УДК 004.75

Krasnov A.E., Nadezhdin E.N., Nikol'skii D.N.

State Institute of Information Technologies and Telecommunications (SIIT&T «Informika»), Moscow, Russia

ARCHITECTURE OF THE PARALLEL SOFTWARE FOR THE SIMULATION OF MULTIDIMENSIONAL PROBLEMS

Abstract

The architecture of the parallel software constructed on the basis of a design pattern «Bridge», and intended for simulation of multidimensional tasks (network dynamics; compression of multidimensional data, pattern recognition) is considered.T he architecture is based on an object-oriented approach to programming and using design patterns. The hierarchy of the classes realizing methods for the most various solvers is created. For parallel programming on a cluster of Intel Core i5-2300 processors and graphics processors GM206, NVIDIA GeForce GTX 950 cards MPI- and CUDA-technologies are used. Examples of solving some multidimensional problems are given: network dynamics, data compression, pattern recognition.

Keywords

Architecture; parallel software; object-oriented approach; design patterns; simulation; multidimensional problems; network dynamics; data compression; pattern recognition; MPI- and CUDA-technologies.

Краснов А.Е., Надеждин Е.Н., Никольский Д.Н.

Государственный научно-исследовательский институт информационных технологий и телекоммуникаций, г. Москва, Россия

АРХИТЕКТУРА ПАРАЛЛЕЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СИМУЛЯЦИИ МНОГОМЕРНЫХ ЗАДАЧ

Аннотация

Рассмотрена архитектура параллельного программного обеспечения, построенная на основе шаблона проектирования «Мост» и предназначенная для симуляции многомерных задач (сетевой динамики, сжатия многомерных данных, распознавания образов). Архитектура основана на объектно-ориентированном подходе в программировании и использовании шаблонов проектирования. Создаётся иерархия классов, реализующих методы для самых разнообразных решателей. Для параллельного программирования на кластере процессоров Intel Core i5-2300 и графических процессоров GM206 используются карты (NVIDIA GeForce GTX 950) MPI- и CUDA-технологии. Приведены примеры решения некоторых многомерных задач: сетевой динамики, сжатия данных, распознавания образов.

Ключевые слова

Архитектура; параллельное программное обеспечение; объектно-ориентированный подход; шаблоны проектирования; симуляция; многомерные задачи; сетевая динамика; сжатие данных; распознавание образов; MPI- и CUDA-технологии.

Introduction

The practice of solving a number of problems related to network dynamics [1, 2], compression of spectral data [3, 4]; boundary problems of mathematical physics [5, 6], invariant recognition of signals and images [7, 8], neural network decision-making methods [9] prompted authors to develop the most suitable parallel software architecture for fast numerical solution of such multidimensional problems.

In this article, we offer a software architecture based on GoF templates (patterns) [10]. These patterns use an object-oriented approach to describe how to solve common problems in software development. In accordance with this approach, a class hierarchy is implemented that implements methods for a wide variety of solvers. For parallel programming on a cluster of Intel Core i5-2300 processors and graphics processors GM206, NVIDIA GeForce GTX 950 cards are used MPI- and CUDA-technologies. The paper also discusses the solution of some network problems using the proposed architecture.

The statement of the problem

One of widespread classifications of patterns of design is classification by extent of specification and level of abstraction of the considered systems. Patterns of design of program systems are divided into the following categories [10]: architectural patterns; design patterns; idioms.

Architectural patterns, being the most high-level patterns, describe the block diagram of a program system in general. The separate functional components of system called by subsystems and also relationship between them are specified in this scheme.

Patterns of design describe schemes of specification of program subsystems and relationship between them, without influencing structure of a program system in general and keeping independence of realization of a programming language. Patterns of GoF belong to this category in which patterns of design of object-oriented systems are understood as the description of interaction of the objects and classes adapted for the solution of the general problem of design in a concrete context [10].

Idioms, being low-level patterns, deal with questions of the realization of any problem taking into account features of this programming language. At the same time often same idioms for different programming languages look differently or don't make sense at all. For example, in C ++ for elimination of possible leakages of memory intellectual indexes can be used [11]. The intellectual index contains the index on the site of dynamically marked out memory which will be automatically freed at an exit from a visibility range. In the environment of Java of such problem just doesn't exist as there automatic assembly of garbage is used. Usually, for use of idioms it is necessary to know features of the applied programming language deeply.

Main objective of the present work is the creation of the flexible parallel software for the solution of a wide class of multidimensional tasks, problems operating with a large number of elements and their connections (links) or relationships.

So, for example, in the simulation problem of network dynamics, the flexibility of the software lies in the ability to scale the task: changes of quantity of knots of the network, structure of their communications; models of functioning of basic elements (switchboards, connectors, controllers) and their parameters [1, 2].

Below architectural patterns for the main class hierarchies of a program system are described.

Archtectural patterns of the parallel software

The scheme of architectural patterns for the main class hierarchies is submitted in the Figure 1.

The development is realized in C ++ [12]. The project architecture is presented in the diagram. The abstract class Solver defines the interface necessary for the full solution of the chosen task.

The creation of specific simulators, representing descendants of the Solver class, is performed in accordance with the Factory method pattern.

The latter allows you to expand the project with your own models. The implementation of descendant's methods of Solver classes is separated from their interface in accordance with the Bridge pattern. The latter allows you to build the project for universal and graphics processors. Implementations for graphics processors are made using Cuda technology.

The work on creating an abstract class Solver begins with an interface for solving the problem: the setting of fields that include parameters of its objects; the definition of basic methods of the solution. Further, descendants of the abstract class Solver-program modules are created for individual interfaces or solvers Solver 1, Solver 2, ..., Solver *N*, as shown in Figure 1.

Also, different libraries of the SolverCuda and SolverCPU subroutines are created, which are combined and brought to the same Solverimpl interface.

To implement the task solvers (Solver 1, Solver 2, ..., Solver *N*) call through Solverimpl separate subroutines.

Developed software is oriented to the following multidimensional tasks:

- the simulation of the network dynamics;
- the simulation of the data compression;
- the simulation of the pattern recognition;
- the simulation of the phase portraits construction and the allocation of invariant pattern’s signs;
- the simulation of three-dimensional evolution of the boundary of a polluted area;
- the simulation of the neural network aggregation of complex system’s parameters.

Simulation of the network dynamics

In the problem of the network dynamics Solver forms (fields) are used [1, 2]:

- Comms – for the description of group of switchboards;
- Conns – for the description of group of connectors;
- Contrs – for the description of group of controllers.

