ISSN 1060-992X, Optical Memory and Neural Networks (Information Optics), 2011, Vol. 20, No. 1, pp. 43–58. © Allerton Press, Inc., 2011.

Prognosis of Dynamical Systems Behavior Based on Cerebellar-Type Neural Technologies

W. L. Dunin-Barkowski[a], **Yu. A. Flerov**[b], **and L. L. Wyshinsky**[b]

[a]*Department of Neuroinformatics, Center of Optical Neural Technologies, Scientific Research Institute for System Analysis, Moscow, Russia*

[b]*A.A. Dorodnitsyn Computing Center, Russian Academy of Sciences, Moscow, Russia*

e-mail: wldbar@gmail.com

Received November 15, 2010; in final form, December 15, 2010

Abstract—We consider a system of multidimensional data prognosis based on the supposed mechanics of short-term prediction of the data in the cerebellum. Presented are the general description of the system, the selected method of numerical calculations and examples of testing the system for the prognosis of data in test mathematical problems, in transportation problem and in energy consumption prediction. Test results have demonstrated that have demonstrated that the cerebellar-based module of data prognosis might an effective and practical tool for multidimensional predictions.

Keywords: artificial neural networks, models of cerebellum, multidimensional data prognosis.

DOI: 10.3103/S1060992X11010073

1. INTRODUCTION

All coming events have been in fact prepared in the past. Experts consider that the 9/11 terrorist attacks were planned at least one year ahead. Of course, few minutes before the events, nothing but the magic could prevent them.

However, the attack could be destroyed should the security services paid attention in proper time for the alarming signals. Although it is hard to blame anybody concrete that they didn't recognized the approaching danger in those times. The present days means of environment monitoring has a limited capacity in forecasting events of that type, while the well-known need in these instruments is high. The catastrophic situations in the modern technogenic world environment arose in many scales—from the global crashes of communication and/or transportation to relatively local power supply blackout, from crisis in finances to natural disasters, etc. The main difficulty in prognosis of large-scale events is in large number of the system variables and in absence of formal models, which can describe interdependence of the variables. For example, for prediction of the stock rates at the New York Stock Exchange or at any other world stock exchange institution the number of the system dimensions is about 10^5–10^7. This number includes: (1) prices of all the current stocks; (2) general financial and economic indices in major world regions, currency exchange rates, major goods prices at several regional markets and, most important, traders' behavior, which is individual and in most cases hardly formalized. Besides, stock prices might substantially depend on the parameters which, at first glance, seem to be very remote from finances, e.g. social events like elections, state leaders' visits and speeches, natural disasters, social uprising, etc. Modern communication media and information storage and processing characteristics of modern (super-)computers enable us with sophisticated means of observation and monitoring of very large sets of events. However, the question is how to deal with this huge amount of information? There are tremendous amounts of information in computerized databases, but only a miserable part of them is actually used for general events analysis and prognosis. As a rule, the number of dimensions in really handled situations is in the order of tens or hundreds, at most. In other words, it seems that data with 10^5–10^8 parameters cannot be processed by the modern super-powerful computers. But human brain, which is behind the modern computers both in memory volumes and in information processing rates, can obviously perform tasks with these large numbers of dimensions. How can it be possible?

The mechanisms of brain operation until now remain among the most intriguing mysteries of the nature. But now, this citadel of enigmas step by step yields its positions to neurosciences. We already have examples of well-established, or highly probable mechanisms of information processing and neural control of organism's functions in several particular regions of the brain [12, 13, 19].

In this paper we analyze the possibility to apply our version of the cerebellar-type algorithms [5, 6] of data prognosis for standard technical problems. The cerebellum is one of the most advanced in evolution parts of the nervous system.

The emerging recently progress in brain understanding hints that the cerebellum, essentially is the *supercomputer* for many brain controlling functional subsystems and performs about 90% of the computational "load", connected, mainly, to short-term prognosis of many body's and mind's live parameters [2, 4, 6, 14–18, 20].

