Features of Intelligent Distributed Sensor Network Higher Level Development

A.Sachenko[1], V.Kochan[1], V.Turchenko[1], T.Laopoulos[2], V.Golovko[3], L.Grandinetti[4]

[1]Institute of Computer Information Technologies, Ternopil Academy of National Economy
3 Peremoga Square, 46004, Ternopil, UKRAINE, Phone: +380 (352) 33-0830,
Fax: +380 (352) 33-0024, E-Mail: sachenko@cit.tane.ternopil.ua

[2]Aristotle University of Thessaloniki, Electronics Lab, Physics Dept.,
54006, Thessaloniki, GREECE, Fax: +3031.998018, Email: laopoulos@physics.auth.gr

[3]Laboratory of Artificial Neural Networks, Departments of Computers,
Brest Polytechnic Institute, Moskowskaja 267, 224017, Brest, Republic of BELARUS,
Fax: +375 162 422127, E-mail: cm@brpi.belpak.brest.by

[4]Department of Electronics, Informatics and Systems, Parallel Computing Laboratory,
87036 Rende - Cosenza, ITALY, Fax: +39 984 494713, E-mail: lugran@unical.it

Abstract

The functions and software structure of the higher level of Intelligent Distributed Sensor Network (IDSN) are considered. The main purpose of software functions is reaching of a high accuracy of the data acquisition and processing using neural networks. The proposed client-server software architecture provides effective using of computing resources of IDSN central computer.

1. Introduction

The analysis of industrial data acquisition systems of the best world manufacture firms [1, 2, 3] has shown, that they are not quite suitable base for construction of distributed measurement and control systems. They do not provide (i) high measurement accuracy (including sensor error) and (ii) construction of network with high universality and adaptability. The increasing of measurement accuracy of physical quantities using industrial sensors (compensating of sensor error during exploitation) is possible by using an artificial intelligence methods [4, 5, 6].

The main principles of IDSN development are considered in [7], where the IDSN possibility of adapting to real external operating conditions using self-adapting and self-training is proposed. Considering this problem further, authors have proposed the intelligent node (IN) structure as a main element of such network in [8]. The IN can be adapted to any network hardware configuration (interfaces, protocols, etc) by using remote reprogramming mode. Besides it is proposed to utilize this IN for sensor errors correction by using calibration or testing on exploitation place and by drift prediction during intertesting interval.

Neural networks (NN) using [9] provide the high prediction quality. It is expedient to execute neural networks training at the higher level (central computer) of the IDSN (the most powerful computing component). The prediction is executed at the middle level (IN), that is maximally comes nearer to sensor [8]. Thus it is important to provide a high-quality neural network training for effective solving of sensor accuracy increasing task. Therefore the features of the IDSN higher level development are considered below.

2. Development Features of the IDSN Higher Level

It is necessary to take into account the features connected with multilevel IDSN structure and distribution of correction procedures on different system levels at development of IDSN higher level.

The higher IDSN level plays an important role in improving of measurement results and technology by using of artificial intelligence methods in accordance with concept of IDSN development [7]. IDSN self-training and self-adaptation to constantly varying measurement

0-7803-5890-2/00/$10.00 © 2000 IEEE

conditions is the main purpose of intelligent functions [10] fulfillment at the higher level for providing of required measurement accuracy. At the same time the measurement result is presented as value of the physical quantity with its error at the middle level. Thus only processed measurement results are circulated in the network that reduces an information flows. It also provides using of measurement results with real error value for all IDSN users.

Proposed structure is expedient for distributed sensor systems and networks. However, rather powerful computing resources should be provided on the higher level for fulfillment of intelligent functions, in particular procedures of self-adaptation and self-training. In this case, the simultaneous training of some set of neural networks is necessary which are individual for each channel of data acquisition. Therefore the high productivity personal computer or high-performance workstation with parallel architecture should be used as central computer.

