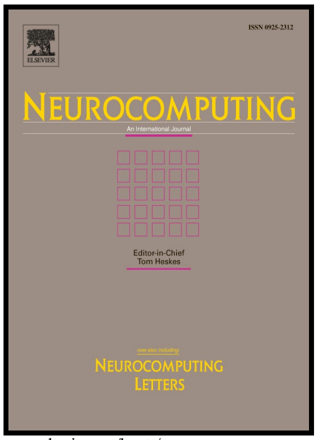


www.elsevier.com/locate/neucom

Adaptive multivariate hybrid neuro-fuzzy system and its on-board fast learning

Ye. Bodyanskiy, O. Vynokurova, G. Setlak, D. Peleshko, P. Mulesa

PII: S0925-2312(16)31558-2
DOI: http://dx.doi.org/10.1016/j.neucom.2016.12.042
Reference: NEUCOM17861

To appear in: *Neurocomputing*

Received date: 13 March 2016
Revised date: 24 November 2016
Accepted date: 13 December 2016

Adaptive multivariate hybrid neuro-fuzzy system and its on-board fast learning

Ye. Bodyanskiy, O. Vynokurova[a,*], G. Setlak[b], D. Peleshko[c], P. Mulesa[d]

[a] *Kharkiv National University of Radio Electronics, 14 Nauky av., 61166, Kharkiv, Ukraine, e-mails:yevgeniy.bodyanskiy@nure.ua, olena.vynokurova@nure.ua*

[b] *Rzeszow University of Technology, 12 Al. Powstancow Warszawy, 35-959, Rzeszow, Poland, e-mail:gsetlak@prz.edu.pl*

[c] *Lviv Polytechnic National University, 12 S. Bandery st., 79013, Lviv, Ukraine, e-mail:dpeleshko@gmail.com*

[d] *Uzhhorod National University, 3 Narodna Square, 88000, Uzhhorod, Ukraine, e-mail:ppmulesa@gmail.com*

Abstract

In the paper the multivariate adaptive hybrid neuro-fuzzy system is proposed that allows to process nonstationary information disturbed by noises in sequential mode and also has smaller number of tuned parameters comparatively with known neuro-fuzzy systems. This proposed system can be used in on-board applications and, first of all, industrial plants, smart homes (energy management, climate control, home electronic devices including security system, etc).

Keywords: Data Mining, Computational Intelligence, Multivariate Hybrid Neuro-Fuzzy System, Adaptive Learning, Prediction, On-Board Applications

1. Introduction

Nowadays the computational intelligence methods and systems are widespread for solving of different Data Mining tasks, problems of intelligent control, prediction, identification, pattern recognition, etc. [1], [2] [3], [4] under conditions of uncertainty, nonlinearity, stochasticity, chaotic states, different kinds of disturbances and noises due to their universal approximation properties and learning

*Corresponding author

Email address: `yevgeniy.bodyanskiy@nure.ua, olena.vynokurova@nure.ua` (Ye. Bodyanskiy, O. Vynokurova)

possibilities based on data that describe the operation of investigated signal, process or plant.

Currently the most known and popular approaches are connected with the artificial neural networks such as multilayer perceptrons that are learned using backpropagation tuning algorithms. Nevertheless, the training set must be defined a priori, and the learning process is implemented using many epochs of the synaptic weights training. In this case, we cannot use such systems for solving tasks in sequential mode, when the data are fed to the inputs in a sequential order in real time.

At this time systems of computational intelligence are used widely in on-board applications and, first of all, industrial plants, smart homes [5], [6], [7], [8], [9], [10], [11] (energy management, climate control, home electronic devices including security system, etc). These tasks need increased learning speed which has to take place in real time, the simplicity of implementation, the possibility of operation under different kind of disturbances and noise conditions and also nonstationary changeable environment.

Generally, implementing of sequential learning process is possible for neural networks, whose output signal depends linearly from tuned synaptic weights, for example, Radial Basis Function Networks (RBFN) [1], [2] [3], [4], [12], [13], [14] and Normalized Radial Basis Function Networks (NRBFN) [15], [16], [17], however their using is often complicated by, so called, curse of dimensionality. In addition, the key moment is not computational complexity, but a problem is obtaining of data sets from the real plant that can be too small for estimating of large synaptic weights number.

Neuro-fuzzy systems that combine the learning abilities of neural networks and transparency- interpretability of the soft computing methods, have a range of advantages ahead of the conventional neural networks. It should be noticed TSK-system [18], [19], [20], [21] and ANFIS [22], [23], [24], [25], whose output signal depends linearly from the synaptic weights and has less number of synaptic weights than RBFN or NRBFN. The more complex hybrid systems of computational intelligence are well-known and have improved approximation properties, for example, the hybrid fuzzy wavelet neural networks [26], [27], [28], but learning algorithms complexity limits their using in sequential mode.

Hence it is necessary to synthesize a multivariate adaptive hybrid neuro-fuzzy system that allows to process nonstationary information that is disturbed by noises in on-board sequential mode, has smaller number of tuned parameters comparatively with known neuro-fuzzy systems, is simple in the computational implementation (due to the paralleling of the information processing) and don't demand previous defining of the training set, i.e. to implement the learning process started with the first observation, which is fed to the system.

2. Neuro-fuzzy systems by Takagi-Sugeno-Kang and Wang-Mendel

The most popular neuro-fuzzy systems, whose output signal depends linearly from tuning synaptic weights, are Takagi-Sugeno-Kang systems (TSK systems) and its simplified version Wang-Mendel system [29], [30] (such system is TSK system of zeroth order). The advantages of Wang-Mendel system are relatively small number of tuning parameters (for example, comparatively to conventional multilayer perceptron) and possibility of using Gauss-Newton optimization procedure for learning of system (the second order optimization algorithms), for example, recurrent least squares method, which characterized by high convergence rate.

The architecture of Wang-Mendel system consists of concatenated layers of information processing: the first layer is fuzzification layer, the second hidden layer is aggregation layer, the third hidden layer is synaptic weights layer, the forth hidden layer is adder units layer, and finally, the output layer is defuzzification layer. The input vector signal $x(k) = (x_1(k), \dots, x_i(k), \dots, x_n(k))^T$ (here $k = 1, 2, \dots$ is discrete instant time) in the first layer, which contained nh membership functions $\mu_{li}(x_i)$ (here $l = 1, 2, \dots, h$, h is a number of membership functions for each scalar input x_i, $i = 1, 2, \dots, n$) is exposed to fuzzification. With the result that nh signals are appeared in the output of the first layer, which are fed to the second hidden layer with h multilayer units, which are realized an aggregation operation.

