

Multisensors Signal Processing Using Microcontroller and Neural Networks Identification

¹ Oleksiy ROSHCHUPKIN, ² Radislav SMID, ¹ Volodymyr KOCHAN,
^{1,3} Anatoly SACHENKO

¹ Research Institute of Intelligent Computer Systems,
Ternopil National Economic University,
3 Peremoga Square 46020, Ternopil, Ukraine

² Faculty of Electrical Engineering, Czech Technical University,
Technicka 2, 166 27 Prague 6, Czech Republic

³ Department of Computer Science and Econometrics,
Silesian University of Technology,
Roosevelta St. 26-28, Zabrze, 41-800, Poland

¹ Tel.: +380352475050 ext.12322, fax: +380352475053

² Tel.: +420224352131, fax: +42023339929

E-mail: o.roshchupkin@chnu.edu.ua, smid@fel.cvut.cz,
vk@tneu.edu.ua, a_sachenko@ieee.org

Received: 15 May 2013 /Accepted: 16 August 2013 /Published: 30 August 2013

Abstract: An approach of multisensor signals processing at microprocessor or 8 bit microcontrollers is showed in this paper. This approach is based on multisensor individual conversion function or individual characteristic curve identification method with using of neural networks. It allow to reduce an amount of calibration points and increase the precision of identification with comparatively to well known methods. The realization of this approach was made on developed system for ultraviolet level measurement based on 8-bit microcontrollers. Copyright © 2013 IFSA.

Keywords: Multisensor, Microprocessor, Microcontroller, Individual conversion function, Individual characteristic curve, Neural networks, Ultraviolet.

1. Introduction

Multiparameter sensors or multisensors (MS) have a widespread application in such industry branches as chemistry, security systems, environment monitoring etc [1]. The MS however have still some imperfections like complexity of output signal processing [2], a significant deviation of the characteristic curve from nominal one [3].

To improve the accuracy, it is recommended to use the individual conversion function or individual characteristic curve (ICC) of MS which can be determined by the MS calibration in real exploitation conditions [4]. But its usage requires some extra work and leads to additional expenses. Just one-parameter sensor (for example, resistance temperature detector) requires the 3 ... 4 calibration points at small deviations of the ICC from nominal one, and this number of the calibration points

increases up to 5 ... 7 at significant deviations of the ICC, for example, for semiconductor and film sensors. In such case the two-parameter film sensor requires $N_2 \approx 25 \dots 49$ calibration points, and the three-parameter sensor requires about 125 ... 343 calibration points correspondly.

To avoid this situation MS calibration results are processed by methods of artificial intelligence [5, 6] for separating the measurable values as well as for reducing the number of calibration points. To reduce the number of calibration points with a purpose of the MS ICC recognition (Fig. 1), it is recommended to use neural networks methods.

The analysis has shown neural networks methods [7] give better results in comparison with existing ones because of its ability of generalization and self-learning. Although it's so not easy to implement the neural networks methods on the computer architectures with limited resources such as microcontrollers (MC) we should note the Neural Networks (NN) require the large computational power at the training stage only. Because according to [8] the computational power reduces in approximately 10^4 times at the NN operating stage, so it is possible to use NN with implementation on MC. In this case the NN training stage can be done on high level of the system with necessary computational power, and the MC of low level is used by the trained NN already.

Therefore a goal of this paper is the microprocessor MS output signals processing with a limited number of the calibration points and the NN training or an adjusting the NN weight coefficients by MC itself.

2. The Neural Processing Method for Identification of Multisensors Individual Characteristic Curve

A proposed method is aimed to reduce a number of MS calibration points. For this purpose in the approximation procedure is added the additional information regarding a character (details) of the MS ICC characteristic curve which are contained in the results of calibration for the MS group of the same type for a large amount of points, for example 49. These above group results are united with the calibration results of MS in the reduced number of points and then, based on these data, it is performed the predicting of calibration results in those points, where the calibration of a given MS wasn't performed. A separate NN is used per each of these points and it is trained to predict the calibration result for a given point based on the results of calibration of the MS group of the same type in a large amount of points [9-11]. For training this separate NN it is used only those calibration points (of the MS group of the same type) which belong to the strictly specified set of points of the calibration regarding a point for which the result of calibration is predicted. In particular it's expedient to use the set of points of the calibration belonging to all lines which could be

drawn across the given calibration point and the adjacent ones. An example is shown on Fig. 1 where the four lines are drawn across a given point 33.