The neural networks using allows providing a high quality of sensors drift prediction [9]. It is known that high accuracy of prediction is reached by the greater size of training sample [11]. However, the technological restrictions are imposed on calibration or testing number in practice. Therefore the volume of training sample is not enough for quality neural network training. Besides the task of sensor accuracy increasing becomes actual just after beginning of sensor exploitation. Thus there is a contradiction at neural networks application. On the one hand, the main purpose of neural network using (by prediction of sensor error) is the increasing of the intertesting interval of data acquisition channels. On the other hand, the calibration (testing) results serve us input data for neural network training. The main point of this contradiction is that: the quality prediction requires a quality neural network training which requires a lot of training input data, that increase the calibration (testing) number.

It is proposed to use the historical data (data about drift of the same type sensors in the similar operating conditions obtained by calibrations or testing) [9] and additional number of neural network [12] for solving of this contradiction (for artificial increasing of data volume for predicting neural network training). The joint usage of these methods is the most expedient. Obviously, that the best prediction quality is provided by predicting neural network training on real data about sensor drift (obtained by calibration or testing). Therefore, the historical data should be replaced by real data during sensor exploitation.

The first calibration of the new sensor allows correcting an initial spread of it conversion characteristic. The second calibration allows receiving the first real value of drift. The purpose of historical data using is the drift prediction at the moments of the future calibrations on the basis of the first real value. For this goal it is expedient to use the separate integrating historical data neural network (IHDNN). The IHDNN should predict a drift value of the new sensor for next calibration on the basis of historical data of current calibration. The number of available historical data of sensor drift defines a structure of IHDNN input layer. At the same time it is necessary to specially form of IHDNN training sample [13]. It is possible to apply a single-layer perceptron model with linear activation function as IHDNN.

The number of sensor drift values (obtained by IHDNN prediction) corresponds to number of historical data that is not enough for quality predicting neural network (PNN) training. It is expedient to use an approximating neural network (ANN) for additional increasing of data volume for predicting neural network training. It is trained on results of IHDNN prediction that corresponds to step of historical data receiving. The sufficient volume of data (with the reduced step) for PNN training is received by approximation. For this purpose the multi-level perceptron as ANN and recurrent neural network as PNN can be used.

There can be a situation, when historical data (or real data) is not available in the beginning of IDSN operation. The hypothetical data using is expedient in this case, which present generalized mathematical models of drift of certain sensor type. These data are formed according to the references, scientific and technology research results, producer information etc. However they characterize drift of the certain sensor at certain operating conditions worse on comparison with historical data.

These main features cause software structure of the higher level central computer considered below.

3. Software Structure of IDSN Higher Level

The higher IDSN level, presented by central computer (CC), operates with data at level of knowledge about measurement object. The general structure of central computer software is presented on Fig. 1. It consists of two levels: the base software level (NN Manager, Expert system and Supervisor) and user software level (IDSN Manager, IDSN Database). The software components (modules) interact between themselves using client-server technology. The base software levels modules execute the function of modules-servers and the user software level modules are the clients as a rule.

The Supervisor is the core of the CC software. It operates invisibly for the user and executes the following system functions: (i) monitoring of all IDSN processes; (ii) providing of interaction between CC software components in real time scale; (iii) remote reprogramming of IN of the middle level [8] etc. The main purpose of Expert system (ES) using is the providing of required accuracy of sensor data acquisition by intelligent functions executing [7]. It forms the following requests: (i) for training of necessary neural networks set to NN Manager; (ii) for transfer of trained neural networks as mathematical models of sensor drift for certain data acquisition channel to Supervisor; etc. The NN Manager executes scheduling and dispatching of some NN set training by the ES request and returns to the ES the results of NN operating. The IDSN Manager is the user program that provides IDSN initialization and configuration of any IN or data acquisition channel as well as an interaction with ES. A user can access to IDSN Database using IDSN Manager. IDSN Database contains an information about all elements of IN and data acquisition channels (sensors, switchboards, analog-to-digital converters etc.).

