Appl. Math. and Comp. Sci., 1996, vol.6, No.3, 447–461

CLASSIFIER-APPROXIMATOR MODULAR NEURAL NETWORK FOR ACCURATE ESTIMATION OF DYNAMIC SYSTEM PARAMETERS

ANDRZEJ MATERKA*

Principles of employing feedforward artificial neural networks for fast and robust estimation of dynamic system parameters are reviewed briefly. In this approach, the neural network approximates the mapping from the system observation space into the parameter space. It is pointed out that for the conventional neural network architectures the network size and time needed for its training increase quickly with the estimated parameter range and with the approximation accuracy level required. Moreover, due to the local minima effect, the training process is likely to become prematurely terminated. To overcome these difficulties, a modular neural network architecture is proposed which comprises classifier and approximator modules, both driven by the system under test observations. With this architecture, the domain of the mapping is partitioned into a number of nonoverlapping regions. The classifier makes a decision as to which predefined region the given observation vector belongs to. This information is then used to select an appropriate weight vector (and possibly the structure) of the approximator module, so as to minimise the parameter estimation error locally, within the region identified. A numerical example is presented to show that the proposed approach offers higher estimation accuracy and huge savings in time required for the training.

1. Introduction

Parametric modelling of dynamic systems is a standard technique in almost every area of engineering and science where physical systems of interest are described by respective parameterised mathematical equations. Such equations form the system model and their parameters often have physical significance. Actual parameter values describe the system for diagnostic purposes. Searching for methods and techniques for fast and reliable parameter estimation based on system observation is a crucial problem in many applications comprising e.g. signal and image processing for medical diagnosis support, electronic circuit fault diagnosis, and plant identification in control engineering. It has recently been postulated (Materka, 1992; 1994; 1995a) that artificial neural networks (ANNs) can be employed for the dynamic system parameter estimation. In this approach, samples of the system-under-test (SUT) response to a predetermined stimulus are applied to the input of a feedforward ANN. The samples form an observation vector of the SUT. The observation vector depends on the

* Institute of Electronics, Technical University of Łódź, Stefanowskiego 18/22, 90-924 Łódź, e-mail: materka@lodz1.p.lodz.pl

unknown system parameters. The estimated parameter values are produced at the ANN outputs. Thus the ANN approximates the mapping from the SUT observation space into the SUT parameter space. This inverse mapping is usually unknown. The neural network learns the mapping during the training process. Once the training has been completed, the mapping, encoded in the ANN weights, can be recalled with no need for any iterative calculations. Thus the proposed technique is highly suitable for real-time SUT parameter tracking. It is an attractive alternative to traditional methods of system parameter estimations, e.g. by least-square-error (LSE) model tuning (Cadzow, 1990). The technique has a big potential for many applications, e.g. for mixed-signal VLSI circuit testing (Materka, 1995b).

To the author's knowledge, the usefulness of using ANNs for the dynamic system parameter estimation was first postulated and proven in (Materka, 1992). In this approach, the neural network performs the function of a nonlinear associative memory (NAM), in principle. A relative high accuracy of parameter estimates was demonstrated with this approach (Materka, 1992; 1994; 1995a). Independently, a linear associative memory (LAM) was proposed in (Kalaba *et al.*, 1992) as a means to obtain preliminary parameter estimates for nonlinear systems. Obviously, the LAM-estimated parameter values are of limited accuracy (Kalaba *et al.*, 1992; Tawfik and Durrand, 1994). Using second-order polynomials resulted in the accuracy improvement over the LAM (Kalaba *et al.*, 1992). Both the LAM and NAM offer higher speed when compared to standard techniques for parameter estimation.

On top of its high speed, the new technique offers higher noise immunity when compared with LSE modelling. Namely, it has been proven analytically and verified numerically that by using ANNs the observation-noise-induced *rms* estimation error can be reduced, provided that noisy SUT observations are used for the ANN training (Materka, 1995c; Materka and Mizushina, 1996). On the other hand, the two techniques are fully equivalent if the ANN is trained in noiseless conditions. Having established the main advantages of the ANN-based approach over the traditional techniques of parameter estimation, the present research is focused on optimising the ANN architecture for fast and accurate approximation of the mapping of interest. This paper proposes a combination of classifier and approximator networks that allow intelligent, high-accuracy parameter estimation. The training of the proposed network architecture is shorter in time and controllable, as opposed to the standard backpropagation approach.