These aggregated signals are fed to the third layer of synaptic weights, which are permanently adjusted under a learning process. In the fourth layer the elementary sum operation of signals from outputs of second and third hidden layers, and finally, in the output layer the defuzzification operation is realized by m elementary division units (here m is the number of system output). In this way, m signals $\hat{y}_p(k)$ ($p = 1, \ldots, m$) are appeared in the output of system, which are response to input signals x_i $(i = 1, 2, \ldots, n)$.

From formal point of view, this neuro-fuzzy system implements nonlinear mapping $x \in \Re^n \Rightarrow \hat{y} \in \Re^m$. This nonlinearity is realized by the first layer with nonlinear activation functions $0 < \mu_{li}(x_i(k)) \leq 1$, where, as usual, bell-shaped functions are used, such as conventional Gaussians

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right). \tag{1}$$

These functions have nonstrictly local receptive fields, due to this fact we can avoid appearing of gaps in fuzzificated space, which is connected with scatter partitioning [16]. It can be also noticed, the centers c_{li} and widths σ_i parameters of Gaussians can be choose either empirically or tuning with computationally tedious error backpropagation algorithms. It is clear, that in this case we cant talk about online learning of system.

A choice of width parameter σ_i can be simplified a little if all input variables are coded in some fixed interval, for example, $0 \leq x_i(k) \leq 1$, that allows to choose this parameter equivalent for all inputs, where the number of membership functions is the same for each input.

The aggregation layer implements a multiplication operator of one-dimensional membership functions for each input in the form

$$\widetilde{x}_l(k) = \prod_{i=1}^{n} \mu_{li}(x_i(k)). \tag{2}$$

In the result of this operation, instead of one-dimensional membership functions, we obtain multidimensional bell-shaped activation functions of radial basis function networks, which allow to implement increasing of input space dimension. If as membership functions are used Gaussians with the same width parameter then output signals of the second hidden layer can be written in form

$$\widetilde{x}_l(k) = \prod_{i=1}^{n} \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right) \tag{3}$$

where $c_l = (c_{l1}, \ldots, c_{li}, \ldots, c_{ln})^T$ is vector of centers parameters of multidimensional activation functions.

In the third layer of synaptic weights, where the main process of learning is implemented, output signals of the second layer are exposed to transformation in form

$$w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k)) = w_{pl}(k-1)\widetilde{x}_l(k) \tag{4}$$

where $w_{pl}(k-1)$ forms $mh \times 1$-vector of synaptic weights, which is computed using $(k-1)$ previous observations, $p = 1, 2, \ldots, m$; $l = 1, 2, \ldots, h$.

In the fourth (the simplest) layer of system, which is formed by adder units, we compute signals in the form

$$\begin{aligned} \sum_{l=1}^{h} w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k)) &= \sum_{l=1}^{h} w_{pl}(k-1)\widetilde{x}_l(k), \\ \sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k)) &= \sum_{l=1}^{h} \widetilde{x}_l(k) \end{aligned} \tag{5}$$

which are fed to the output layer of defuzzification, where the output signals are computed in form

$$\begin{aligned} \hat{y}_p(k) &= \frac{\sum_{l=1}^{h} w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k))}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))} = \frac{\sum_{l=1}^{h} w_{pl}(k-1)\widetilde{x}_l(k)}{\sum_{l=1}^{h} \widetilde{x}_l(k)} = \\ &\sum_{l=1}^{h} w_{pl}(k-1) \frac{\widetilde{x}_l(k)}{\sum_{l=1}^{h} \widetilde{x}_l(k)} = \\ &\sum_{l=1}^{h} w_{pl}(k-1) \frac{\prod_{i=1}^{n} \mu_{li}(x_i(k))}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))} = \\ &\sum_{l=1}^{h} w_{pl}(k-1)\varphi_l(x(k)) = w_p^T(k-1)\varphi(x(k)) \end{aligned} \tag{6}$$

where $\varphi_l(x(k)) = \prod_{i=1}^{n} \mu_{li}(x_i(k)) \left(\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k)) \right)^{-1}$,
$w_p(k-1) = (w_{p1}(k-1), \ldots, w_{pl}(k-1), \ldots, w_{ph}(k-1))^T$,
$\varphi(x(k)) = (\varphi_1(x(k)), \ldots, \varphi_l(x(k)), \ldots, \varphi_h(x(k)))^T$.

It can be noticed, that nonlinear transformation, which is realized by Wang-Mendel neuro-fuzzy system, is similar to one that is implemented by normalized radial basis function network, but contains smaller tuning parameters. This fact allows to increase a speed operation of learning process and to simplify a computational implementation.

3. Multivariable Hybrid Neuro-Fuzzy System

Decreasing of number of tuning parameters is provided by using a scatter partitioning of input space. At that it is necessary to notice that in this case in areas, which are disposed from centers of multidimensional membership-activation functions

$$\prod_{i=1}^{n} \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right) \tag{7}$$

the provided quality of approximation can be nonsufficient.

The approximation quality can be improved using, for example, grid partition of input space but at that the number of tuning parameters increases rapidly, i.e. the neuro-fuzzy systems advantages are lost ahead of the conventional neural network.

For improving the approximation properties of neuro-fuzzy system we can introduce, so called, nonlinear synapses in the third hidden layer instead of usual synaptic weights w_{pl}, $p = 1, 2, \ldots, m$, $l = 1, 2, \ldots, h$. These nonlinear synapses are building elements of neo-fuzzy neuron [31], [32], [33], which is enough simple and effective real-time system of computational intelligence, which is aimed at operating in on-board applications [33]. The neuro-fuzzy system based on neo-fuzzy neurons was proposed in [34] and its simplified versions in [35], [36], [37]. These systems confirmed their efficiency for many tasks connected with Dynamic Data Mining and Data Stream Mining.

Here, it is necessary to notice that systems based on nonlinear synapses and neo-fuzzy neurons are the single-output systems while it is necessary to use multi-inputs – multi-outputs description for a lot of real tasks' solution. In generally we can solve many tasks using some number of the parallel connected single output systems. This approach was proposed in [9] where for solving of smart house tasks the group of parallel ANFIS have been used. At that, therefore, the implementation of such systems is getting more complicate and the number of tuning parameters is increased.