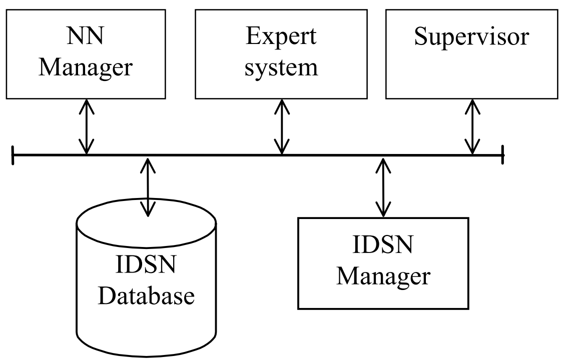

Fig. 1. The general structure of CC software

The general algorithm of module-server operation is presented on Fig. 2. Each module-server has own message queue. The interaction between modules is executed by sending of the certain input query to message queue of the appropriate module-server. After finishing of input query processing the appropriate module-server sends the answer query to the server or client that initiated it operating.

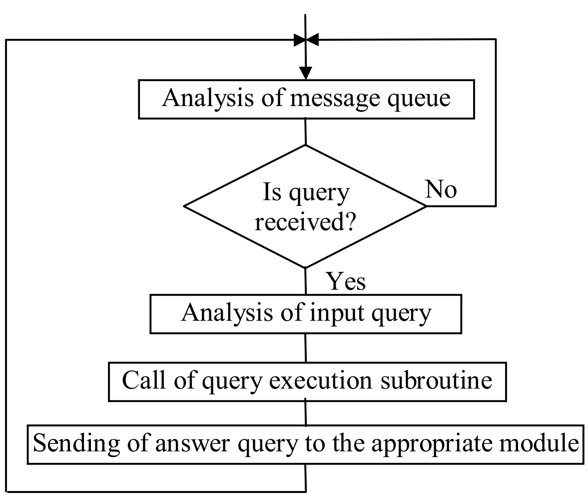

Fig. 2. The general algorithm of module-server operation

The necessity of client-server architecture using is stipulated by IDSN computer power distribution (see chapter 2). It is expediently to execute the NN training on higher level and to provide NN using (prediction, approximation etc.) on the middle IDSN level. The training process requires various real time for various NN architectures, in particularly IHDNN, ANN and PNN. For example, the average training time for IHDNN, ANN, PNN is 6 min, 2 min and 10 min accordingly (on the computer Pentium®II-350). Besides the training time greatly depends on parameters of training sample and selected NN architecture. Therefore CC software should be operated in real time scale for effective utilization of its computing resources that is provided by client-server architecture using. The real time monitoring is provided by one of Supervisor subroutines. Besides such modular principle of CC software construction provides easiness of expandability, development and replacement of new modules at the base software level as well as at user software level.

The Expert system provides method of data acquisition accuracy increasing using NN (see chapter 2) on the base software level. Let us consider it below. The Supervisor or user (using IDSN Manager) can send query for start of accuracy increasing procedure. The general algorithm of ES operation is presented on Fig. 3, where those subprograms are shown which provide the accuracy increasing method only.

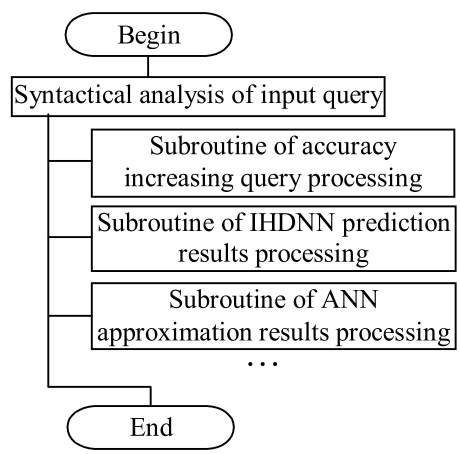

Fig. 3. The general algorithm of ES operation

The subroutines of input query processing are executed after it syntactical analysis. The subroutine of accuracy increasing query processing (Fig. 4) first defines the availability of historical data for given data acquisition channel and then calls the subroutines of historical or hypothetical data operating accordingly.

