EFFECTIVE ELEMENTAL BASE OF SWITCHING NEUROSTRUCTURES

Alexander N. Sychev

Embassy of the Russian Federation in Latvia, Riga, LV-1010, Latvia

Abstract

Designs of logical elements and artificial neurons are described, which are simple in technical and software implementation. Walsh spectral representations are widely used.

Keywords: *filters, Fibonacci neurons, noise filters*

1. INTRODUCTION

Neural structures (NS), being information devices, are to provide noise tolerance reception and selection of the useful part of the incoming information in case of its redundancy. It is quite obvious that effective NS shall have appropriate design and contain different blocs which perform characteristic information processing functions.

Biological prototypes, for example, have areas of the nervous system that functionally solve various tasks, and have different types of neurons. It would be a mistake to create artificial NS, containing the elements of the same type connected in a standard way, for different situations. It would also be ineffective to limit oneself by a mathematical presentation of NS only. For example, it is monotonous to present the elements of NS by the structural vectors of the input scales and a threshold, in accordance with sigmoid or threshold function of the exitation. The situation is getting worse as such presentations possess low functional abilities and their application allows the user to solve a limited number of tasks. One of the ways how to overcome this drawback is to attract numerous elements, but such a decision will demonstrate both lack of creativity and lack of a good engineering idea.

The aim of this work is to give a description and substantiation of additional constructive decisions in the artificial NS, which possess efficiency both in the software and hardware implementation, involving the language of Walsh's spectral representations.

2. SWITCHING NEURAL STRUCTURE

Switching neural structure (SNS) analyzed in the work [1], represents itself as a network of channel switches, uniting a certain ensemble of neurons functioning independently from each other.

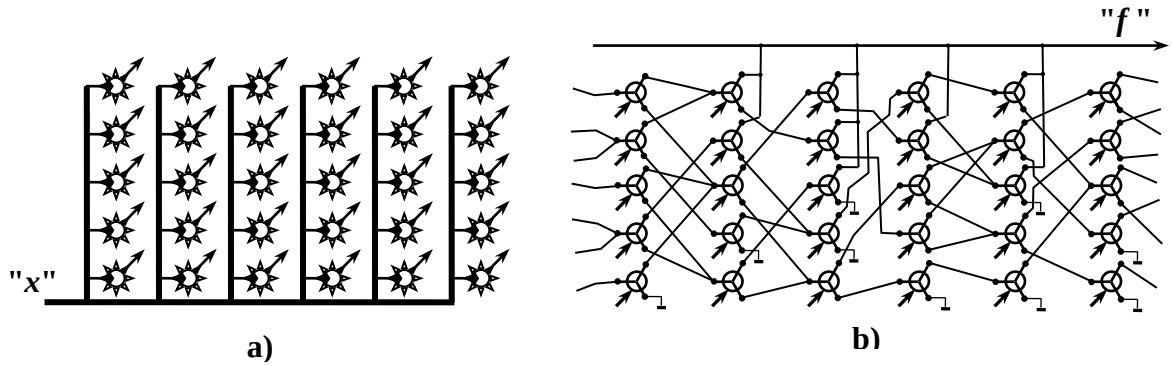


Fig. 1. Schematic representation of the region in the switching neuron network:

a) an ensemble/ set of neurons, controlling network switches,

b) the network of channel switches "1×2", corresponding to this ensemble.

The arrows indicate the circuits of neuron control signal transmission to the switches in the corresponding positions.

According to [1], every switch is associated with a certain part $X' \subset X$ in the area of X definition of the logical function $f(x)$, realized by the network. Neurons, which control the switches, are adjusted independently. At the same time, the resulting neuron setting forms function $\nu(x)$ that approximates $f(x)$ on the corresponding fragment X' most accurately.

Thus, SNS gives an opportunity to simplify, to a great extent, and make the process of neural structure synthesis more effective by the division of the definition area (or learning sequence). It is also necessary to note that the synthesis of the structure becomes more visual and controlled. At the same time, any learning sequence without contradictions can always be realized by the corresponding SNS.

3. LINEAR FILTERS

Neural structure is an informative device, so a preliminary input signal conversion is valid for it, as well. The aim of the conversion might represent, for example, optimization of the neural network or increase in the efficiency of the separate indices of its functioning. By analogy with other devices various aims of the input transformations can be achieved with the help of filter application.

