Computer Assisted Mechanics and Engineering Sciences, 13: 575–582, 2006.

Simulation of hysteresis loops for a superconductor using neural networks with Kalman filtering

Agnieszka Krok
Cracow University of Technology, PhD Course at Faculty of Civil Engineering
ul. Warszawska 24, 31-155 Kraków, Poland

(Received in the final form October 18, 2006)

Kalman filtering is used as a learning method for the training of Feed-forward Layered Neural Networks (FLNN) and Recurrent LNNs (RLNN). These networks were applied to the simulation of hysteresis loops obtained by the experiment on a cable-in-conduit superconductors by the test carried out in a cryogenic press [8]. The training and testing patterns were taken from nine selected, characteristic hysteresis loops. The formulated FLNN: 4-4-5-1 gives the computer simulation of higher accuracy than the standard network FLNN: 4-7-5-1 discussed in [5].

Keywords: Kalman filtering, neural network, simulation, conductor, hysteresis loops

1. Introduction

The cable-in-conduit superconductors made from Nb3Sn material for the International Thermonuclear Experimental Reactor, ITER [9], is investigated. The cable geometry includes sub-cables, and strands in 5 cable blocks, steel jacket and cooling tube. Transverse loading of the conductor due to Lorentz forces is the cause of the mechanical effects and variations of the electrical electrical resistance and contact patterns in the cable, hoop, transverse and shear stresses [8]. These effects cause electromagnetic loss and t decrease of the conductor functioning efficiency. The conductor is examined in a cryogenic press and the cyclic loading with progressive maximal force is applied in order to find the response of the material. One-dimensional model is used. Corresponding laboratory tests, reported in [8], were performed on an isolated cable, Fig. 1a. Measurements of displacements d under applied compressive force F were obtained for 38 cycles during the force driven experiment. It was stated that the first five loops are rather sharply different from the later ones (Fig. 1b) because of the initial irreversible settlement of the strands.

An analytical mechanical model of the conductor was formulated in [7] on the base of Ishlinsky's elastoplastic material. The model turned out to be difficult for the computational purposes since the parameters of each branch of the hysteresis loop have to be calibrated. A much more simple simulation was proposed in [5] where the standard feed-forward layered neural networks were used. In order to increase the accuracy of neural approximation the Kalman filtering (KF) [3] was used in the present paper.

2. Kalman filtering as a NN learning method

Extended KF is based on two equations: (1) process equation, (2) measurement equation. These equations are modified into the following form to use in ANNs,

$$\{\mathbf{w}_i(k+1), v_i(k+1)\} = \{\mathbf{w}_i(k), v_i(k)\} + \boldsymbol{\omega}(k), \tag{1}$$

$$\mathbf{y}(k) = \mathbf{h}\left(\mathbf{w}(k), \mathbf{v}(k), \mathbf{x}(k)\right) + \boldsymbol{\nu}(k), \tag{2}$$

