more than 15 years experience as an engineer, planner, professor, and consultant in U.S.A. and Kuwait. He holds an M.S.E. degree in construction engineering and management in 1985 from the University of Michigan, Ann Arbor; an M.S. degree in computer science/AI and expert systems in 1988 and a Ph.D. degree in civil engineering in 1989 from Stanford University.

https://doi.org/10.1017/S089006049812303X Published online by Cambridge University Press