As long as the neural network implements a certain logical (Boolean) function, then it is appropriate to apply the spectral representation according to Walsh in order to find efficient filters. It was stressed in the work of [2] that the group of Walsh functions is isomorphic in relation to the group of linear Boolean functions (BF). This fact allows one to solve the tasks related to the linearization of BF, applying the language of Walsh functions, and, vice versa, the decomposition by Walsh linear functions may be considered as the decomposition by linear BF. The choice of coefficient $S_f(\omega)$ in the spectrum S_f of the logical function (LF) f, is determined in accordance with the argument ω of the chosen linear filter coefficient (linear LF) $\lambda(x) = \omega_1 x_1 \oplus \omega_2 x_2 \oplus \ldots \oplus \omega_n x_n$, where the binary argument is $\omega = (\omega_1, \omega_2, \ldots, \omega_n)$.

Let, for example, the Karnaugh table set some LF f from $n = 6$ variables. Its range S according to Walsh is also represented by the corresponding Karnaugh table. For the table f the upper 3 rows of code lines represent (from the top down) arguments x_1, x_2, x_3, and for the table S, - arguments

$\omega_1, \omega_2, \omega_3$. Accordingly, the side 3 rows of code lines represent (from left to right) arguments x_4, x_5, x_6 and $\omega_4, \omega_5, \omega_6$.

$$f = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix} \quad S_f = \begin{vmatrix} 26 & -6 & 0 & 0 & 2 & -22 & 0 & 0 \\ -6 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ -2 & -2 & 0 & 0 & -2 & 6 & 0 & 0 \\ -2 & 6 & 0 & 0 & -2 & -2 & 0 & 0 \\ -2 & -2 & 0 & 0 & -2 & 6 & 0 & 0 \\ -10 & -2 & 0 & 0 & 6 & 6 & 0 & 0 \\ 2 & 2 & 0 & 0 & -6 & 2 & 0 & 0 \\ -6 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \end{vmatrix}$$

In accordance with the work [3] it is possible to identify 5 spectral dominants $S_f(\omega)$ with the linear independent arguments $\omega', \omega'', \omega''', \omega^{iv}, \omega^{v}$. They are highlighted with circles in the table S .Arguments of the dominants form matrix σ as a result of their linear transformation of the input argument x, the result of linearization is $x' = \sigma \otimes x$ (*mod 2*).

$$\sigma = \begin{vmatrix} \omega' \\ \omega'' \\ \omega''' \\ \omega^{iv} \\ \omega^{v} \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{vmatrix}$$

By the appropriate choice of the structural vector $A = (p; a_1, a_2, a_3, a_4, a_5) = (5;3,1,1,1,2)$ the realization of LF with one neuron and two input linear filters $\lambda' = x'_1 = (x_1 \oplus x_2 \oplus x_3)$ and $\lambda'' = x'_5 = (x_4 \oplus x_5 \oplus x_6)$ is provided.

$$x' = (x'_1, x'_2, x'_3, x'_4, x'_5) = (x_1 \oplus x_2 \oplus x_3, x_6, x_4, x_3, x_4 \oplus x_5 \oplus x_6)$$

$$f(x) = \begin{cases} 1, & if \sum_{i=1\div5} a_i x'_i \geq 5 \\ 0, & in\ other\ cases \end{cases}$$

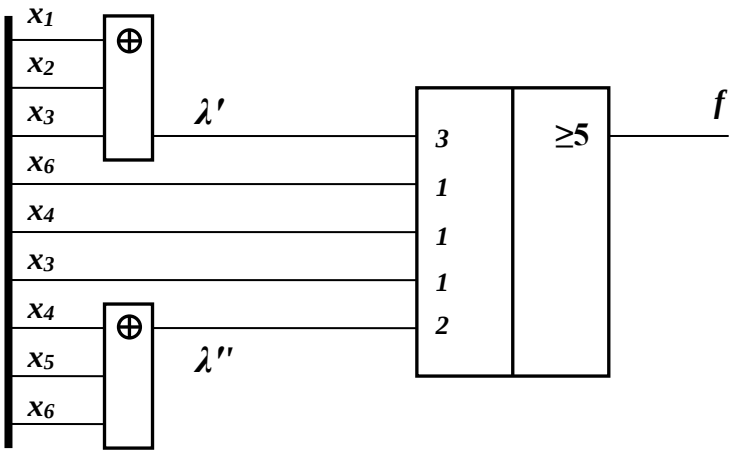


Fig. 2. Realization of LF $f(x)$ by a neuron and two linear filters at the input.