In the connection with this, we propose here adaptive multivariate hybrid neuro-fuzzy system, which is characterized by the comparatively small number of adjustable parameters, allows to tune parameters in real time under nonstationary and stochasticity of processed information conditions and don't demand using the error backpropagation procedures for its learning.

Fig.1 shows the architecture of proposed multivariate hybrid neuro-fuzzy system.

It can be noticed that the first two layers of proposed system coincide with fuzzification and aggregation layers of Wang-Mendel system. Whereas these layers process information like R-neuron of radial basis function networks (on fig.1 the respective blocks are denoted by $R_1, R_2, \ldots, R_h$). The output signals of these blocks can be written in the form

$$\widetilde{x}_l(k) = \prod_{i=1}^{n} \mu_{li}(x_i(k)). \tag{8}$$

Further values $\widetilde{x}_l(k)$ are fed to the blocks outputs $MNS_1, MNS_2, \ldots, MNS_h$, which are pertinently multidimensional nonlinear synapses. These nonlinear synapses along with adder units of the fourth hidden layer form an architecture of so called generalized neo-fuzzy neuron [36], [37].

Generalized neo-fuzzy neuron (GNFN) is the multidimensional version of neo-fuzzy neuron [31], [32], [33] and implements nonlinear mapping in the form

$$f_p(\widetilde{x}(k)) = \sum_{l=1}^{h} f_{pl}(\widetilde{x}_l(k)) \tag{9}$$

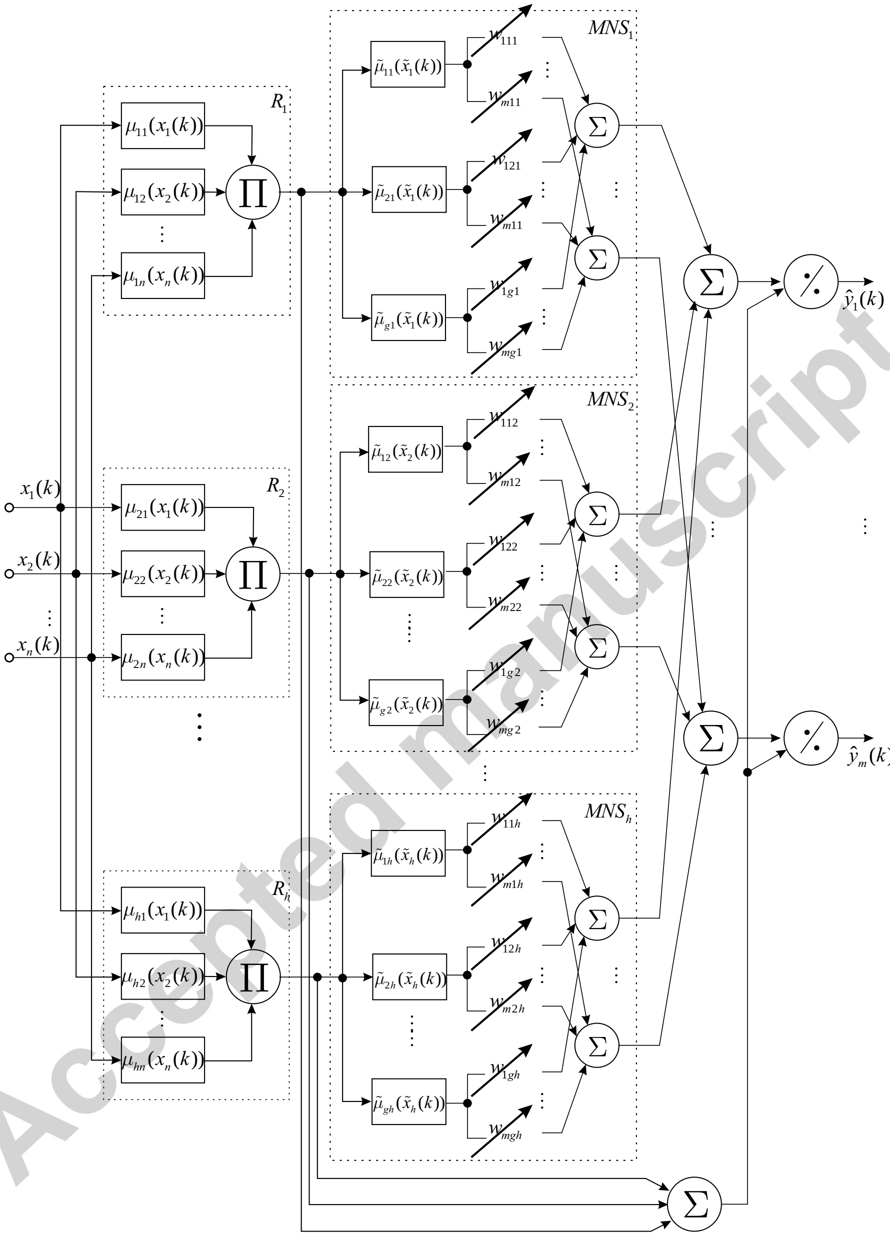


Figure 1: Adaptive multivariate hybrid neuro-fuzzy system (AMHNFS)

140 where $f_p(\widetilde{x}(k))$ is pth output signal of GNFN $(p = 1, 2, \ldots, m)$, $\widetilde{x}(k) = (\widetilde{x}_1(k), \ldots, \widetilde{x}_l(k), \ldots, \widetilde{x}_h(k))^T$. In each nonlinear synapses MNS_l the

fuzzification operation by using g membership function $\widetilde{\mu}_{jl}(\widetilde{x}_l)$, $l = 1, 2, \ldots, g$ and tuning operation of mg synaptic weights w_{pjl} are implemented.

It is important to notice that in the system under consideration the output signal MNS_l depends linearly from the tuning synaptic weights w_{pjl}, that allow to optimize the learning process with respect to high speed. So the signal in the output of each multidimensional nonlinear synapses can be written in the form

$$f_{pl}(\widetilde{x}_l(k)) = \sum_{j=1}^{g} w_{pjl}(k-1)\widetilde{\mu}_{jl}(\widetilde{x}_l(k)) \tag{10}$$

and signal in each output of GNFN (9) can be represented in the form

$$f_p(\widetilde{x}(k)) = \sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}(k-1)\widetilde{\mu}_{jl}(\widetilde{x}_l(k)), p = 1, 2, \ldots, m. \tag{11}$$

Analyzing of expressions (10), (11) it can be noticed that GNFN is nonlinear modification of multivariate generalized additive model [16], [38], [39]. The main advantages of such model is simplicity of computational implementation and possibility to parallelize the information analyzing process that allows to increase system response speed as a whole.