2. NEURAL NETWORKS AND MULTI-DIMENSIONAL DATA PROGNOSIS

Starting with 1980-s, the so called neural network computational methods started successfully compete with the traditional "mathematically correct" methods in dealing with many problems of information processing and systems control [8, 9, 21]. It should be noted that in most cases neural network methods are being applied to the prognosis (recognition, interpolation, etc.) of signals. We can also emphasize that the thorough comparison of the "neural" methods with the standard statistical methods shows that they have hardly overlapping fields of maximal efficiency [3].

The best "recommendation" of the artificial neural networks as a practical tool for the prognosis problems is the fact that they have deserved credit in practical financial applications, such as estimations of credit reliability, in procedures banking loans decisions, the active funds distribution, prognosis of cash flows, currency reserves control, etc. [1]. Neural networks are the best in cases when the underlying dependencies are nonlinear and there are no a priory data about the core processes. Thus, it could be summarized that application of neural networks in data prognosis is now a well-established branch of the modern computational technologies.

At this point, it should be emphasized that the overwhelmed majority of the applied "neural network" methods uses network constructions, based on so called formal neurons. That probably means that some effective neural mechanisms do not need more complicated properties of the live neural systems. Others might need them. And the cerebellum may use in its efficient functioning rather complicated properties of the real neurons.

Until recently, the cerebellar-type artificial control systems were used for the most obviously "cerebellar" applications, like robotic movements control. However, it becomes more and clearer that the cerebellum is involved practically in all functions of the control of the organism [16–18]. All the sub-regions of the cerebellar cortex have essentially the same structure, not depending on the concrete functions, in which they are involved. Here we are evaluating the hypothesis that the cerebellum all the time computes the forthcoming values of huge number of variables of the organism, which represents its state. Also, we hypothesize that one of the secrets of the cerebellar effectiveness lies in the fact that it predicts all the variables simultaneously, making use of their interdependence. The possibility of the successful momentous prediction of many variables at a time arises from the fact of interdependence of many variables and from the high connectivity between the standard cerebellar predicting modules. So, our objective in these research efforts is to test, if this possibility might be real.

The prognosis, interpolation and extrapolation of the data (including pattern recognition) always constituted the task no. 1 in the artificial neural network technologies. The mechanisms of solution of these problems, which has been recently revealed, are substantially new compared to the traditional machine learning. The basic structure of the cerebellar module is a closed loop, consisting of one *liana cell*, several (ten, on average) Purkinje *cells*, which are controlled by liana cell via the *climbing fibers*. The Purkinje cells, in turn, send signals back to the liana cell (Fig. 1).

For simplicity, in this paper we consider the cerebellar loop with only one Purkinje cell. Each liana cell module predicts only one controlled variable or function. External signals for the module is delivered by $X(t)$. They are coded by mossy fibers and granule cells and constitute the input of the Purkinje cells in a form of the vector $G(t)$. The connection weights of granule cells on the Purkinje cell, $w(t)$, are the subject of permanent changes in interaction with the signals from liana cell. The Purkinje cell in this model is just an adder of the input signals, weighted with the synaptic weights vector. The output of the Purkinje cell, $p(t)$, should model the external input to the liana cell, $y(t)$. To provide such modeling of $y(t)$, it is necessary to tune correctly the vector of synaptic weights $w(t)$. The tuning is provided by the feedback loop, which includes the Purkinje cell in a process of learning. To make the prognosis possible, $X(t)$ should contain the information, which controls the value of $y(t)$. The liana cell integrates deflection of prognosis with the actual values of and excites, when the weighted integral of prognosis exceeds a threshold value. However,