Programming of separate realization of elements of the network is executed with use of a pattern of design of Bridge. So, the abstract class Solver has the protected field pImpl which indicates concrete realization of methods of a class. On the chart it is methods of the final classes SolverCuda and SolverCPU. Methods of the class SolverCuda are intended for performance on a cluster of graphic cards and use CUDA and MPI technologies. Methods of the class SolverCPU are intended for performance on a cluster of universal processors and use MPI and OpenMP technologies.

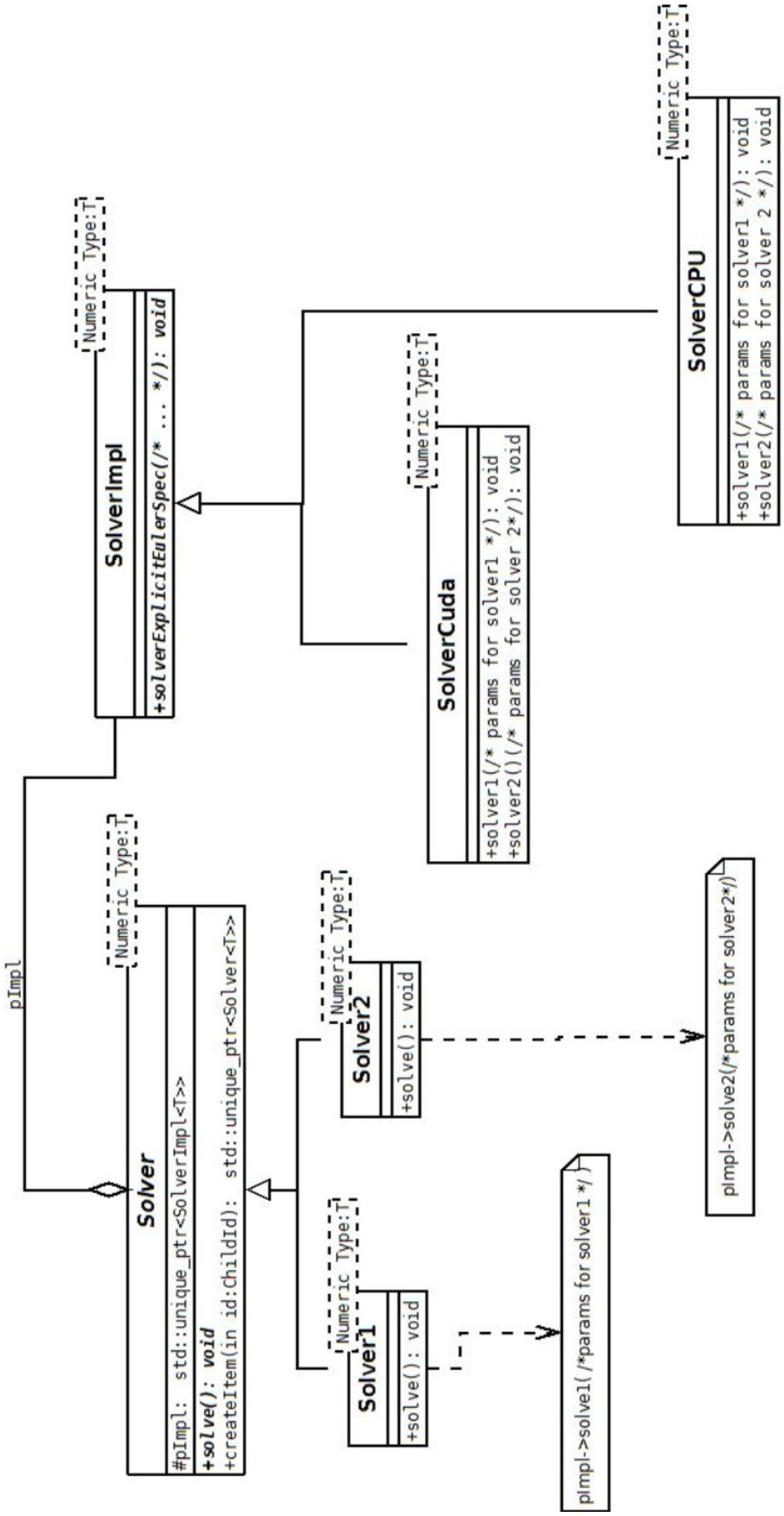
Numeric Type:T
SolverImpl
+solverExplicitEulerSpec(/* ... */): void
Numeric Type:T
SolverCPU
+solver1(/* params for solver1 */): void
+solver2(/* params for solver 2 */): void
Numeric Type:T
SolverCuda
+solver1(/* params for solver1 */): void
+solver2()(/* params for solver 2*/): void
pImpl
Numeric Type:T
Solver
#pImpl: std::unique_ptr<SolverImpl<T>>
+solve(): void
+createItem(in id:ChildId): std::unique_ptr<Solver<T>>
Numeric Type:T
Solver2
+solve(): void
Numeric Type:T
Solver1
+solve(): void
pImpl->solve1(/*params for solver1*/)
pImpl->solve2(/*params for solver2*/)

Figure 1. Parallel software architecture

The abstract class Solver, in compliance with a design pattern Factory method, contains the CreateItems method for generation of various specialized objects of the final classes realizing concrete methods of elements of network. So, the class Comms can generate a number of the successors who are carrying out the modeling of switchboards under the known neurophysiological laws: NeursIzhik – for Izhikevich's models; NeursXX – for Hodzhkin-Huxley's models; and another.

For realization of codes of the software it is used C^{++} 14 [12].

The created software is the development of software for simulation of the network dynamics [2]. Based on the architecture under consideration, the work of the so-called cybernetic network was implemented, the functional-structural topology of which is shown in Figure 2. This topology includes the following subnets and their structural elements.

Information subnet formed by commutators («*Comm*», *N*) and connectors («*Conn*», *N* x *N*).

Commutator (switchboard) «*Comm*» has input and output communication ports connected hubs with internal and external inputs and outputs, as well as ports of external control.

Connector «*Conn*» communicates between switches, and has input and output communication ports, as well as ports of external and internal control.

Management subnet formed by controllers («*Contr*», $M \geq N$).

Controllers «Contr» have input and output communication ports, as well as ports of external and internal control. These ports are connected via communication hubs channels.

Setting up the structural elements of a heterogeneous network carried out via the external control trunk - Trunk of an external control.

In information subnet there is an exchange of information streams of I_{comm} and I_{conn}, forming the data transformed by switchboards and connectors on the respective internal trunks of switchboards and connectors.