Moreover it is expedient to use the point's sets that belong to the groups of calibration points adjacent to a predictable point, for example a square around the point 33 (see Fig. 1).

To improve the prediction accuracy for points in which the calibration has not been executed all prediction results of these points are merged by NNs (which have been trained on the calibration results of MS group of the same type for the same given set of calibration points).

As a result we have a significant reduction in the required number of calibration points. This value of decreasing is about 82 % and it is shown on Fig. 1, the actual calibration was carried out in 9 points ICC instead of 49 points.

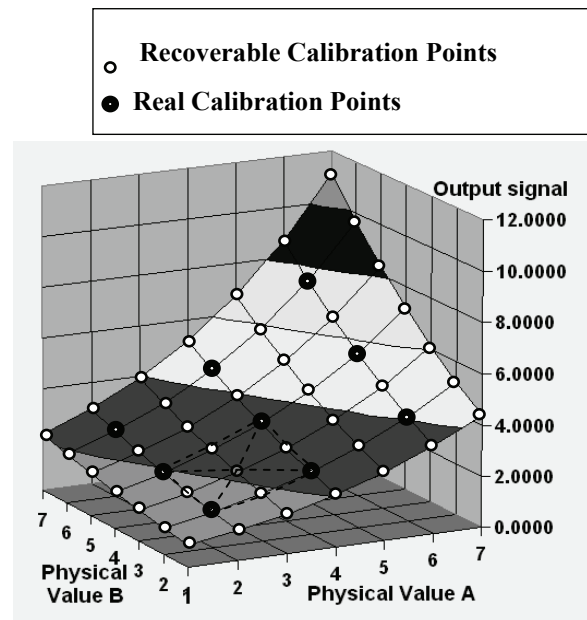


Fig. 1. ICC of MC and placement of real calibration points and recoverable calibration points.

In studies of the method, the MS individual characteristic curve is described as the product of polynomials [12]:

$$Y_{NOM} = (A \times (X_1 + B)^k + C \times (X_1 + B)) \times (D \times (X_2 + E)^l + F \times (X_2 + E)) \times G, \quad (1)$$

where Y_{NOM} is the nominal output signal of MS; $A \dots G$, k , l is the coefficients and exponents respectively; X_1 , X_2 is the measurable physical quantities A and B , respectively.

Obviously, the additive and multiplicative errors of the MS can be corrected without using the neural networks. Therefore, the main objective of this work is studying the influence of the nonlinear component

for MS error on the result correction by using the proposed method.

Therefore, to study the residual (non-corrected) error let's describe this nonlinear component by polynomials of different degree. Since the MS error for the various physical quantities can differ quantitatively as well as qualitatively, it's logically to study the errors various combinations using the equation:

$$Y = Y_{NOM} \pm n\Delta(\pm K_1(i-4)^a \pm K_2(j-4)^b), \quad (2)$$

where Y is the output signal of MS; Y_{NOM} is the nominal output signal of MS; n is the number of variants for research, it is taken as $n = 100$ (i.e. 50 experiments for each polarity); Δ is the quantization step of MS error, it is taken 0.1 % (i.e. maximum error MS for each physical quantity gives 5 %); K_1 ,

K_2 is the coefficients that characterize the nonlinearity error function MS, and equal to 1 %. The output signal of MS is very sensitive to random errors. To study the resistance to such errors let's introduce the following random error (which is distorting the MS output signal)

$$Y_N = Y + Rnd(K_3), \quad (3)$$

where K_3 is the coefficient, which determines the random error amplitude.

The coefficient K_3 is chosen so that the random error does not exceed 0.5 % of the MS nominal output signal. The results of imitation modeling [10] for error prediction of ICC where the real calibration has not been executed are shown in Table 1.

Table 1. Values for point 34 with level of random error 0.5 %.