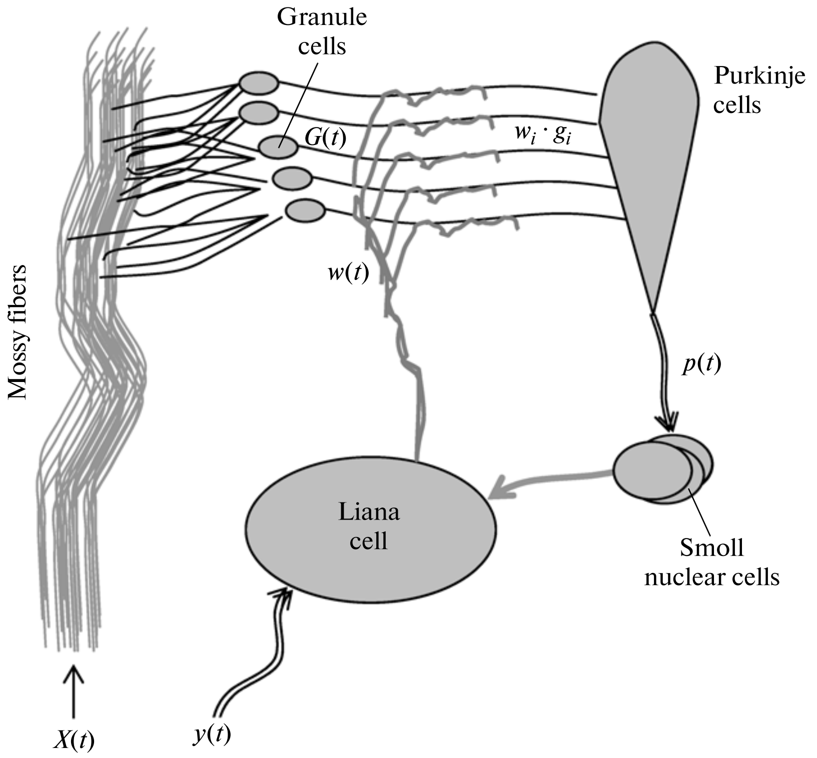

Fig. 1. The scheme of the basic prognosis module of the cerebellum.

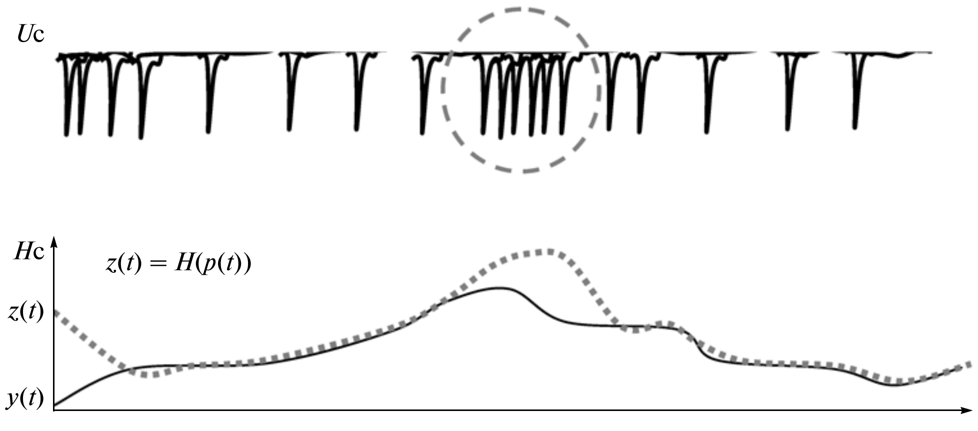

Fig. 2. The activity of liana cell depends on the balance between the feedback and feed forward signals at its input.

the negative deflection of integrated signal at the input of liana cell imposes its long silence, which also affects the connection vector $w(t)$ as explained below. The system demonstrates the dynamic stochasticity behavior. In the stationary state, when $y(t) \sim p(t)$, the liana cell fires with a stationary rate with stochastic deflections from the mean value. In the stationary state the weight vector, $w(t)$, is either constants, or slowly drifts, when there are many values of $w(t)$, which provide the correct values of $p(t)$. However, if the input signals $X(t)$, or $y(t)$, or both of them are not tuned to each other with the appropriate vector, $w(t)$, then the liana cell abruptly changes its activity pattern, and generates obviously non-random burst of impulses, or silent pause (Fig. 2). The vector $w(t)$ during such disturbances undergoes fast changes. The peaks and/or troughs of liana cell activity might thus signal to the rest of the organism that something unusual happens in these periods in the system, and that the extra control of behavior might be needed. Figure 2 illustrates, what has been stated above.