The subroutine of historical data operating (Fig. 5) finds a number of historical values of sensor drift N. The subroutine forms i training samples for each set of historical values. Subroutine sends the query to NN

Manager message queue for training of i-variant of IHDNN after forming of i training sample. Thus, the given subroutine forms training samples for all N values of sensor drift and sends to NN Manager N queries for IHDNN training.

The NN Manager (after input query receiving) trains some IHDNN architectures and sends the best prediction result to ES as answer query. The ES executes subroutine of IHDNN prediction results processing (see Fig. 3) after analyzing of input query. The subroutine of IHDNN prediction results processing (Fig. 6) saves the prediction results into IDSN Database and checks if all N prediction results are saved into Database. The given subroutine forms the query to the NN Manager for ANN training if all N prediction results are saved into IDSN Database. The ANN training sample consists of N prediction results of sensor drift in such case.

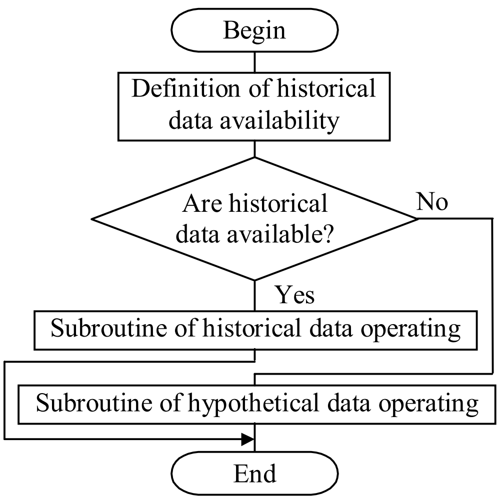

Fig. 4. The general algorithm of accuracy increasing query processing subroutine

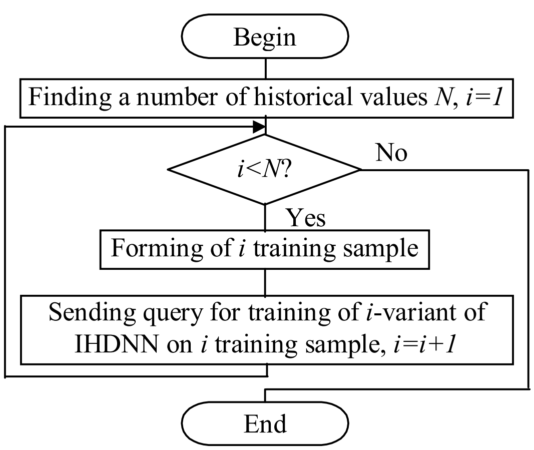

Fig. 5. The general algorithm of subroutine of historical data operating

The NN Manager (after input query receiving) trains some ANN architectures and sends the best approximation result to ES as answer query. The ES executes subroutine of ANN approximation results processing (see Fig. 3) after analyzing of input query. The subroutine of ANN approximation results processing saves this result into IDSN Database and forms the answer query to Supervisor about readiness of mathematical model of sensor data accuracy increasing on given data acquisition channel. This routine sends query to Supervisor about writing of mathematical model to appropriate IN of the middle level. The PNN using is also carried out by call of NN Manager.