2. System Parameter Estimation Using Neural Networks

Consider a system under the test response $y(t)$ to a stimulus $x(t)$, where t denotes the independent variable. Figure 1 shows a system excited by a constant-amplitude step stimulus, where the independent variable is time. The response depends on system parameters θ_k, $k = 1,2,\ldots,p$. The parameters remain constant within an observation interval $[t_1, t_n]$. The objective is to find $\theta = [\theta_1, \theta_2, \ldots, \theta_p]^T$, $\theta \in \Theta \subset \mathbb{R}^p$, given a set of observations

$$y_i = y(t_i) = f(t_i, \theta) + \varepsilon_i, \quad i = 1,2,\ldots,n \tag{1}$$

where Θ is the parameter space, $\Theta = \{\theta : \theta_k^- \leq \theta_k \leq \theta_k^+,\ k = 1, 2, \ldots, p\}$, $f(\cdot)$ is a deterministic SUT function and i stands for observation noise. Noiseless observations will be assumed in this paper, i.e. the variance of the zero-mean random variables ε_i will be zero, $\sigma^2 = 0$. Further work is planned to investigate in detail the noise influence on the proposed modular ANN architecture performance.

The most popular, perhaps, technique of dynamic system parameter estimation employs the least-square error principle to tune the system model by means of minimising a norm of the error between the system and its model responses to the stimulus $x(t)$. This is illustrated by Fig. 2. The norm of the error is a nonlinear function of model parameters and therefore time-consuming iterative calculations have to be carried out each time a new observation vector is acquired. This method is of limited use for fast parameter identification.

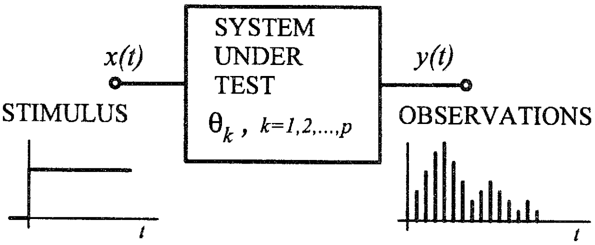

Fig. 1. System under test excited by a stimulus $x(t)$.

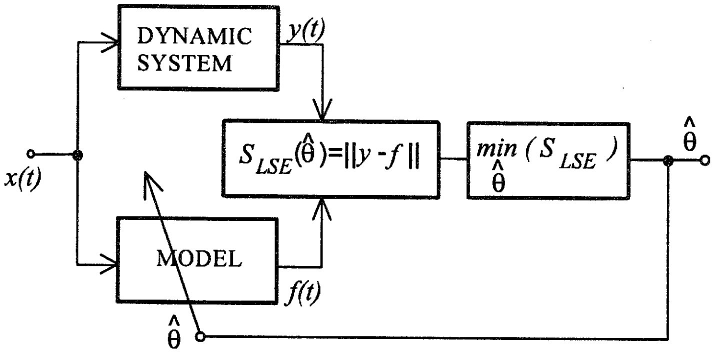
DYNAMIC SYSTEM
y(t)
$S_{LSE}(\hat{\theta})=||y-f||$
$\min_{\hat{\theta}}(S_{LSE})$
$\hat{\theta}$
x(t)
MODEL
f(t)
$\hat{\theta}$

Fig. 2. Principle of system parameter estimation using the LSE technique.

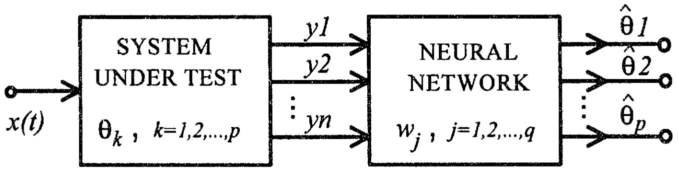

Fig. 3. Neural-network-based estimation of dynamic system parameters: $x(t)$ – stimulus, $\theta = (\theta_1, \ldots, \theta_p)^T$ – unknown parameters, $y = (y_1, \ldots, y_n)^T$ – measured observations, $w = (w_1, \ldots, w_q)^T$ – neural network weights, $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_p)^T$ – estimated parameters.

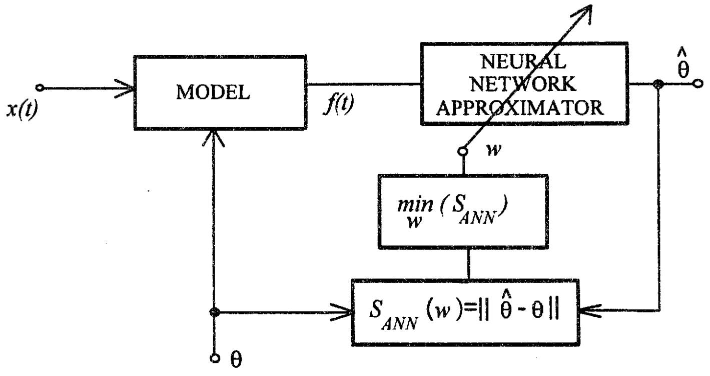

Fig. 4. Principle of ANN training for system parameter estimataion.