Additional $(m+1)$-th adder unit of the fourth hidden layer computes the value of signal

$$\sum_{l=1}^{h}\prod_{i=1}^{n} \mu_{jl}(x_i(k)) = \sum_{l=1}^{h} \widetilde{x}_l(k) \tag{12}$$

like Wang-Mendel neuro-fuzzy system.

And, finally, in output layer, which is formed by m division units, the resulting output signals of system are compute in the form

$$\begin{aligned} \hat{y}_p(k) &= \frac{\sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}(k-1)\widetilde{\mu}_{jl}(\widetilde{x}_l(k))}{\sum_{l=1}^{h}\widetilde{x}_l(k)} = \\ &= \frac{\sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}(k-1)\widetilde{\mu}_{jl}\left(\prod_{i=1}^{n}\mu_{li}(x_i(k))\right)}{\sum_{l=1}^{h}\prod_{i=1}^{n}\mu_{li}(x_i(k))} = \\ &= \sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}(k-1)\frac{\widetilde{\mu}_{jl}(\widetilde{x}_l(k))}{\sum_{l=1}^{h}\widetilde{x}_l(k)} = \\ &= \sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}(k-1)\widetilde{\varphi}_{jl}(\widetilde{x}(k)) = w_p^T(k-1)\widetilde{\varphi}(\widetilde{x}(k)) \end{aligned} \tag{13}$$

where $\widetilde{\varphi}_{jl}(\widetilde{x}(k)) = \widetilde{\mu}_{jl}(\widetilde{x}_l(k)) \left(\sum_{l=1}^{h} \widetilde{x}_l(k)\right)^{-1} =$
$= \widetilde{\mu}_{jl}\left(\prod_{i=1}^{n} \mu_{li}(x_i(k))\right) \left(\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))\right)^{-1}$, $w_p(k-1) = (w_{p11}(k-1), w_{p21}(k-1), \ldots, w_{pg1}(k-1), w_{p12}(k-1), \ldots, w_{pjl}(k-1), \ldots, w_{pgh}(k-1))^T$, $\widetilde{\varphi}(\widetilde{x}(k)) = (\widetilde{\varphi}_{11}(\widetilde{x}(k)), \widetilde{\varphi}_{21}(\widetilde{x}(k)), \ldots, \widetilde{\varphi}_{jl}(\widetilde{x}(k)), \ldots, \widetilde{\varphi}_{gh}(\widetilde{x}(k)))^T$.

In such a way, proposed neuro-fuzzy system combines the advantages of multidimensional generalized additive model by Hastie-Tibshirani and Wang-Mendel neuro-fuzzy system. The main advantages of proposed system comparatively with their prototypes are possibility of computation parallelizing together with high approximation properties, and also its safety from appearing of gaps in input space, which is connected with scatter partitioning.

As already mentioned, in the proposed system the conventional Gaussians are used as membership functions both in the first layer and in the third hidden one. It should be noticed that if in the third hidden layer in nonlinear synapses MNS_l we will use membership functions $\widetilde{\mu}_{jl}(\widetilde{x}_l)$, which satisfy to condition of unity partitioning (such as, for example, B-splines, triangular, trapezoidal and etc. membership functions), instead of conventional Gaussians then we can exclude from the architecture of proposed system the $(m+1)$th adder unit of the fourth hidden layer and m divisionary units of output layer because the defuzzification operation in this case is implemented automatically by generalized neo-fuzzy neuron, which in this case forms the output layer of this system.

4. On-Board Fast Learning of Multivariate Hybrid Neuro-Fuzzy System

The learning process of the proposed system is connected with the tuning of GNFN synaptic weights, which form the output layers of system under consideration.

For the on-board learning of conventional NFN its authors [33] used the gradient procedure, which minimizes learning criterion

$$\begin{aligned}E(k) &= \frac{1}{2}(y(k)-\hat{y}(k))^2 = \frac{1}{2}e^2(k) = \\ &= \frac{1}{2}\left(y(k)-\sum_{l=1}^{h}\sum_{j=1}^{g} w_{jl}\widetilde{\varphi}_{jl}(\widetilde{x}(k))\right)^2\end{aligned} \tag{14}$$

and can be written in the form

$$\begin{aligned}&w_{jl}(k) = w_{jl}(k-1) + \eta e(k)\widetilde{\varphi}_{jl}(\widetilde{x}(k)) = \\ &= w_{jl}(k-1) + \eta\Big(y(k)-\sum_{l=1}^{h}\sum_{j=1}^{g} w_{jl}(k-1)\widetilde{\varphi}_{jl}(\widetilde{x}(k))\Big)\widetilde{\varphi}_{jl}(\widetilde{x}(k))\end{aligned} \tag{15}$$

where $y(k)$ is reference signal, $e(k)$ is learning error, η is constant learning rate parameter.

For the tuning of synaptic weights of GNFN in [37] the one-step criterion

$$\begin{aligned}E_p(k) &= \frac{1}{2}(y_p(k)-\hat{y}_p(k))^2 = \frac{1}{2}e_p^2(k) = \\ &= \frac{1}{2}\left(y(k)-\sum_{l=1}^{h}\sum_{j=1}^{g} w_{pjl}\widetilde{\varphi}_{jl}(\widetilde{x}(k))\right)^2\end{aligned} \tag{16}$$

and the gradient algorithm by Kaczmarz-Widrow-Hoff, which can be written in the form using agreed notation

$$\begin{aligned}&w_{pjl}(k) = w_{pjl}(k-1) + \frac{e_p(k)\widetilde{\varphi}_{jl}(\widetilde{x}(k))}{\sum_{j=1}^{g}\sum_{l=1}^{h}\widetilde{\varphi}_{jl}^2(\widetilde{x}(k))} = \\ &= w_{pjl}(k-1) + \frac{(y_p(k)-\hat{y}_p(k))\widetilde{\varphi}_{jl}(\widetilde{x}(k))}{\sum_{j=1}^{g}\sum_{l=1}^{h}\widetilde{\varphi}_{jl}^2(\widetilde{x}(k))} = \\ &= w_{pjl}(k-1) + \frac{\Big(y_p(k)-\sum_{j=1}^{g}\sum_{l=1}^{h} w_{pjl}(k-1)\widetilde{\varphi}_{jl}(\widetilde{x}(k))\Big)\widetilde{\varphi}_{jl}(\widetilde{x}(k))}{\sum_{j=1}^{g}\sum_{l=1}^{h}\widetilde{\varphi}_{jl}^2(\widetilde{x}(k))}\end{aligned} \tag{17}$$

were used.