The input of each switch «*Comm*» via internal Trunk of switches is connected to the output of the group of connectors «*Conn*».

The information output of each switch «*Comm*» is connected via internal Trunks of switches with a group of connectors. For example, the output of the switch $Comm_1$ - the information dada flow $Icomm_1$ effects on inputs of connectors $Conn_{1,1}$, ..., $Conn_{1,N}$.

In the management subnet the exchange by streams of ***Sconn*** and ***C*** data via the Trunk1, Trunk2 of internal management is occurred (Figure 2). Data flow ***Sconn*** describes the state of connectors, and *C* describes the control by connectors from the controllers.

The exchange by information data flows *Icontr* between all controllers carried out via the internal $Trunk_3$ (Figure 2).

Since the number *M* of controllers are much smaller than the number *N*x*N* of connectors, each controller controls a group (set) of connectors. For example, connectors of the group {1,1}, ..., {1, *N*} ($Conn_{1,1}$, ..., $Conn_{1,N}$) are operated by controller $Contr_1$ which generally performs various controls $\boldsymbol{C}_{1,1}$, ..., $\boldsymbol{C}_{1,N}$. In turn, each *m*-th controller receives information $\boldsymbol{S}_{\{m\}}$, which characterizes the state of a group {*m*} of connectors connected to it (*m* = 1, ..., *M*).

In a set of experiments the effect of controllers on the connectors in the network with the correct regular topology (complete graph), which is most resistant to variations of relations [13], was studied. As a dynamic model of the switches functioning the model of «Izhikevich» neuron [14] was used. For example, in the diagram (Figure 3, a) all weights are the same and equal to 100, and in the diagram (Figure 4b) for 10% of bonds weights are set equal to 1.

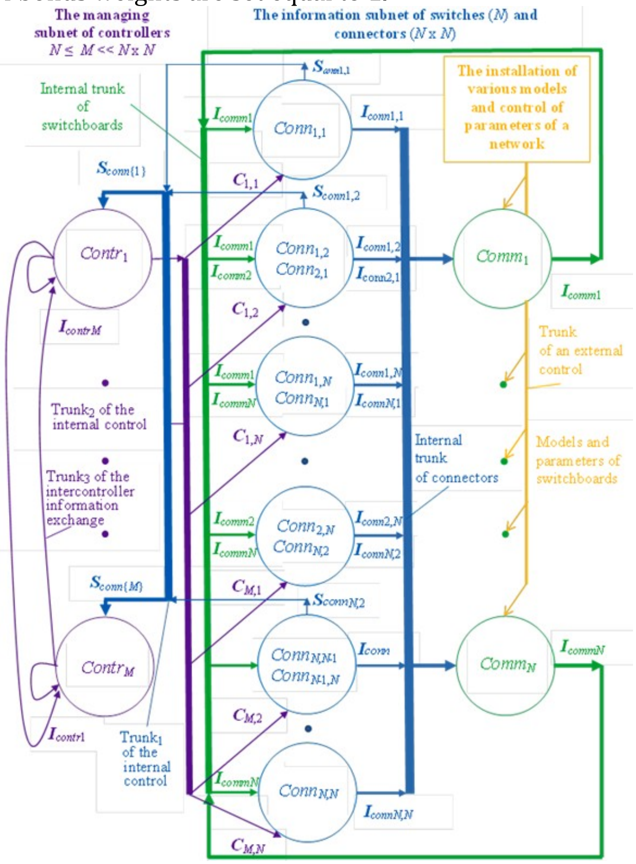
The managing subnet of controllers
N ≤ M << N x N
The information subnet of switches (N) and connectors (N x N)
Internal trunk of switchboards
The installation of various models and control of parameters of a network
S_conn{1}
S_conn1,1
S_conn1,2
I_comm1
I_conn1,1
Conn1,1
C1,1
Contr1
Conn1,2
Conn2,1
I_comm2
I_conn1,2
I_conn2,1
Comm1
I_comm1
C1,2
I_contrM
Conn1,N
ConnN1
I_commN
I_conn1,N
I_connN,1
Trunk of an external control
Trunk2 of the internal control
C1,N
Internal trunk of connectors
Models and parameters of switchboards
Trunk3 of the intercontroller information exchange
Conn2,N
ConnN2
I_conn2,N
I_connN,2
S_conn{M}
C_M,1
S_connN,2
ContrM
ConnN,N-1
ConnN-1,N
I_conn
CommN
I_commN
C_M,2
I_contr1
Trunk1 of the internal control
ConnN,N
I_connN,N
C_M,N

Figure 2. Functional - structural topology of the cybernetic network taking into account the exchange of information and control data flows

Simulation of the data compression

To solve the problems of analysis of multidimensional data in order to cluster them and to reveal regularities, it is necessary to make significant data compression. To this end, based on the approach discussed above, software was developed to reduce the dimensionality of data based on their projecting algorithms into orthogonal subspaces.

Figure 4 shows the network topology that implements a neural-like algorithm for mapping the original optical spectral data to orthogonal subspaces [3, 4]. This topology is some topological subset of the topology, shown in Figure 3.

The dimension's reducing of optical spectra is made for the whole of its set – the set $\{\boldsymbol{S}\}^k$ (k = 1, 2, ..., K), consisting of K elements $\boldsymbol{S}^k = (S^k_1, S^k_2, ..., S^k_N)^T$, each of which includes discrete components with the amplitudes S^k_n (n = 1, 2, ..., N).

Each S^k_n component of any k-th optical spectrum is multiplexed into several channels, the number of which is much smaller than the number of spectral components (in Figure 3 three channels *X*, *Y*, *Z* are shown).

The software registers in the first channel *X* variations in the values of all optical spectra components, and, from these variations, initially adjusts the increase/decrease component's amplitudes of the first channel.

As an example, the variations are recorded in the form of the values $\Delta_n = |S^{max}_n - S^{min}_n|$, determined by the maximum S^{max}_n and the minimum S^{min}_n values of the amplitude's scatter of all components ($n = 1, 2, ..., N$).

Further, all variations Δ_n ($n = 1, 2, ..., N$) are compared with the originally assigned threshold Δ_{1thr},, and when exceeding this threshold, the samples of the discrete signal of the first reference channel are formed by the first notch filtering:

$$W_X = (w_{X1}, w_{X2}, \dots, w_{XN})^T ,$$

where

$$w_{XN} = \begin{cases} 1, if\ \Delta_n \Delta_{1thr}, \\ 0, otherwise . \end{cases}$$

These values adjust the values of the amplification / attenuation coefficients w_{Xn} of component's amplitudes of the optical spectra in the first channel *X* (Figure 3).

