Neural prediction of response spectra from mining tremors using recurrent layered networks and Kalman filtering

A. Krok*, Z. Waszczyszyn

Cracow University of Technology, Institute of Computer Methods in Civil Eng., Warszawska 24, 31–155 Krakow, Poland

Abstract

Acceleration response spectra (ARS) for mining tremors in the Upper Silesian Coalfield are generated using artificial neural networks trained by means of Kalman filtering. The target ARS were computed on the basis of measured accelerograms. It was proved that the recurrent layered network, trained by the recurrent decoupled extended kalman filter (RDEFK) algorithm is numerically much more efficient than the standard feed-forward NN learnt by DEKF. It is also shown that the considered KF algorithms are better than the traditional Rprop learning method.

Keywords: Acceleration response spectrum; Mining tremor; Reccurent layered neural network; Kalman filtering

1. Introduction

Response spectra caused by paraseismic excitations (in the paper, mining tremors only are analyzed) are used for the design of buildings in mining regions, as well as for evaluation of damage resistance of actual buildings [1]. The monitoring of paraseismic excitation at every building is impossible so either recommended design response spectra or average response spectra, computed an the base of earlier measured accelerograms at the buildings, are used.

In recent years artificial neural networks (ANNs) have been applied for the computation of acceleration response spectra (ARS) [2–4]. An attempt to predict ARS in two Polish mining regions was carried out by Kuzniar et al. [1]. Corresponding ANNs were designed on the base of accelerograms of the surface waves measured on the ground level at selected buildings for known values of epicentre distances and energy of mining tremors.

The analysis of this problem was developed by Krok et al. in [5], where the Kalman filtering (KF) was introduced as a refined learning method of the feed-forward multilayer perceptron, using the DEKF algorithm.

The present paper is a continuation of [5]. Instead of DEKF, its modification, recurrent DEKF (RDEKF) is applied. The modification is coupled with a recurrent-layer version of Elman [6], called RLNN. Similarly, as in [5], an autoregressive (time-delayed) input is used and the training and testing patterns are based on the records measured in the Upper Silesian Coalfield (USC), Poland.

* Corresponding author. Tel.: +48 12 2667965; Fax: +48 12 6282034; E-mail: agakrok@poczta.fm

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

2. Neural Kalman filter

2.1. Network architectures

Basic equations and algorithms of Kalman filtering, used for ANN training, correspond to the NN architecture. Following Haykin et al. [7], two architectures, related to the standard feed-forward layered network FLNN (multilayer perceptron) and the recurrent layered network RLNN (Fig. 1), are considered.

2.2. Kalman filtering

Extended KF is based on two equations [7]: (1) process equation, and (2) measurement equation, modified for using in RLNN into the following form:

$$\{\mathbf{w}_i(k+1), \mathrm{v}_i(k+1)\} = \{\mathbf{w}_i(k), \mathrm{v}_i(k)\} + \boldsymbol{\omega}(\mathbf{k}) \quad (1)$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{w}(k), \mathbf{v}(k), \mathbf{x}(k)) + \nu(k) \quad (2)$$

where k is the discrete pseudo-time parameter; i is the number of neurons in ANN; $\mathbf{w}\,(k) = \{\mathbf{w}_i\,(k), \mathrm{v}_i\,(k)\; i = 1,$

2, ... , n} is the state vector (one-column matrix) corresponding to the set of vectors, $\mathbf{w}$, of synaptic weights and biases, and neuron outputs, v_i, for n neurons of NN; $\mathbf{h}$ is the nonlinear vector-function of input–output relation; $\mathbf{x}$, $\mathbf{y}$ are the input/output vectors and $\boldsymbol{\omega}(\boldsymbol{k})$, $\boldsymbol{\nu}(\boldsymbol{k})$ the Gaussian process and measurement noises with zero mean and known covariance matrices. The term 'extended KF' (EKF) is used because a nonlinear output–input relation is considered in (2) by the introduction of the vector-function $\mathbf{h}$.

2.3. Algorithms RDEKF and DEKF

The algorithm is called 'recurrent' because the time-delay internal input v_{k-1} is used in the network shown in Fig. 1(b). Decoupling is performed with respect to each neuron $i = 1, 2, \ldots, n$. The algorithm RDEKF was formulated in [7] on the base on Eqs. (1) and (2) as a modification of algorithms given in [5].