Here it should be noticed that the Kaczmarz-Widrow-Hoff algorithm, which is the optimal gradient procedure, provides only the linear convergence to the optimal values. Furthermore the properties of this learning algorithm are getting worse when the processed signals are disturbed by noises that are presented in real systems.

For the providing both smoothing and tracking properties for learning process we can use multi-steps weighted learning criterion and algorithms which are based on Gaussian-Newtonian optimization procedures of second order.

Introducing into consideration the $(m \times 1)$-vectors $y(k) = (y_1(k), \ldots, y_p(k), \ldots, y_m(k))^T,\ \hat{y}(k) = (\hat{y}_1(k), \ldots, \hat{y}_p(k), \ldots, \hat{y}_m(k))^T$, $(m \times gh)$ - matrix of synaptic weights in the form

$$W(k) = \begin{pmatrix} w_{111}(k) & w_{121}(k) & \cdots & w_{1gh}(k) \\ w_{211}(k) & w_{221}(k) & \cdots & w_{2gh}(k) \\ \vdots & \vdots & \ddots & \vdots \\ w_{m11}(k) & w_{m21}(k) & \cdots & w_{mgh}(k) \end{pmatrix} \tag{18}$$

and learning criterion

$$\begin{aligned} E(k) &= \frac{1}{2}\sum_{j=1}^{k} \alpha^{k-j} \|y(j) - \hat{y}(j)\|^2 = \frac{1}{2}\sum_{j=1}^{k}\sum_{p=1}^{m} \alpha^{k-j} e_p^2(j) = \\ &= \frac{1}{2}\sum_{j=1}^{k} \alpha^{k-j} \|y(j) - W\widetilde{\varphi}(\widetilde{x}(j))\|^2 \end{aligned} \tag{19}$$

we can use the exponentially weighted recurrent least squares method as the learning algorithm in the form

$$\left\{ \begin{aligned} W(k) &= W(k-1) + \frac{(y(k) - W(k-1)\widetilde{\varphi}(\widetilde{x}(k)))\widetilde{\varphi}^T(\widetilde{x}(k))P(k-1)}{\alpha + \widetilde{\varphi}^T(\widetilde{x}(k))P(k-1)\widetilde{\varphi}_{(}\widetilde{x}(k))}, \\ P(k) &= \frac{1}{\alpha}\Big(P(k-1) - \frac{P(k-1)\widetilde{\varphi}(\widetilde{x}(k))\widetilde{\varphi}^T(\widetilde{x}(k))P(k-1)}{\alpha + \widetilde{\varphi}^T(\widetilde{x}(k))P(k-1)\widetilde{\varphi}(\widetilde{x}(k))}\Big) \end{aligned} \right. \tag{20}$$

(here $0 < \alpha \le 1$ is forgetting parameter) that provides the compromise between tracking and filtering properties of learning algorithm, which therefore can be numerically unstable under large number of tuning parameters.

In this case using of the learning algorithm that has both tracking (for non-stationary signals processing) and filtering (for noised signals processing) properties is more preferable so we can write the tuning procedure in the form [40]

$$\begin{cases} W(k) = W(k-1) + r^{-1}(k)e(k)\widetilde{\varphi}^T_{(}\widetilde{x}(k)), \\ r(k) = \alpha r(k-1) + \|\widetilde{\varphi}(\widetilde{x}(k))\|^2, \\ 0 \leq \alpha \leq 1, \end{cases} \tag{21}$$

which is stable for any values of forgetting parameter α. This learning algorithm coincides with Kaczmarz-Widrow-Hoff optimal multivariate learning algorithm when $\alpha = 0$ and stochastic approximation learning algorithm when $\alpha = 1$.

5. Experimental Results

5.1. Prediction of Chaotic Multivariate Time Series

Proposed adaptive multivariate hybrid neuro-fuzzy system was simulated using nonstationary chaotic time series modeling, which describes the, so-called, Rössler system. The Rössler attractor is the attractor of a system of three non-linear ordinary differential equations originally studied in [41], [42] in the form

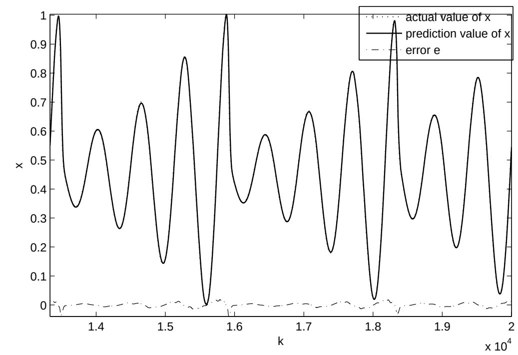
$$\begin{cases} \frac{dx}{dt} = -y - z, \\ \frac{dy}{dt} = x + ay, \\ \frac{dz}{dt} = b + z(x - c) \end{cases} \tag{22}$$

where $a = 0.15$, $b = 0.2$, $c = 10$.

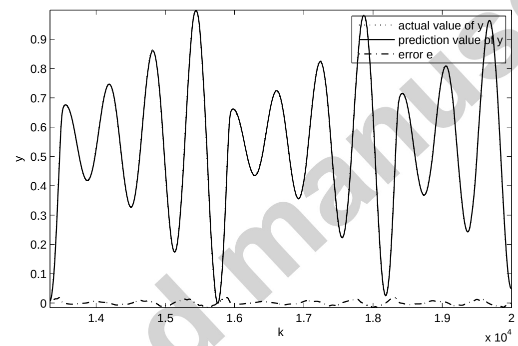
These differential equations describe a continuous-time dynamical system that exhibits chaotic dynamics associated with the fractal properties of the attractor. This attractor has some similarities to the Lorenz attractor, but has only one manifold.

The fig.2 and fig.3 show the results of Rössler attractor modeling.

