Досліджуються питання застосування штучних нейронних мереж при вирішенні задач ідентифікації та управління нелінійними динамічними системами. Досліджено характеристики мережі, що є результатом застосування апарату нечіткої логіки в класичній нейронної мережі CMAC і яка отримала назву FCMAC – Fuzzy Cerebral Model Arithmetic Computer. Отримані результати підтверджені імітаційним моделюванням процесів ідентифікації і управління нелінійними динамічними системами

Ключові слова: штучна нейронна мережа, фаззі-CMAC, ідентифікація, моделювання, непряме адаптивне управління, хешування

Исследуются вопросы применения искусственных нейронных сетей при решении задач идентификации и управления нелинейными динамическими системами. Исследованы характеристики сети, являющейся результатом применения аппарата нечеткой логики в классической нейронной сети CMAC и получившей название FCMAC – Fuzzy Cerebral Model Arithmetic Computer. Полученные результаты подтверждены имитационным моделированием процессов идентификации и управления нелинейными динамическими системами

Ключевые слова: искусственная нейронная сеть, фаззи-CMAC, идентификация, моделирование, непрямое адаптивное управление, хеширование

UDC 004.852
DOI: 10.15587/1729-4061.2018.128270

ADAPTIVE CONTROL OVER NON-LINEAR OBJECTS USING THE ROBUST NEURAL NETWORK FCMAC

O. Rudenko
Doctor of Technical Sciences,
Professor, Head of Department*
E-mail: oleg.rudenko@hneu.net
O. Bezsonov
Doctor of Technical Sciences, Professor*
E-mail: oleksandr.bezsonov@hneu.net
O. Lebediev
PhD, Associate Professor
Department of electronic computers
Kharkiv National University of Radio Electronics
Nauka ave., 14, Kharkiv, Ukraine, 61166
E-mail: oleh.lebediev@nure.ua
*Department of information systems
Simon Kuznets Kharkiv National
University of Economics
Nauky ave., 9-A, Kharkiv, Ukraine, 61166

1. Introduction

Artificial neural networks (ANNs) that are increasingly common at present are particularly effective when solving problems on the identification and control of non-linear dynamic objects in real time. ANNs are also widely used for processing and filtering of signals and images and for resolving a number of other tasks, which in one form or another employ approximation of complex non-linear dependences

$$y(x)=f(x)+\xi, \tag{1}$$

where y is the output signal; $x=(x_1, x_2,..., x_N)^T$ is the vector of input signals; f is an unknown nonlinear function; ξ is the disturbance with zero mathematical expectation; T is the symbol of transposition.

Among the existing large number of network structures, solving the specified problems mainly involve a multilayer perceptron (MP), radial-basis (RBN) and neural fuzzy (NFN) networks.

All of these ANNs are based on the approximation of the examined function by a certain system of basis functions $f_i(x)$. In this case, the approximated function is represented as a neural network, containing, in addition to the input and output layers, one or more hidden layers. Each of these layers consists of a certain number of neurons whose activating functions are the assigned basis functions

$$\hat{y}=\sum_{i,j} w_i f_i(x,\hat{w}_j), \tag{2}$$

where w_i are the weights of neurons in the output layer; $\hat{w}_j$ are the weights of neurons in the hidden layer.

In this case, the approximation problem comes down to determining network parameters through its training.

2. Literature review and problem statement

By using a mathematical model of the cerebellar cortex, developed in [1], author of papers [2, 3] proposed the artificial neural network *CMAC Cerebellar Model Articulation Controller*. This network, owing to the high speed of learning and a small volume of the required memory, achieved by special encoding of information, is especially promising for the implementation in microcontroller systems of control over non-linear objects, as well as when solving several other practical tasks. However, the use of traditional CMAC, which makes it possible, in contrast to MP, RBN, to perform a piecewise-constant approximation, is impossible for the tasks of indirect adaptive control in which it is required to calculate partial

derivatives based on the input control signals. This drawback can be eliminated by integrating a CMAC network with an apparatus of fuzzy logic [4]. A similar approach was employed in papers [5–8]. However, these papers applied traditional training of a network, which, on the one hand, is sufficiently proven when solving rather simple problems, and on the other hand, is very inefficient if there are interferences ξ with a distribution different from Gaussian.

As we know, training ANN implies determining a vector of its parameters and is reduced to minimizing certain functionality (learning criterion) caused by the approximation error

$$e(k) = y(k) - \hat{f}(k),$$

whose form depends on statistical properties of the interference. Most of the currently known algorithms for training neural networks are based on applying rigid and difficult-to-verify conditions associated with a hypothesis about normality of the distribution law of interference and substantiated by references to the central limit theorem and represents some modifications of the method of least squares (LSM).

A traditional CMAC network is trained using an error back propagation algorithm, minimizing quadratic functional from error $e(k)$ based on the presentation of training pairs $(x(k), y(k))$, $k = 1,2,...$ [2, 3, 9]. The LSM solution, obtained in this case, which is asymptotically optimal with a minimum variance in the class of unbiased estimates, is based on the assumption that interference ξ is not correlated and follows a normal distribution law. However, this assumption is usually wrong under actual conditions because an *a priori* information on distribution ξ is typically unavailable, or the interference is clogged with a non-Gaussian noise. Due to this, some measurements are at a relatively large distance from the primary data volume and form the so-called "tails".

The instability of LSM estimation in the presence of such interference led to the development of alternative, robust estimation in statistics, the purpose of which was to eliminate the impact of large errors.

It should be noted that a quite common model of clogging is the Tukey-Huber model [10]

$$\rho(\xi) = (1-\varepsilon)\rho_0(\xi) + \varepsilon q(\xi), \quad (3)$$

where $\rho_0(\xi)$ is the density of the respective main distribution; $q(\xi)$ is the density of the clogging (random distribution); $\varepsilon \in [0,1]$ is a parameter describing the degree of clogging of the main distribution.

When using model (3), it is also commonly assumed that both $\rho_0(\xi)$, and $q(\xi)$, are Gaussian with zero mathematical expectations and different variances.

However, for many information-processing tasks (interpolation, estimation and modeling of attributes, related to spatial distribution, the presence of pulse noise with long "tails", etc.), there is often the need to account for asymmetry in the distribution of original data (and interferences).

If the information about belonging of interference ξ to some specific class of distributions is known, then, by minimizing an optimal criterion, which is a logarithm of distribution function of the interference, taken with opposite sign, one can obtain an estimate for maximum likelihood. If such information is not available, then, in order to evaluate the desired vector of parameters, one should apply any nonquadratic criterion that would enable the robustness of estimate to be obtained.

Among the main types of robust estimations, M-, L-, and R-estimates, training tasks most commonly employ the M-estimation proposed in [10].

3. The aim and objectives of the study

The aim of present study is to develop a robust approach to training CMAC and to investigate the properties of a network, obtained as a result of combining a traditional CMAC with the apparatus of fuzzy logic and a robust M-estimation, the robust FCMAC – Fuzzy CMAC. This would make it possible to implement a robust approach to solving problems on the identification of and control over non-linear dynamic systems, as well as enable the use of a FCMAC network in systems with strong perturbations.

To accomplish the aim, the following tasks have been set:

– to synthesize a structure of the fuzzy CMAC network (FCMAC);

– to develop a robust algorithm for configuring FCMAC;

– to perform a test simulation to evaluate the effectiveness of application of the proposed method for the identification of non-linear dynamic objects.

4. Architecture of the robust neural network FCMAC

The main part of FCMAC is a linguistic description of steps, carried out in accordance with the current state of the object. While a traditional CMAC simulates a physical system or a real process, FCMAC makes it possible to include in the operation algorithm an expert knowledge and can be considered as a real time expert system [11–13].

Let is consider the principle of operation of the network, an example of which is shown in Fig. 1.