The equations, which describe the system were given in [5] and are described in detail below. It can be stated that the principles of the system tuning are very simple, however their effectiveness might be explained with the large number of dimensions, which the system can handle. In the live (human) cerebellum each liana cell controls (on average) 10 Purkinje cells. Each Purkinje cell is connected to up to 500000 parallel fibers (so, one liana cell is served by 1000000 granule cells). The total number of human liana cells is also about 1000000. Different liana cells work in parallel, but independent from each other. Their feedback loop networks might crisscross at small nuclear cells, which they might have in common. In principle with high dimensionality of the system the intersection of the loops might help, when the prognosis in different liana cells might have something common. In this case, the problem solution (the prognosis) might be in fact distributed over the system. Alternately, it might be considered that the computing resources of different liana cells are unified for solution of the multidimensional problem.

3. THE BLOCK-DIAGRAM OF THE MODULE FOR DATA PROGNOSIS

The above general description of the cerebellar module will be used for formulation of the mathematical model of the module for data prognosis. So, let us consider a dynamical system. Let $X = \{x_1, x_2, ..., x_n\}$ be the vector of the observable parameters of the system. Besides, let us consider that the system has a set of characteristics (functions), $Y = (y_1, y_2, ..., y_m)$, which also are observable and are somehow related to X. Let be all the observations synchronized, i.e. all the variables x_i and y_i are obtained in the same time moments:

$$\begin{gathered} t^1, x_1^1, x_2^1, \ldots, x_n^1, y_1^1, y_2^1, \ldots, y_m^1 \\ t^2, x_1^2, x_2^2, \ldots, x_n^2, y_1^2, y_2^2, \ldots, y_m^2, \\ \ldots, \\ t^k, x_1^k, x_2^k, \ldots, x_n^k, y_1^k, y_2^k, \ldots, y_m^k, \ldots. \end{gathered} \tag{1}$$

The dependence of the characteristics on parameters in general case might be represented as a function Φ:

$$Y^{k+1} = \Phi(X^k, X^{k-1}, X^{k-2}, \ldots, X^2, X^1).$$

The function Φ might be deterministic or random, but we will consider that its properties are unknown. Moreover, for real complex systems there could be principal difficulties for their formalization. The main method of analysis of such dependencies might be observation of the parameters and building up hypotheses related to their behavior. By hypotheses we mean computing of the expected values of the characteristics $Y = \{y_1, y_2, ..., y_i\}$. Let $Z^{k+1} = \{z_1^{k+1}, z_2^{k+1}, ..., z_m^{k+1}\}$ he the computed values of the appropriate characteristics at the time step $k + 1$. These values are computed on the basis of the whole history of observations:

$$Z^{k+1} = P(X^k, X^{k-1}, X^{k-2}, X^1, X^0).$$

If the system is predictable, the difference between the expected and the really observed values, $(\|y_i^k - z_i^k\|)$, starting with some k should be in some predetermined range. Of course, the difference will depend on many factors: on the properties of the system, on the observation time (if it is long enough), and, ultimately on the quality of the prognosis system.

Figure 3 shows the block-diagram of the prognosis module. Notations:

- $x_1, x_2, ..., x_n$—the observable parameters,
- $y_1, y_2, ..., y_m$—real values of the characteristics,
- $z_1, z_2, ..., z_m$—expected values of the characteristics,
- MF—mossy fibers, which deliver the coded values of X,
- GC—granule cells,
- PC$_i$—Purkinje cells,
- NC$_i$—nuclear cells,
- CFC$_i$—liana cells (climbing fiber cells).