Figure 3 illustrates the idea of using neural networks for the system parameter identification (Materka, 1992; 1994; 1995a). The observation vector y forms the input to a feedforward ANN which produces estimates $\hat{\theta} = \hat{\theta}(w, y)$ of the unknown parameters at its output, where $w \in \mathbb{R}^q$ is a weight vector (Haykin, 1994). Thus the ANN approximates a mapping from the observation space into the parameter space. It is assumed that this mapping exists, which can be ensured by a proper selection of stimulus and observation moments (Materka, 1994; 1995a). The ANN learns the mapping during training. The aim of the ANN training process is to minimise the mean-square approximation error over a set of examples $\theta^{(l)} \in \Theta$, $l = 1, 2, \ldots, N$, by adjusting the weights w. This is illustrated in Fig. 4. For each given vector $\theta^{(l)}$, the model response $f(t, \theta(l))$ is calculated, sampled and applied at the input of the ANN. The neural network produces parameter estimates at its output. These estimates depend on the weight vector w. The weights w are adjusted to minimise the norm of the error between the actual and estimated parameters, over a set of examples covering the parameter space. The training itself is a nonlinear minimisation process and may consume a lot of computer time. However, the training has to be performed only once for a given SUT and its parameter space Θ. After training, the ANN weights are frozen. This corresponds to the *recall mode* of the ANN operation, leading to the arrangement shown in Fig. 3. Thus the actual system response forms the input to the neural network in this mode. The network of feedforward type produces parameter estimates at its output in a relatively short time. This time is equal to the time-delay related to signal propagation from the ANN input to the output and can be made very short by employing a hardware-implemented ANN. Thus high speed is an advantageous feature of the technique. No iterative calculations are performed in the recall mode.

Another useful property of the proposed approach originates from the definition of the error function that is minimised in the training mode. Namely, the error is defined based on the difference between the actual and estimated *parameters*. On the contrary, in the LSE approach shown in Fig. 2, the error is based on differences between the actual and modelled circuit *responses*. This distinction leads to different properties of the two estimators in the presence of noise. It has been shown (Materka, 1995c; Materka and Mizushina, 1996) that if the ANN is trained on noisy observations, the noise-induced identification error can be significantly lower in the recall mode, compared with the error of LSE model tuning.

The properties of the proposed technique have been investigated by means of computer simulations (Materka, 1995d; Materka and Strzelecki, 1995) and measurements (Materka, 1995a; 1995b) to confirm the analytically predicted performance. Linear (Materka, 1995a; 1995b; 1995d) and nonlinear (Materka and Mizushina, 1996; Materka and Strzelecki, 1995; Materka, 1995e) system parameters have been successfully identified by employing a popular multilayer perceptron ANN architecture (Haykin, 1994). In particular, the CUT response time delay can be an identified parameter (Materka, 1995a; 1995b; 1995e) which is normally not easy to identify using traditional methods (Unbehaunen and Rao, 1987). An interesting application of ANN approximators to finding parameters of CMOS circuits by processing transients in the power supply current has been described in (Materka and Strzelecki, 1995) This approach can reduce the number of circuit pins required for internal parameter identification. The speedup of the technique compared with the LSE model fitting, with ANNs simulated by a PC (i.e. serial) computer, ranged from 500 to 40000, depending on the complexity of CUT equations (Materka, 1995a; 1995c; 1995e). One can say that basic properties of the technique, which itself has a big potential for a number of applications, have been understood. Present research is focused on searching for efficient means of its practical implementation. Some of the related topics are pointed out in the following section.

3. Classifier-Aproximator ANN Architecture

It has been observed that for popular ANN architectures, e.g. for a multilayer perceptron (Haykin, 1994), the number of neurones required to maintain a given level of error increases with the number of unknown system parameters and with the parameter range (Materka, 1992; 1994; 1995a). Similarly, system parameters that depend on the observations in a highly nonlinear manner require larger-size ANNs when compared with the parameters that are linear functions of the SUT response. with the ANN complexity increased, the size of the weight vector increases, too. As a result, the convergence of the training process, which is basically a numerical minimisation of a multivariable error function, becomes slower. Moreover, with the increased number of optimised variables (weights) there appear new local minima of the error function. Their presence increases chances for premature termination of the training process. Due to these effects, the concept illustrated in Fig. 3 is of limited practical use for systems with a large parameter number p and for cases where the parameter space is of high volume (i.e. where each parameter range is wide). (Similar effects take place in situations where a Taylor series expansion is used to find approximate values of a continuous function of a single variable α over a given interval of α (Fortuna *at al.*, 1982). As the length of the interval increases, more Taylor-series terms are needed to maintain a given level of the approximation error.) Another factor that has to be taken into account at the increased number of ANN weights (ANN degrees of freedom) relates to the necessity of increasing the number of training examples (Materka, 1993); otherwise the new degrees of freedom would not be under control. This factor also contributes to the increased demand for the training time.