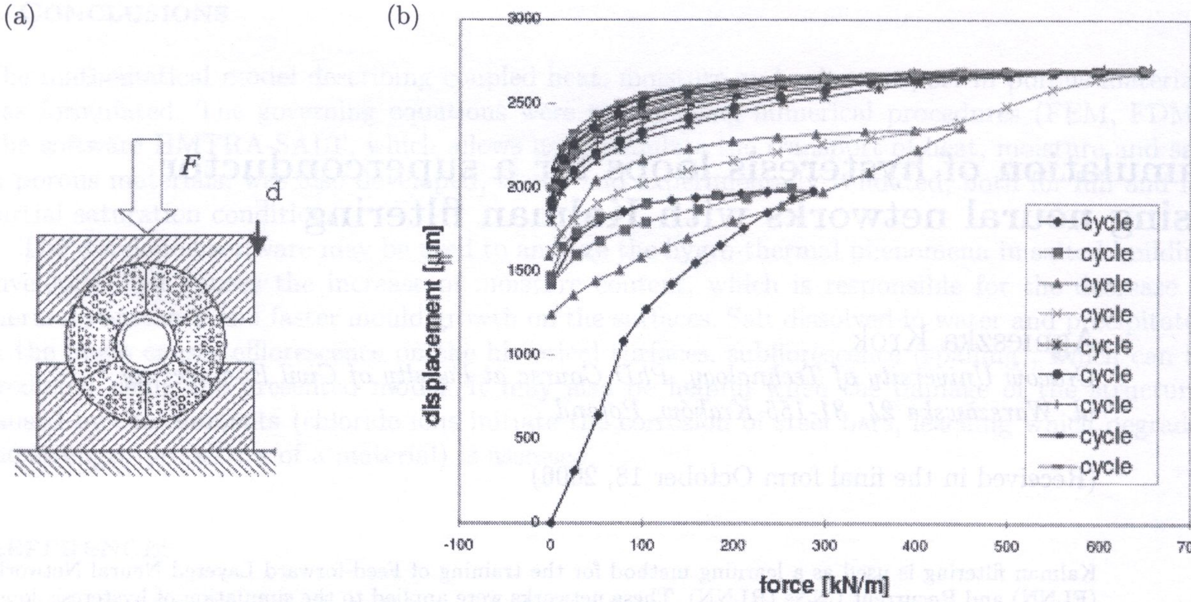

Fig. 1. (a) the tested cable, (b) hysteresis loops of the tested cable

where: k – discrete pseudo-time parameter, i – the number of neuron in ANN; $\mathbf{w}(k) = \{\mathbf{w}_i(k), v_i(k) \mid i = 1, 2, \ldots, n\}$ – state vector corresponding to the set of vectors $\mathbf{w}$ of synaptic weights and biases, and neuron outputs $\mathbf{v}$ for n neurons of ANN; $\mathbf{h}$ – non-linear vector-function of input-output relation; $\mathbf{x}/\mathbf{y}$ – input /output vectors; $\boldsymbol{\omega}(k), \boldsymbol{\nu}(k)$ – Gaussian process and measurement noises with mean and covariance matrices defined by

$$
\begin{aligned}
&E[\boldsymbol{\omega}(k)] = E[\boldsymbol{\nu}(k)] = 0, \\
&E[\boldsymbol{\omega}(k)\,\boldsymbol{\omega}^T(l)] = \mathbf{Q}(k)\,\delta_{kl}\,, \\
&E[\boldsymbol{\nu}(k)\,\boldsymbol{\nu}^T(l)] = \mathbf{R}(k)\,\delta_{kl}\,.
\end{aligned}
\tag{3}
$$

Two algorithms were formulated basing on Eqs. (1) and (2). The algorithm DEKF (Decoupled Extended Kalman Filter) corresponds to the Feed-forward Layered Neural Network and RDEKF (Recurrent DEKF) formulated for learning the Recurrent Network RLNN (Recurrent Layered NN), cf. [3].

2.1. Algorithm RDEKF

KF algorithm induces decoupling state vector into groups. In the paper the decoupling level to the NN neurons (nodes $i = 1, 2, \ldots, n$) is performed. The Recurrent Decoupled Extended Kalman Filter (RDEKF) takes the following form on the base of Eqs. (1) and (2):

$$
\begin{aligned}
\mathbf{A}(k) &= \left[\mathbf{R}(k) + \sum\nolimits_{i=1}^{g} (\mathbf{H}_i^{\mathrm{rec}})^T(k)\,\mathbf{P}_i(k)\,\mathbf{H}_i^{\mathrm{rec}}(k)\right]^{-1}, \\
\mathbf{K}_i(k) &= \mathbf{P}_i(k)\,\mathbf{H}_i^{\mathrm{rec}}(k)\,\mathbf{A}(k), \\
\boldsymbol{\varepsilon}(k) &= \mathbf{y}(k) - \hat{\mathbf{y}}(k), \\
\{\hat{\mathbf{w}}_i(k{+}1), \hat{\mathbf{v}}_i(k{+}1)\} &= \{\hat{\mathbf{w}}_i(k), \hat{\mathbf{v}}_i(k)\} + \mathbf{K}_i(k)\,\boldsymbol{\varepsilon}(k), \\
\mathbf{P}_i(k{+}1) &= \mathbf{P}_i(k) - \mathbf{K}(k)\,\mathbf{H}_i^{\mathrm{rec}}(k)\mathbf{P}_i(k) + \mathbf{Q}_i(k),
\end{aligned}
\tag{4}
$$