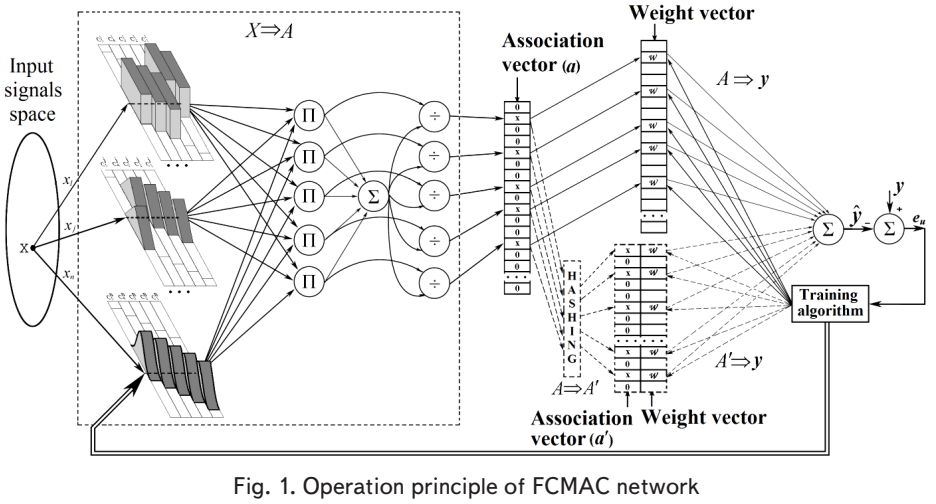


Fig. 1. Operation principle of FCMAC network

Because sensors and controlling elements use definite values, the network employs two subsystems: a fuzzification subsystem (converts definite values to fuzzy values) at the input and a defuzzification subsystem (converts fuzzy values to definite values) at the output. Fuzzification is executed by using an input membership functions and defuzzification can be implemented applying special procedures, which will be described below.

Fuzzy rules describe possible states of the system and their corresponding actions. The rules implemented by the fuzzy neural network FCMAC take the form

$$\text{IF } x_1 \text{ IS } X_{1,i} \text{ AND}... x_N \text{ IS } X_{N,i} \text{ THEN } \hat{f}_i = w_i,\ i = 1,...,\rho,$$

where $X_{1,i},...,X_{N,i}$ are linguistic values in the antecedent of the i-th rule; f_i is the function in the consequent of the i-th rule; w_i are the adjustable parameters in the consequent of the i-th rule; ρ is a parameter, assigned *a priori*, which defines the number of active membership functions (quantization steps); N is the dimensionality of the input signal.

Thus, each component of the input signal X is sent to the appropriate association matrix in the fuzzification unit containing R_i quantization levels. In addition, each level of quantization of the i-th component has the same number ρ of quantization steps (C_1^i, C_2^i, ..., C_ρ^i). Each quantization step includes $\rho^* \leq \rho$ of quantization regions, denoted by some intermediate variables and which represent certain pre-selected membership functions. N-dimensional signal $\boldsymbol{x}$, arriving at the network input, activates $N\rho$ membership functions (each component x_i excites ρ membership functions – one for each level of quantization in the corresponding association matrix).

The association matrices outputs produce signals whose magnitude is determined by the value of the accepted membership function.

To solve the problem on indirect adaptive control, it is required that t-norm should be differentiable for input variables and for parameters of membership functions. This condition should also be met when FCMAC network employs learning methods that make it possible to tune parameters of membership functions. To this end, a t-norm used in the network shown in Fig. 1 is the product, however, if a network is designed to solve the problem on approximation of functions or the task of indirect control with a reference model, it is possible to apply other t-norms.

When a t-norm used is the product, the degree of fulfillment of the i-th rule is calculated according to expression

$$\bar{\Phi}_i(x) = \prod_{j=1}^{N} \varphi_{ij}(x_j),\quad i = \overline{1,\rho}, \tag{4}$$

where $\varphi_{ij}(x_j)$ is the membership function at the j-th input in the antecedent of the i-th rule.

After computing the degree of compliance with the rules, their normalization is performed in accordance with formula

$$\Phi_i(x) = \frac{\prod_{j=1}^{N} \varphi_{ij}(x_j)}{\sum_{k=1}^{\rho} \prod_{j=1}^{N} \varphi_{ij}(x_j)},\ i = \overline{1,\rho}. \tag{5}$$

Components of the resulting vector $\Phi(x)$ are recorded to the corresponding cells of association vector a, which is the code of the received input signal X.

Dimensionality of the association vector a defines the number of weights (configurable parameters) of the network and can be derived from formula

$$n_{\max} = \left\rceil \rho \left(\frac{R-1}{\rho} + 1 \right)^N \right\lceil, \tag{6}$$

where R is the applied number of quantization levels of input signals; N is the dimensionality of the input vector; $\rceil\bullet\lceil$ denotes rounding toward the nearest larger integer.

One can see from (6) that the volume of memory required for the implementation of a network dramatically increases with an increase in the dimensionality of vector of input signals N. Consequently, it is quite difficult to implement a FCMAC network to solve the tasks on identification of and control over objects with a large dimensionality. Given this, in order to reduce the memory used, the hashing of information is employed [14–16], at which the resulting association vector $\mathbf{a}$ is converted, by using a certain hashing algorithm, into the new vector $\mathbf{a}' = H(\mathbf{a})$ with a smaller dimensionality. Here $H(\bullet)$ is the transformation performed by a hashing algorithm. The number of nonzero components in vector $\mathbf{a}'$ can be smaller than ρ (corresponding units and connections at hashing are shown in the bottom right part of Fig. 1 with dotted lines).

A network output is computed as a weighted sum of all nonzero association vector components $\mathbf{a}$ (in the absence of hashing), or $\mathbf{a}'$ (if it is applied), that is defuzzification is carried out at this stage.

Thus, in a general case, a CMAC network performs transformations

$$\text{S: X} \Rightarrow \text{A}, \tag{7}$$

$$H: A \Rightarrow \text{A}', \tag{8}$$

$$P: \text{A}' \Rightarrow \mathbf{y}, \tag{9}$$

where X is the N-dimensional space of continuous input signals; A is the n-dimensional space of associations; A' is the space of associations transformed by the hashing algorithm; $\mathbf{y}$ is the vector of output signals.

Transformation (7) corresponds to the coding of information

$$\mathbf{a}=S(\mathbf{x}), \tag{10}$$

(8) – to hashing

$$\mathbf{a}' = H(\mathbf{a}),$$

and (9) – to the output signal computation

$$\hat{y} = P(\mathbf{a}') = (\mathbf{a}')^{\text{T}}\mathbf{w} = (H(\mathbf{a}))^{\text{T}}\mathbf{w}, \tag{11}$$

where T is the transposition.

Expression (11) describes the transformation taking place in a traditional CMAC using the hashing of information (not always applied). If a network employs the apparatus of fuzzy logic, transformation (11) takes the form

$$\hat{y} = H(\mathbf{a}^{\text{T}}\boldsymbol{\Phi}(\mathbf{x}))\mathbf{w}, \tag{12}$$

where

$$\Phi(x)=\begin{bmatrix}\Phi_1(x) & 0 & ... & 0\\ 0 & \Phi_2(x) & ... & 0\\ ... & ... & ... & ...\\ 0 & 0 & 0 & \Phi_n(x)\end{bmatrix}.$$

For a traditional CMAC, $\mathbf{\Phi(x)=I}$, where $\mathbf{I}$ is the identity matrix.

4. 1. Choice of membership functions

The choice of membership functions significantly affects approximating properties of a FCMAC network.

As already noted, traditional CMAC executes continuous piecewise approximation, which is a consequence of its using the rectangular membership functions. When choosing membership functions of this type, computing costs would be minimal. Components of the association vector in this case can accept values of either "0" or "1". Consequently, this greatly reduces a network response time to the arrived input signal, that is, the rate of network learning will be maximum.