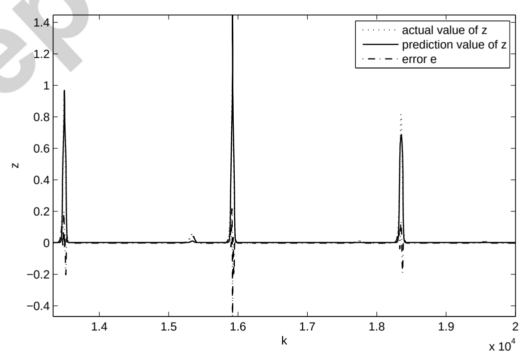
The data sample consists of 4500 samples, where 3000 sample - training set (which is fed for processing in sequential mode), 1500 sample - testing set. The initial parameters of hybrid system and its learning algorithm were determined as: initial parameters value of centers and widths of activation functions were generated using subtractive clustering method. This method allows to obtain



(a) x parameter



(b) y parameter



(c) z parameter

Figure 2: Result of chaotic time series modelling

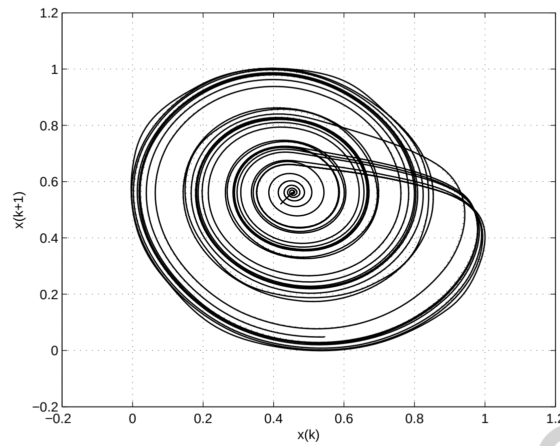


Figure 3: Result of Rössler attractor modeling

both initial values of the coordinate matrix centers, and the vector with components which determine influence range of cluster center. Parameters of adaptive multivariate hybrid neuro-fuzzy system were taken $n = 3, h = 6, g = 3, m = 3$.

Besides, the situation when the centers of the membership functions were distributed in uniformly mode. In this case the quality of learning process is worse, but the learning process was performed started from the first observation in real-time mode.

The root mean-square error (RMSE) and mean absolute percentage error (MAPE) was used as criterion for the prediction quality.

Tab. 1 shows comparative analysis of Rössler attractor modeling results based on proposed adaptive multivariate hybrid neuro-fuzzy system with other approaches which were described in the literature.

Thus as it can be seen from experimental results the proposed adaptive multivariate hybrid neuro-fuzzy system with the learning algorithm having the same number of adjustable parameters and smaller learning time ensures the best quality of emulation in comparison with multivariate neuro-fuzzy system of Wang-Mendel and multivariate neo-fuzzy neuron.

Table 1: Comparative analysis of Rössler attractor modeling results

Neural network/ Learning algorithm	Signal	RMSE	MAPE	Num. of param.
AMHNFS (the centers of membership functions were distributed in uniformly mode) / Proposed learning algorithm	x	0.009	2.9%	54
	y	0.0089	2.8%	
	z	0.039	7.5%	
AMHNFS (the centers of membership functions were distribured by subtractive clustering method) / Proposed learning algorithm	x	0.0078	2.6%	54
	y	0.0072	2.1%	
	z	0.023	6.7%	
Multivariate neuro-fuzzy system of Wang-Mendel / Gradient learning algorithm	x	0.018	5.3%	84
	y	0.017	5.2%	
	z	0.054	9.2%	
Multivariate neo-fuzzy neuron/ Recurrent least squares learning algorithm	x	0.0099	3.0%	74
	y	0.0098	3.1%	
	z	0.048	8.3%	

5.2. Prediction of Energy Consumption Multivariate Time Series

The second experiment has been connected with prediction of hourly energy consumption time series in the one of Germany federal lands [43].

The inputs number of proposed adaptive multivariate hybrid neuro-fuzzy system is $n = 12$ so the input vector can be written in the form $[x_1(k+h), x_2(k+h), x_3(k+h)] = (x_1(k), x_1(k-h), \Delta x_1(k), x_1(k+h-24), x_1(k+h-168), [k/42], [k/168], [k/(366*24)], x_2(k), x_2(k-h), x_3(k), x_3(k-h))$, where h is horizon of prediction, k is discrete instant time, $x_1(k+h)$ is the prediction value of energy consumption, $x_1(k)$ is the current value of energy consumption, $x_1(k-h)$ is the value of energy consumption h-steps ago, $x_1(k+h-24)$ is value of energy consumption twenty-four hours ago from the prediction value,

$x_1(k+h-168)$ is value of energy consumption one week ago from the prediction value, $[k/24]$ is the hour number in a day, $[k/168]$ is the hour number in a week, $[k/(366*24)]$ is the hour number in a year, $x_2(k+h)$ is the prediction value of the dry bulb temperature, $x_2(k)$ is the current value of dry bulb temperature, $x_2(k-h)$ is the value of dry bulb temperature h-steps ago, $x_3(k+h)$ is the prediction value of the dew point temperature, $x_3(k)$ is the current value of dew point temperature, $x_3(k-h)$ is the value of dew point temperature h-steps ago. The initial values of the synaptic weights were taken zero. Parameters of adaptive multivariate hybrid neuro-fuzzy system were taken $n = 12, h = 20, g = 12, m = 3$.

Fig.4 shows the results of energy consumption time series prediction, fig. 5 shows the results of dry bulb temperature time series prediction and fig. 6 shows the results of dew point temperature time series prediction. The two curves, representing the actual (dot line) and forecasting (solid line) values, are almost indistinguishable.

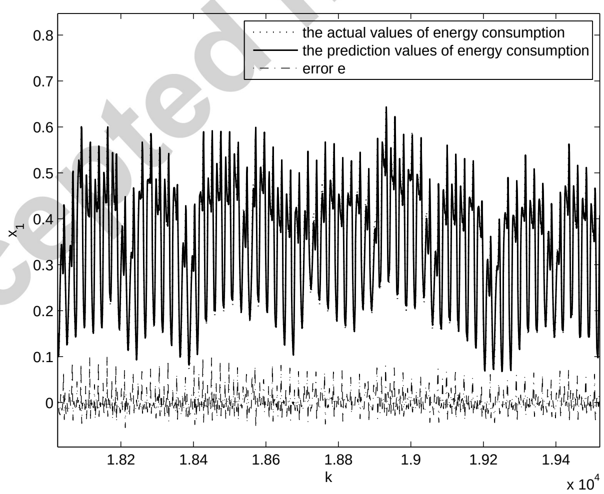


Figure 4: Result of energy consumption time series prediction

The Tab. 2 shows the comparative analysis of energy consumption time series prediction based on the different approaches.