where: $\mathbf{K}_i(k)$ – Kalman gain matrix; $\mathbf{P}_i(k)$ – approximate error covariance matrix; $\boldsymbol{\varepsilon}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$ – error vector, where $\mathbf{y}(k)$ is the target vector for the kth presentation of a training pattern; $\hat{\mathbf{w}}_i(k)$, $\hat{\mathbf{y}}(k)$ – estimate of non recurrent weight vector and output vector. The variable $\hat{\mathbf{v}}_i$ is the component associated with the recurrent connection in the ANN. $\mathbf{H}_i^{\rm rec}$ is the matrix of current linearization of Eq. (2):

$$\mathbf{H}_i^{\rm rec} = -\frac{\partial \mathbf{h}_i(k, \mathbf{w}, \mathbf{v})}{\partial \{\mathbf{w}_i, v_i\}}\,, \tag{5}$$

computed at $\{\hat{\mathbf{w}}_i(k), \hat{\mathbf{v}}_i(k)\}$, where $\hat{\mathbf{w}}_i(k)$, $\hat{\mathbf{v}}_i(k)$ are estimators.

2.2. Algorithm DEKF

The terms $\hat{\mathbf{v}}_i(k+1)$ and $\hat{\mathbf{v}}_i(k)$ could be omitted in Eqs. (1)–(2) and then Eq. (5) takes the form

$$\mathbf{H}_i(k) = \frac{\partial \mathbf{h}_i(k, \mathbf{w})}{\partial \mathbf{w}_i}\,. \tag{6}$$

computed at $\hat{\mathbf{w}}_i(k)$.

In both algorithms DEKF and RDEKF the considered parameters for white (Gaussian) noise were adopted:

$$R_k = 7\,\exp\left(-\frac{s-1}{50}\right), \qquad \mathbf{Q}_k = 0.001\,\exp\left(-\frac{s-1}{50}\right)\mathbf{I}, \tag{7}$$

where $\mathbf{I}$ is the identity matrix whose dimension depends on the state vector dimension in particular network layer, s is the number of the learning epoch.

3. Neural simulation of hysteresis loops

The experiment, reported in [8], was carried out for 38 hysteresis loops. From among them nine representative loops (Nos. 1–6, 11, 21, 38) were used in the present paper, similarly as in [5], for the training and testing of FLNNs of different architectures.

3.1. Investigated ANN output and input variables

The following ANNs are examined:

a) Feed-forward Layered Neural Network (FLNN), see Fig. 2a,

b) Recurrent Layered Neural Network (RLNN), see Fig. 2b.

A single output corresponds to the displacement under the applied force, Fig 1a,

$$y = d(k+1). \tag{8}$$

The following input vectors were considered:

i1) $\mathbf{x} = \{\, F(k),\ d(k),\ F(k+1)\,\}$ (9a)

i2) $\mathbf{x} = \{\, F(k),\ d(k),\ F(k+1),\ 1 - \frac{k}{244}\,\}$ (9b)

i3) $\mathbf{x} = \{\, d(k),\ F(k+1),\ 1 - \frac{k}{244}\,,\ l(k)\,\}$ (9c)

The components of the vector $\mathbf{x}$ are: $F(k)$, $F(k+1)$ – values of applied force; k, $k+1$ ($k = 1, \ldots, 244$) – current and subsequent discrete time parameters; $1 - k/244$ – decreasing numbering of patterns. For k appropriate for each loop, parameter $l(k)$ is associated with the number of measurements for this loop, separately from other loops. $l(k)$ takes the subsequent values: $1 - 1/N(1)$, $1 - 2/N(2)$, $\ldots$, $1 - N(l)/N(l)$, where $N(l)$ is the total number of measurements in the lth loop, see Fig. 3.