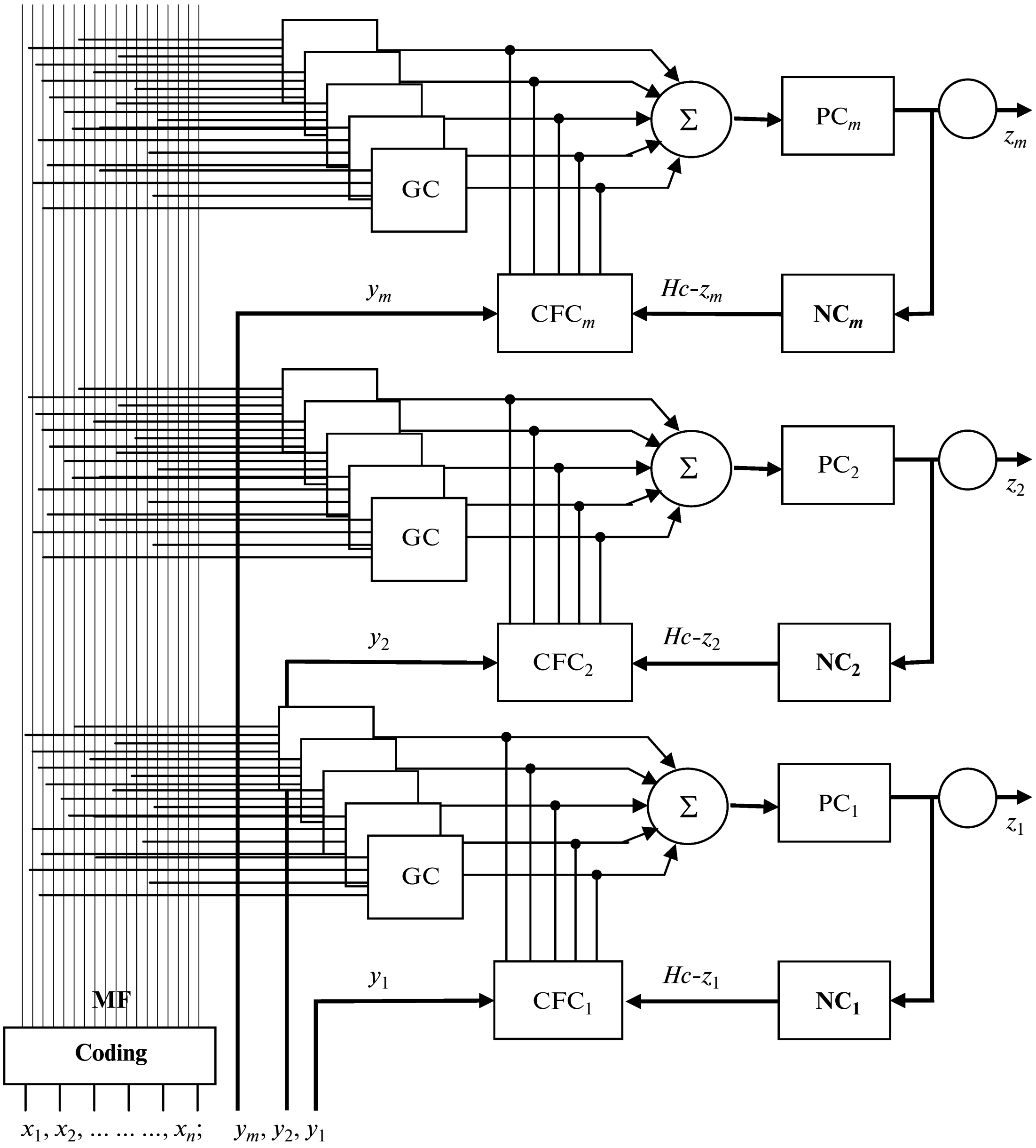

Fig. 3. Block-diagram of the prognosis module.

The box "Coding" transforms the values of the parameters x_1, x_2, ..., x_n, of the observable object into the signals of the mossy fibers. The number of mossy fibers might totalize to hundreds of thousands. The exact value of this number (the dimensionality of the vector X) is determined by the number of degrees of freedom of the plant, the granularity of variable quantization, and the needed precision of their sampling. The methods of coding are connected to the methods of prognosis and are substantial for the whole process. In principle, the problem needs a separate treatment. Some details, related to this subject are given in the Appendix. At this stage, we will constrain ourselves by one of the possible coding method—the so-called method of bell-shaped functions.