An essential problem of the RDEKF algorithm is the computation of matrix $\mathbf{H}_i^{\text{rec}}$ of recurrent linearization, computed at $\{\mathbf{w}_i, \mathbf{v}_i\}$, where: $\mathbf{w}_i$(k), $\mathbf{v}_i$(k) are a priori estimators. The linearization in the present paper was performed by the back-propagation in time procedure.

The algorithm DEKF, related to the feed-back layer network (FLNN), Fig. 1(a), can be easily formulated as a special case of RDEKF. This is caused by canceling of the feed-back links $\mathbf{v}_i$ (k) in Eqs. (1) and (2).

3. Surface vibrations from mining tremors in USC region

The accelerograms of surface waves were taken from USC region in Poland and corresponding values of the tremor energy $E \in [2.10^4, 4.10^6]$ J and epicentre distance $r_e \in [0, 1200]$ m evaluated by seismic stations situated nearby.

The dimensionless ARS were computed in [1], using the definition $\beta(T) = S_a(T; \mathrm{E}, r_e)/a_{\max}$, where: S_a [m/s^2] are computed ARS; $a_{\max}$[m/s^2] $= \max_t a(t)$ is maximal acceleration, and $T(t)$[s] $= 1/f(t)$ is the period of vibration for natural frequency $f(t)$ [Hz].

A set of 145 ARS was taken from [10] in discretized form $\beta_k = \beta(T_k)$ for pseudo-time parameters $k = 1, 2, \ldots, 198$. This makes a set of $P = 145 \times 198 = 28710$ patterns. The same sets composed of $ARSL = 113$ spectra and $ARST = 32$ spectra, corresponding to those randomly selected in [1] were used for the networks learning and testing, respectively. This made $L = 113 \times 198 = 22374$ and $T = 32 \times 198 = 6336$ patterns, correspondingly.

4. Neural analysis

4.1. Application of RDEKF and DEKF algorithms

The input vector $\mathbf{x} = \{\beta_{k-1}, E, r_e\}$ and scalar output $y = \beta_k \equiv \beta(T_k)$ were adopted. The inputs E and r_e were transformed by the function ln and then all inputs and outputs were scaled to the range [0,1]. The training was performed by our own computer simulator written in MATLAB language and KF procedures written in C++. After extensive numerical experiments the network of architecture RLNN: 3–15–1 was designed, assuming bipolar sigmoidal activation function in neurons of the hidden layer and identity activation function for the output.

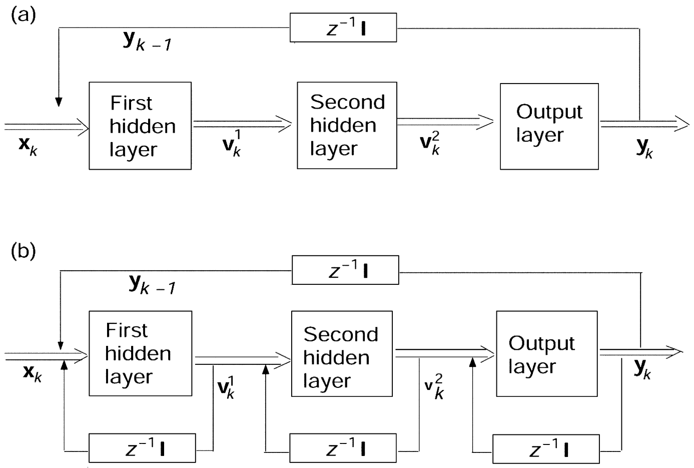

Fig. 1. Multilayer NNs with an autoregressive input y_{k-1}: network FLNN with only feed-forward transmission of signals (a) and recurrent network RLNN with internal time-delay connection (b).