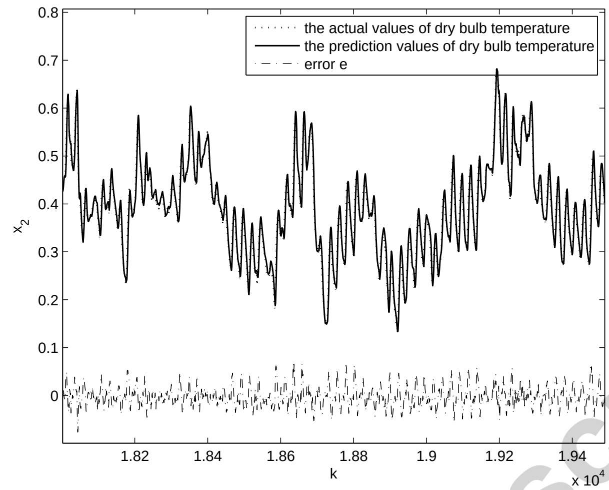


Figure 5: Result of dry bulb temperature time series prediction

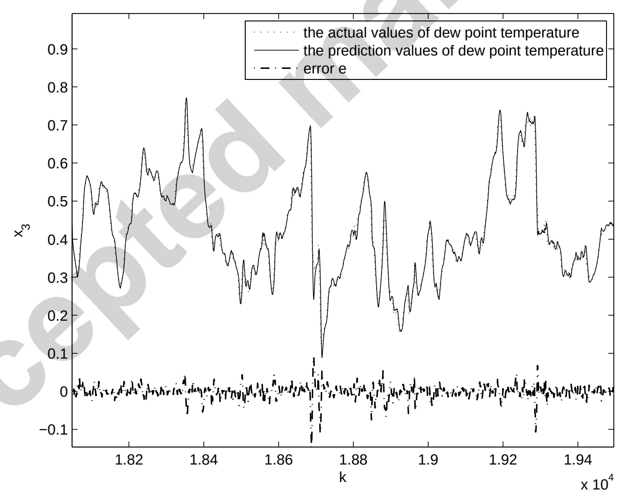


Figure 6: Result of dew point temperature time series prediction

For demonstration of proposed approach effectiveness we considered case, when the processed signal was corrupted by noise with Gaussian distribution. For this the energy consumption time series x_1 was corrupted $\tilde{x}_1 = x_1 + \xi$, $(\xi \sim N(0,1))$.

Table 2: The comparison analysis of energy consumption time series prediction

Neural network/ Learning algorithm	Param.	RMSE	MAPE
Adaptive multivariate hybrid neuro-fuzzy system / Proposed learning algorithm	x_1	0.028	5.7%
	x_2	0.024	3.1%
	x_3	0.018	2.2%
Multivariate neuro-fuzzy system of Wang-Mendel / Gradient learning algorithm	x_1	0.078	7.6%
	x_2	0.072	6.9%
	x_3	0.043	5.7%
Multivariate neo-fuzzy neuron/ Recurrent least squares learning algorithm	x_1	0.058	4.6%
	x_2	0.042	5.1%
	x_3	0.023	4.7%
Multivariate adaptive neuro-fuzzy inference system / Gradient learning algorithm	x_1	0.06	4.9%
	x_2	0.05	6.1%
	x_3	0.034	5.1%

The Tab. 3 shows the comparative analysis of energy consumption time series prediction based on the different approaches, when processed time series was corrupted by noise with Gaussian distribution.

Thus as it can be seen from experimental results the proposed adaptive multivariate hybrid neuro-fuzzy system with its learning algorithm provides the best quality of prediction among considered approaches, when the learning process was performed in on-board mode. It can be seen that in the case with noised signal we have little worse results, but it is better result among systems under consideration.

6. Conclusion

The adaptive multivariate hybrid neuro-fuzzy system that connects advantages of the neuro-fuzzy system by Wang-Mendel and the multivariate generalized additive models by Hastie-Tibshirani, is proposed. Such system is charac-

Table 3: The comparison analysis of energy consumption time series prediction which was corrupted by Gaussian noise

Neural network/ Learning algorithm	Param.	RMSE	MAPE
Adaptive multivariate hybrid neuro-fuzzy system / Proposed learning algorithm	$\tilde{x}_1$	0.037	6.0%
	x_2	0.033	3.9%
	x_3	0.027	3.0%
Multivariate neuro-fuzzy system of Wang-Mendel / Gradient learning algorithm	$\tilde{x}_1$	0.086	8.1%
	x_2	0.081	7.7%
	x_3	0.053	6.4%
Multivariate neo-fuzzy neuron/ Recurrent least squares learning algorithm	$\tilde{x}_1$	0.068	5.1%
	x_2	0.053	5.9%
	x_3	0.034	5.2%
Multivariate adaptive neuro-fuzzy inference system / Gradient learning algorithm	$\tilde{x}_1$	0.11	5.5%
	x_2	0.10	6.9%
	x_3	0.043	5.9%

terized by the simplicity of computational implementation, improving approximation properties, high-speed of learning process and is intended to solve wide range tasks of intelligent control, identification, forecasting etc., which are connected with the nonstationary noised stochastic and chaotic signal processing in on-board mode (i.e. the observations are fed to the system sequentially in real time). Such system can be used in on-board applications and, first of all, industrial plants, smart homes (energy management, climate control, home electronic devices including security system, etc).

[1] L. Rutkowski, Computational Intelligence: Methods and Techniques, Berlin: Springer-Verlag, 2008.

[2] C. Mumford, L. Jain, Computational Intelligence Collaboration, Fusion and Emergence, Berlin: Springer-Verlag, 2009.

[3] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher,

P. Held, Computational Intelligence. A Methodological Introduction, Berlin: Springer-Verlag, 2013.

[4] K.-L. Du, M. Swamy, Neural Networks and Statistical Learning, London: Springer-Verlag, 2014.

[5] J. Augusto, C. Nugent (Eds.), Designing Smart Homes. The Role of Artificial Intelligence, Vol. 4008 of Lecture Notes in Artificial Intelligence, Springer-Verlag Berlin Heidelberg, 2006.

[6] A. Badlani, S. Bhanot, Smart home system design based on artificial neural networks, in: Proceedings of the World Congress on Engineering and Computer Science (WCECS 2011), San Francisco, USA, Vol. I, 2011, pp. 19–21.