FV B \ FV A	$+x^2$	$-x^2$	$+x^3$	$-x^3$	$+x^4$	$-x^4$
$+x^2$	0.016 % / 0.010 %	0.015 % / 0.01 %	0.016 % / 0.007 %	0.018 %/ 0.013 %	0.012 %/ 0.002 %	0.021 %/ 0.013 %
$-x^2$	0.015 % / 0.005 %	0.016 % / 0.009 %	0.016 % / 0.007 %	0.028 %/ 0.005 %	0.017 %/ 0.007 %	0.016 %/ 0.01 %
$+x^3$	0.016 % / 0.007 %	0.016 % / 0.007 %	0.12 % / 0.009 %	0.014 %/ 0.01 %		
$-x^3$	0.018 %/ 0.013 %	0.028 %/ 0.005 %	0.027 % / 0.005 %	0.019 %/ 0.008 %		
$+x^4$	0.012 %/ 0.002 %	0.017 %/ 0.007 %				
$-x^4$	0.021 %/ 0.013 %	0.016 %/ 0.01 %				

3. Data Processing at Microcontrollers Using Neural Networks

The NN implementation on MC PIC Microchip was described in [13]. It uses the hyperbolic tangent as the activation function and pseudo-floating point format for proceedings 16-bit numbers. Taking an approach [13] as basic one, the authors developed a modified algorithm for MS data processing. It's implemented on MC AVR Atmel with reasonable performance and extended set of instructions.

A proposed algorithm can be divided into two major phases: first phase – MS data processing at the top level (Software), and the second phase – MS data processing on the low level (Hardware). At the first phase the description of NN architecture in MATLAB environment was made, then a transformation of NN architecture into C or Assembler code was reached. In the second phase all calculations were carried out in MC. Here each neuron is calculated sequentially, one after another, and then calculation results are stored in the memory and passed in the processing unit of the activation function.

The algorithm has two main modes: teaching mode and operating mode. The operating mode sets as default, which means the NN has already been taught and weights coefficients have been adapted for to input signal correction. On the output of NN we have a recalibrated sensor signal.

The teaching mode includes the procedure of Feed Forward algorithm and procedure of weights coefficients adaptation (Back Propagation algorithm) [14] with using teaching pairs which are stored in MC data memory.

As the activation function is used the sigmoid activation function (with optimization for MC [15])

$$OUT = \frac{1}{1 + e^{-NET}}, \quad (4)$$

where NET is the neuron summed signal that is fed to the unit of the activation function.

The AVR Atmel as well as PIC MC has no tools for calculation of the exponent, and the MC ATMEGA8 uses the one byte fixed numbers only. So it is necessary to create the subroutines for floating point numbers processing and apply the Taylor series for calculation of the exponent with using tools of the

AVR Assembler language. A number of 3 bytes (Fig. 2) is assigned for floating point numbers.

The normalization of the mantissa is done by setting a comma previous to the first information units. Six bits are assigned for the order of the number X so this number can be located in the range of 2^{-64} to 2^{+64} .

To run the floating point numbers the functions of addition, subtraction, multiplication and division are created. For this purpose all operations are performed on variables Float1 and Float2 and the calculation result is stored in variable Float2.

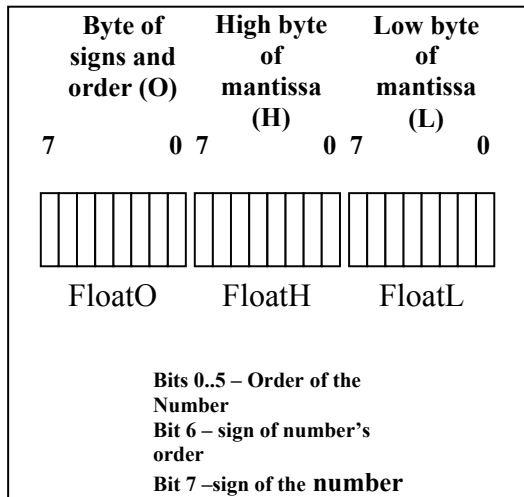


Fig. 2. The floating number format of variable Float.

The operating mode (the fine-tuning of the weight coefficients) is the similar to training mode but as input data it is chosen the weight coefficients that were calculated in training mode.

The Fig. 3 is illustrated the location vectors of the output signals of layers L, the matrixes of weight coefficients W, the adaptation vectors of weights D and training pairs.