The training process was controlled by decrease of the *MSEV* error:

$$MSEV(s) = \frac{1}{V}\sum_{p=1}^{V}(\bar{d}_p - \bar{y}_p)^2 \tag{3}$$

where: s is the number of epoch; $V = L, T$ is the number of patterns in the training or testing sets, respectively and $\bar{d}_p, \bar{y}_p$ are the target and computed output values for pth pattern. The corresponding *MSE* errors and statistical parameters are listed in Table 1. The obtained errors are significantly smaller than those computed in [9] by the feed-forward network FLNN. In Table 1 it can be seen that using the RDEKF algorithm in network RLNN after $S = 24$ epochs, the admissible error $\epsilon_{\mathrm{adm}} = 1\times10^{-4}$ is attained.

Fig. 2 shows selected spectra computed by the recurrent network using RDEKF algorithm and by feed-forward network learnt by DEKF (training spectrum No.113 is denoted by ARS l #113 and testing spectrum No.11 as ARS t #11). It can be seen that these spectra have shapes similar to the graphics of target spectra (ARS computed from measured accelerograms). Nearly all the neurally computed spectra give the approximation of target curves from below if DEKF was applied. In the case of the RDEKF this conclusion is valid for low values of vibration periods (in a very non-smooth part of ARS graphics).

The neural predictions of the target spectra discussed above are shown in Fig. 3 for the two testing spectra, Nos. 5 and 17. They are shown for the range $T_k \in [0.02, 0.308]$ s that corresponds to frequencies $f_k \in [3.25, 50]$ Hz. These ranges cover spectra of medium-height flat buildings analyzed in Ciesielski et al [8]. On the base of measurements carried out at 13 five-storey buildings of various construction types the basic vibration periods were computed for the range [0.155, 0.294] s.

4.2. Application of Rprop learning method

In order to compare numerical efficiency of Kalman filtering the computation was also performed by a traditional learning method. Following [1] the resilient-propagation (Rprop) method and MATLAB neural network toolbox [9] was used in the training of the same networks FLNN and RLNN of structure 3–15–1. In the case of RLNN the Rprop used in [1] was modified in [5]

Table 1
Errors and statistical parameters for networks FLNN and RLNN

Network (algorithms)	Numbers of epoch S	Errors $MSEV\times10^3$ L	T	Statistical parameters r_T	St ε_T
RLNN	24	1.01	0.88	0.9861	0.2114
(RDEKF)	100	0.41	0.37	0.9874	0.1381
FLNN	188	1.00	1.17	0.9877	0.2445
(DEKF)	500	0.87	1.02	0.9877	0.2282
RLNN	1000	0.42	0.46	0.9870	0.1535
(RpropR)	10000	0.40	0.45	0.9877	0.1519
FLNN	1000	0.57	1.20	0.9683	0.2509
(Rprop)	10000	0.42	0.56	0.9845	0.1681

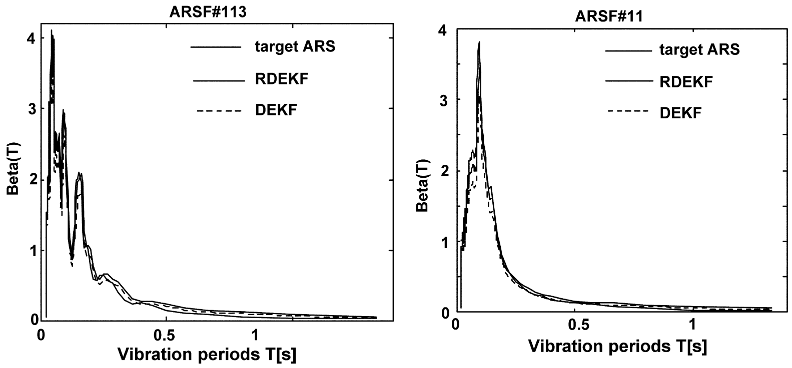

Fig. 2. Target and neurally computed spectra for selected accelerograms from the training set (#113) and testing set (#11).