Notch filtering allows you to remove non-informative components from optical spectra, leaving only components responsible for their difference.

Then, in the second channel *Y*, using Hilbert filtering, the first auxiliary discrete signal is formed:

$$\boldsymbol{W}_1 = (\boldsymbol{w}_{11}, \boldsymbol{w}_{12}, \dots, \boldsymbol{w}_{1N})^T ,\ w_{1n} = \sum_k h_{n-k} w_{Xk} ,$$

$$\sum_n w_{1n} w_{Xn} = 0,$$

where $h_k = \frac{2}{\pi k}[1 - \cos(\pi k)]$ – is the impulse response of Hilbert [15].

Further, absolute sample values $|w_{1n}|$ ($n = 1, 2, ..., N$) of the first auxiliary discrete signal are registered, and samples of the second auxiliary discrete signal $\boldsymbol{W}_2 = (w_{21}, w_{22}, ..., w_{2N})^T$ are formed by the second notch filtering in the form of unitary units:

$$w_{2n} = \begin{cases} 1,\ \text{if}\ ((w_{1n} > 0)\ И\ (|w_{1n}| \geq \Delta_{2thr})), \\ \text{-}1,\ \text{if}\ ((w_{1n} < 0)\ И\ (|w_{1n}| \geq \Delta_{2thr})), \\ 0, otherwise . \end{cases}$$

The second notch filtering contributes to the isolation of features in the components of optical spectra related to individual classes (classes that combine similar optical spectra).

Thus, if the first notch filtering separates the components responsible for the difference, the second separates the components responsible for the similarity of the optical spectra.

The samples of the reference discrete signal $\boldsymbol{W}_Y = (w_{Y1}, w_{Y2}, ..., w_{YN})^T$of the second channel are formed on the basis of the Gramm-Schmidt orthogonalization process.

The amplification / attenuation coefficients w_{Yk} of component's amplitudes of the optical spectra in the second channel *Y* are adjusted in proportion to sample'svalues of the reference discrete signal $\boldsymbol{W}_Y$.

In subsequent channels, the corresponding reference discrete signals orthogonal to all the reference signals of the previous channels are formed. So, for the third channel *Z*, the third reference discrete signal $\boldsymbol{W}_Z = (w_{Z1}, w_{Z2}, ..., w_{ZN})^T$ is formed. The values of the amplification / attenuation coefficients w_{Zk} of the amplitude values of the optical spectral components in the third channel *Z* are adjusted in proportion to the values of the reference discrete signal samples $\boldsymbol{W}_Z$.

As shown in Figure 4, the reduced images of all *K* optical spectra are formed in each channel by the spatial accumulation of the corresponding filtered components:

$$X_k = \sum_n^N w_{Xn} S^k_n ;\quad Y_k = \sum_n^N w_{Yn} S^k_n ;\quad Z_k = \sum_n^N w_{Zn} S^k_n ,$$