In this paper, a modular ANN architecture is proposed to overcome the difficulties experienced, without loosing the high speed and high noise immunity of the ANN-based technique. The architecture is illustrated in Fig. 5 where the ANN is split into two parts: the classifier module and the approximator module. Both modules take system observations as their input signals. The domain of the mapping, in the observation space, is interpreted as a union of m non-overlapping n-dimensional regions S_i, $i = 1, 2, \ldots, m$. The approximator network is optimised to provide high approximation accuracy separately for each region, resulting in m weight vectors w^i, $i = 1, 2, \ldots, m$. Any parameter range corresponding to any of the regions is narrower than the corresponding parameter range for the whole observation domain. Thus the mapping of interest becomes less complex within each region — in a sense it is a smoother function there (locally). It requires a smaller number of neurons, smaller number of adjustable weights and smaller number of training examples to achieve a given level of approximation error. The task of the approximator training becomes decomposed into m separate tasks of training simpler neural networks. This approach gives savings in total time required for the training and makes the whole training process controllable.

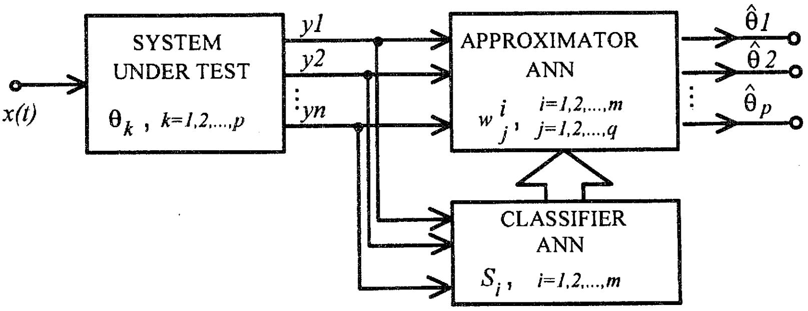

Fig. 5. Proposed modular ANN architecture for system parameter estimation.

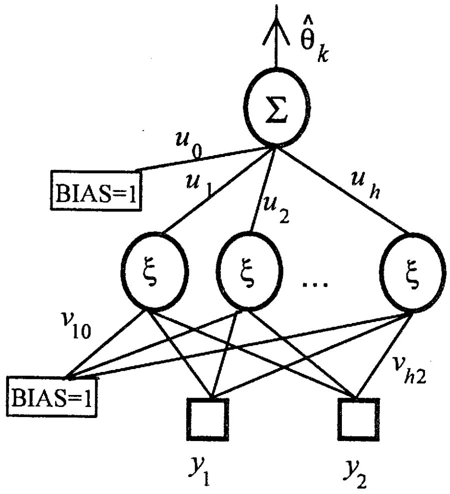

Fig. 6. A single-hidden layer MLP ANN architecture.

In the recall mode, any observation vector measured is first assigned (by the classifier network) to one of the predefined regions. Next, the corresponding weight vector is loaded to the weight memory of the approximator module. This module is then used to estimate system parameters given the observation vector at its input.

4. Computer Simulation

As an example, consider a two-parameter multi-exponential system

$$f(t,\theta) = 25\left[\exp\left(-\frac{t}{\theta_1}\right) - \exp\left(-\frac{t}{\theta_2}\right)\right] \tag{2}$$

over the parameter space Θ : $6.0 \le \theta_1 \le 14.0$, $0.2 \le \theta_2 \le 1.0$. This system was discussed for the three-parameter case in (Materka, 1994; Materka, 1995a). The observations were taken at $t_1 = 0.558$ and $t_2 = 2.613$. Without any loss of generality, the assumption of $p = 2$ facilitates graphical presentation of the results. The MLP architecture (Haykin, 1994) shown in Fig. 6, was first used with linear neurons in the output layer for global parameter estimation by means of ANNs (Fig. 3). Each parameter requires one network of the type presented in Fig. 6. Hence, in this numerical experiment, the k-th parameter of system (2) was estimated as

$$\hat{\theta}_k(w_k, y) = u_{k0} + \sum_{j=1}^{h} u_{kj}\xi\left(v_{kj0} + \sum_{i=1}^{n=2} v_{kji}y_i\right) \tag{3}$$

where $w = (w_1,\dots,w_q)^T = (u_{k0}, u_{k1},\dots,u_{kh}, v_{k10}, v_{k11},\dots,v_{khn})^T$ is the weight vector, $k = 1,2$ whereas $\xi(\cdot)$ denotes the sigmoidal function of the MLP neurons and h is their number. For the ANN training and testing 32 and 100 example values of each parameter were respectively selected to cover the parameter space, uniformly over each parameter range. Thus the training and testing set sizes were $N = 1024$ and $N' = 10000$, respectively. A two-stage training procedure was employed composed of the standard back-propagation algorithm (Haykin, 1994) followed by the multivariable function minimisation (Materka, 1992; 1994; 1995a). All the calculations were performed using a PC 486/66 computer.