This method asserts to each parameter a number (actually, from 5 to 100) of mossy fibers. The sets of mossy fibers for different parameters should have zero intersection. We will consider that all values of the parameter are inside the segment $[x_{\min}, x_{\max}]$. And the coding of x is implemented in the following way. The segment $[x_{\min}, x_{\max}]$ is devided into n subsegments (n is the number of mossy fibers, allocated to the

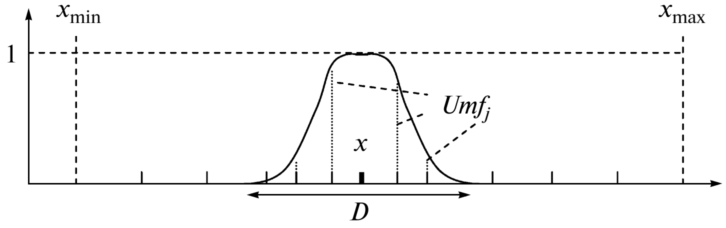

Fig. 4. Bell-shaped coding.

variable x) with the descrete values $\{Q_j\}$, where $j = 0, ..., n$ and $Q_0 = x_{\min}$, $Q_n = x_{\max}$. Values of x in the range $[x_{\min}, x_{\max}]$ are coded by the next vector of mossy fibers activity:

$\{Umf_j\}, j = 0, ..., n$. Umf_j takes values from the interval [0, 1] and is defined by the following expressions:

$$\{Umf_j\} = 0.5(1 + \cos(\omega(Q_j - x)/(x_{\max} - x_{\min}))),$$

if $-\pi \le (\omega(Q_j - x)/(x_{\max} - x_{\min})) \le \pi$.

$Umf_j = 0$ otherwise.

Here $x_{\min} \le x \le x_{\max}$, $\omega = 2\pi(x_{\max} - x_{\min})/D$, where D—is the range of the "fuzzification" of the parameter x (Fig. 4).

The set of mossy fibers of the model, **MF**, is defined as the sum of sets of mossy fibers for each parameters of the modeling object. The number of mossy fibers, $N_{\mathrm{MF}} = \Sigma_i n_i$. For example, if the number of the problem parameters is 1000 and the number of discretes for each variable is 100, then $N_{\mathrm{MF}} = 100000$.

GC—are blocks of granule cells. They implement the secondary coding of the input signals as follows. Let, say, the number of granule cells be 100000. For each granule cell, using the random number generator we select a number (3–8) of mossy fibers from **MF**. The output of a granule cell is determined by activity of mossy fibers, which are selected for that granule cell.

The model of the granule cell is a leaky integrator with the time constant T_{GC},

$$T_{\mathrm{GC}}\frac{dU_{\mathrm{GC}}}{dt} = -U_{\mathrm{GC}} + \sum_{j \in S} U_{\mathrm{MF}_j}. \tag{gc}$$

Here, in the right-hand side the signals, coming from the mossy fibers to the granule cells are summed up. The output of granule cells forwards to the Purkinje cells.

GC ⟶ PC link. Each Purkinje cell gets input from NGC granule cells and from one liana cell. Let $NGC = 50000$ (a half of the total number of granule cells). For each Purkinje cell its corresponding granule cells are chosen randomly (once and forever). The Purkinje cell equations are following:

$$T_{\mathrm{PC}}\frac{dU_{\mathrm{PC}}}{dt} = -U_{\mathrm{PC}} + \sum_i w_i U_{GC_i} \tag{pc}$$

$$V_{\mathrm{PC}} = SG(U_{\mathrm{PC}})$$

$$SG(x) = x, \text{ if } x > 0, \text{ and } SG(x) = 0, \text{ if } x \le 0.$$

NC—the blocks of nuclear cells. Their equations:

$$T_{\mathrm{NC}}\frac{dU_{\mathrm{NC}}}{dt} = -U_{\mathrm{NC}} + V_{\mathrm{PC}} \tag{nc}$$

$$V_{\mathrm{NC}} = SG(U_{\mathrm{NC}})$$

The nuclear cells also are leaky integrators and they implement a delay of signals from Purkinje cells back to liana cell.