$$k = 1, 2, ..., K.$$

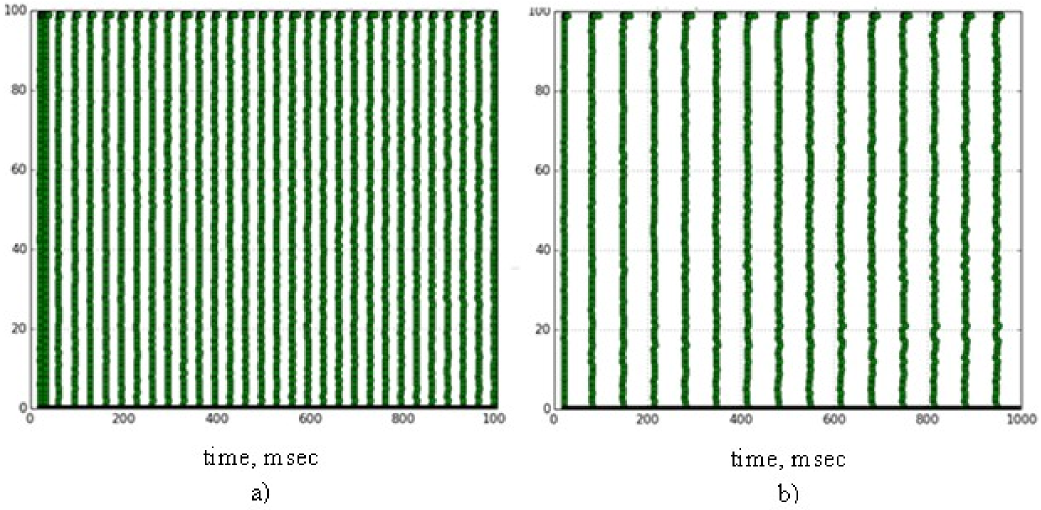

Figure 3. The activity of neurons at different scales links

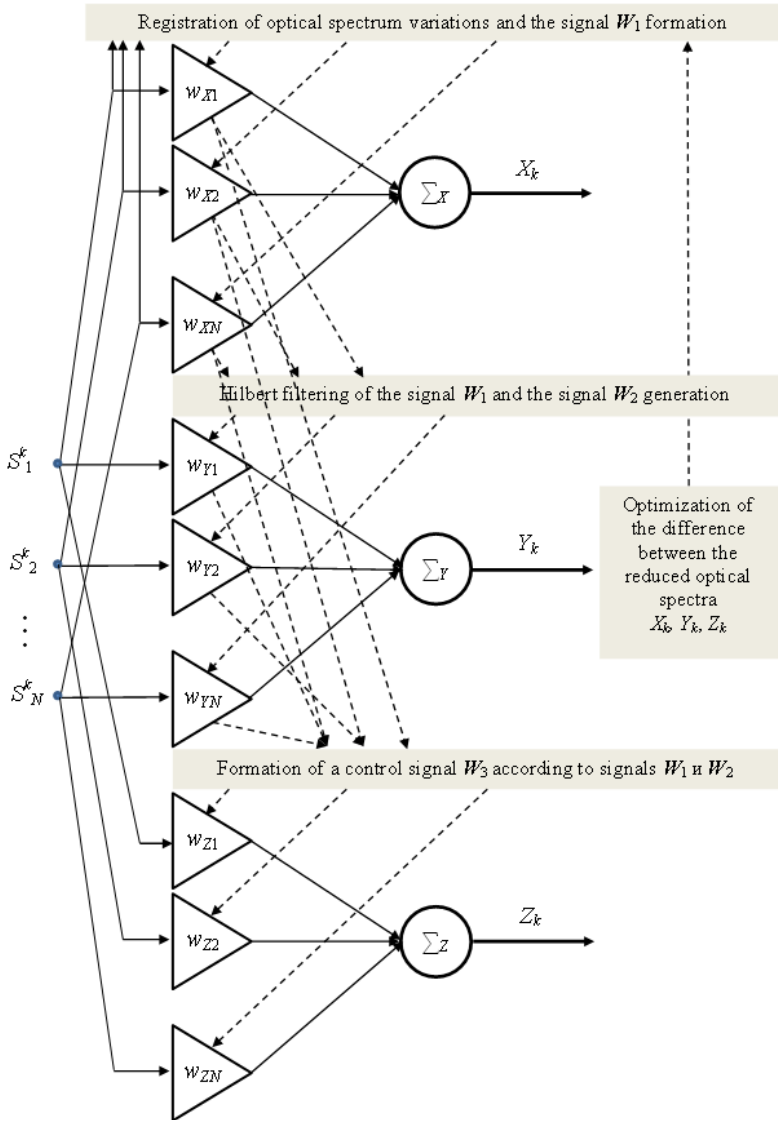
Registration of optical spectrum variations and the signal W_1 formation
w_{X1}
w_{X2}
w_{XN}
$\sum_X$
X_k
Hilbert filtering of the signal W_1 and the signal W_2 generation
S^k_1
S^k_2
S^k_N
w_{Y1}
w_{Y2}
w_{YN}
$\sum_Y$
Y_k
Optimization of the difference between the reduced optical spectra X_k, Y_k, Z_k
Formation of a control signal W_3 according to signals W_1 и W_2
w_{Z1}
w_{Z2}
w_{ZN}
$\sum_Z$
Z_k

Figure 4. Functional - structural topology of the cybernetic network for the data compression

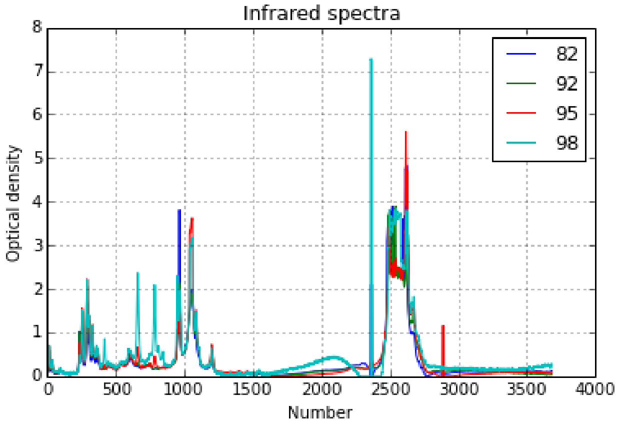

Figure 5. Examples of four different infrared spectra of automotive gasolines (with octane numbers 82, 92.95, 98)

In the described procedure of the decreasing of the dimensionality K of optical spectra, each of the spectra, the multidimensional element $\boldsymbol{S}^k$, will correspond to its reduced three-dimensional pattern X_k, Y_k, Z_k (k = 1, 2, ..., K).

For the purpose of the spectra visualizing by the increasing of the difference in the reduced patterns, it is iteratively changing thresholds Δ_{1thr} and Δ_{2thr} in the first two channels, increasing any distinction criterion (for example, the minimum of minima distances between points X_k, Y_k, Z_k or the multiple discriminant of the Fisher criterion).

Figure 5 shows examples of four different infrared spectra of automotive gasolines (with octane numbers 82, 92.95, 98) and Figure 6 shows examples of their reduced three-dimensional patterns.

These examples clearly demonstrate the advantages of the proposed method. Comparison of the obtained result with the results on the identification of infrared spectra [16] showed that the reduction in the dimension does not worsen the discriminatory properties of reduced spectra, makes it possible to simplify their observation and interpretation, and does not limit the possibility of further use for solving various problems of identification, clustering and recognition, parameter’s estimation.

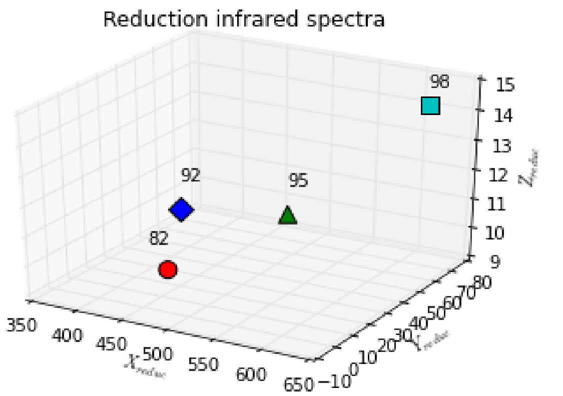

Figure 6. Reduced three-dimensional patterns of infrared spectra

Simulatin of the pattern recogniion

Reduced spectra can then be used for their recognition / discrimination using an effective neural network technology that allows working with signals against a background of significant interference (noises) [17].

So, on the basis of the above architecture, software was implemented for the decision network, which is trained simply by presentation of standards, and does not use the deeplearningalgorithm by the method of back propagation of the error [17].

Figure 7 shows a simplified fragment of the structure of such a decisive network, explaining the principle of its functioning. As can be seen, one of the 6 standards (**S**$_1$, ...,**S**$_6$) recorded in its first layer is fed to the network input. Each reference signal is made up of 6 components. In each layer, nonlinear elements (Nonlinearity) compare the incoming signal **X** with the corresponding standard **S**^{k} through a nonlinear measure of similarity cos^{40}(∠ **XS**^{k}), где ∠ **XS**^{k}, where ∠ **XS**^{k} is the angle between the vectors **X** and **S**^{k}.

In the 2nd and 3rd layers, the responses of previous layers to the corresponding reference signals are used as reference standards. As shown by experiments, this provides high reliability of the signal's recognition in the network.

Figure 7a shows the operation of the network to distinguish one of the 6 reference signals (first) in the absence of interference. It can be seen that the network recognizes the first reference signal fed to the input with 100% similarity measures across all layers.