Realization of LF by the given network, containing only neurons, will demand the increased number of them. Furthermore, a linear filter constriction is simpler than that of a neuron, and software matrix calculations, corresponding to the filter functioning, are also much simpler.

4. LOCAL AND MULTILOCAL FILTERS

In the work [3] it was noted that with the increase in the dimension of the problem as, for example, in case of big data, the information importance of the coefficients $S_f(\omega)$ of the logical function range $f(x)$ declines. Consequently, the efficiency of simple linear filters' application, corresponding to argument ω of the spectral coefficient, also diminishes. The solution could be found by applying multiple linear filters of the type $\varphi(x) = \prod_{i=1}^{k} \lambda^i(x)$, where a plain linear filter $\lambda^i(x) = \lambda_1^i x_1 \oplus \lambda_2^i x_2 \oplus \ldots \oplus \lambda_n^i x_n = \omega_1^i x_1 \oplus \omega_2^i x_2 \oplus \ldots \oplus \omega_n^i x_n$. Taking into consideration the spatial character of spectral coefficients, the multiple linear filter should be named a local linear filter. In the theory of communication, a narrow band filter can serve as its analogue. The local filter allows one to select (out of the area of function definition) monochrome or near to monochrome state the area in which the meaning of the function is either constant or near to it. Different θ of the function in this area will possess a prevailing absolute meaning, in relation to other regions, $|\theta|$, $\theta = \log_2(k_1 / k_0)$,where k_1, k_0 according to the number of single and zero values of the function.

In hardware implementation the local filter is represented by a scheme of several simple linear filters, the outputs of which are united by the logical element „AND". As it has been noted in the work [3], realization of the local filter may implement no more than n of two input adders by module 2. An example of filter for $n = 6$ variables are given here:

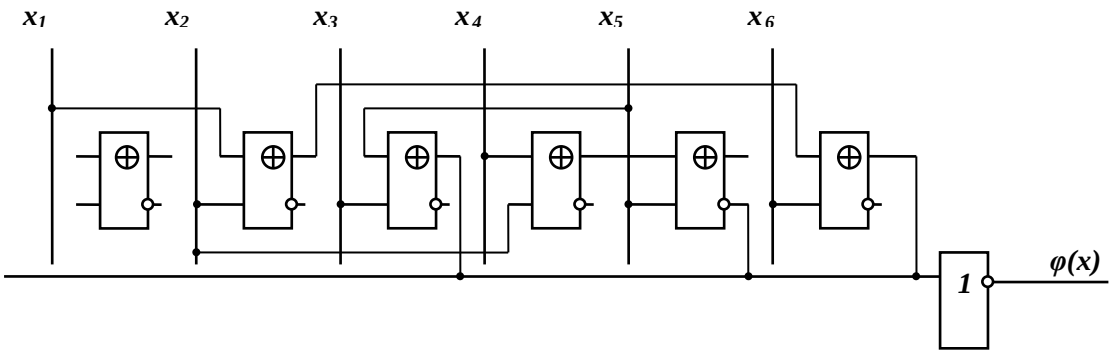


Fig. 3. A structural scheme of a local filter

$$\varphi(x) = (\overline{x_3 \oplus x_5})\overline{(x_1 \oplus x_2 \oplus x_6)}(x_2 \oplus x_4 \oplus x_5) = \overline{(x_3 \oplus x_5) \vee (x_1 \oplus x_2 \oplus x_6) \vee \overline{(x_2 \oplus x_4 \oplus x_5)}}$$

If we take software terms, then a local filter is realized by the conjunction of discharges (several of which can be pre-inverted) of the linearized argument $x^\sigma = \sigma \otimes x$. Matrix lines σ, at that, form arguments ω^i of the spectral coefficients that determine simple linear filters λ^i.