CFC—the block of liana cells. There are two signals at the input of the liana cell: the variable, which is controlled in the process of observation and the output of the nuclear cells. The work of liana cell consists of permanent spiking with an average rate of 1 impulse per second. The equation of liana cell is following:

$$T_{\mathrm{CFC}}\frac{dU_{\mathrm{CFC}}}{dt} = -U_{\mathrm{CFC}} + V_{\mathrm{NC}} + y(t) \qquad \text{(cfc)}$$

for $U_{\mathrm{CFC}} < H_{\mathrm{C}}$.

Here $y(t)$ is the current value of the variable, which is controlled by the given liana cell.

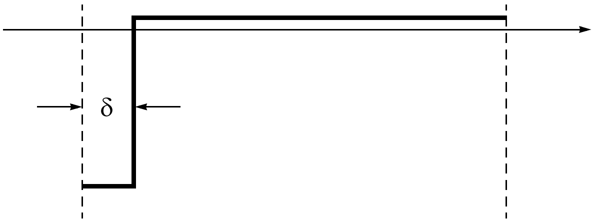

Fig. 5. The function $\chi(\tau)$.

The variables w_i in the equation (pc) are the changeable synaptic weights of the granule cells, the main "game-players" of the system. They are permanently changing, according to the following equations:

$$\frac{dw_i}{dt} = \varepsilon\chi(t - t^*(t))U_{\mathrm{GC}_i}. \qquad \text{(ww)}$$

The function $\chi(\tau)$ presents the factor, which determines the rate of synaptic weights changes depending from the time elapsed since the last excitation of the liana cell. This function time course is given at the Fig. 5.

$$\chi(\tau) = \begin{cases} -A, & \text{if } \tau \le \delta, \\ b, & \text{if } \tau > \delta. \end{cases} \qquad \text{(xx)}$$

The value $T_{\mathrm{equ}} = A(\delta/b)$ corresponds to the equilibrium (stationary) interval of liana cell firing. When liana cell fires with the interspike interval T_{equ}, then, "on average" synaptic weights from granule cells to the Purkinje cells, which are controlled by a given liana cell, are kept constant.

The liana cell gets excited at moments, t^*, when $U_{\mathrm{CFC}}(t)$ attains the level of H_{C} (Fig. 6). These moments define the moments of impulse generation by liana cells. After the t^*, the value of liana cell membrane potential is nullified: $U_{\mathrm{CFC}}(t^* + 0) = 0$.

The output of the prognosis module is defined by the formula $z(t) = H_{\mathrm{C}} - U_{\mathrm{PC}}(t)$.

At the block-diagram (Fig. 2) the number of outputs coincides with the number of the observable parameters. In practice, however, the number of values, which need prediction, might be substantially less, than the number of all observable parameters. Because of that, the number of Purkinje cells and of liana cells might be less than n. For tuning of the system properties we are starting with the problem, which predicts only one parameter with one Purkinje cell and one liana cell.

4. NUMERICAL SCHEME FOR DIFFERENTIAL EQUATIONS INTEGRATION

The previous section gives the mathematical model of the main block of the system, which implements the prognosis function. We give here the description of the integration method, which we have used for integration of differential equations.

The numerical scheme uses the following relations:

$$U^i_{\mathrm{GC}}(t + \mathrm{d}t) = U^i_{\mathrm{GC}}(t) + (\Sigma_j U^{ij}_{\mathrm{mf}}(t) - U^i_{\mathrm{GC}}(t))\mathrm{d}t/T_{\mathrm{GC}}, \qquad \text{(gc-n)}$$