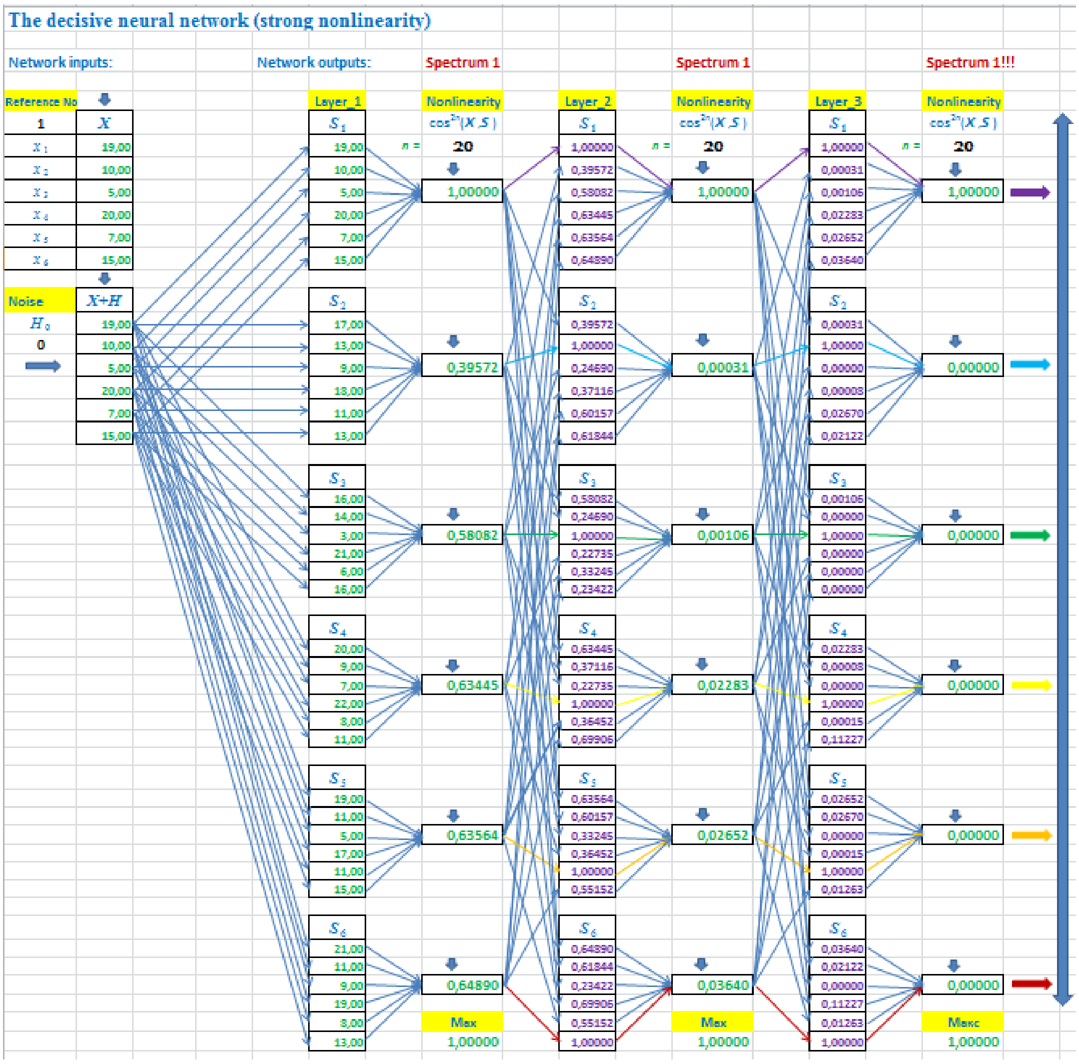
The decisive neural network (strong nonlinearity)
Network inputs:
Network outputs:
Spectrum 1
Spectrum 1
Spectrum 1!!!
Reference No
1
X
x 1 19,00
x 2 10,00
x 3 5,00
x 4 20,00
x 5 7,00
x 6 15,00
Noise
H 0
0
X+H
19,00 10,00 5,00 20,00 7,00 15,00
Layer_1
S 1: 19,00 10,00 5,00 20,00 7,00 15,00
S 2: 17,00 13,00 9,00 18,00 11,00 13,00
S 3: 16,00 14,00 3,00 21,00 6,00 16,00
S 4: 20,00 9,00 7,00 22,00 8,00 11,00
S 5: 19,00 11,00 5,00 17,00 11,00 15,00
S 6: 21,00 11,00 9,00 19,00 8,00 13,00
Nonlinearity
cos2n(X,S)
n = 20
1,00000
0,39572
0,58082
0,63445
0,63564
0,64890
Max
1,00000
Layer_2
S 1: 1,00000 0,39572 0,58082 0,63445 0,63564 0,64890
S 2: 0,39572 1,00000 0,24690 0,37116 0,60157 0,61844
S 3: 0,58082 0,24690 1,00000 0,22735 0,33245 0,23422
S 4: 0,63445 0,37116 0,22735 1,00000 0,36452 0,69906
S 5: 0,63564 0,60157 0,33245 0,36452 1,00000 0,55152
S 6: 0,64890 0,61844 0,23422 0,69906 0,55152 1,00000
Nonlinearity
cos2n(X,S)
n = 20
1,00000
0,00031
0,00106
0,02283
0,02652
0,03640
Max
1,00000
Layer_3
S 1: 1,00000 0,00031 0,00106 0,02283 0,02652 0,03640
S 2: 0,00031 1,00000 0,00000 0,00008 0,02670 0,02122
S 3: 0,00106 0,00000 1,00000 0,00000 0,00000 0,00000
S 4: 0,02283 0,00008 0,00000 1,00000 0,00015 0,11227
S 5: 0,02652 0,02670 0,00000 0,00015 1,00000 0,01263
S 6: 0,03640 0,02122 0,00000 0,11227 0,01263 1,00000
Nonlinearity
cos2n(X,S)
n = 20
1,00000
0,00000
0,00000
0,00000
0,00000
0,00000
Макс
1,00000

Figure 7a. Decisive neural network with strong nonlinearity (no noise)

Figure 7b shows the operation of the network to distinguish one of the 6 reference signals (first) in the presence of interference. It can be seen that not all network layers correctly recognize the first reference signal fed to the input.

In the experiments, the variation of the components of the reference signals was changed, adding to them the randomly distributed interference, and the network response was checked. Figure 8 shows the dependence of the recognition error on the amplitude of the noise with a variation of the reference signals of 13%.

The considered signal discrimination technology was originally developed for the control of protective nanocrystalline labels [17] on their optical spectra, but its integration with the technology of spectral compression [3, 4], by virtue of its generality, can be used to distinguish any signals observed against strong interference.