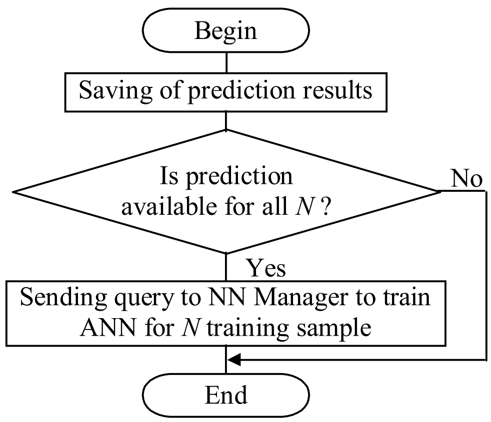

Fig. 6. The general algorithm of subroutine of IHDNN prediction results processing

Let us consider a situation when there are no historical data for the given IDSN channel (see Fig. 4). The subroutine of hypothetical data operating (Fig. 7) first defines the hypothetical function according to sensor type installed into channel. If results of definition are successful the training sample using selected function on required interval (depending on sensor parameters) is formed. The subroutine sends query to NN Manager for training of PNN on formed training sample after this. If the subroutine can not define hypothetical data it sends to Supervisor the appropriate answer query about impossibility of mathematical model choice for sensor drift correction of the given IDSN data acquisition channel. The interaction between ES and NN Manager is carried out similarly as for ANN in a case of PNN.

The IDSN Manager and IDSN Database operate at the user software level. The IDSN Database consists of many files. The main of them are "Sensors", "Switchboards", "Measuring circuits", "Lower level devices", "INs", IDSN configuration", "Calibration/testing" etc. For example attributes of "Lower level devices" file (Fig. 8) are considered. The information indexing and searching is executed in database by using attributes with asterisk(*).

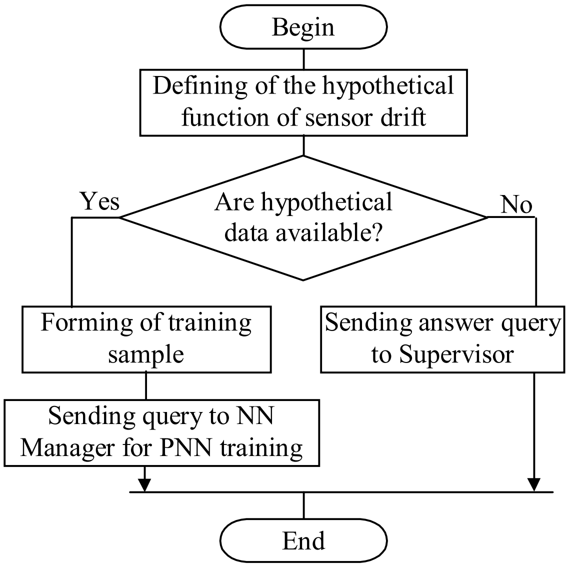

Fig. 7. The general algorithm of subroutine of hypothetical data operating

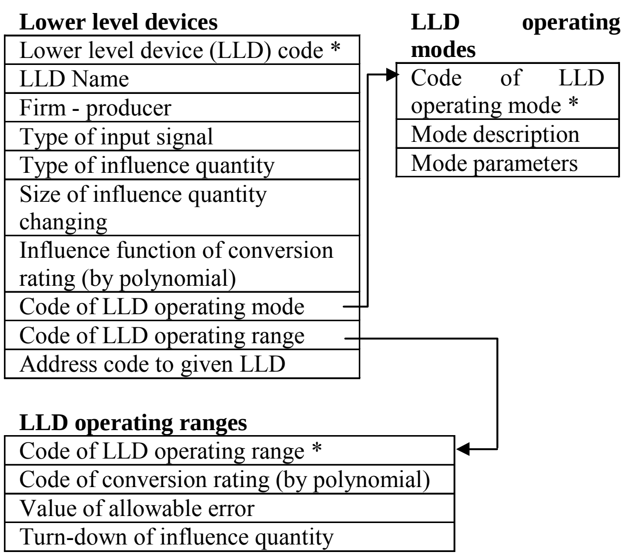

Fig. 8. An example of one file from IDSN Database

The IDSN Manager has mode of input parameters of all IDSN devices for providing of IDSN initialization and configuration of any data acquisition channel or intelligent node. For example, the dialogue window of sensor parameters input subroutine is presented on Fig. 9. The user can enter:

- The title of the sensor and firm - producer;
- The conversion rating of the sensor as polynomial and as NN (NN type, architecture, weight coefficients and neurons thresholds, training algorithm);
- Type of output sensor signal and its changing range in SI units;
- The physical quantity which is necessary to use for receiving of the output value of the parametrical sensor;
- The type, changing range and allowable error of influence quantity.