The estimation accuracy increased with the number of neurons h, at the expense of an increased training time, as demonstrated in Fig. 7. The maximum absolute testing error less than an arbitrarily selected small value of 1.0% of the parameter nominal value ($\theta_1 = 10.0$) was obtained using ANNs with $h = 7$ (i.e. with $q = 29$ weights). A further increase in h did not produce any significant error reduction. Instead, the training time increased to about 3 hours of calculations and the results of the training process became somewhat unpredictable. These effects were attributed to the more pronounced local minima problem and to the fact that (3) did not describe any orthogonal basis for function approximation.

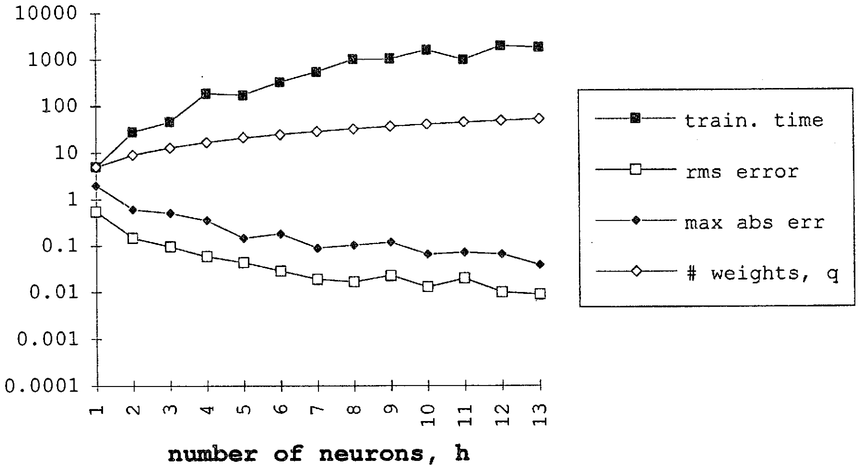

Fig. 7. Results of global estimation of the parameter θ_1 using an MLP with sigmoidal neurons.

To investigate the potential performance of the proposed architecture of Fig. 5, a Kohonen-type classifier ANN was used as a Learning Vector Quantiser (LVQ) (Haykin, 1994). In the case of the example system (2), the classifier had two input nodes, each corresponding to a respective element of the two-dimensional observation vector. The number of output nodes was equal to $m = 9$, which was the number of regions (cells) the whole observation domain was partitioned into. Each output node was allocated to a code book vector $\boldsymbol{y}_i^* = (y_{i1}^*, y_{i2}^*)^T$, $i = 1, 2, \ldots, m$.

For the training of the classifier network, the system parameter vectors θ were repeatedly taken at random from the parameter space Θ. The corresponding observation vectors $\boldsymbol{y}(\theta) = (y_1, y_2)^T$ were then calculated using (2). For each random value of θ, the Euclidean distance

$$d(i) = \sqrt{(y_1 - y_{i1}^*)^2 + (y_2 - y_{i2}^*)^2} \tag{4}$$

between the corresponding observation vector and each of the code book vectors $\boldsymbol{y}_i^*$, $i = 1, 2, \ldots, m$ was computed. Let

$$c = \arg \min_{i \in [1,m]} \left\{ d(i) \right\} \tag{5}$$

be the index of a codebook vector which is the closest one to the observation vector $\boldsymbol{y}$. This codebook vector $\boldsymbol{y}_c^*$ was modified to train the classifer, as follows (Haykin, 1994):

$$\boldsymbol{y}_c^{*\text{new}} = \boldsymbol{y}_c^{*\text{old}} + \eta \left(\boldsymbol{y} - \boldsymbol{y}_c^{*\text{old}} \right) \tag{6}$$

where $\eta > 0$ is a small constant. The other codebook vectors, $\boldsymbol{y}_i^*$, $i = 1, 2, \ldots, m$, $i \neq c$ were not changed for a given observation vector. Figure 8 illustrates an initial distribution of $m = 9$ codebook vectors that correspond to $m = 9$ points (θ_1, θ_2) which form a uniform rectangular lattice over the parameter space Θ. The distribution of the codebook vectors obtained after 500,000 training iterations for $\eta = 0.005$ is shown in Fig. 9.

In the recall mode of the classifier, the distance (4) was calculated for each observation vector acquired. The vector $\boldsymbol{y}$ was classified as belonging to an observation region S_c, where c denotes the index of an observation vector of lowest distance to $\boldsymbol{y}$. The mapping $\theta = \theta(\boldsymbol{y})$ was approximated locally within the region S_c by the approximator in Fig. 5. In the case of Fig. 3, as discussed earlier and illustrated in Fig. 7, the approximation was performed globally, over the whole domain of the mapping in the observation space.

