GENERALIZED REGRESSION NEURO–FUZZY NETWORK

Ye. Bodyanskiy, N. Teslenko

Abstract: Generalized Regression Neuro-Fuzzy Network, which combines the properties of conventional Generalized Regression Neural Network and Adaptive Network-based Fuzzy Inference System is proposed in this work. This network relates to so-called "memory-based networks", which is adjusted by one-pass learning algorithm.

Keywords: memory-based networks, one-pass learning, Fuzzy Inference Systems, fuzzy-basis membership functions, neurons at data points, nonlinear identification.

ACM Classification Keywords: F.1 Computation by abstract devices - Self-modifying machines (e.g., neural networks), I.2.6 Learning - Connectionism and neural nets, G.1.2. Approximation – Nonlinear approximation.

Introduction

Nowadays neural networks have wide spreading for identification, prediction and nonlinear objects control problems solving. Neural networks possess universal approximating abilities and capabilities for learning by the data that characterize the functioning of investigating systems. The situation becomes sharply complicated in the case, when the data are fed in real time, their processing must be simultaneous with functioning of the plant, and the plant is nonstationary. It's clear, that conventional multilayer perceptron, that is universal approximator, isn't effective in this case, so Radial Basis Functions Networks (RBFN) can be used as its alternative [1-3]. These networks are also universal approximators, and their output is linearly dependent on tuned synaptic weights. In this case, recurrent least squares method or its modifications can be used for their real time learning. These procedures are second-order optimization algorithms, which provide quadratic convergence to the optimal solution. At the same time, practical application of RBFN is bounded by so-called curse of dimensionality as well as appearance of "gaps" in the space of radial-basis functions (RBF) that lead to appearance of regions where all neurons of the network are inactive.

So-called, space partition of unity, implemented by Normalized Radial Basis Functions Networks (NRBFN), in which output signal is normalized by the sum of outputs of all neurons, is used to avoid such a "gaps" [4]. Given networks are learned using recurrent gradient algorithms that have slow rate of convergence and possibility of getting to the local minima as their common drawback.

Thus, these neural networks and many others that use recurrent learning procedures and united by general name "optimization-based networks" may be inefficient in problems of adaptive identification, prediction and real-time control, when the information is fed for processing with sufficiently high frequency. In this case, these networks have not time to learn and are unable to follow changing parameters of a plant.

The so-called "memory-based networks" are the effective alternative to "optimization-based networks" and Generalized Regression Neural Network (GRNN), proposed by D. F Specht [5], is the brightest representative of these networks.

At the basis of this network lies the idea of Parzen windows [6], kernel estimates of Nadaraya-Watson [7-9] and nonparametric models [10]. Its learning consists of one-time adjustment of multidimensional radial-basis functions (RBF) at points of unit centered hypercube, which are specified unambiguously by the learning set. Therefore, these networks can be referred to, so-called, just-in-time models [4], which are adjusted by one-pass learning algorithm. Being similar to NRBFN by the architecture, GRNN learns much faster, placing the centers of RBF at the points with coordinates that are determined by input signals of a plant using principle “neurons at the data points” [11] and with RBF heights, which coincide with corresponding values of plant output signal. High learning rate of GRNN provides their effective using in the real-time problems solving [12,13].

For the solving of nonlinear plant identification problem

$$y(k) = F(x(k))$$

where $y(k)$, $x(k)$– scalar and $(nx1)$-vector of output and input signals correspondingly in the instant time $k=1,2,\ldots$, $F(\bullet)$ – unknown nonlinear operator of the plant, it is necessary to form learning sample $\{x^*(k), y^*(k)\}$, $k=1,2,\ldots,l$, whereupon it is possible to get the estimate $\hat{y}(k)$ of the plant response $y(k)$ to arbitrary input signal x in the form

$$\hat{y}(x) = \frac{\sum_{k=1}^{l} y^*(k)\varphi(D(k))}{\sum_{k=1}^{l} \varphi(D(k))} \tag{1}$$

where $D(k)$ – distance measure in accepted metrics between x and $x^*(k)$, $\varphi(\bullet)$ – some kernel function, usually, Gaussian. Conventionally the Euclidean metrics is used as a distance

$$D^2(k) = \sum_{i=1}^{n} \left(\frac{x_i(k) - x_i^*(k)}{\sigma(k)} \right)^2$$

(here $\sigma(k)$ – scalar parameter, which determines the receptive field radius of kernel function $\varphi(\bullet)$), although in more common case it is possible to use Minkowski metrics

$$D^p(k) = \sum_{i=1}^{n} \left| \frac{x_i - x_i^*(k)}{\sigma(k)} \right|^p, \; p \geq 1 .$$

Thus, GRNN converges asymptotically to optimal nonlinear regression surface with the growing of learning sample size [9].