As the data input in the training mode are used the training pairs X and Y which are orderly written in the MC data memory (starting from the address 138) in the following format:

$$X_1^1, \dots, X_m^1, Y_1^1, \dots, Y_k^1, X_1^2, \dots, X_m^2, \dots, Y_1^n, \dots, X_m^n, Y_1^n, \dots, Y_k^n, \quad (5)$$

where m – inputs quantity of NN, k – outputs quantity of NN, n – quantity of training pairs.

The data input of NN in operating mode are the X vectors which are the output values of the converted sensors signals, and weight coefficients that were adjusted in training mode. The output data in training mode are arrays of weight coefficients, and in operating mode – the calibrated sensor signals.

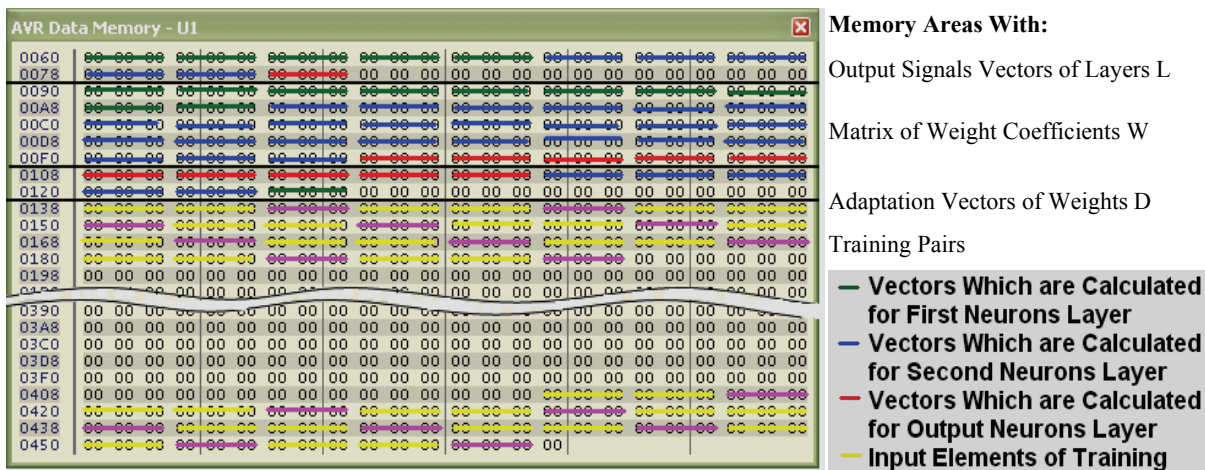


Fig. 3. AVR microcontroller data memory.

4. Experimental Research

In PROTEUS visual simulation environment the NN and ADC were simulated. As the sensors, the potentiometers were connected to the MC that provided the output voltage control in the range of 0 to 5 V.

Training of the three layer perceptron (with two inputs and five neurons per each hidden layer) is run by MC's program.

Previously the 15 test training pairs are loaded in the memory. During the NN training the gradual adaptation of the weight coefficients is run up to achieve a given accuracy of the output signal or unless the training approaches up to the final epoch. Taking into account the large numbers of epochs that

pass up to achieve the required accuracy and changing state of weight matrixes so the important part of an observation are the boundary cycles of the training algorithm as well as the changes of the adaptation of the weights coefficients.

Matrixes of weight coefficients are stored in the random access memory from the address 90 till 107 (Fig. 4), the addresses are shown in hexadecimal view.

Address	0060	006C	0078	0084	0090	009C	00A8	00B4	00C0	00CC	00D8	00E4	00F0	00FC
03 CE 9D 00	85 88 01 F7	16 00 87 87	03 CE 9D 01	C8 66 02 8F	14 00 AE C0	01 DF 44 00	81 8A 02 EE	14 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
42 C0 00 42	C0 00 42 80	00 41 80 00	41 80 00 12	E1 49 42 80	00 42 C0 00	C2 C0 00 42	80 00 41 80	00 41 80 00	2F B8 C7 42	80 00 42 C0	00 C2 C0 02	42 80 00 C1	80 D5 41 80	00 8C 8C 62
42 80 00 42	C0 00 42 C0	00 42 80 00	41 80 00 41	80 00 00 00	00 C2 80 00	42 C0 00 C2	C0 00 42 80	00 C1 80 00	41 80 00 00	00 00 C2 80	00 11 01 A9	AE 95 81 11	F3 C6 00 00	00 00 00 00

Fig. 4. Changes (highlighted) of weighting coefficients values matrixes in data memory within VSM Proteus.