A FCMAC network can employ, for example, the following functions as a membership function [17]:

$$\Phi_i(x)=\exp\left\{-\frac{\|x-\mu_i\|^2}{\sigma_i^2}\right\}, \quad (13)$$

$$\Phi_i(x)=\left(1-\frac{(x-\mu_i)^2}{\sigma_i^2}\right)e^{\frac{(x-\mu_i)^2}{\sigma_i^2}}, \quad (14)$$

$$\Phi_i(x)=\exp\left\{-\frac{|x-\mu_i|}{\sigma_i}\right\}, \quad (15)$$

$$\Phi_i(x)=\frac{2(x-\mu_i)}{\sigma_i}\exp\left\{-\frac{(x-\mu_i)^2}{\sigma_i^2}\right\}, \quad (16)$$

where $\|\cdot\|$ is the Euclidean norm; μ_i is the center of the i-th quantization region; σ_i is the size of the i-th quantization region.

In addition, a FCMAC network quite often employs, as membership functions, the B-splines of various orders [14, 15] whose apparent advantage is the possibility for recurrent calculation of both the B-splines, in accordance with formula

$$B_{n,j}(x)=$$
$$=\left[\frac{x-\lambda_{j-n}}{\lambda_{j-1}-\lambda_{j-n}}\right]B_{n-1,j-1}(x)+\left[\frac{\lambda_j-x}{\lambda_j-\lambda_{j-n+1}}\right]B_{n-1,j}(x), \quad (17)$$

and their derivatives of the δ-th order

$${}^{(\delta)}B_{n,j}(x)=\frac{(n-1)}{(n-\delta-1)}\times$$
$$\times\left\{\left[\frac{x-\lambda_{j-n}}{\lambda_{j-1}-\lambda_{j-n}}\right]^{(\delta)}B_{n-1,j-1}(x)+\left[\frac{\lambda_j-x}{\lambda_j-\lambda_{j-n+1}}\right]^{(\delta)}B_{n-1,j}(x)\right\}, \quad (18)$$

where ${}^{(\delta)}B_{n,j}(x)$ is the derivative of δ-th order from B-spline of the n-th order in the j-th quantization region.

Here

$$B_{0,j}(x)=\begin{cases}1, \text{ if } x\in[\lambda_{j-1},\lambda_j);\\ 0, \text{ otherwise;}\end{cases} \quad {}^{(0)}B_{0,j}(x)=\begin{cases}1, \text{ if } x\in[\lambda_{j-1},\lambda_j);\\ 0, \text{ otherwise;}\end{cases}$$

$${}^{(\delta)}B_{n,j}(x)=\left[\frac{{}^{(\delta-1)}B_{n-1,j-1}(x)}{\lambda_{j-1}-\lambda_{j-n}}\right]-\left[\frac{{}^{(\delta-1)}B_{n-1,j}(x)}{\lambda_j-\lambda_{j-n+1}}\right];$$

λ_j is the j-th knot of the spline (quantization region center).

Thus, upon determining active interval $(\lambda_{j-1},\lambda_j]$ for the first-order B-spline, these expressions can be used to retrieve the values of all B-splines of higher orders and, if necessary, their derivatives.

Thus, a traditional CMAC is matched by the zero-order B-spline; when choosing the first-order B-spline we obtain triangular membership functions; the fourth-order B-spline is similar to choosing the Gaussian membership function.

It should be noted, however, that although the most commonly used Gaussian membership function allows very simple calculation of derivatives and possess the property of local excitation, it is difficult to clearly identify their excitation limits, which is important in order to realize a FCMAC network. It is possible, in order to eliminate this shortcoming, to use a modified Gaussian function

$$\varphi_i(x)=\begin{cases}\exp\left\{-\frac{(\lambda_2-\lambda_1)^2/4}{(x-\lambda_1)(\lambda_2-x)}\right\} \text{ for } x\in(\lambda_1,\lambda_2);\\ 0, \text{ otherwise.}\end{cases} \quad (19)$$

One can see from expression (19) that this function is strictly defined in interval (λ_1, λ_2), which simplifies the scaling of basis functions when changing such network parameters as R and ρ.

Because the value for the association vector components is obtained by multiplying the corresponding values of membership functions, it is required to consider the following fact. Such membership functions as the B-splines of the fourth, and higher, orders, as well as the Gaussian function and a modified Gaussian function, accept the values, near the bound of a quantization region, close to zero. Hence, it follows that the values for the association vector components will also tend to zero, which would have a negative effect on the properties of the network. To avoid the specified drawback, it is possible to use as membership functions the trigonometric functions, for example, the cosinusoidal function, of the form

$$\varphi_i(x_j)=\begin{cases}\cos\left(\frac{\pi}{\rho r_j}(x_j-\lambda_i)\right) \text{ for } {}_j\in\left(\lambda_i-\frac{\rho r_j}{2},\lambda_i+\frac{\rho r_j}{2}\right];\\ 0, \text{ otherwise,}\end{cases} \quad (20)$$

where λ_i is the i-th center of quantization region; r_j is the step of quantization for the j-th component of the input signal.

Trigonometric functions also make it possible to easily calculate their derivatives. It is easy to make sure that in this case the minimum values for the association vector components will slightly increase.

Fig. 1 shows a circuit of FCMAC for the n-dimensional case, which can possess different membership functions for different variables. In this figure, the basis functions, for

the sake of simplicity, are displayed as volumetric shapes, although, as demonstrated by (13)–(20), they are flat.

5. Robust algorithm for training a FCMAC neural network

Determining all network parameters, that is, in a general case, vector

$$\theta(k)=(a_0(k), w_1(k), \mu_1^T(k), \sigma_1(k),..., w_N(k), \mu_N^T(k), \sigma_N(k))^T,$$

is carried out by training it with a trainer.

In this case, an estimation criterion (learning criterion) may be represented as follows:

$$F[e(k)]=\sum_{i=1}^{k}\rho(e(i)), \tag{21}$$

where $\rho(e(i))$ is a certain loss function.

The task of training implies finding an estimate $\hat{\theta}$, defined as a solution to the extreme problem on minimum

$$F(\theta)=\min F[e(k)] \tag{22}$$

or as a solution to the system of equations

$$\frac{\partial F(e)}{\partial\theta_j}=\sum_{i=1}^{k}\rho'(e(i))\frac{\partial e(i)}{\partial\theta_j}=0, \tag{23}$$

where

$$\rho'(e(i))=\frac{\partial\rho(e(i))}{\partial e(i)}$$

is a function of influence.

If we introduce a weight function

$$\omega(e)=\frac{\rho'(e)}{e},$$

the system of equations (23) can be written as follows:

$$\sum_{i=1}^{k}\omega(e(i))e(i)\frac{\partial e(i)}{\partial\theta_j}=0, \tag{24}$$

and minimization of functional (21) will be equivalent to the minimization of the weighted quadratic functional

$$\min\sum_{l=1}^{k}\omega(e_l)e_l^2. \tag{25}$$

When choosing

$$\rho(e(i))=\frac{1}{2}e^2(i),$$

the influence function is $\rho'(e(i))=e(i)$, that is, it increases linearly with an increase in $e(i)$, which explains the instability of LSM estimation against emissions and interference whose distributions possess large tails. A variety of robust learning algorithms for ANN, minimizing (21), are given in papers [20–24].

Training a traditional CMAC, which employs rectangular basis functions, is executed at every cycle following the presentation of training pairs $\{\mathbf{x}(k),\mathbf{y}(k)\}$, where $\mathbf{y}(k)$ is the value of function corresponding to $\mathbf{x}(k)$. The training implies correcting only those of its ρ weights that match singular components of the association vector for a given vector $\mathbf{x}(k)$. In this case, a learning rule for all i, j, for which $a_i(k)=a_j(k)=1$, takes the form

$$w_j(k+1)=w_j(k)+\gamma\left(y(k)-\frac{1}{\rho}\sum_{i=1}^{n}w_i(k)\right), \tag{26}$$

where $\gamma\in(0,1]$ is a parameter that affects the speed of learning.

When using membership functions that take the form other than rectangular, a given algorithm can be written as follows:

$$w(k+1)=$$
$$=w(k)+\gamma(k)\left(\frac{y(k)-a^T(k)\Phi(x)w(k)}{\|\Phi(x)a(k)\|^2}\Phi(x)a(k)\right), \tag{27}$$

where $\gamma(k)$ is a certain, in a general case, variable parameter.