GRNN learning process can be organized easily in real time. In this case the learning pairs $x^*(k)$, $y^*(k)$ are fed to the network sequentially, forming new radial-basis function-neurons. At the same time, the distance between newly formed and already existing functions is estimated gradually. If this distance is smaller than threshold value r, that is defined in advance, new neuron isn’t included in the network. The main problems concerned with GRNN using are defined by possible curse of dimensionality. Growing of the learning sample size l and the difficulties with correct definition of parameter r, which is sufficiently difficult to choose and interpret in multidimensional space, are the causes of it.

Neuro-Fuzzy Systems (NFS) are the natural expansion of artificial neural networks [14-15]. They combine the neural networks learning abilities with transparence and interpretability of the Fuzzy Inference Systems (FIS). Generally, FIS represents fuzzy models, which are learned by observations data of plant inputs and outputs, using univariate Fuzzy Basis Functions (FBF) instead of multidimensional RBF. In common case FBF are bell-shaped (usually Gaussian) membership functions, which are used in Fuzzy Logic. Using of bell-shaped FBF allows us to combine local features of the kernel functions with the properties of sigmoidal activation functions that provide global approximation properties [16]. Having the approximating abilities of RBFN [15], NFS subject to curse of dimensionality with less degree, that provides them advantage in comparison with neural networks.

Among Neuro-Fuzzy Systems (NFS) Adaptive Network-based Fuzzy Inference System (ANFIS) have got wide spread [17]. ANFIS has five-layer architecture, whose synaptic weights are tuned similarly to RBFN. The adjusting possibility of FBFs using error back-propagation algorithm is provided in this system too. ANFIS and many other similar neuro-fuzzy systems [4,15,16] are typical representatives of the optimization-based networks family, which are characterized by insufficient learning rate.

Lattice-based Associative Memory Networks (LAMN) [18, 19] are the representatives of memory-based networks, whose output signal is formed on basis of univariate bell-shaped functions uniformly distributed on axes of n-dimensional input space. As a result of aggregation operation multidimensional FBFs are formed, whose centers are also uniformly distributed in multidimensional space, and their layout doesn't depend on characteristics of learning sample.

The goal of this work is the development of Generalized Regression Neuro-Fuzzy Network (GRNFN), which represents by itself NFS and learns as GRNN that provides it approximating properties of ANFIS with learning rate of memory-based networks.

The Generalized Regression Neuro-Fuzzy Network architecture

The architecture of Generalized Regression Neuro-Fuzzy Network is illustrated on Fig. 1 and consists of five sequentially connected layers. First hidden layer is composed of l blocks with n FBF in each and realizes fuzzification of the input variables vector. Second hidden layer implements aggregation of membership levels that are computed in first layer, and consists of l multiplication blocks. Third hidden layer – the layer of synaptic weights that are defined in special way. Fourth layer is formed by two summation units and computes the sums of output signals from the second and third layers. Finally, normalization takes place in fifth (output) layer, as a result of which, the output network signal is computed.

One can see, that the architecture of GRNFN coincides with the architecture of L.-X. Wang–-J.M. Mendel neuro-fuzzy system [20], which, in turn, is the modification of zero-order T. Takagi–M. Sugeno fuzzy inference system [21]. However, if NFS is learned using one or another optimization procedures, GRNFN is adjusted using one-pass learning algorithm.