For example, the software implementation of the local filter $\varphi(x)$, which you can see in Figure 3, is carried out by matrix multiplication and subsequent multiplication of the discharge values.

$$x^{\sigma} = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{vmatrix} \otimes \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{vmatrix} = \begin{vmatrix} x_3 \oplus x_5 \\ x_1 \oplus x_2 \oplus x_6 \\ x_2 \oplus x_4 \oplus x_5 \end{vmatrix} = \begin{vmatrix} x_1^{\sigma} \\ x_2^{\sigma} \\ x_3^{\sigma} \end{vmatrix}, \qquad \varphi(x) = \overline{x_1^{\sigma}} \cdot \overline{x_2^{\sigma}} \cdot x_3^{\sigma}.$$

It is obvious that in case of big data, that is, where LF has a big size n, the application of separate local filters will not lead to a an efficient solution. Here it might be more appropriate to unite them into groups,-multilocal filters

$$\psi(x) = \bigcup_{i=1}^{k} \varphi_i(x) = \varphi_1(x) \vee \varphi_2(x) \vee \ldots \vee \varphi_k(x).$$

In the work [1] there is an example of the randomly given LF of $n = 14$ variables, and the example of its realization by the switching neuron network, in which the network switches are operated by the multilocal filters with the number $k = 7 \div 10$, is given.

5. NEURONS OF FIBONACCI

The logical elements "AND", "OR", used for creating local and multilocal filters, represent elementary artificial neurons, the structural vectors of which are equal to $A_{AND} = (k;\underset{1}{1},\underset{2}{1},\underset{3}{1},\ldots,\underset{k}{1})$ and $A_{OR} = (1;\underset{1}{1},\underset{2}{1},\underset{3}{1},\ldots,\underset{k}{1})$ accordingly. This might prove to be insufficient for the complex logical functions of these elements even if linearization is applied. It is appropriate to find elements with wider opportunities. They are to provide the neuron network's efficiency, and at the same time, have a simple construction both in software and hardware.

Neuron of Fibonacci (NF), analyzed in the work [4] is one of such elements. A cascade element connection "INHIBIT" (Figure 4a) serves as the basis of its construction. From the point of hardware implementation, the positive characteristic of NF is the opportunity to realize a lot of elements in the form of the programmed logical matrix that possesses invertors between programmed contacts, Figure 4b. Besides that, one may use dual pairs NF in order to manage the channel switches in SNS, and provide management of the switches of the type 1×3, (Figure 4c). Naturally, the linearization of the programmed matrix NF is possible. It may be executed in the block of input buffer cascade.