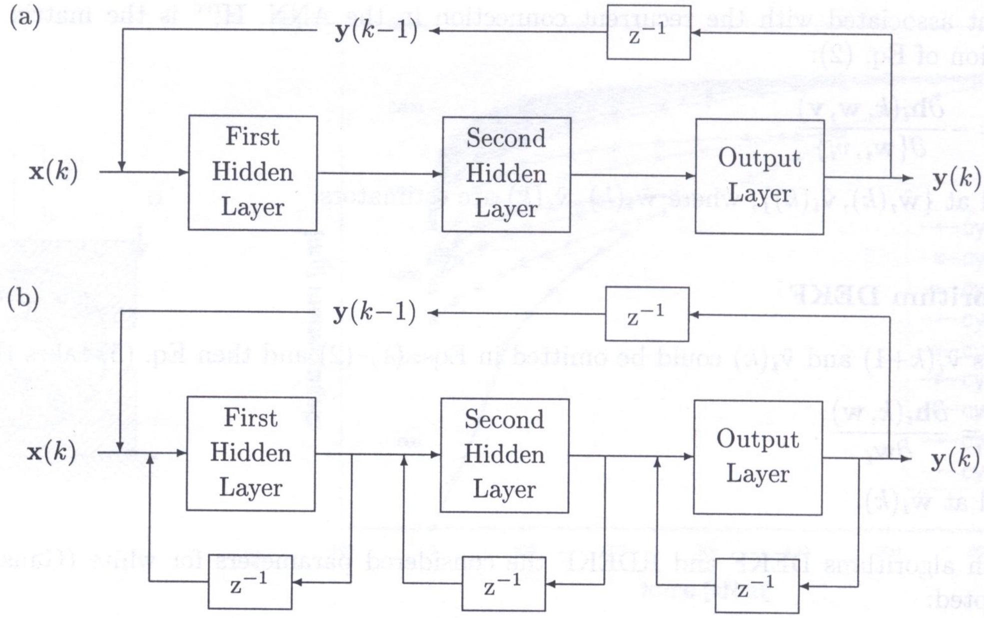

Fig. 2. Architectures of neural Networks (a) FLNN, (b) RLNN structures

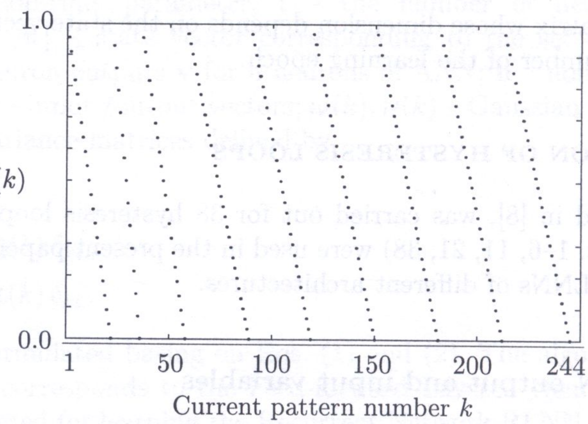

Fig. 3. Values of $l(k)$ for learning and testing patterns

3.2. Learning and testing patterns

The data sets were arranged into the following three groups:

a) 25% or 50% of all patterns, randomly selected from the whole data set (this group corresponds to irregularly placed patterns). For selecting 25% patterns $T = 61$ patterns were used for testing and $L = 244 - 61 = 183$ randomly placed patterns (with respect to k) used for learning. In case of selecting 50% of patterns: $T = L = 122$.

b) The first seven loops containing $L = 180$ patterns for the $k = 1, \ldots, 180$ were used for the learning and $T = 244 - 180 = 64$ patterns for $k = 181, \ldots, 244$ for testing,

c) Two first loops containing $T = 35$ patterns, corresponding to $k = 1, \ldots, 35$ were used for the testing and remaining $L = 244 - 35 = 209$ patterns were explored for learning.

The accuracy of neural approximation was measured by the MSE error (Mean Squared Error)

$$MSEV = \frac{1}{V}\sum_{k=1}^{V}(\mathbf{y}(k) - \hat{\mathbf{y}}(k))^2, \tag{10}$$

where: $V = L, T$ – the number of learning and testing patterns, respectively; $\mathbf{y}(k)$, $\hat{\mathbf{y}}(k)$ – the target and computed output values for kth pattern scaled to the interval [0, 1].