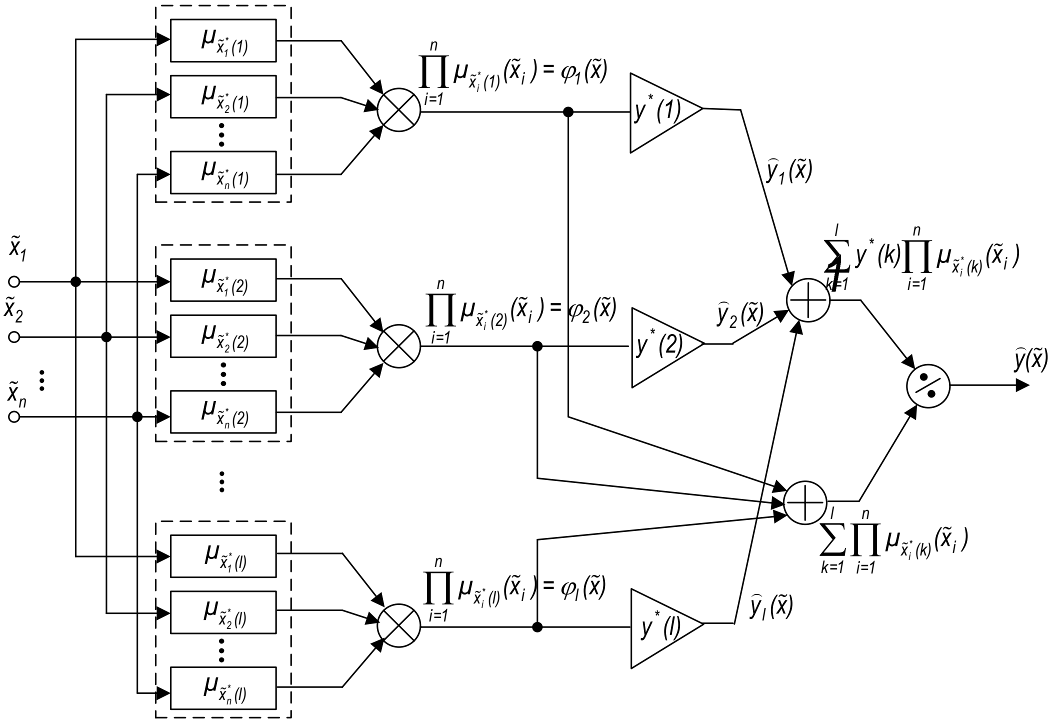


Fig.1 – Generalized Regression Neuro-Fuzzy Network.

Generalized Regression Neuro-Fuzzy Network learning

Since GRNFN belongs to memory-based networks, its learning is based on principle “neurons at data points” that makes it extremely easy and fast.

Learning sample vectors $x^*(1),...,x^*(k),...,x^*(l)$ are normalized in advance on unit centered hypercube so, that

$$x_i^{*min} \le x_i^*(k) \le x_i^{*max}, \; i = 1,2,...,n,$$

$$-0,5 \le \tilde{x}_i^*(k) \le 0,5 .$$

Mutual recalculation is made according to the next expressions

$$\tilde{x}_i^*(k) = \frac{x_i^*(k) - x_i^{*min}}{x_i^{*max} - x_i^{*min}} - 0,5 ,$$

$$x_i^*(k) = (\tilde{x}_i^*(k) + 0,5)(x_i^{*max} - x_i^{*min}) + x_i^{*min} .$$

For each vector from the learning sample $\tilde{x}^*(k) = (\tilde{x}_1^*(k), \tilde{x}_2^*(k), ..., \tilde{x}_n^*(k))^T$ in the first hidden layer own set of fuzzy-basis membership functions $\mu_{\tilde{x}_1^*(k)}, \mu_{\tilde{x}_2^*(k)}, ..., \mu_{\tilde{x}_n^*(k)}$ is formed, so that centers of $\mu_{\tilde{x}_i^*(k)}$ coincide with $\tilde{x}_i^*(k)$, $k=1,2,…,l$. The process of FBF formation is illustrated on Fig. 2. Note that GRNFN contains nl fuzzy-basis functions, that can’t lead to the curse of dimensionality.

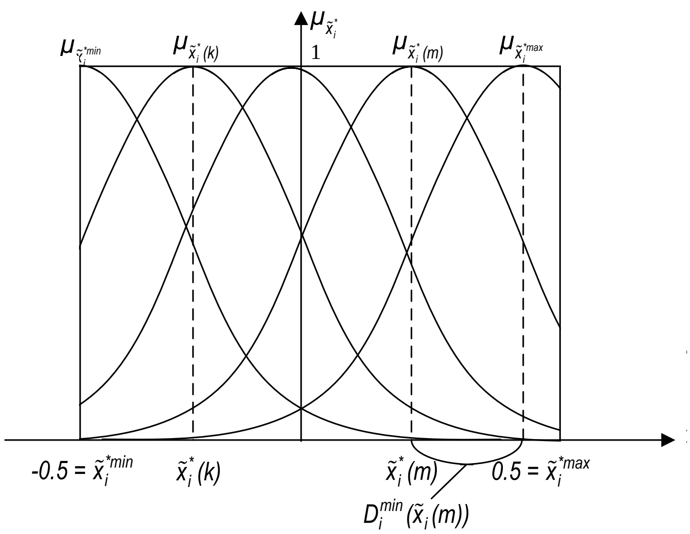


Fig.2 – Fuzzy-basis membership functions.