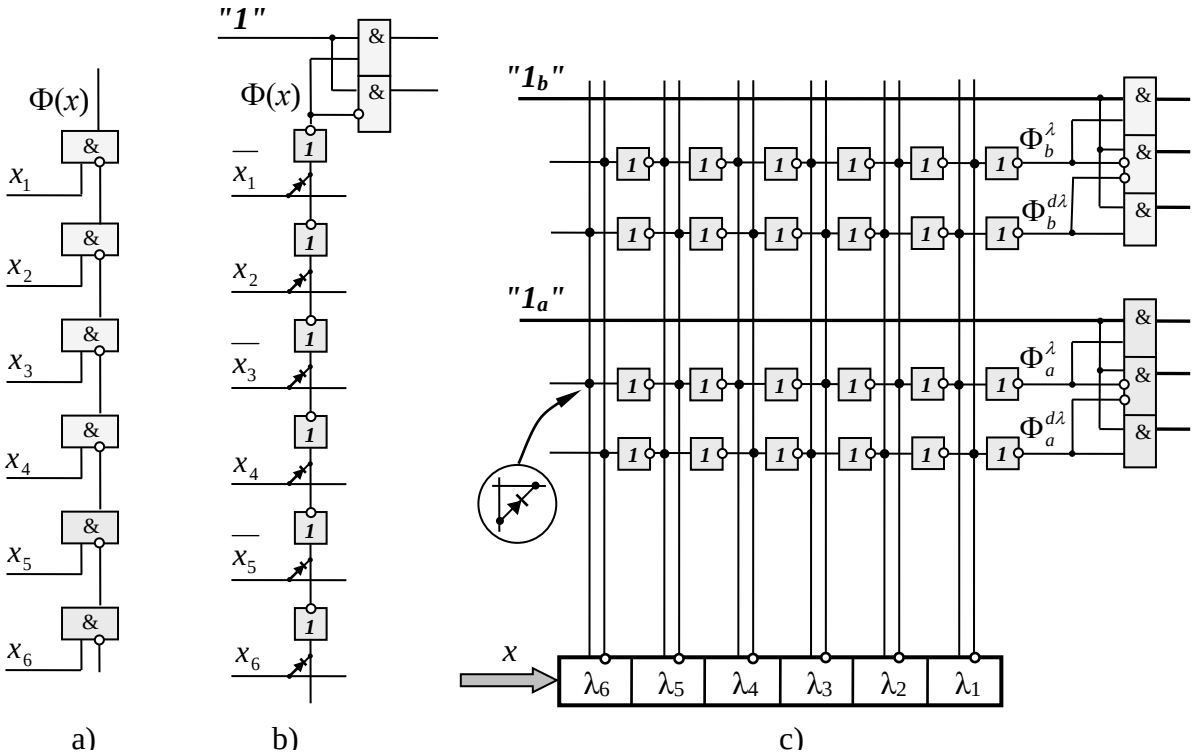


Fig. 4. a) a logical model of Φ-neuron in the form of serial connection of the elements INHIBIT;
b) the logical circuit of a ***1×2*** channel switch and a control *Φ*-neuron
$\Phi(x) = x_1(x_2 \vee x_3(x_4 \vee x_5 x_6))$, performed by series connections of diodes and inverters;
c) a ***1×3*** channel switch circuit controlled by a linearized programmable matrix of Φ-neuron
dual pairs $(\Phi_a^{\lambda}, \Phi_a^{d\lambda})$ and $(\Phi_b^{\lambda}, \Phi_b^{d\lambda})$:

$$\Phi_a^{\lambda}(x) = \overline{\lambda}_1(\overline{\lambda}_2 \vee \lambda_3(\lambda_4 \vee \lambda_5\lambda_6)), \quad \Phi_a^{d\lambda}(x) = \lambda_1(\lambda_2 \vee \overline{\lambda}_3(\overline{\lambda}_4 \vee \overline{\lambda}_5\overline{\lambda}_6)),$$

$$\Phi_b^{\lambda}(x) = \lambda_1(\overline{\lambda}_2 \vee \overline{\lambda}_3(\lambda_4 \vee \overline{\lambda}_5\overline{\lambda}_6)), \quad \Phi_b^{d\lambda}(x) = \overline{\lambda}_1(\lambda_2 \vee \lambda_3(\overline{\lambda}_4 \vee \lambda_5\lambda_6)),$$

where $\lambda_i = \lambda_i(x) = \lambda_{i1}x_1 \oplus \lambda_{i2}x_2 \oplus \ldots \oplus \lambda_{in}x_n$, $i = 1 \div 6$, $\overline{\lambda}_i$ - is its inversion.

A structural vector A_Φ of Fibbonacci neuron with n inputs represent by itself a row of Fibbonachi from $n+1$ member. So, for example, for $n = 6$,

$$A_\Phi = (p; a_1, a_2, \ldots, a_6) = (13;8,5,3,2,1,1).$$

Logical function $f_\nu^\sigma(x)$ of the linearized neuron of Fibonacci, being a linearized threshold function, possesses the definition of complete monotony, with precision up to linear transformation. That is why, in a number of cases NF are able to provide a more accurate solution of the task of given LF approximation in comparison with multilocal filters.

In general case, the software implementation of the linearized Fibonachi neuron includes matrix calculation $x^\sigma = \sigma \otimes x$ with the operations of consequent inverting of x^σ vector discharge. For example, for $n = 6$ $f_\nu^\sigma(x) = \overline{x_1^\sigma x_2^\sigma x_3^\sigma x_4^\sigma x_5^\sigma x_6^\sigma}$. It is obvious that such method of calculation NF proves to be quicker and more economical than the calculation of the input neuron's sums values.

6. NOISE FILTERS

The logical function, representing a class of recognizable images, has the character of a noisy binary signal. The neural network, the task of which is recognition, must have the elements approximating the representing function in different fragments within the area of its definition.

For example, in some biological prototypes, eye retina contains neural structures of "on", "off"-type. It is noted in [5], that they perform the function of noise filters while the process of image recognition is going on in the visual cortex. In the logical element base, noise filters are modelled with the help of a certain element's input signal mixing. It is realized by summing of the output signals of two or more elements over *mod2*.

Taking into account the demand for the simplicity of realization, the system of two elements, united by the binary adder at the output, could be offered for the role of noise filters.