A proposed approach has implemented in the developed system for measuring the level of ultraviolet radiation (Fig. 5) with the following components: multisensor which generates photocurrent for both a first ($I_{photo\ 1}$) and second ($I_{photo\ 2}$) channels and it measures temperature of these channels ($t^{\circ}1$ and $t^{\circ}2$); temperature and

photocurrent-to-voltage conversion unit (VCU) (U_{photo1} , U_{photo2}), and ($U\ t^{\circ}1$, U (U_{photo1} , U_{photo2}), and ($U\ t^{\circ}1$, $U\ t^{\circ}2$) respectively; analog to digital converter (16 bit sigma delta ADC) (ADC) and other components. The MC controls all connected units, and MS signals that come to ADC through Values Converter Unit. Then the digitized data are transmitted to the calculation unit of NN which calculates the values for both channels and stores weight coefficients in Database and then data are transferred to the unit updating of the calibration coefficients (UCC). Finally the Display indicates the measured and calculated values.

5. Conclusions

A multisensors signal processing approach based on the novel method of the multisensor individual characteristic identification using neural networks and microcontroller was proposed. It allows reducing a number of calibration points and increasing the identification accuracy of the individual characteristic. This approach has implemented in the developed system for measurement the level of the ultraviolet radiation using 8-bit microcontrollers.

Experimental research confirmed the prediction accuracy of multisensor testing results is improved in about 10 times in a comparison with existing methods.

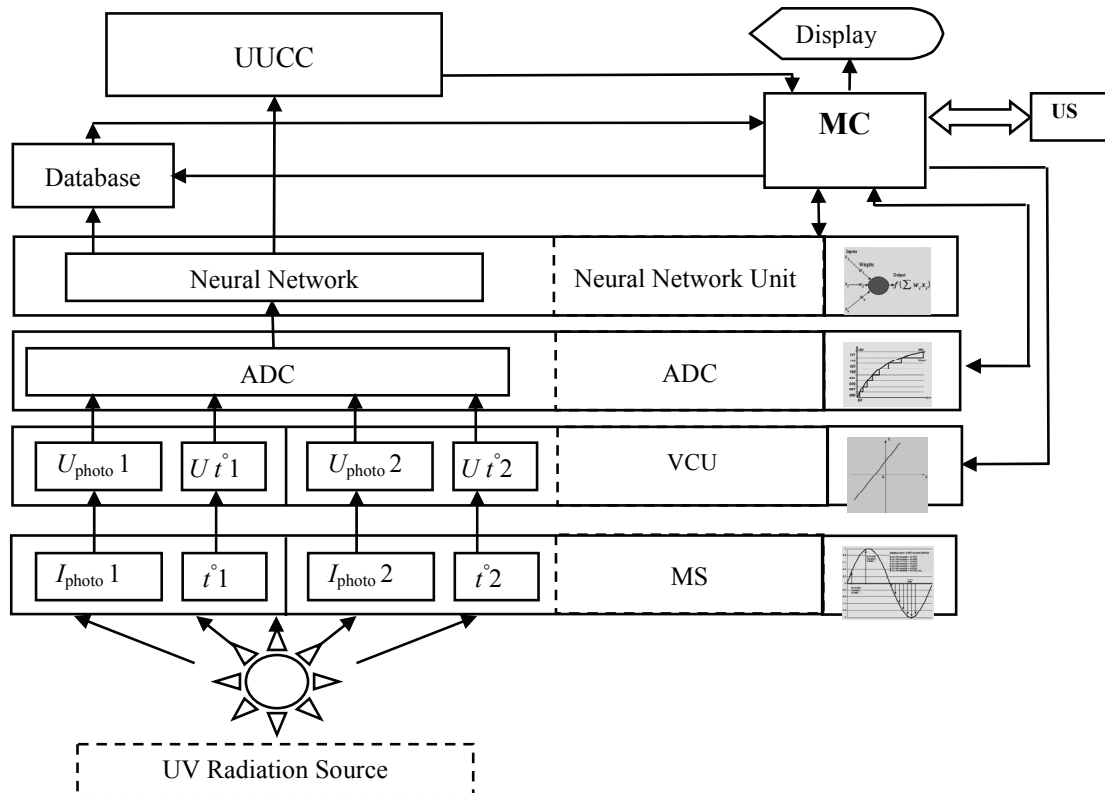


Fig. 5. Functional Diagram of System for Measurement the Level of Ultraviolet Radiation based on Multisensor.