Theoretically, any kernel function with non-strictly local support can be used as FBF. It allows avoiding of appearance of "gaps" [9]. As such a function one can recommend generalized Gaussian

$$\mu_{\tilde{x}_i^*(k)}(\tilde{x}_i) = \left(1 + \left|\frac{\tilde{x}_i^*(k) - \tilde{x}_i}{\sigma_i(k)}\right|^{2b}\right)^{-1}, \; b \ge 0,5, \qquad (2)$$

that is the bell-shaped function, whose shape is defined by the scalar parameter b [15]. Let's also note, that b defines the metrics $D^{2b}(k)$ too. As for choosing of the width parameter $\sigma_i(k)$, standard recommendation leads to the idea [8], that it must ensures small overlapping of neighboring FBFs. Easy to see, that for Gaussian this recommendation leads to estimate

$$\sigma_i(k) < \frac{l-1}{2 \div 3}.$$

At the same time with FBFs forming in first hidden layer, the synaptic weights are being tuned in the third hidden layer and they are supposed to be equal to the signals of learning sample $y^*(k)$.

Thus, when arbitrary signal $\tilde{x}$ is fed to the input of GRNFN in the first hidden layer membership levels $\mu_{\tilde{x}_i^*(k)}(\tilde{x}_i)$, $i=1,2,\ldots,n$, $k=1,2,\ldots,l$ are computed, in the second layer their aggregation is made by forming multidimensional FBFs

$$\varphi_k(\tilde{x}) = \prod_{i=1}^{n}\left(1 + \left|\frac{\tilde{x}_i^*(k) - \tilde{x}_i}{\sigma_i(k)}\right|^{2b}\right)^{-1}, \; k = 1,2,\ldots,l,$$

in the third layer products $\hat{y}(\tilde{x}) = y^*(k)\varphi_k(\tilde{x})$ are determined, fourth layer computes the values of signals $\sum_{k=1}^{l} y^*(k)\varphi_k(\tilde{x})$ and $\sum_{k=1}^{l}\varphi_k(\tilde{x})$, and, finally, in the output layer the estimate

$$\hat{y}(\tilde{x}) = \frac{\sum_{k=1}^{l} y^*(k)\varphi_k(\tilde{x})}{\sum_{k=1}^{l} \varphi_k(\tilde{x})} = \frac{\sum_{k=1}^{l} y^*(k)\prod_{i=1}^{n}\mu_{\tilde{x}_i^*(k)}(\tilde{x}_i)}{\sum_{k=1}^{l}\prod_{i=1}^{n}\mu_{\tilde{x}_i^*(k)}(\tilde{x}_i)},$$

is forming, which coincides with (1) with the only difference, that instead of radial-basis functions multidimensional fuzzy-basis functions are used, that were formed of univariate FBF.

The scheme of fuzzy inference, which is realized by GRNFN can be presented as a logic equations system

$$IF(\tilde{x}_1.IS.A_1(1)).AND.(\tilde{x}_2.IS.A_2(1)).AND.....AND.(\tilde{x}_n.IS.A_n(1)), \quad THEN \quad \hat{y}_1(\tilde{x}) = y^*(1)$$

$$\vdots$$

$$IF(\tilde{x}_1.IS.A_1(k)).AND.(\tilde{x}_2.IS.A_2(k)).AND.....AND.(\tilde{x}_n.IS.A_n(k)), \quad THEN \quad \hat{y}_k(\tilde{x}) = y^*(k)$$

$$\vdots$$

$$IF(\tilde{x}_1.IS.A_1(l)).AND.(\tilde{x}_2.IS.A_2(l)).AND.....AND.(\tilde{x}_n.IS.A_n(l)), \quad THEN \quad \hat{y}_l(\tilde{x}) = y^*(l)$$

where the operator $A_i(k)$ is represented by the membership function (2). Hence, using of neuro-fuzzy approach allows ensuring of obtained results interpretation.

The GRNFN learning process can proceed both in batch mode, when learning sample $\left\{x^*(k), y^*(k)\right\}$ is specified apriori and in real time, when pairs $x^*(k)$, $y^*(k)$ are given sequentially, forming multidimensional FBFs φ_k. It is sufficiently easy to organize the exclusion process of slight information pairs. If for some observation $\tilde{x}^*(m)$ next condition is held

$$\max_i D_i^{min}(\tilde{x}_i(m)) < r < (l-1)^{-1} \quad (3)$$

(here $D_i^{min}(\tilde{x}_i(m))$ – the least distance between $\tilde{x}_i(m)$ and earlier formed neighboring centers of FBFs), then $\tilde{x}^*(m)$ doesn't form function φ_m and is removed from the consideration. Note, that for univariate situation the threshold parameter r and the distance D_i^{max} are significantly easier to define, then in multidimensional case of GRNN.