The properties of algorithm (27) depend largely on the choice of the parameter $\gamma(k)$. It is easy to demonstrate that the optimal value for this parameter that provides a maximum learning rate in the absence of interference will be equal to 1. To ensure the convergence of algorithm (27) in the presence of measurement interference, parameter $\gamma(k)$ must meet Dvoretzky conditions [25]. In this case, parameter $\gamma(k)$ should not decrease too rapidly, in particular, it would suffice to select $\gamma(k)$ of the type

$$\gamma(k)=\gamma^{\alpha k}, \tag{28}$$

where $\gamma(k)\in(0,1)$, $0<\alpha<<1$.

M-estimation also represents the evaluation of $\hat{\theta}$, defined as a solution to extreme problems (22) or as a solution to the system of equations (23), however, the loss function $\rho(e_i)$ is chosen other than quadratic. Studying different classes of interference distributions made it possible to obtain for these classes the least favorable distributions, that is, those that minimize the Fischer information. The use of these distributions, in turn, determines the form of a loss function and allows obtaining robust estimations that are applicable for almost any interference distributions [26, 27].

Currently, there are many such functions $\rho(e)$. Table 1 gives several typical representatives of functionals and their derivatives, employed in the robust estimation.

The gradient network learning algorithm will take the form

$$\hat{\theta}(k)=\hat{\theta}(k-1)+\gamma(k)\frac{\partial F(e(k))}{\partial\theta_j} \tag{29}$$

or

$$\hat{\theta}(k)=\hat{\theta}(k-1)+\gamma(k)\rho'(e(k))\frac{\partial e(k)}{\partial\theta_j}, \tag{30}$$

where $\gamma>0$ is the parameter that affects learning rate and which can be selected different for various network parameters.

Algorithms for configuring specific network parameters when selecting, for example, the Gaussian membership functions, will take the following form:

$$w_j(k+1)=w_j(k)-\gamma(k)e(k)a_j(k)\prod_l e^{-\left(\frac{x_i(k)-m_l(k)}{\sigma_l(k)}\right)^2}; \tag{31}$$

$$\sigma_{ij}(k+1) =$$

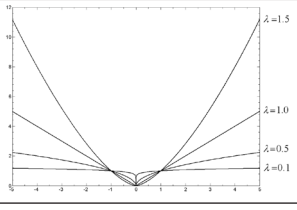
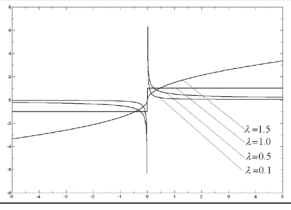
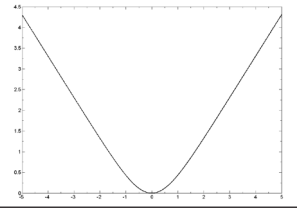
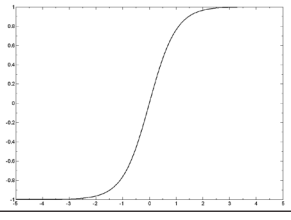
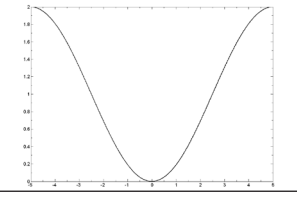
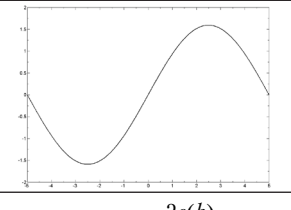
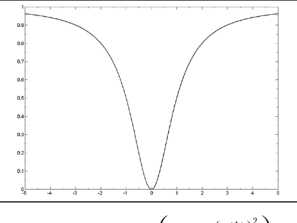
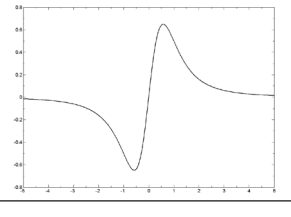
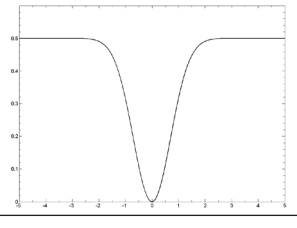
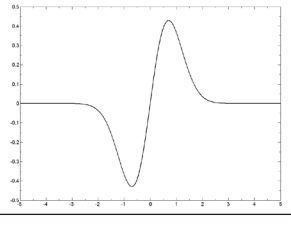
$$= \sigma_{ij}(k) - \gamma(k)e(k)a_j(k)w_j \left(\prod_l e^{-\left(\frac{x_i(k)-m_l(k)}{\sigma_l(k)}\right)^2} \right) \frac{2\left(x_i(k)-m_j(k)\right)^2}{\sigma_j^3(k)}; \quad (32)$$

$$m_{ij}(k+1) =$$

$$= m_{ij}(k) - \gamma(k)e(k)a_j(k)w_j \left(\prod_l e^{-\left(\frac{x_i(k)-m_l(k)}{\sigma_l(k)}\right)^2} \right) \frac{2\left(x_i(k)-m_j(k)\right)}{\sigma_j^2(k)}. \quad (33)$$

Table 1

Functionals and their derivatives used in the robust estimation

Functional	Derivative
$\rho[e(k)] = \lvert e(k)\rvert^{\lambda}$	$\rho'[e(k)] = \lambda \lvert e(k)\rvert^{\lambda-1} sign(e(k))$
	
$\rho[e(k)] = \ln\cosh\left(e(k)\right)$	$\rho'[e(k)] = \tanh\left(e(k)\right)$
	
$\rho[e(k)] = \begin{cases} 1-\cos\left(\frac{\pi e(k)}{c}\right), \lvert e(k)\rvert \le c; \\ 2, \lvert e(k)\rvert > c. \end{cases}$	$\rho'[e(k)] = \begin{cases} \frac{\pi}{c}\sin\left(\frac{\pi e(k)}{c}\right), \lvert e(k)\rvert \le c; \\ 0, \lvert e(k)\rvert > c. \end{cases}$
	
$\rho[e(k)] = \frac{e^2(k)}{1+e^2(k)}$	$\rho'[e(k)] = \frac{2e(k)}{\left(1+e^2(k)\right)^2}$
	
$\rho[e(k)] = 0.5c^2\left(1-e^{-\left(\frac{e(k)}{c}\right)^2}\right)$	$\rho'[e(k)] = e(k)e^{-\left(\frac{e(k)}{c}\right)^2}$
	

It is easy to construct network learning algorithms when selecting the membership functions of other forms. However, as already noted, some of the membership function (B-splines of the fourth, and higher, orders, the Gaussian function, and the modified Gaussian function) possess, near the border of a quantization region, values that are close to zero. Therefore, the values for the association vector components will also approach zero, which will have a negative effect on the properties of the network. In this case, the use of normalization (5) is ineffective; moreover, even the introduction of certain regularization to learning algorithms may not yield the desired result.

In addition, when correcting the centers of membership functions, one may find that these centers are beyond the respective intervals. Therefore, their values for these intervals will be equal to zero and a corresponding region of quantization of the input signal will not participate in the processing of information. An attempt to impose constraints on the range of acceptable changes in the values of centers leads to a significant complication in correction algorithms and prolong the process of network learning. Therefore, it is often more effective to correct only the weight parameters of the network.

Note that training only the weight parameters of a network is performed when using B-splines.

5. 1. Identification

A problem on the identification of a nonlinear dynamic object that is represented by the NARMAX model [31]

$$\tilde{y}(k+1) = f[y(k),..., y(k-m), u(k),..., u(k-n), k] + \xi(k+1), \quad (34)$$

where $\tilde{y}(i)$, $u(i)$ are the output and input signals of the object at time i, respectively; m, n are the orders of delay for the output and input channels, respectively; $f[\bullet]$ is an unknown nonlinear function; ξ is the interference in the measurement of the output signal, implies the estimation of function $f(\bullet)$ based on the measurement of the input $u(k)$ and the output $\tilde{y}(k+1)$ variables.