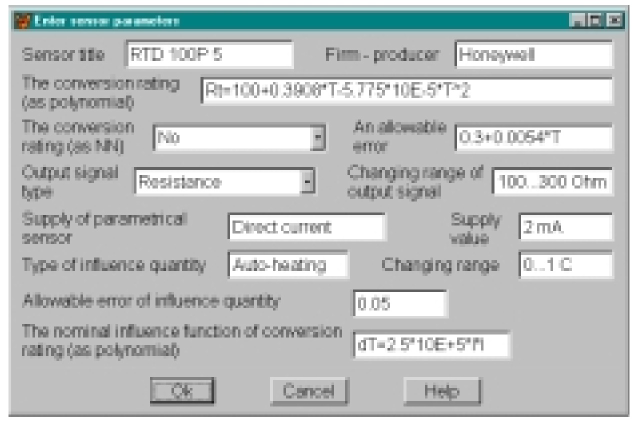

Fig. 9. The dialogue window of the sensor parameters input in the IDSN Manager

The dialogue window of IDSN data acquisition channel configuration is presented on Fig. 10. User configures the certain data acquisition channel using this subroutine. All data about components of this channel should be entered into IDSN Database before. Besides here it is possible to setup parameters of accuracy increasing method for the given channel at once.

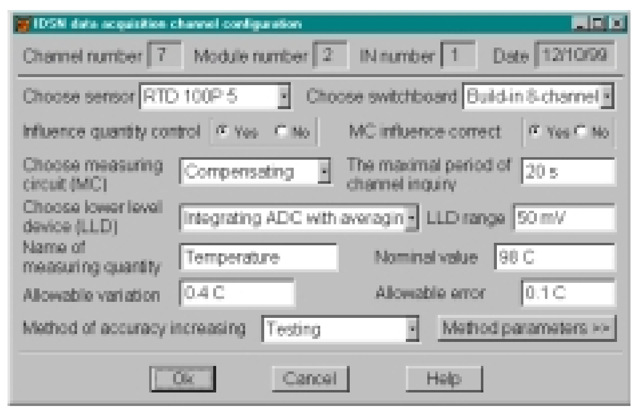

Fig. 10. The dialogue window of IDSN data acquisition channel configuration

4. Experimental Researches

The method of accuracy increasing was executed with experimental data using by operation of three neural networks (IHDNN, ANN and PNN). The 10 curves of sensor drift were used as historical. The 10E-7 sum-squared error is reached at IHDNN training. The percentage error of prediction exceeds 10% only in one case. For estimation of ANN the 5 points of historical data calibrations are used. The 2.4E-7 sum-squared error is reached at training of the ANN model with 5 hidden neurons and logistic activation function. The maximum percentage error does not exceed 2%. The result of approximation contains 25 points on each curve of historical data where PNN (model with 10 input neurons, 10 hidden neurons with logistic activation function and one output linear neuron) was trained up to 7.8E-8 sum-squared error. The maximum percentage error does not exceed 11% at PNN prediction. Thus, the proposed method of accuracy increasing allows extending of an intertesting interval 10 times with total percentage error 23%.

The training of neural networks was conducted sequentially and simultaneously for estimation of efficiency of CC resources using. The personal computer with one processor Celeron™400 was used as CC. The training time of each PNN did not exceed 11 minutes for 10 PNN at sequential training and the total training time was 39 minutes (see Table 1). The training time of each PNN was increased and the maximum time did not exceed 46 minutes at simultaneous training (all 10 neural networks have started practically in one moment of time).