Operation of GRNFN can be organized simply in the continuous adaptation mode that is essentially important for nonstationary objects identification and control. Here it is possible to use two approaches. The first is – on the sliding window of l observations, when while learning pairs $x^*(l+1)$, $y^*(l+1)$ are being fed to the input of the network, in the first and third layers the pair of $\mu_{\tilde{x}_i^*(1)}$ and $y^*(1)$ is removed, and instead of it the membership function $\mu_{\tilde{x}_i^*(l+1)}$ and weight $y^*(l+1)$ are formed. The second approach is based on inequality (3). In this case newly received pair $x^*(m)$, $y^*(m)$ isn't removed, but replaces the nearest to it in the "old" data.

As far as the learning process operates almost immediately, there is no problem with following properties of tuning algorithm at all.

Numerical experiment

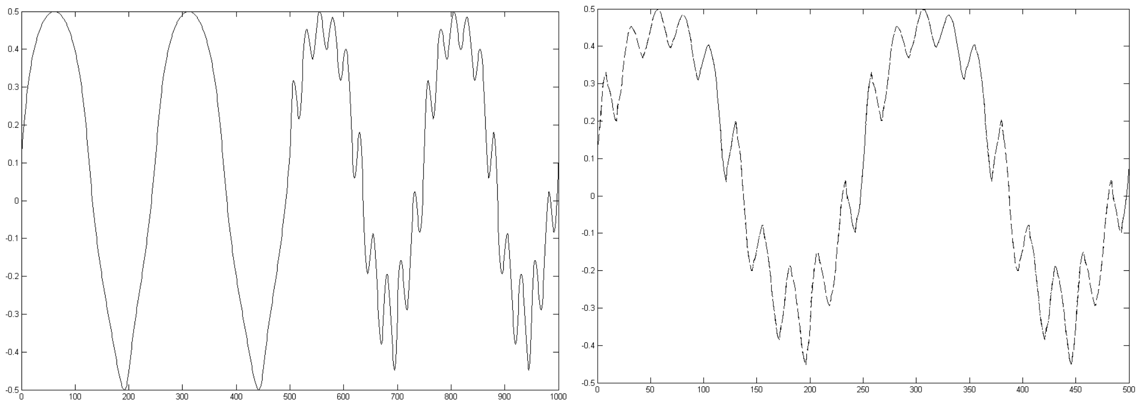
In this experiment, the plant is assumed to be of the form [22]:

$$y(k+1) = f(y(k), y(k-1), y(k-2), u(k), u(k-1)) ,$$

where the unknown function *f* has the form

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} .$$

The input to the plant is given by $u(k) = sin(2\pi k/250)$ for $k \le 500$ and $u(k) = 0.8sin(2\pi k/250) + 0.2sin(2\pi k/25)$ for $k > 500$, in all 1000 signals. Fig.3(a) shows the output of the plant.



(a) (b)

Fig. 3. (a) Outputs of the plant. (b) Outputs of the GRNFN (dash-dot line) and GRNN (dashed line) practically coincide.

Two experiments were made. In the first experiment, GRNFN was constructed and learned by first 500 signals, which organized learning sample. After that, the next 500 signals were fed to the network for testing its performance. In addition, this problem was solved using conventional GRNN. The results are shown in Fig.3(b) for last 500 instants. One can see that output signals of GRNFN and GRNN practically agree with test signals and with each other, but numerical analysis shows that GRNFN has accuracy higher by 2%. In the second experiment the distances between all learning signals were computed and compared with threshold value. Only 378 of 500 signals exceeded preassigned threshold value, and they organized learning sample. In this case, GRNFN has the same accuracy. Hence, it is logically to conclude that GRNFN needs less number of signals to be learned in comparison with GRNN.

Conclusions

Generalized Regression Neuro-Fuzzy Network, that is generalization of conventional GRNN and adaptive fuzzy inference systems, is proposed in this work. Network is characterized by computational simplicity, interpretability of the results and ensures high accuracy in the nonlinear nonstationary systems prediction and identification problems.