We shall denote a vector of the generalized input signal with a dimensionality of $(n+m)x1$ as

$$x(k) = [y(k), y(k-1),..., y(k-m), u(k), u(k-1),...,u(k-n)],$$

record equation (34) in the following form:

$$\tilde{y}(k+1) = f[x(k), k] + \xi(k+1), \quad (35)$$

that is, reduce to form (1), applied at a neural-network representation.

Thus, the problem on identification is reduced in this case to training a network, which implies configuring its weight parameters based on the minimization of a certain functional in the identification error

$$\mathbf{e}_i(k+1) = y(k+1) - \hat{y}(k+1).$$

Examples of using a CMAC network for the identification of nonlinear objects can be found, in particular, in [29–31].

5. 2. Control problem

The task on control over a non-linear object (34) implies finding a control action $u(k)$, which minimizes an error of control

$$e_u(k+1)=y^*(k+1)-y(k+1),$$

where y^* is the desired value of the output signal. This is equivalent to the problem of optimization

$$\min_{u(k)}\left|y^*(k+1)-y(k+1)\right|$$

or

$$\min_{u(k)}\left|y^*(k+1)-\mathrm{f}\left[\mathbf{y}(k),\ldots,y(k-m+1),u(k),\ldots,u(k-n+1)\right]\right|.$$

When function $f[\bullet]$ is unknown, it is required to solve a problem on identification, which implies obtaining an estimate $\hat{f}[\bullet]$, which is used in the control law. Thus, a control problem is reduced to solving the optimization problem

$$\min_{u(k)}\left|y^*(k+1)-\hat{\mathrm{f}}\left[y(k),\ldots,y(k-m+1),u(k),\ldots,u(k-n+1)\right]\right|.$$

The task to control, based on a traditional CMAC network, the objects whose description represented particular cases (34), was addressed in many papers, specifically in [32–36].

If one selects quadratic $\mathbf{e}_u^2(k+1)$, as a criterion, the law of control that implements indirect adaptive control without a reference model represents a certain gradient procedure

$$u(k+1)=u(k)+\gamma\nabla_u\hat{y}(k)e_u(k+1), \quad (36)$$

where γ is a parameter that affects the algorithm configuration rate:

$$\nabla_u\hat{\mathbf{y}}(k)=\frac{\partial\hat{\mathbf{y}}(k)}{\partial\mathbf{u}(k)}=\sum_{i=1}^{n}a_i w_i\frac{\partial\Phi_i}{\partial\mathbf{u}(k)}.$$

Because in this case the use of a traditional CMAC network with membership functions of rectangular shape is not possible, implementation (36) may be based on FCMAC whose membership functions are differentiable.

6. Discussion of results of experimental study of the proposed method for identification and control

In the course of studying the proposed method for the identification and control of nonlinear dynamical systems using a FCMAC network, we conducted a series of experiments. The aim was to demonstrate the effectiveness of using a robust FCMAC network to identify noisy dynamic objects.

Experiment 1. We consider a problem on the identification of a nonlinear dynamic object described by equation [28]

$$y(k)=0.5075\beta(k)\sin\left(\frac{16u(k-1)+8y(k-1)}{0.7\left(3+4u^2(k-1)+4y^2(k-1)\right)}\right)+$$
$$+0.2u(k-1)+0.2y(k-1), \quad (37)$$

using a FCMAC network with different basis functions and applying hashing algorithms.

Fig. 2, 3 show results of the object identification (37) using the rectangular and Gaussian membership functions, respectively. Fig. 2, *a* and Fig. 3, *a* show the surfaces, restored by a FCMAC network without application of hashing algorithms. The volume of memory required in this case amounted to 708 cells. Fig. 2, *b* and Fig. 3, *b* demonstrate results of the identification using the hashing, corresponding to selecting in the algorithm (19) m=500; Fig. 2, *c* and Fig. 3, *c* – m=400; Fig. 2, *d* and Fig. 3, *d* – m=350.

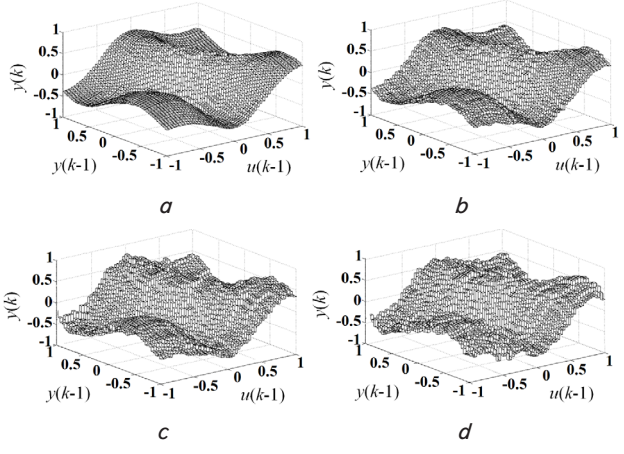


Fig. 2. Results of the identification of object (37) using rectangular membership functions: *a* — surface restored without the use of hashing; *b* — results of the identification using hashing at m=500; *c* — at m=400; *d* — at m=350

Experiment 2. We solved a problem on the identification of a very noisy object described by equation

$$y(k)=\sin(u(k))+\frac{y(k)}{1+y^2(k)}+\xi(k). \quad (38)$$

The surface described by equation (38) is shown in Fig. 4, *a*. An output signal of the object was noisy due to an interference, uniformly distributed in interval [−0.4, 0.4], resulting in the noise equal to 20 % of the useful signal (Fig. 4, *b*). To solve the problem on identification, we used a FCMAC neural network with the following parameters: R=100, ρ=20, N=708 (volume of the required memory), membership functions are cosinusoidal. The network was presented with 100,00 training pairs. Fig. 4, *c* shows identification results when using algorithm (27) with $\gamma(k)=1$; Fig. 4, *d* – with $\gamma(k)$, modified in accordance with formula (28) at $\alpha=0.01$ and $\gamma=0.99$. The results of modeling show that algorithm (27) with a variable parameter $\gamma(k)$ produces significantly better results and almost completely eliminates the negative impact of an interference.

Experiment 3. We solve a problem on the identification and indirect control, without a reference model, of a very noisy object (38). Network parameters and noise parameters were taken similar to those from experiment 2. However, we chose the form of a membership function in a given example in a different way: membership functions for the network input signal $y(k)$ that we selected were rectangular, while for the input control signal $u(k)$ – cosinusoidal.

The network was trained on 100,00 training pairs. Fig. 5, *a* show results of the identification using algorithm (27) with $\gamma(k)=1$, Fig. 5, *b* – using algorithm (27) with a variable parameter $\gamma(k)$ in line with (28) at $\alpha=0.01$ and $\gamma=0.95$.