Table 1. Training Time for 10 PNNs at Sequential (─) and Simultaneous (═) Training

NN	1	2	3	4	5	6	7	8	9	10
─	10,1	3,9	1,2	6,8	1,4	3,9	3,7	0,8	1,8	5,4
═	32,7	26,8	15	45,2	20,9	8,7	30,1	7	20,8	38,4

The effective utilization of CC computing resources is achieved at sequential training in case of uniprocessor computer using as CC (in the majority of industrial solutions). However the losses of time are rather insignificant at simultaneous NN training. At the same time it is expedient to provide a possibility of forced interruption of NNs training process at receiving of variant of neural network with rather good training parameters. In this case the probability of time economies is higher at simultaneous NN training then at sequential training.

Conclusions

The IDSN high-level software is developed using client-server technology. It provides: (i) fulfillment of intelligent functions in relation to measurement process [10]; (ii) fulfillment of procedures of accuracy increasing of sensors data acquisition and processing using neural networks; (iii) an effective utilization of CC resources in real time scale.

Acknowledgment

Authors would like to thank INTAS, grant reference number INTAS-OPEN-97-0606.

References

[1] http://www.fluke.com/products/home.asp?SID=7&AGID=6&PID=5308.

[2] http://content.honeywell.com/sensing/control/mc/VPR100.stm.

[3] http://www.dataq.com/di730.htm.

[4] L. Finkelstein, "Measurement and Instrumentation Centre", *Measurement,* vol. 14, No 1, 1994, pp.23-29.

[5] C.Alippi, A.Ferrero, V.Piuri, "Artificial Intelligence for Instruments & Applications", *IEEE I&M Magazine*, June 1998, pp.9-17.

[6] P. Daponte, D. Grimaldi, "Artificial Neural Networks in Measurements", *Measurement*, vol. 23, 1998, pp.93-115.

[7] A.Sachenko, V.Kochan, V.Turchenko, "Intelligent Distributed Sensor Network", *Proc. of 15th IEEE IMTC/98,* St.Paul, USA, 1998, pp.60-66.

[8] A.Sachenko, V.Kochan, V.Turchenko, V.Tymchyshyn, N.Vasylkiv, "Intelligent Nodes for Distributed Sensor Network", *Proc. of 16th IEEE IMTC/99*, Venice, Italy, 1999, pp.1479-1484.

[9] Golovko V., Grandinetti L., Kochan V., Laopoulos T., Sachenko A., Turchenko V., "Sensor Signal Processing Using Neural Networks", *Proc. of IEEE Region 8 Intern. Conference Africon'99*, Cape Town, South Africa, Sep 29-Oct 1, 1999, pp.339-344.

[10] V.Golovko, L.Grandinetti, V.Kochan, T.Laopoulos, A.Sachenko, V.Turchenko, V.Tymchyshyn, "Approach of an Intelligent Sensing Instrumentation Structure Development", *Proc. of IEEE Int. Workshop on Intelligent Signal Processing*, 1999, Budapest, Hungary, pp.336-341.

[11] Kroese B, An Introduction to Neural Networks, Amsterdam, *University of Amsterdam*, 1996, 120 p.

[12] V.Golovko, J.Savitsky, A.Sachenko, V.Kochan, V.Turchenko, T.Laopoulos, L.Grandinetti, "Intelligent System for Prediction of Sensor Drift", *Proc. of Inter. Confer. on Neural Networks and Artificial Intelligence ICCNAI'99*, Brest, Belarus, 1999, pp.126-135.

[13] A.Sachenko, V.Kochan, V.Turchenko, V.Golovko, J.Savitsky, T.Laopoulos, "Method of the training set formation for neural network predicting drift of data acquisition device", *Patent of Ukraine*, application number 2000010010, in press