[7] S.-H. Lee, S.-J. Lee, K.-I. Moon, An ANFIS control system for smart home, International Journal of Advancements in Computing Technology 5 (11) (2013) 464–470.

[8] M. Reaz, Artificial intelligence techniques for advanced smart home implementation, Acta Technica Corvininesis - Bulletin of Engineering 6 (2) (2013) 51–57.

[9] S.-H. Lee, S.-J. Lee, K.-I. Moon, Smart home security system using multiple ANFIS, International Journal of Smart Home 7 (3) (2013) 121–132.

[10] L. G. Fahad, A. Khan, M. Rajarajan, Activity recognition in smart homes with self verification of assignments, Neurocomputing 149 (Part C) (2015) 1286–1298.

[11] J. Zhang, Y. Shan, K. Huang, Isee smart home (ISH): Smart video analysis for home security, Neurocomputing 149 (Part B) (2015) 752–766.

[12] H. W. Chi Zhang, L. Xie, Y. Shen, K. Zhang, Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework, Neurocomputing 205 (2016) 53–63.

[13] Y. Cui, J. Shi, Z. Wang, Lazy quantum clustering induced radial basis function networks (LQC-RBFN) with effective centers selection and radii determination, Neurocomputing 175 (Part A) (2016) 797–807.

[14] Q. Ha, H. Wahid, H. Duc, M. Azzi, Enhanced radial basis function neural networks for ozone level estimation, Neurocomputing 155 (2015) 62–70.

[15] O. Nelles, Nonlinear Systems Identification, Berlin: Springer, 2001.

[16] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning. Data Mining, Inference and Prediction, N.Y.: Springer Science+ Business Media, LLC, 2009.

[17] G. Bugmann, Normalized gaussian radial basis function networks, Neurocomputing 20 (1-3) (1998) 97–110.

[18] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. on Systems, Man and Cybernetics 15 (1) (1985) 116–132.

[19] M. Sugeno, G. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1) (1988) 15–33.

[20] H. Takagi, I. Hayashi, NN-driven fuzzy reasoning, Int. J. of Approximate Reasoning 5 (3) (1991) 191–212.

[21] C.-F. Juang, S.-T. Huang, F.-B. Duh, Mold temperature control of a rubber injection-molding machine by tsk-type recurrent neural fuzzy network, Neurocomputing 70 (1-3) (2006) 559–567.

[22] R.-S. Jang, ANFIS: Adaptive network based fuzzy inference systems, IEEE Trans. on Systems, Man and Cybernetics 23 (3) (1993) 116–132.

[23] C.-T. S. J.-S.R. Jang, Neuro-fuzzy modeling and control, in: Proc. of the IEEE, Vol. 83, 1995, pp. 378–406.

[24] R.-S. Jang, C. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence, Upper Saddle River: Prentice Hall, 1997.

[25] C.-H. Sua, C.-H. Cheng, A hybrid fuzzy time series model based on ANFIS and integrated nonlinear feature selection method for forecasting stock, Neurocomputing 205 (2016) 264–273.

[26] R. Abiyev, O. Kaynak, Fuzzy wavelet neural networks for identification and control of dynamic plants - a novel structure and a comparative study, IEEE Trans. on Industrial Electronics 55 (2) (2008) 3133–3140.

[27] Y. Bodyanskiy, O. Vynokurova, Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification, Information Sciences 220 (2013) 170–179.

[28] M. Rana, I. Koprinska, Forecasting electricity load with advanced wavelet neural networks, Neurocomputing 182 (2016) 118–132.

[29] L.-X. Wang, J. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least-squares learning, IEEE Trans. on Neural Networks 3 (5) (1992) 807–814.

[30] L.-X. Wang, Adaptive fuzzy systems and control: design and stability analysis, Upper-Saddle River: Prentice Hall, 1994.

[31] T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi, A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior, in: Proc. 2-nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", Iizuka, Japan, 1992, pp. 477–483.

[32] E. Uchino, T. Yamakawa, Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, Boston: Kluwer Academic Publishers, 1997, Ch. Soft computing based signal prediction, restoration and filtering, pp. 331–349.

[33] T. Miki, T. Yamakawa, Computational Intelligence and Application, Piraeus: WSES Press, 1999, Ch. Analog implementation of neo-fuzzy neuron and its on-board learning, pp. 144–149.

[34] Y. Bodyanskiy, G. Setlak, D. Peleshko, O. Vynokurova, Hybrid generalized additive neuro-fuzzy system and its adaptive learning algorithms, in: Proc of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2015), Warsaw, Poland, 2015, pp. 328–333.

[35] Y. Bodyanskiy, G. Setlak, I. Pliss, O. Vynokurova, Hybrid neuro-neo-fuzzy system and its adaptive learning algorithm, in: Proc. of Xth IEEE International Scientific and Technical Conference "Computer Science and Information Technologies", Lviv, Ukraine, 2015, pp. 111–114.

[36] R. Landim, B. Rodrignes, S. Silva, W. Matos, A neo-fuzzy-neuron with real-time training applied to flux observer for an induction motor, in: Proc. Vth Brasilian Symp. on Neural Networks. - Los Alamitos, CA: IEEE Computer Society, 1998, pp. 67–72.

[37] A. M. Silva, W. Caminhas, A. Lemos, F. Gomide, A fast learning algorithm for evolving neo-fuzzy neuron, Applied Soft Computing Journal 14, Part B (2014) 194–209.

[38] T. Hastie, R. Tibshirani, Generalized Additive Models, London: Chapman and Hall, 1990.

[39] K. Bossley, M. Brown, C. Harris, Neuro-fuzzy model construction for modelling of non-linear processes, in: Proc. of 3rd European Control Conference, Rome, Italy, Vol. 3, 1995, pp. 2438–2443.

[40] Y. Bodyanskiy, O. Tyshchenko, W. Wojcik, Multivariate non-stationary time series predictor based on an adaptive neuro-fuzzy approach, Elektronika 54 (8) (2013) 10–13.

[41] O. Rössler, An equation for continuous chaos, Physics Letters 57A (5) (1976) 397–398.

[42] O. Rössler, An equation for hyperchaos, Physics Letters 71A (2,3) (1979) 155–157.

[43] P. Otto, Y. Bodyanskiy, V. Kolodyazhniy, A new learning algorithm for a forecasting neuro-fuzzy network, Integrated Computer-Aided Engineering 10 (4) (2003) 399–409.