Fig. 5, *c*, *d* show the results of control over object (38). The desired object's output signal change law was assigned in the following way:

$$y^*(k)=\begin{cases}0.7\sin(\pi x/100), & \text{for } x\in[0,400];\\ -0.5, & \text{for } x\in[401,600];\\ -0.5+0.003x, & \text{for } x\in[601,1000].\end{cases}$$

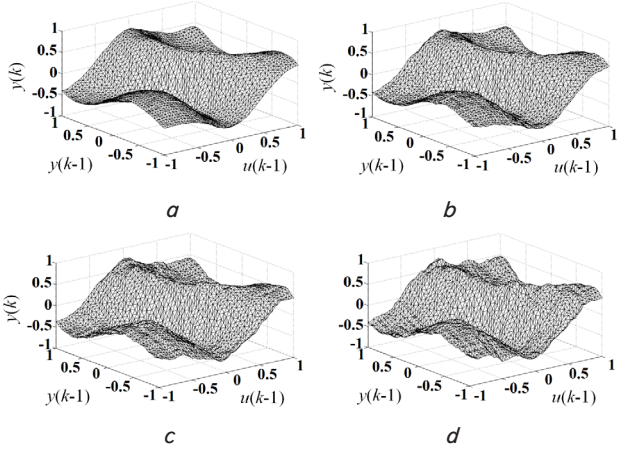


Fig. 3. Results of the identification of object (37) using the Gaussian membership functions: *a* – surface restored without the use of hashing; *b* – results of the identification using hashing at m=500; *c* – at m=400; *d* – at m=350

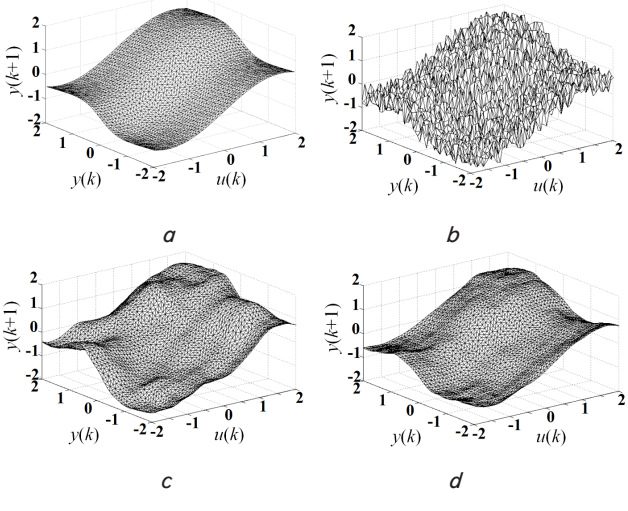


Fig. 4. Results of the identification of a very noisy object (38): *a* – reference surface; *b* – noisy surface; *c* – results of the identification using algorithm (27) with $\gamma(k)$=1; *d* – with $\gamma(k)$ modified in accordance with formula (28)

Here, a solid line marks the desired output signal, dotted – actual output signal, solid with circles – control signal.

The results of modeling show algorithm (27) with $\gamma(k)=\gamma^{0.01k}$ produces significantly better results, almost completely eliminating the negative influence of an interference, which positively affects the results of control.

A choice of different basis functions for different input signals was justified because, first, the training time reduced significantly, second, computational costs decreased, and, third, the calculation rate of control signals grew. A slightly increased level of the identification error did not in this case affect the results of control.

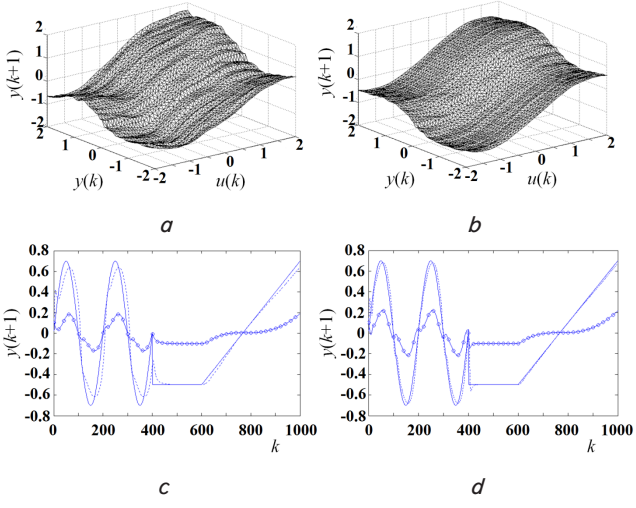


Fig. 5. Results of the identification and indirect control, without a reference model, of a very noisy object (38): *a* – results of the identification using algorithm (27) with $\gamma(k)$=1; *b* – with parameter $\gamma(k)$ that changes in line with (28); *c*, *d* – results of control

Experiment 4. In this experiment we solved a problem on the identification and control of a multidimensional object (MIMO) described by the following equations:

$$y_1(k)=\frac{15u_1(k-1)y_2(k-1)}{2+50\left[u_1(k-1)\right]^2}+$$
$$+0.5u_1(k-1)-0.25y_2(k-1)+0.1;$$

$$y_2(k)=\frac{\sin(\pi u_2(k-1)y_1(k-1))+2u_2(k-1)}{3}. \qquad (39)$$

To solve a given problem, we used a FCMAC network with four inputs and two outputs. The network had the following parameters: R=100, ρ=20, N=25,067 (volume of the required memory), membership functions for all network inputs are cosinusoidal. The network was trained using 10^6 training pairs.

Results of the work of a neural controller that implements algorithm (36) are shown in Fig. 6. In all these figures, a dotted line shows the desired output signal $y_i^*(k)$, solid – actual $\hat{y}_i(k)$, a line with circles – a corresponding change in control signal $u_i(k)$ (i=1, 2). The required values for output signals, for the example of solving a control problem, shown in Fig. 6, *a*, *b*, were assigned as follows:

$$y_1^*(k)=0.4\sin(\pi k/100)+0.1\cos(\pi k/200);$$

$$y_2^*(k)=\begin{cases}0.2 & \text{for } k=\overline{1,500};\\ 0.1 & \text{for } k=\overline{501,1000},\end{cases}$$

for the example shown in Fig. 6, *c*, *d*, signal $y_1^*(k)$ remained unchanged and a saw-shaped signal was chosen to serve as $y_2^*(k)$.

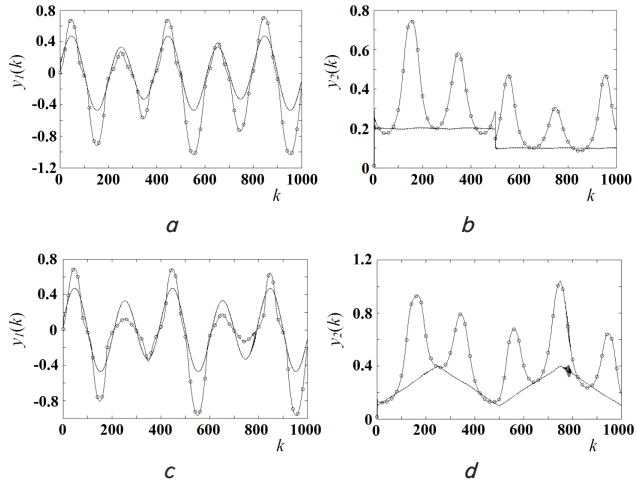


Fig. 6. Results of the work of a neural controller: *a* — for the first channel for a sinusoidal reference signal; *b* — for the second channel for a sinusoidal reference signal; *c* — for the first channel for a saw-shaped reference signal; *d* — for the second channel for a saw-shaped reference signal

Experiment 5. We solved a problem of control over a nonlinear object [28]

$$y(k+1)=f[y(k),y(k-1)]+u(k), \quad (40)$$

where

$$f[y(k),y(k-1)]=\frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)}, \quad (41)$$

using a reference model and hashing algorithms.

Control using a reference model rendered the following form to the neural network model

$$\hat{y}(k+1)=\hat{f}[y(k),y(k-1)]+u(k). \quad (42)$$

As the input signal $u(k)$, we selected a stationary random sequence with a uniform law of distribution in interval [−5, 5]. This object's identification was carried out based on 20,000 training pairs at the following network parameters: R=100, ρ=20, membership functions are cosinusoidal (17), length of the association vector totaled 708 bits.

A model of the following form was assigned as the reference model

$$y^*(k+1)=0.4y^*(k)+0.1y^*(k-1)+r(k) \quad (43)$$

with a reference input signal $r(k)=\sin(\pi k/150)$.

Because control action $u(k)$ is linearly included in equation (40), its value can be calculated as follows:

$$u(k)=-\hat{f}\left[y(k),\, y(k-1)\right]+ \\ +0.4y(k)+0.1y(k-1)+r(k), \quad (44)$$

where $\hat{f}(\bullet)$ is the non-linearity (41), restored by using a neural network model (42).

Results of the experiment are shown in Fig. 7, 8. Fig. 7, *a* shows the surface described by equation (34); Fig. 7, *b* – the surface restored by the neural network CMAC without hashing the information (M=708). Fig. 7, *c, d* shows results of the identification using a hashing algorithm at, respectively, m=500, m=400, and m=350.