To compare the complexity of the approximation task for the two architectures (shown in Fig. 3 and Fig. 5, respectively), consider the higher-order residuals of the mapping for the parameter θ_1 of system (2):

$$\Delta_1 = \theta_1 - \theta_{c1}^* - \left.\frac{\partial \theta_1}{\partial y_1}\right|_{\boldsymbol{y}_c^*} (y_1 - y_{c1}^*) - \left.\frac{\partial \theta_1}{\partial y_2}\right|_{\boldsymbol{y}_c^*} (y_2 - y_{c2}^*) \tag{7}$$

The residual Δ_1 is the difference between the actual parameter value and its value approximated by a truncated Taylor series expansion of the mapping around the appropriate codebook vector. If the mapping were linear, the residual (7) would be zero, $\Delta_1 \equiv 0$. In such a case, a linear combiner ANN would perfectly perform the approximation task. Figures 10 and 11 compare the values of Δ_1 calculated for the global approximator (Fig. 3) and for the proposed architecture (Fig. 5) employing a Kohonen net LVQ classifier, respectively. One can see that employing the classifier makes the mapping closer to a linear relationship and reduces the demand for higher-order approximation capabilities.

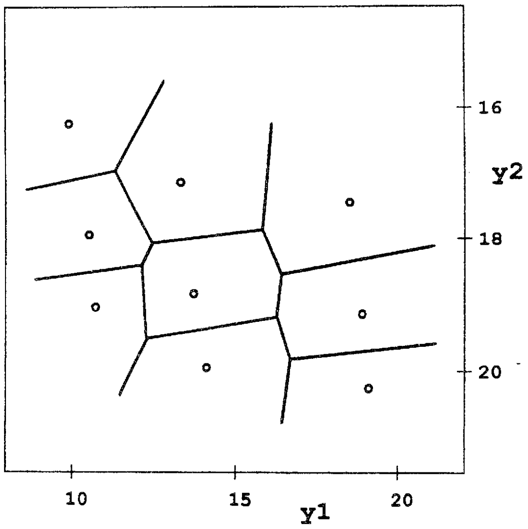

Fig. 8. Code book vectors and the tessellation of the observation plane corresponding to a lattice of uniformly distributed points in the parameter space.

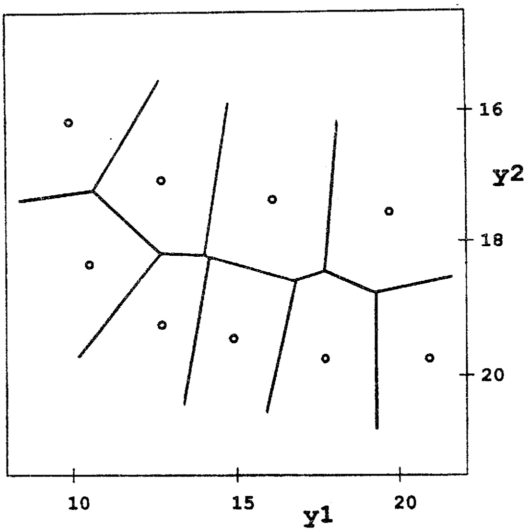

Fig. 9. Code book vectors and the tessellation of the observation plane corresponding to the Kohonen net learning vector quantizer solution.

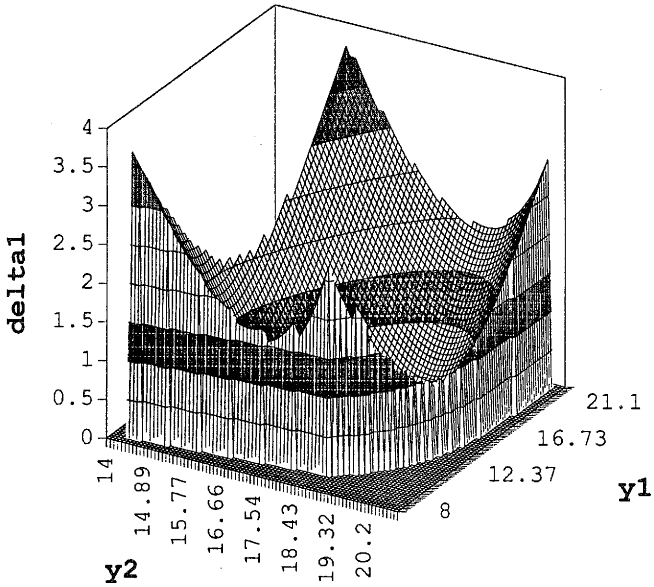

Fig. 10. Residuals (7) for global estimation of parameter θ_1 by ANN of Fig. 3.