3.3. Results obtained by ANN with different input vectors

Results of FLNN and RLNN learning are shown in Tables 1, 2 and 3. *MSEL* and *MSET* values are obtained after 500 epochs of learning. After extensive introductory computations the early stop was found – the number of stopping epochs $S = 500$. The lowest values from 100 times randomly selected initial weights of the networks are presented. Additional input $1 - k/244$ related to the number of learning pattern enabled us to reach lower errors level comparing the networks with the same numbers of neurons in hidden layers H1 and H2, but smaller number of inputs. For instance, the network FLNN 4-4-5-1 was able to simulate hysteresis loops much better than the network FLNN 3-4-5-1 for all kinds of training and testing sets. The best results of learning for all types of learning and testing sets were obtained while introducing in the input vector both $1 - k/244$ and $l(k)$ parameters.

3.3.1. Case a) of selecting the testing set

In Table 1 there are shown the results of learning and testing for the testing set containing 25% or 50% randomly chosen points, selected from the whole loading history. The presented results were obtained for the ANN input vector numerically more effective. The behaviour of the material under given loading is simulated correctly. It is visible that the ratio of the number of learning and testing patterns practically does not effect the accuracy of neural prediction.

3.3.2. Case b) of selecting the testing set

One layered and two layered FLNN and RLNN were tested, see Table 2. The number of network parameters (*NNP*) ranges from 146 (case 4) to 51 (case 6). Different input vectors were examined. The best result was obtained for the input vector $\mathbf{x} = [\,d(k),\, F(k{+}1),\, 1 - k/244,\, l(k)\,]$, containing both pattern numbering: i.e. $1 - k/244$ and $l(k)$ for $k = 1, \ldots, 244$. Adding the recurrent connection does not improve the accuracy of displacement simulation. Study case 8, corresponding to the