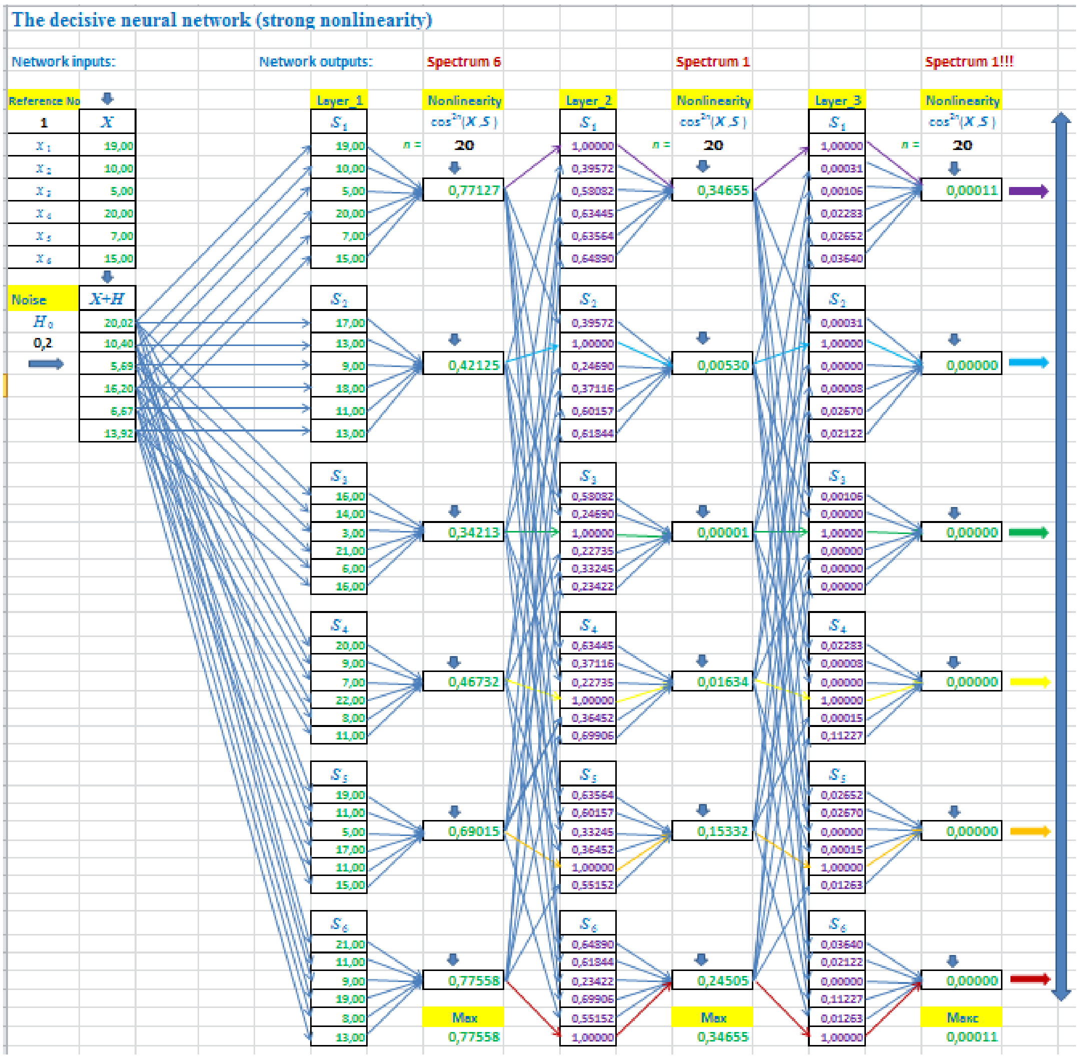
The decisive neural network (strong nonlinearity)
Network inputs:
Network outputs:
Spectrum 6
Spectrum 1
Spectrum 1!!!
Reference No
1
X
x1 19,00
x2 10,00
x3 5,00
x4 20,00
x5 7,00
x6 15,00
Noise
X+H
H0
0,2
20,02
10,40
5,69
16,20
6,67
13,92
Layer_1
S1 19,00 10,00 5,00 20,00 7,00 15,00
S2 17,00 13,00 9,00 18,00 11,00 13,00
S3 16,00 14,00 3,00 21,00 6,00 16,00
S4 20,00 9,00 7,00 22,00 8,00 11,00
S5 19,00 11,00 5,00 17,00 11,00 15,00
S6 21,00 11,00 9,00 19,00 8,00 13,00
Nonlinearity
cos2n(X,S)
n = 20
0,77127
0,42125
0,34213
0,46732
0,69015
0,77558
Max
0,77558
Layer_2
S1 1,00000 0,39572 0,58082 0,63445 0,63564 0,64890
S2 0,39572 1,00000 0,24690 0,37116 0,60157 0,61844
S3 0,58082 0,24690 1,00000 0,22735 0,33245 0,23422
S4 0,63445 0,37116 0,22735 1,00000 0,36452 0,69906
S5 0,63564 0,60157 0,33245 0,36452 1,00000 0,55152
S6 0,64890 0,61844 0,23422 0,69906 0,55152 1,00000
Nonlinearity
cos2n(X,S)
n = 20
0,34655
0,00530
0,00001
0,01634
0,15332
0,24505
Max
0,34655
Layer_3
S1 1,00000 0,00031 0,00106 0,02283 0,02652 0,03640
S2 0,00031 1,00000 0,00000 0,00008 0,02670 0,02122
S3 0,00106 0,00000 1,00000 0,00000 0,00000 0,00000
S4 0,02283 0,00008 0,00000 1,00000 0,00015 0,11227
S5 0,02652 0,02670 0,00000 0,00015 1,00000 0,01263
S6 0,03640 0,02122 0,00000 0,11227 0,01263 1,00000
Nonlinearity
cos2n(X,S)
n = 20
0,00011
0,00000
0,00000
0,00000
0,00000
0,00000
Макс
0,00011

Figure 7b. Decisive neural network with strong nonlinearity (with noise)

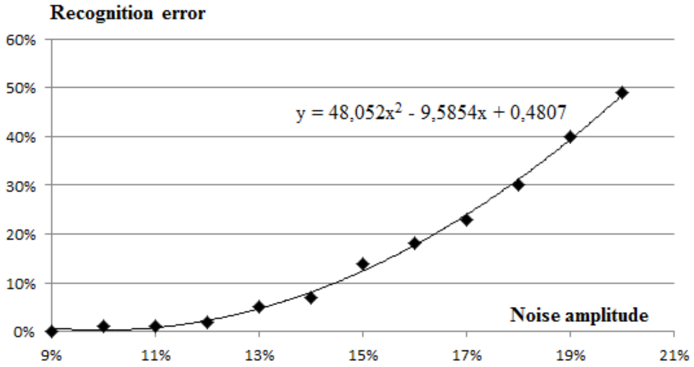

Figure 8. Dependence of the recognition error on the amplitude of the noise (with the analytic trend)

Conclusion

Using the Bridge structural pattern allowed to separate the user interface of the hierarchy of solvers from their implementation on various platforms (CPU and GPU). The latter allows performing independent development of solver implementation and extending the functional interface of the system.