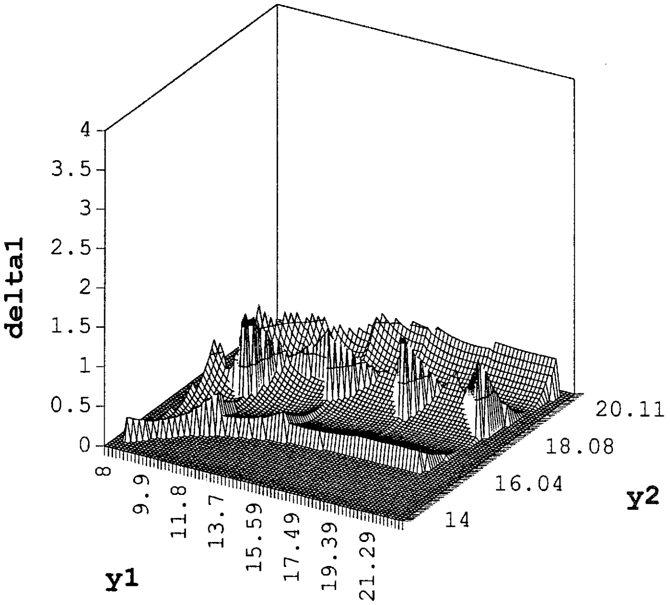

Fig. 11. Residuals (7) for global estimation of parameter θ_1 by ANN of Fig. 5.

Two ANN types for function approximation in the arrangement of Fig. 5 were compared, an MLP with $h = 2$ neurons and an architecture based on rational function (RF) approximation (Leung and Haykin, 1993). The k-th parameter, $k = 1, 2$ at the output of the RF network was estimated in the i-th observation region as

$$h_k(w_k^i, y) = \frac{w_{k1}^i + w_{k2}^i y_1 + w_{k3}^i y_2 + w_{k4}^i y_1^2 + w_{k5}^i y_1 y_2 + w_{k6}^i y_2^2}{1 + w_{k7}^i y_1 + w_{k8}^i y_2} \tag{8}$$

A comparison of the performance of the two approximators applied to the parameter estimation task can be made referring to Figs. 12 and 13, as well as to Table 1. The MLP architecture is clearly less efficient than the RF network, both in terms of the

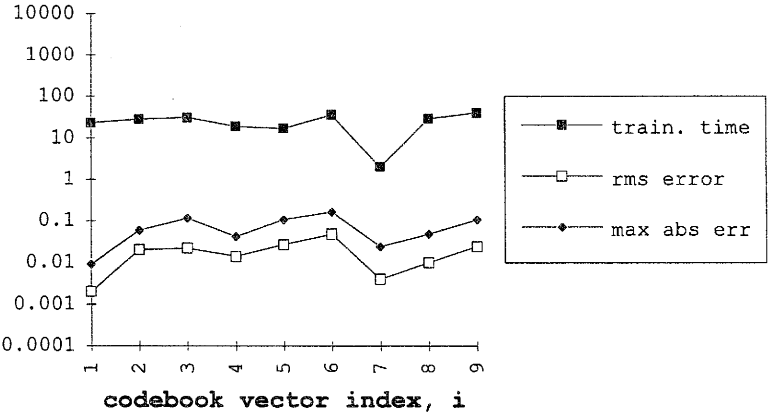

Fig. 12. Results of θ_1 parameter estimation using the architecture of Fig. 5 with an MLP approximator.

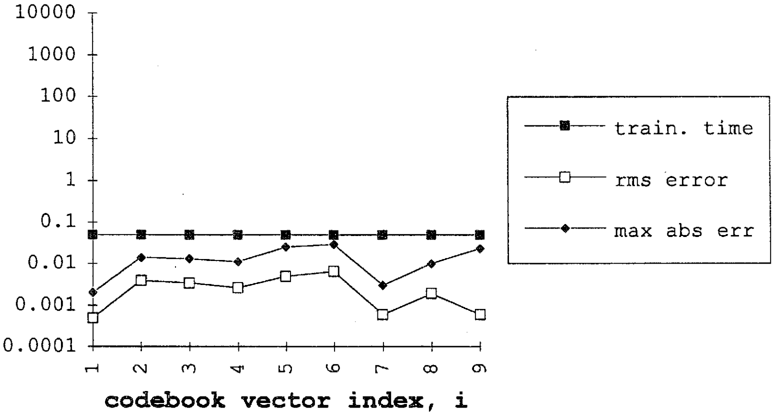

Fig. 13. Results of θ_1 parameter estimation using the architecture of Fig. 5 with an RF-ANN approximator.

Tab. 1. Computer-simulated estimation of θ_1 by the ANN of Fig. 5 using a Kohonen net classifier with $m = 9$ classes.

Mean values over all classes

Approximator architecture	training time [s]	rms error	max absolute error
MLP ($q = 9$)	25.0	0.0190	0.075
RF ($q = 8$)	0.05	0.0034	0.014

Maximum values
(worst-case results over the whole parameter space)

Approximator architecture	training time [s]	rms error	max absolute error
MLP ($m = 9$)	40.0	0.0480	0.166
RF ($q = 8$)	0.05	0.0065	0.029

accuracy level achieved and the time required for the training, in the example of system (2) under consideration. (As in the case of Fig. 3, the performance of the proposed architecture was tested at $N' = 10000$ points uniformly covering the parameter space.) Similar results were obtained for a classifier using codebook vectors of Fig. 8, instead of those shown in Fig. 9. This indicates that in some cases the observation space can be appropriately tessellated based simply on a lattice of points uniformly covering the parameter space.