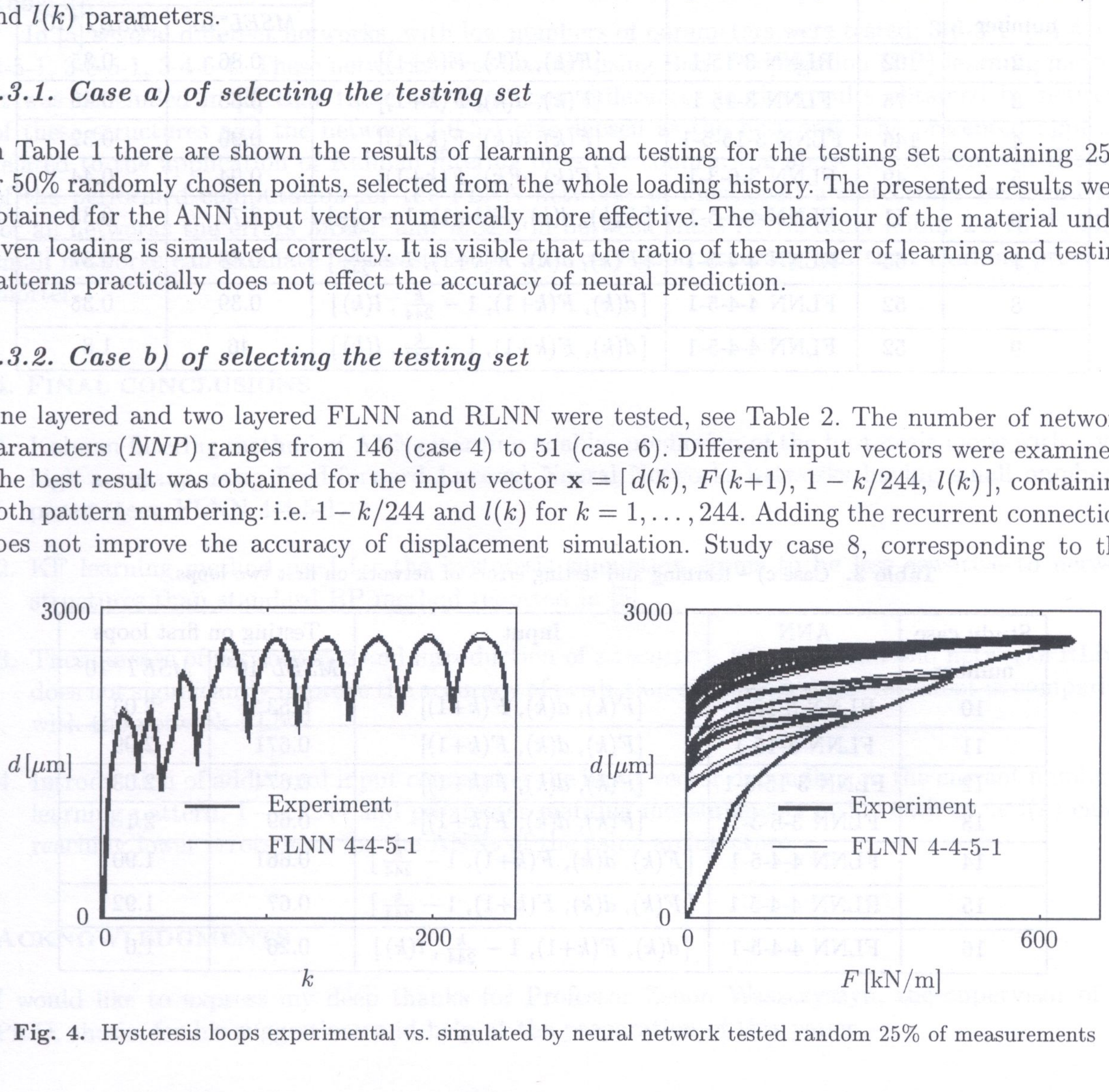

Fig. 4. Hysteresis loops experimental vs. simulated by neural network tested random 25% of measurements

Table 1. Case a) – learning and testing errors of network on randomly chosen 25% and 50% of measurement points

Study case number for (%)*	ANN	Input	Testing on 25% or 50%	
			$MSEL$*10^3	$MSET$*10^3
1 (25%)	FLNN 4-4-5-1	$[d(k),\, F(k{+}1),\, 1-\frac{k}{244},\, l(k)]$	0.27	0.37
1 (50%)	FLNN 4-4-5-1	$[d(k),\, F(k{+}1),\, 1-\frac{k}{244},\, l(k)]$	0.36	0.30

* percentage of selected testing patterns

Table 2. Case b) – learning and testing errors of network on two final loops

Study case number	NNP	ANN	Input	Testing on final loops	
				$MSEL$*10^3	$MSET$*10^3
2	92	RLNN 3-15-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.86	0.35
3	76	FLNN 3-15-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.88	0.46
4	146	FLNN 3-15-5-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.90	0.32
5	49	FLNN 3-6-3-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.94	0.44
6	51	FLNN 4-4-5-1	$[F(k),\, d(k),\, F(k{+}1),\, 1-\frac{k}{244}]$	0.78	0.35
7	63	RLNN 4-4-5-1	$[F(k),\, d(k),\, F(k{+}1),\, 1-\frac{k}{244}]$	0.88	0.37
8	52	FLNN 4-4-5-1	$[d(k),\, F(k{+}1),\, 1-\frac{k}{244},\, l(k)]$	0.39	0.35
9	52	FLNN 4-4-5-1	$[d(k),\, F(k{+}1),\, 1-\frac{k}{244},\, l(k)]$	46	1.8

Table 3. Case c) – learning and testing errors of network on first two loops

Study case number	ANN	Input	Testing on first loops	
			$MSEL$*10^3	$MSET$*10^3
10	RLNN 3-15-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.53	2.03
11	FLNN 3-15-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.671	2.09
12	FLNN 3-15-5-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.674	2.03
13	FLNN 3-6-3-1	$[F(k),\, d(k),\, F(k{+}1)]$	0.69	2.1
14	FLNN 4-4-5-1	$[F(k),\, d(k),\, F(k{+}1),\, 1-\frac{k}{244}]$	0.661	1.90
15	RLNN 4-4-5-1	$[F(k),\, d(k),\, F(k{+}1),\, 1-\frac{k}{244}]$	0.67	1.92
16	FLNN 4-4-5-1	$[d(k),\, F(k{+}1),\, 1-\frac{k}{244},\, l(k)]$	0.26	1.0

network FLNN 4-4-5-1, having small number of parameters $NNP = 57$, seems to by the most effective, see Table 2. A striking result is that the study case 9, corresponding to the network without autoregressive input $d(k)$, gives very high errors of neural learning and prediction, cf. [4].