Fig. 8, *a* shows results of control without using hashing algorithms; Fig. 8, *b* – those that correspond to selecting in algorithm (19): m=500; Fig. 8, *c* – m=400; Fig. 8, *d* – m=350. In all figures, a dashed line corresponds to the desired output signal, solid – to the actual one when sending to the input of the object of control signal computed from formula (44) and shown in figures with a solid line with circles. Control error, as demonstrated by the figures, increases when the amount of the required memory decreases.

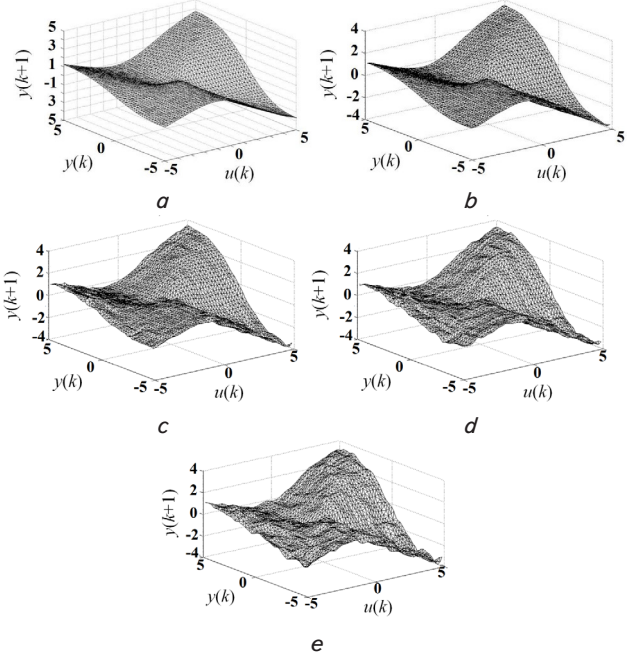


Fig. 7. Results of experiment 5: *a* — surface described by equation (34); *b* — surface restored without hashing the information; *c* — results of the identification using hashing at m=500; *d* — at m=400; *e* — at m=350

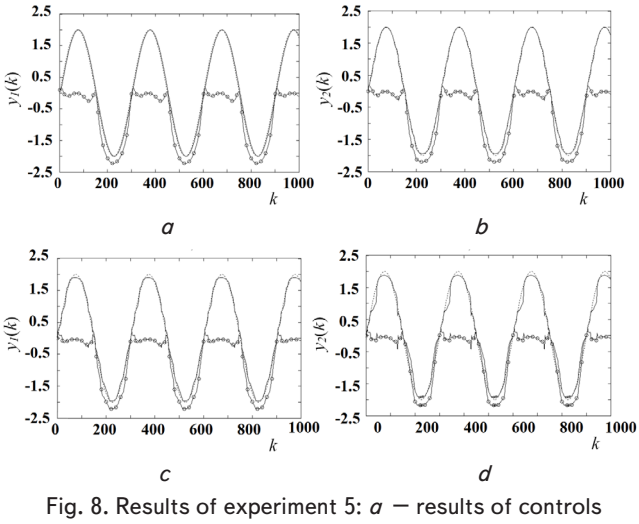


Fig. 8. Results of experiment 5: *a* — results of controls without using the hashing; *b* — using the hashing at m=500; *c* — at m=400; *d* — at m=350

As shown by the results of present study, the proposed structure of a robust FCMAC neural network, which is the result of combining a traditional CMAC network with the

apparatus of fuzzy logic and the robust M-estimation, makes it possible to realize a robust approach to the problems on identification and control of nonlinear dynamic systems in the presence of non-Gaussian noise. The choice of membership functions of FSMAS depends on the specific task to be solved, in particular, in order to synthesize control systems with a reference model, one should select membership functions of a rectangular shape, and when solving a problem on indirect control without a reference model – trigonometric functions or Gaussian functions. A distinctive feature of a given approach is the possibility to apply various types of membership functions for different input signals of FCMAC.

Despite significant positive results of the application of a given network in the synthesis of indirect control systems with or without a reference model under conditions of non-Gaussian noise, there are a number of problems that arise in the practical application of the proposed robust FCMAC network: the following ones are worth mentioning.

First, still unresolved is the task on the most efficient encoding of information, implying the selection of network parameters ρ and R, because, on the one hand, an increase in the values of these parameters leads to improved accuracy of signal representation, on the other hand, to an increase in the time for obtaining the model and the inconvenience of using this approach when solving problems in real time.

Second, there is currently no recommendations for choosing the most effective learning criterion, while solutions obtained at the minimization of different criteria have different properties. It seems therefore appropriate to undertake a study into comparative analysis of the effectiveness of the application of different criteria for different problems, as well as the development of relevant recommendations.

Third, using in a given network such a powerful apparatus as the hashing of information can significantly reduce the amount of memory required for network implementation, however, leads, because of the emergence of collisions, to the deterioration in the approximating properties of the network. Increasing the dimensionality of the problem that is being solved, that is an increase in the dimensionality of input signals N, predetermines the hashing of the information employed, as it often is not possible to realize a FCMAC network without reducing the amount of memory.

Finally, fourth, it should be noted that in the application of gradient algorithms in order to train a network and solve the tasks of identification and control, there is a problem related to the selection of the optimal values of parameter $\gamma(k)$, which is part of expressions (29)–(33), also (36), which ensure maximal speed of algorithms convergence (minimum time for training). It is obvious that the values of these algorithm's parameters will depend on the statistical properties of signals and interference. But because the latter are often not known, it is necessary to apply certain recurrent procedures to obtain the estimates (mathematical expectations, variances, covariances, etc.) and use these estimates in order to select or correct algorithms' parameters.

7. Conclusions

1. We have proposed a structure of the robust FCMAC neural network, which is the result of combining a traditional CMAC with the apparatus of fuzzy logic and the robust M-estimation. This neural-network structure makes it possible to implement a robust approach to solving the problems on identification and control of nonlinear dynamic systems.

2. The following practical recommendations were devised for choosing the membership functions of a FCMAC neural network.

– If a high speed and a relatively low accuracy of approximation are required, one should choose membership functions of a rectangular shape. The same choice is effective in the synthesis of control systems with a reference model.

– If a high accuracy of approximation is required, it is necessary to select B-splines of the second or fourth order, the Gaussian functions, or trigonometric functions, as membership functions.

– To solve a problem on indirect control without a reference model, it is required to use either Gaussian function or a trigonometric function as membership functions.

– In order to reduce computing costs and increase network performance, it is possible to employ different forms of membership functions for different input signals of FCMAC.

3. We have designed a robust algorithm to configure a FCMAC network that makes it possible to utilize a given network in control systems in the presence of non-Gaussian noise.

4. A study was conducted into information hashing algorithms in a F FCMAC network. The study revealed that hashing, while ensuring a significant reduction in the amount of memory required to implement a network, leads, because of collisions, to the deterioration in the approximating properties of the network. This has an adverse effect, in a greater extent, on the accuracy of solution to the identification problem, and, to a lesser degree, to the problem of control.

5. We have conducted simulation modeling, which confirmed that a FCMAC network, when choosing the appropriate membership functions, can be applied for the synthesis of indirect control systems with and without a reference model. However, it is more efficient to use it in control systems with the reference model. This sharply reduces the number of training pairs and simplifies the coding due to the narrower range of the applied values for input signals. It is also possible to significantly reduce the amount of required memory owing to hashing the information; while executing the control algorithm, there are no any problems associated with the application of gradient procedures (jamming at local extrema, plateaus, etc.). In the cases when the reference model is not assigned, or if the structure of an object is not known and control is implemented without a reference model, it is rather effective to employ a FCMAC network that uses differentiable membership functions.