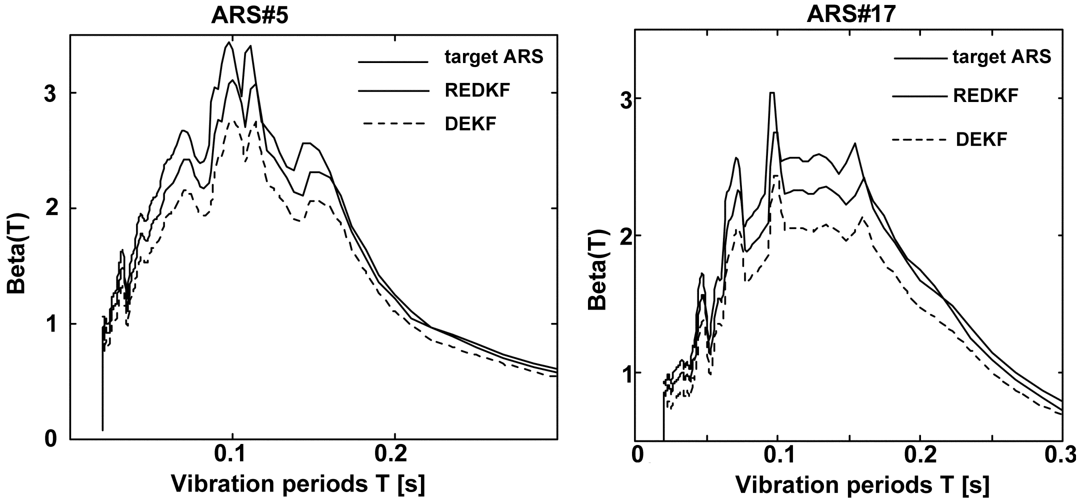

Fig. 3. Target and neurally computed spectra for the selected testing accelerograms #5 and #17.

to procedure Rprop-R in order to take into account the time-delay link v_i (k). Table 1 shows errors computed for the number of epochs $S = 10000$. It can be seen that there are not great differences in errors obtained in networks RLNN and FLNN trained by means of Rprop.

In [1] the network FLNNs without the autoregressive input were applied (instead of β_{k-1} the input T_k was used). Despite much bigger networks the neural approximation was much worse than in the case of the FLNN with the autoregressive input.

To end the discussion it should be said that the Kalman filtering, used as a new learning method, is time consuming. When applying DEKF and 100 epochs the CPU time was about 38% higher than the time needed to carry out 10000 epochs using the Rprop learning method.

5. Final conclusions

1. The Kalman filtering method of network learning enables us to increase accuracy of neurally predicted ARS from mining tremors.
2. Formulation of RDEKF algorithm for the learning of the recurrent layered NN appears to be much more numerically efficient than using DEKF in the feed-forward layered NN.
3. Introducing the autoregressive input (time-delayed input) significantly improves the speed-up of the training process
4. An interesting feature of neural prediction related to the approximation of a part of target ARS from below and above needs further research.

Acknowledgments

Financial support by the Foundation for the Polish Science, Subsidy No. 13/2001 'Application of artificial neural networks to the analysis of civil engineering problems' is gratefully acknowledged.

References

[1] Kuźniar K, Maciag E, Waszczyszyn Z. Computation of response spectra from mining tremors using neural networks. Soil Dynamics and Earthquake Engineering 2004 (in press).
[2] Cheung K, Popplewell N. Neural network for earthquake selection in structural time history analysis. Earthquake Engineering and Structural Dynamics 1994;23:303–396.
[3] Ghaboussi J, Lin CJ. New method of generating spectrum compatible accelerograms using neural networks. Earthquake Engineering and Structural Dynamics 1998;27:377–396.
[4] Lee SC, Han SW. Neural-network-based models for generating artificial earthquakes and response spectra. Computers and Structures 2002;80:1627–1638.
[5] Krok A, Waszczyszyn Z. Prediction of response spectra from mining tremors by neural Kalman filtering (in Polish). In: Proc. 50th Polish Conf of Civil Eng, Krynica–Warsaw, 2004;II:93–100.
[6] Pham PT, Liu X. Neural Networks for Identification, Prediction and Control, London: Springer-Verlag, 1995.
[7] Haykin S, editor, Kalman Filtering and Neural Networks, New York: John Wiley & Sons, 2001.
[8] Ciesielski R, Kuźniar K, Maciag E, Tatara T. Empirical formulae for fundamental natural periods of buildings with load-bearing walls (in Polish). Archives of Civil Engineering 1992;38:291–299.
[9] Neural Network Toolbox for Use with MATLAB, User's Guide, Version 4, Natick MA: The Math Works Inc, 2001.