The CreateItem() method built into the interface of the abstract solver class, in accordance with the generating pattern Factory method, allows us to expand the system with new types of solvers using the polymorphism mechanism.

Carried out experimental studies on the application of the architectural approach to a number of multidimensional network problems showed its effectiveness and ease of use both with the expansion of functionals within a specifically solved multidimensional problem, and in the transition from the one task to another.

References

1. Kalachev A.A., Krasnov A.E., Nadezhdin E.N., Nikol'skiy D.N., Repin D.S. Geterogennaya mnogosvyaznaya set' aktivnykh elementov. Innovatsionnye, informatsionnye i kommunikatsionnye tekhnologii // Sbornik trudov XIII Mezhdunarodnoy nauchno-prakticheskoy konferentsii. / Pod red. S.U. Uvaysova. – M.: Assotsiatsiya vypusknikov i sotrudnikov voenno-vozdushnoy inzhenernoy akademii im. prof. Zhukovskogo, 2016. S. 277-279.
2. Kalachev A.A., Krasnov A.E., Nadezhdin E.N., Nikol'skiy D.N., Repin D.S. Model' geterogennoy seti dlya simulyatsii neyrodinamicheskikh zadach. // Sovremennye informatsionnye tekhnologii i IT-obrazovanie. 2016, Tom 12, №1. S. 80-90. (ISSN2411-1473).
3. Krasnov A.E., Krasnikov S.A., Chernov E.A. Method of the compression of phase portraits of generalized spectral data for solving their clustering and recognition //In math. of International Scientific-Practical Conference «Innovative Information Technologies» (Prague, 21–25 april 2014), 2014, Volume 2, section 2. – M.: HSE, P. 664–670.
4. Krasnov A.E., Nikol'skiy D.N., Kalachev A.A. Snizhenie razmernosti spektral'nykh dannykh neyropodobnym algoritmom //Svidetel'stvo o gosudarstvennoy registratsii programmy dlya EVM, Rossiyskaya federatsiya, № 2017612195, 2017.
5. Nikol'skii D. N. Mathematical simulation of the evolution of a liquid-liquid interface in piecewise inhomogeneous layers of complex geological structure // Computational Mathematics and Mathematical Physics. June 2013, Volume 53, Issue 6, P. 858-865. doi:10.1134/S0965542513060146.
6. Nikol'skii D. N. Three-dimensional evolution of the boundary of a polluted area in a bounded piecewise homogeneous porous material//Computational Mathematics and Mathematical Physics. May 2011, Volume 51, Issue 5, P. 855-861; doi: 10.1134/S0965542511050125.
7. KrasnovA.E., Krasnikov S.A., Kompanets I.N. Correlation-statistical methods of distinguishing complicated and noisy spectra // Journal of Optics A: Pure and Applied Optics. 2002, Volume 4, № 3. P. 329-337.
8. Krasnov A.E., Golovkin M.E. Metody vydeleniya invariantnykh priznakov izobrazheniy // Aktual'nye problem sovremennoy nauki. 2016, №4 (89). P. 209–212.
9. Krasnov A.E., Nadezhdin E.N., Nikol'skiy D.N., I dr. Neyrosetevoy podkhod k otsenivaniyu effektivnosti funktsionirovaniya organizatsii na osnove agregirovaniya pokazateley ee deyatel'nosti // Informatizatsiya nauki I obrazovaniya. Yanvar' 2017, №1. S. 141-154.
10. Gamma E., Khelm R., Dzhonson R., VlissidesDzh. Priemy ob"ektno-orientirovannogo proektirovaniya. – St. Peterburg: SPb.Piter, 2016. – 366 s.
11. Bjarne S. The C++ Programming Language, - 4th Edition. Addison-Wesley, 2013. – 1368 p.
12. Scott M. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media, 2014. 384 p.
13. Pu C.-L., Pei W.-J., Michaelsond A. Robustness analysis of network controllability // Physica, 2012, A 391. P. 4420-4425.
14. Izhikevich E.M. Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting // Cambridge. Massachusetts. London. England: The MIT Press, 2007. – 497 p.
15. Klingspor M. Hilbert transform: Mathematical theory and applications to signal processing //LiTH - MAT - EX, 2015. – 76 p.
16. Krasnov A.E., idr. Osnovy spektral'noy komp'yuternoy kvalimetrii zhidkikh sred / Pod redaktsiey prof. Krasnova A.E. - M: «Yurisprudentsiya», 2006. – 264 p.
17. Kazakov K.V., Kalachev A.A., Krasnov A.E., Nikol'skiy D.N., Shevelev S.A. Sravnenie effektivnostey razlicheniya signalov na fone sil'nykh pomekh na osnove mnogokriterial'noy i neyrosetevoy tekhnologiy. Innovatsionnye, informatsionnye i kommunikatsionnye tekhnologii // Sbornik trudov XIII Mezhdunarodnoy nauchno-prakticheskoy konferentsii. / Pod red. S.U. Uvaysova. – Moskva: Assotsiatsiya vypusknikov I sotrudnikov voenno-vozdushnoy inzhenernoy akademii im. prof. Zhukovskogo, 2016. S. 257-259.

Поступила: 25.03.2017

Об авторах:

Краснов Андрей Евгеньевич, доктор физико-математических наук, профессор, главный научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», a.krasnov@informika.ru

Надеждин Евгений Николаевич, доктор технических наук, профессор, главный научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», e.nadezhdin@informika.ru

Никольский Дмитрий Николаевич, кандидат физико-математических наук, доцент, ведущий научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», d.nikolsky@informika.ru

Note on the authors:

Krasnov Andrey, Doctor of Physical and Mathematical Sciences, Professor, Research fellow, State Institute of Information Technologies and Telecommunications, a.krasnov@informika.ru

Nadezhdin Evgeniy, doctor of technical sciences, professor, Research fellow, State Institute of Information Technologies and Telecommunications, e.nadezhdin@informika.ru

Nikol’skii Dmitriy, Candidate of Physical and Mathematical Sciences, Associate Professor, Senior research fellow, State Institute of Information Technologies and Telecommunications, d.nikolsky@informika.ru