The results presented in Fig. 7, compared with those shown in Fig. 13 and in Table 1, demonstrate advantages of the modular classifier-approximator architecture as a means for accurate, real-time estimation of dynamic system parameters. Namely, by employing the proposed modular arrangement, the parameter estimation error was reduced by a factor of 5, approximately, with a simultaneous reduction of the time needed for the ANN training. Similar advantages are expected in the case of parameter estimation of other types of dynamic systems. Applying the proposed technique to systems whose number of unknown parameters is larger than $p = 2$ is one of the current research topics.

5. Conclusion

A modular ANN architecture has been proposed for parameter estimation of dynamic systems. By using this architecture, the duration of the training process can be reduced significantly while maintaining high estimation accuracy. A numerical example of a multi-exponential system model was presented to confirm the usefulness of the architecture. Two ANN types were compared through a multivariable function approximation. It was found that for a given number of adjustable weights, the popular multilayer perceptron (MLP) network gives less accurate parameter estimation and requires much longer training times as compared to networks based on the concept of rational function (RF) approximation. Interestingly, the RF networks can be trained by solving a set of linear equations, which takes much less time than nonlinear programming does and helps to avoid the problem of local minima. Thus the architecture of choice is a classifier module using e.g. a Kohonen network and an RF-type ANN as an approximation module. Further work will focus on the analysis of noise immunity of the modular architecture, on issues related to its hardware implementation, as well as on the application of the technique to parameter estimation of more complex systems.

References

Cadzow A. (1990): *Signal processing via least-square modelling.* — IEEE ASSP Magazine, v.7, No.4, pp.12–31.

Fortuna Z., Macukow B. and Wąsowski J. (1982): *Numerical Methods.* — Warsaw: WNT (in Polish).

Haykin S. (1994): *Neural Networks — A Comprehensive — Foundation.* — New York: Macmillan.

Kalaba B., Lichtenstein Z., Simchony T. and Tesfatsion L. (1992): *Linear and nonlinear associative memories for parameter estimation.* — Information Sciences, v.61, No.1 pp.45–66.

Leung H. and Haykin S. (1993): *Rational function neural network.*— Neural Computation, v.5, No.9 pp.928–938.

Materka A. (1992): *Application of neural networks to dynamic system parameter estimation.*— Proc. 14th Ann. Int. Conf. *IEEE Eng. Med. Biol. Soc.*, v.3, Paris, France, pp.1042–1045.

Materka A. (1993): *Smoothness constraints for training neural-network-based parameter estimators.*— 5th Australian Conf. *Neural Networks*, Brisbane, Australia, pp.210–213.

Materka A. (1994): *Application of artificial neural networks to parameter estimation of dynamical systems.* — Proc. IEEE Conf. *Instr. Meas. Tech.*, Hamamatsu, Japan, pp.123–126.

Materka A. (1995a): *Time delay estimation: Least-square error versus neural-network-based techniques.* — 8th Int. Symp. System-Modelling-Control, Zakopane, Poland, v.3, pp.79–84.

Materka A. (1995b): *New method of dynamic system parameter identification using artificial neural networks.* — Quartely of Electronics and Telecomunication, v.41, No.1, pp.45–68 (in Polish).

Materka A. (1995c): *Neural-network technique for parametric testing of mixed-signal circuits.* — Electronics Letters, v.31, No.3, pp.183–184.

Materka A. (1995d): *On noise-induced error of system parameter estimation using artificial neural networks.* — 8th Int. Sym. *System-Modelling-Control*, Zakopane, Poland, v.3, pp.85–90.

Materka A. (1995e): *Parametric fault identification in mixed-signal circuits using feed-forward neural networks.* — 2nd Advanced Training Course *MixVLSI'95*, Kraków, Poland, pp.84–89.

Materka A. and Mizushina S. (1996): *Parametric signal restoration using artificial neural networks.* — IEEE Trans. Biomedical Engineering, v.43, No.4, pp.357–372.

Materka A. and Strzelecki M. (1995): *Parametric testing of CMOS circuits by ANN-based processing of transient power-supply current.* — IEEE Int. *Mixed Signal Testing* Workshop, Grenoble, France, pp.223–227.

Unbehauen H. and Rao G. (1987): *Identification of Continuous Systems.* — Amsterdam: North-Holland.

Tawfik B. and Durrand D.M. (1994): *Nonlinear parameter estimation by linear association: application to five-parameter passive neuron model.* — IEEE Trans. Biomedical Engineering, v.41, No.5, pp.461–469.