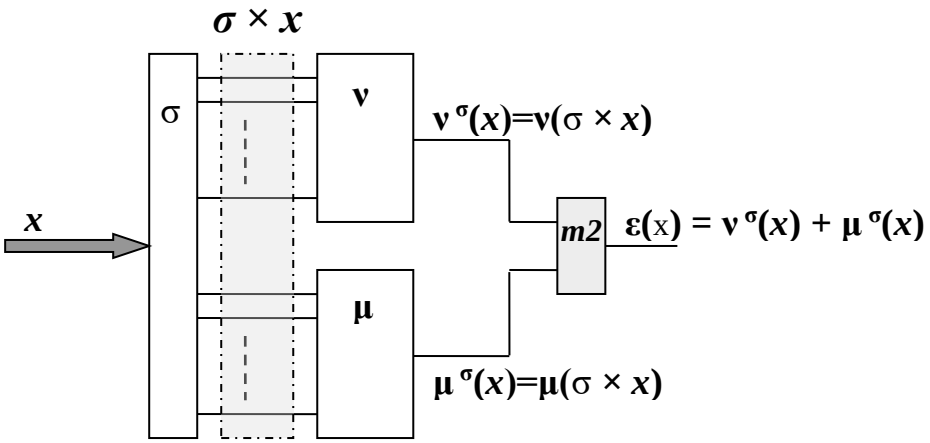


Fig. 5. Linearized neural noise filter ε

ν, μ – artificial neurons,

x – an input signal of noise filter, argument of filter function,

σ – a block of linear transformations by *mod 2*,

m2 – adder by *mod2*.

It should be noted that the application of three and more neurons in the noise filter is impractical. The success of the function approximation by means of noise filter is estimated by the trim absolute value on the corresponding fragment. Let us take θ_1 - as a different of the initial function $f(x)$ on the fragment $X_1 \subset X$ of its domain X, that is exuded by a filter $\nu(x)$ c with one neuron ν: $X_1 = \{x : \nu(x) = 1\}$. Having determined a second mixing neuron μ, by the method from [6], we shall get a noise filter $\varepsilon(x)$, that determines a somewhat different fragment $X_2 \neq X_1$, $X_2 = \{x : \varepsilon(x) = 1\}$, but with a different $\theta_2 : |\theta_2| > |\theta_1|$ increased by its absolute value. As a result, an increment of absolute value of the different $\delta_2^\theta = |\theta_2| - |\theta_1|$ is insured. By adding one more mixing neuron μ' to the noise filter, we shall get a modified fragment X_3 with a different θ_3, accordingly $|\theta_3| > |\theta_2|$ и $\delta_3^\theta = |\theta_3| - |\theta_2|$. But at that $\delta_3^\theta < \delta_2^\theta$, and at that, the difference in the majority of cases is significant. The use of every subsequent neuron in the noise filter will be accompanied by the decreasing absolute value increment of the different $|\theta|$, which is technically impractical.

Following the aim of providing the minimization of technical costs, local filters, multi local filters and linearized neurons of Fibonacci could be used as neural elements ν and μ in noise filters.

7. CONCLUSION

A set of logical elements, listed in the research, combined with the analytical capabilities of Walsh's spectral representations, provide the simplicity, both in the software and hardware realization of the neural structures. Their operation is based on matrix computational algorithms in the field of deductions for module2 and logical operations over separate discharges of the linearized arguments. This provides a substantial simplification of neural structures and a significant increase in the speed of their functioning. They also provide the basis for solving a wide range of tasks connected with artificial intelligence technologies.

REFERENCES

1. Sychev, A 2012, 'Composition of switching logical networks in neurostructures', *Automatic Control and Computer Sciences*, vol. 46, no. 6, pp. 248-260.
2. Karpovsky, M & Moskalev, E 1973, *Spectral methods of analysis and synthesis of discrete device*, Energy, Leningrad.
3. Sychev, A 2011, 'Logic Neuron Model Using Walsh Spectral Representations', *Automatic Control and Computer Sciences*, vol. 45, no. 5, pp. 241-258.
4. Sychev, A 2017b, 'The Effect of Presynaptic Inhibition in Switching Neurostructures', *Automatic Control and Computer Sciences*, vol. 51, no. 6, pp. 453-462.
5. Sychev, A 2019, 'Logical Modelling and Correlation Analysis of the Retina Neural Structures with "ON"("OFF")-Center', *Journal of International Scientific Publications: Materials, Methods & Technologies*, vol. 13, pp. 292 – 300.
6. Sychev, A 2018, 'The Use of Noise Filters in Switching Neural Structures', *Journal of International Scientific Publications: Materials, Methods & Technologies*, vol. 12, pp. 49 – 59.