Acknowledgments

This work has carried out within a National project entitled Environmental Monitoring HW & SW System for Measurement of Ultraviolet Radiation Level supporting by a firm TENZOR, Ukraine.

References

- [1]. G. L. Dempsey, N. L. Alt, B. A. Olson, J. S. Alig, Control sensor linearization using a microcontroller-based neural network, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 4, October 12-15, 1997, pp. 3078-3083.
- [2]. R. G. Jackson, Novel sensors and sensing, *Taylor & Francis*, London, UK, 2004, 512 p.
- [3]. M. A. Khan et al., High-responsivity photoconductive ultraviolet sensors based on insulating single-crystal GAN epilayers, *Applied Physics Letters*, Vol. 60, 1992, pp. 2917.
- [4]. A. S. Bangalore, G. W. Small, R. J. Combs, R. B. Knapp, R. T. Kroutil, Automated detection of trichloroethylene by Fourier transform infrared remote sensing measurements, *Analytical Chemistry*, Vol. 69, No. 2, pp. 118-129.
- [5]. D. Michie, D. J. Spiegelhalter, C. C. Taylor, Machine learning, neural and statistical classification, *Technometrics*, Ellis Horwood, New York, Vol. 37, Issue 4, 1994, pp. 459.
- [6]. V. Dorosh, O. Roshchupkin, I. Turchenko, V. Kochan, A. Sachenko, Identification method of multisensor conversion characteristics based on neural networks, in *Proceedings of the International Conference TCSET'2010 X-th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, Lviv-Slavsko, Ukraine, 2010, CD, A. 314. (in Ukrainian).
- [7]. I. Turchenko, V. Kochan, A. Sachenko, Neural-based recognition of multi-parameter sensor signal described by mathematical model, *International Journal of Computing*, Vol. 3, Issue 2, 2004, pp. 140-147. (in Ukrainian).
- [8]. I. Turchenko, O. Osolinsky, V. Kochan, A. Sachenko, R. Tkachenko, V. Svyatnyy, M. Komar, Approach to neural-based identification of multisensor conversion characteristic, in *Proceedings of the 5th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS '09)*, Rende, Italy, 2009, pp. 27-31.
- [9]. O. Yu. Roshchupkin, V. V. Kochan, A. O. Sachenko, Method of multisensor conversion function identification, *Application for a Patent of Ukraine for Inventions #a201113840*, declared 24.11.2011. (in Ukrainian).
- [10]. Oleksiy Roshchupkin, Radislav Smid, Volodymyr Kochan, Anatoly Sachenko, Reducing the calibration points of multisensors, in *Proceedings of the 9th IEEE International Multi-Conference on Systems, Signals and Devices (SSD '2012)*, Chemnitz, Germany, March 20-23, 2012, pp. 1-6.
- [11]. O. Roshchupkin, A. Sachenko, V. Kochan, Neural processing of multisensor signals at the 8-bit microcontroller, in *Proceedings of the 6th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS '2011)*, Prague (Czech Republic), 2011, pp. 383-387.
- [12]. I. Turchenko, O. Osolinsky, V. Kochan, A. Sachenko, R. Tkachenko, V. Svyatnyy and M. Komar, Approach to neural-based identification of multisensor conversion characteristic, *5th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS '09)*, Rende, Italy, 2009, pp. 27-31.
- [13]. J. Nicholas Cotton and M. Bogdan Wilamowski, Compensation of sensors nonlinearity with neural networks, in *Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 1210-1217.
- [14]. Yu Hen Hu, Jenq-Neng Nwang, Handbook of neural network signal processing, *CRC Press*, USA, 2002, 384 p.
- [15]. D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, *Parallel Distributed Processing*, Vol. I Foundations, *MIT Press*, Cambridge, MA, 1986, pp. 315-328.