Bibliography

[1] Moody J., Darken C.J. Fast learning in networks of locally-tuned processing units// Neural Computation.- 1989.-1.-P.281-294.

[2] Park J., Sandberg I.W. Universal approximation using radial-basis-function networks// Neural Computation.-1991.-3.-P.246-257.

[3] Schilling R.J., Carrol J.J., Al-Ajlouni A.F. Approximation of nonlinear systems with radial basis function neural networks// IEEE Trans. on Neural Networks.-2001.-12.-P.1-15.

[4] Nelles O. Nonlinear System Identification.-Berlin: Springer, 2001.-785p.

[5] Specht D.E. A general regression neural network// IEEE Trans. on Neural Networks.-1991.-2.-P.568-576.

[6] Parzen E. On the estimation of a probability density function and the mode//Ann. Math. Stat.-1962.-38.-P.1065-1076.

[7] Nadaraya E.A. About nonparametric probability density and regression estimates// Probability theory and its Application.-1965.-10.-№1.-P199-203.

[8] Bishop C.M. Neural Networks for Pattern Recognition.- Oxford: Clarendon Press, 1995.-482p.

[9] Friedman J., Hastie T., Tibshirani R. The Elements of Statistical Learning. Data Mining, Inference, and Prediction.- Berlin: Springer, 2003.-552p.

[10] Zhivoglyadov V.G., Medvedev A.V. Nonparametric algorithms of adaptation.-Frunze: Ilim, 1974.-135p. (in Russian).

[11] Zahirniak D.R., Capman R., Rogers S.K., Suter B.W., Kabrisky M., Pyati V. Pattern recognition using radial basis function network// Proc. 6-th Ann. Aerospace Application of AI Conf.- Dayton, OH, 1990.-P.249-260.

[12] Seng T.L., Khalid M., Yusof R., Omatu S. Adaptive neuro-fuzzy control system by RBF and GRNN neural networks// J. of Intelligent and Robotic Systems.- 1998.-23.-P.267-289.

[13] Guo X.-P., Wang F.-L., Jia M.-X. A sub-stage moving window GRNN quality prediction method for injection molding process// "Lecture Notes in Computer Science"- V3973.- Berlin-Heidelberg: Springer-Verlag, 2006.-P.1138-1143.

[14] Jang J.-S. R., Sun G.-T. Neuro-fuzzy modeling and control// Proc. IEEE.-1995.-83.-P.378-406.

[15] Jang J.-S. R., Sun G.-T., Mizutani E. Neuro-Fuzzy and Soft Computing.- Upper Saddle River, NJ: Prentice Hall, 1997.-614p.

[16] Cios K.J., Pedrycz W. Neuro-Fuzzy algorithms// In: "Handbook on Neural Computation" – Oxford: University Press, 1997.-D1.3:1-D1.3:7.

[17] Jang J.-S. R. ANFIS: Adaptive-Network-based Fuzzy Inference Systems// IEEE Trans. on Systems, Man, and Cybernetics.-1993.-23.-P.665-685.

[18] Brown M., Harris C.J. Neural networks for modeling and control/ In: Eds. by C.J. Harris "Advances in Intellectual Control".- London: Taylor and Francis, 1994.-P.17-55.

[19] Wang H., Liu G.P., Harris C.J., Brown M. Advanced Adaptive Control.- Oxford: Pergamon, 1995.- 262p.

[20] Wang L.-X., Mendel J.M. Fuzzy basis functions, universal approximation, and orthogonal least squares learning//IEEE Trans. on Neural Networks.-1992.-3.-P.807-814.

[21] Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control// IEEE Trans. on Systems, Man, and Cybernetics.-1985.-15.-P.116-132.

[22] Narendra K.S., Parthasarathy K. Identification and control of dynamical systems using neural networks// IEEE Trans. on Neural Networks.-1990.-1.-P.4-26.

Authors' Information

Yevgeniy Bodyanskiy – Kharkiv National University of Radio Electronics, Doctor of Technical Sciences, Professor of Artificial Intelligence Department, Head of Control Systems Research Laboratory, IEEE Senior Member; Postal address: CSRL, Ofiice 511, Lenin Av., 14, Kharkiv, 61166, Ukraine; e-mail: bodya@kture.kharkov.ua.

Nataliya Teslenko - Kharkiv National University of Radio Electronics, post-graduate student, research scientist of Control Systems Research Laboratory; Postal address: CSRL, Ofiice 511, Lenin Av., 14, Kharkiv, 61166, Ukraine; e-mail: ntntp@ukr.net.