References

1. Marr D. A theory of cerebellar cortex // The Journal of Physiology. 1969. Vol. 2002, Issue 2. P. 437–470. doi: 10.1113/jphysiol.1969.sp008820
2. Albus J. S. A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC) // Journal of Dynamic Systems, Measurement, and Control. 1975. Vol. 97, Issue 3. P. 220. doi: 10.1115/1.3426922
3. Albus J. S. Data Storage in the Cerebellar Model Articulation Controller (CMAC) // Journal of Dynamic Systems, Measurement, and Control. 1975. Vol. 97, Issue 3. P. 228. doi: 10.1115/1.3426923

4. Wang L.-X. Fuzzy systems are universal approximators // Proc. IEEE Int. Conf. On Fuzzy Systems. San Diego, 1992. P. 1163–1170. doi: 10.1109/fuzzy.1992.258721
5. Medical sample classifier design using fuzzy cerebellar model neural networks / Li H.-Y., Yeh R.-G., Lin Y.-C., Lin L.-Y., Zhao J., Lin C.-M., Rudas I. J. // Acta polytechnica Hungarica. 2016. Vol. 13, Issue 6. P. 7–24. doi: 10.12700/aph.13.6.2016.6.1
6. Lee C.-H., Chang F.-Y., Lin C.-M. An Efficient Interval Type-2 Fuzzy CMAC for Chaos Time-Series Prediction and Synchronization // IEEE Transactions on Cybernetics. 2014. Vol. 44, Issue 3. P. 329–341. doi: 10.1109/tcyb.2013.2254113
7. Chung C.-C., Lin C.-C. Fuzzy Brain Emotional Cerebellar Model Articulation Control System Design for Multi-Input Multi-Output Nonlinear // Acta Polytechnica Hungarica. 2015. Vol. 12, Issue 4. P. 39–58. doi: 10.12700/aph.12.4.2015.4.3
8. Xu S., Jing Y. Research and Application of the Pellet Grate Thickness Control System Base on Improved CMAC Neural Network Algorithm // Journal of Residuals Science & Technology. 2016. Vol. 13, Issue 6. P. 150.1–150.9.
9. Cheng H. The Fuzzy CMAC Based on RLS Algorithm // Applied Mechanics and Materials. 2013. Vol. 432. P. 478–782. doi: 10.4028/www.scientific.net/amm.432.478
10. Huber P. J., Ronchetti E. M. Robust Statistics. 2nd ed. Wiley, 2009. 380 p.
11. Jou C.-C. A fuzzy cerebellar model articulation controller // [1992 Proceedings] IEEE International Conference on Fuzzy Systems. 1992. doi: 10.1109/fuzzy.1992.258722
12. Nie J., Linkens D. A. FCMAC: A fuzzified cerebellar model articulation controller with self-organizing capacity // Automatica. 1994. Vol. 30, Issue 4. P. 655–664. doi: 10.1016/0005-1098(94)90154-6
13. Knuth D. Sorting and Searching. The Art of Computer Programming. Vol. 3. Menlo Park, Calif.: Addison Wesley, 1973. 506 p.
14. Wang Z.-Q., Schiano J. L., Ginsberg M. Hash-coding in CMAC neural networks // Proceedings of International Conference on Neural Networks (ICNN'96). 1996. doi: 10.1109/icnn.1996.549156
15. Rudenko O. G., Bessonov A. A. Heshirovanie informacii v neyronnoy seti SMAS // Upravlyayushchie sistemy i mashiny. 2004. Issue 5. P. 67–73.
16. Ching-Tsan C., Chun-Shin L. CMAC with General Basis Functions // Neural Networks. 1996. Vol. 9, Issue 7. P. 1199–1211. doi: 10.1016/0893-6080(96)00132-3
17. Lane S. H., Handelman D. A., Gelfand J. J. Theory and development of higher-order CMAC neural networks // IEEE Control Systems. 1992. Vol. 12, Issue 2. P. 23–30. doi: 10.1109/37.126849
18. Wang S., Lu H. Fuzzy system and CMAC network with B-spline membership/basis functions are smooth approximators // Soft Computing – A Fusion of Foundations, Methodologies and Applications. 2003. Vol. 7, Issue 8. P. 566–573. doi: 10.1007/s00500-002-0242-2
19. Rudenko O. G., Bessonov A. A. M-obuchenie radial'no-bazisnyh setey s ispol'zovaniem asimmetrichnyh funkciy vliyaniya // Problemy upravleniya i informatiki. 2012. Issue 1. P. 79–93.
20. Rudenko O. G., Bessonov A. A. Robastnoe obuchenie radial'no-bazisnyh setey // Kibernetika i sistemnyy analiz. 2011. Issue 6. P. 38–46.
21. Rudenko O. G., Bezsonov A. A. Robust Learning Wavelet Neural Networks // Journal of Automation and Information Sciences. 2010. Vol. 42, Issue 10. P. 1–15. doi: 10.1615/jautomatinfscien.v42.i10.10
22. Vazan M. Stohasticheskaya approksimaciya. Moscow: Mir, 1972. 289 p.
23. Cypkin Ya. Z. Osnovy informacionnoy teorii identifikacii. Moscow: Nauka, 1984. 320 p.
24. Cypkin Ya. Z., Polyak B. T. Ogrublennyy metod maksimal'nogo pravdopodobiya // Dinamika sistem. 1977. Issue 12. P. 22–46.
25. Narendra K. S., Parthasarathy K. Identification and control of dynamical systems using neural networks // IEEE Transactions on Neural Networks. 1990.Vol. 1, Issue 1. P. 4–27. doi: 10.1109/72.80202
26. Rudenko O. G., Bessonov A. A. Neyronnaya set' SMAS i ee primenenie v zadachah identifikacii i upravleniya nelineynymi dinamicheskimi ob'ektami // Kibernetika i sistemniy analiz. 2005. Issue 5. P. 16–28.
27. Liao Y.-L., Peng Y.-F. Applications of Prediction and Identification Using Adaptive DCMAC Neural Networks // International Journal of Computer and Information Engineering. 2011. Vol. 5, Issue 6. P. 677–682.
28. Zhao J., Lin L.-Y., Lin C.-M. A General Fuzzy Cerebellar Model Neural Network Multidimensional Classifier Using Intuitionistic Fuzzy Sets for Medical Identification // Computational Intelligence and Neuroscience. 2016. Vol. 2016. P. 1–9. doi: 10.1155/2016/8073279
29. Commuri S., Lewis F. L. CMAC neural networks for control of nonlinear dynamical systems, structure, stability, and passivity // Proc. IEEE Int. Symp. Intell. Control. San Francisco, 1995. P. 123–129.
30. Commuri S., Lewis F. L., Jagannathan S. Discrete-time CMAC neural networks for control applications // Proceedings of 1995 34th IEEE Conference on Decision and Control. New Orleans, 1995. doi: 10.1109/cdc.1995.478453
31. Jagannathan S. Discrete-time CMAC NN control of feedback linearizable nonlinear systems under a persistence of excitation // IEEE Transactions on Neural Networks. 1999. Vol. 10, Issue 1. P. 128–137. doi: 10.1109/72.737499
32. Lin C.-M., Peng Y.-F. Adaptive CMAC-Based Supervisory Control for Uncertain Nonlinear Systems // IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics). 2004. Vol. 34, Issue 1. P. 1248–1260. doi: 10.1109/tsmcb.2003.822281
33. A Novel Wavelet-based-CMAC Neural Network Controller for Nonlinear Systems / Lee C.-H., Wang B.-H., Chang H.-H., Pang Y.-H. // The 2006 IEEE International Joint Conference on Neural Network Proceedings. Vancouver, BC, Canada, 2006. P. 2593–2599. doi: 10.1109/ijcnn.2006.247136
34. Aved'yan E. D., Hormel' M. Povyshenie skorosti skhodimosti processa obucheniya v special'noy sisteme associativnoy pamyati // Avtomatika i telemekhanika. 1991. Issue 12. P. 100–109.
35. Fuzzy CMAC structures / Mohajeri K., Zakizadeh M., Moaveni B., Teshnehlab M. // 2009 IEEE International Conference on Fuzzy Systems. 2009. doi: 10.1109/fuzzy.2009.5277185