3.3.3. Case c) of selecting the testing set

This discussed case corresponds to adopting the first two loops for the network testing. Networks learned on regular loops and tested on irregular first and second loops gave higher level of $MSET$ errors when comparing with ANNs testing on final loops. The fitting of simulated cycles to the experimental cycles is better than those computed in [5]. This concerns especially the first four cycles. The abnormal behaviour of the material during the first stage of the loading is simulated very well. Similarly to b) the network FLNN 4-4-5-1 with the input vector $\mathbf{x} = [\, d(k),\ F(k+1),\ 1-k/244,\ l(k)\,]$ input vector (case 16, Table 3) is the most efficient.

3.4. Numerical accuracy of ANN of different architectures

Extensive numerical experiments lead to the conclusion that in the considered problem the accuracy of neural approximation is not strongly affected by the number of neurons in hidden layers (see Table 2).

In [5] several different networks, with low numbers of parameters were tested: 3-6-3-1, 3-5-4-1, 3-4-5-1, 3-6-5-1, 3-4-6-1. These networks were learned using Back Propagation (BP) learning method. It was concluded in [5] that there are significant differences in the results obtained by networks of these structures and the network 3-6-5-1 was chosen as the best one. The presented approach related to the application of Kalman filtering does not confirm the above conclusion. On the base of the performed computation for the FLNN networks of mentioned architectures we found that for all networks the errors $MSET$ and $MSEL$ in between these ANNs differ about $2 * 10^{-4}$. These small networks can simulate hysteresis loops with very good accuracy and their learning process is shorter.

4. Final conclusions

1. Kalman filtering method of ANNs learning enables prediction of the hysteresis loops with a very high accuracy using Feed-forward Layered Neural Networks networks having small number of parameters, FLNN 4-4-5-1.

2. KF learning method used for the hysteresis simulation seems to be less sensitive to network structures than standard BP method reported in [5].

3. The increase of network size and introduction of a recurrent KF algorithm and networks RLNNs does not significantly improve the accuracy of prediction of concerned displacement in comparison with the network FLNN.

4. Introduction of additional input concerning the input vector depending on the current number of learning pattern, $1 - k/244$ and parameter marking measurements inside each cycle $l(k)$ enable reaching lower errors level for the ANNs of the same architecture.

Acknowledgments

I would like to express my deep thanks for Professor Zenon Waszczyszyn, the supervisor of my Ph.D. thesis, for his suggestions and help at the preparation of this paper.

References

[1] U. Galvanetto, V. Naumov, V. Palmov, B.A. Schrefler. Analysis of the mechanical behavior of cable-in-conduit superconductors under transverse cyclic loading. *Int. J. Comput. Civ. Struct. Eng*, **1**(2): 1–10, 2000.

[2] S. Haykin. *Neural Networks, A Comprehensive Foundation*, 2nd ed. MacMillan College Publ., Engle-wood Cliffs, NJ, 1999.

[3] S. Haykin (ed.). *Kalman Filtering and Neural Networks.* Wiley, New York, 2001.

[4] A. Krok, Z. Waszczyszyn. Kalman filtering for neural prediction of response spectra from mining tremors. In: T. Burczynski, W. Cholewa, W. Moczulski (eds), *Recent Development in Artificial Intelligence Methods, AI METH2004*. AI-Meth Series, 157–162, Gliwice, 2004.

[5] M. Lefik. One-dimensional model of cable-in-conduit supercoductors under cyclic loading using artificial neural networks. In: *Fusion Engineering and Design.* Elsevier Science Volume, 60/2, 105–117, 2002.

[6] M. Lefik. *Zastosowania sztucznych sieci neuronowych w mechanice i w inżynierii.* Zeszyty Nauk. Polit. Łódzkiej, 2005, (in Polish)

[7] M. Lefik, B.A. Schlefler. Artificial neural networks for parameter identifications for elasto-plastic model of superconductiong cable under cycling loading. *Comp. Struct.*, **80**: 1699–1713, 2002.

[8] A. Nijuhuis, N.H.W. Noordman, H.H.J. ten Kate. *Mechanical and Electrical Testing of an ITER CS1 Model Coil Conductor under Transverse Loading in a Cryogenic Press.* Preliminary Report. Univ. of Twente, March, 1998.

[9] www.iter.org, official ITER project website.