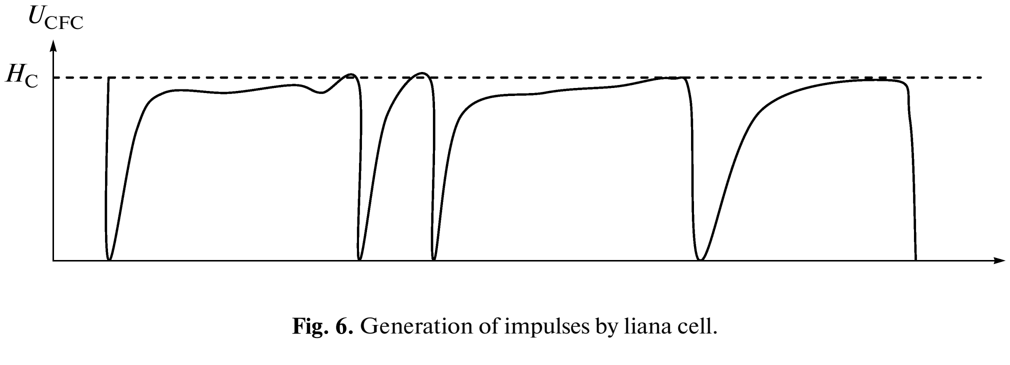

Fig. 6. Generation of impulses by liana cell.

for i = 1, ..., N_{GC};

$$w^i(t+\mathrm{d}t) = w^i(t) + \varepsilon\chi(\tau(t))U^i_{\mathrm{GC}}(t)\mathrm{d}t; \qquad \text{(ws-n)}$$

for $i = 1, ..., N_{\mathrm{GC}}$;

here ε—is a fixed constant, which needs tuning,

$\chi(\tau) = -A$ if $\tau < \delta$; otherwise $\chi(\tau) = b$; A, δ, b—constants;

$$ww^i(t) = \mathrm{SG}(ww^i(t)),$$

for $i = 1, ..., N_{\mathrm{GC}}$;

here $\mathrm{SG}(x) = x$, if $x > 0$, and $\mathrm{SG}(x) = 0$, if $x \le 0$;

$$I_{\mathrm{PC}}(t) = \Sigma_i U^i_{\mathrm{GC}}(t)ww^i(t); \qquad \text{(ipc-n)}$$

$$U_{\mathrm{PC}}(t+\mathrm{d}t) = U_{\mathrm{PC}}(t) + (I_{\mathrm{PC}}(t) - U_{\mathrm{PC}}(t))\mathrm{d}t/T_{\mathrm{PC}}; \qquad \text{(pc-n)}$$

$$V_{\mathrm{PC}}(t) = \mathrm{SG}(U_{\mathrm{PC}}(t)); \qquad \text{(vpc-n)}$$

$$U_{\mathrm{NC}}(t+\mathrm{d}t) = U_{\mathrm{NC}}(t) + (V_{\mathrm{PC}}(t) - U_{\mathrm{NC}}(t))\mathrm{d}t/T_{\mathrm{NC}}; \qquad \text{(nc-n)}$$

$$V_{\mathrm{NC}}(t) = \mathrm{SG}(U_{\mathrm{NC}}(t)); \qquad \text{(vnc-n)}$$

if $U_{\mathrm{CFC}}(t) < H_{\mathrm{C}}$:

$$U_{\mathrm{CFC}}(t+\mathrm{d}t) = U_{\mathrm{CFC}}(t) + (y_m(t) + V_{\mathrm{NC}}(t) - U_{\mathrm{CFC}}(t))\mathrm{d}t/T_{\mathrm{CFC}}, \qquad \text{(cnc-n)}$$

$$\tau(t+\mathrm{d}t) = \tau(t) + \mathrm{d}t;$$

otherwise:

$$U_{\mathrm{CFC}}(t+\mathrm{d}t) = 0,$$

$$\tau(t+\mathrm{d}t) = 0;$$

$$z_m(t) = H_{\mathrm{C}} - U_{\mathrm{PC}}.$$

Time constants in these equations are the tunable parameters of the system. We choose their initial values on a basis of the published data on the cerebellar neurons:

$$T_{\mathrm{GC}} = 0.03, \quad T_{\mathrm{PC}} = 0.03, \quad T_{\mathrm{NC}} = 0.1, \quad T_{\mathrm{CFC}} = 0.06.$$

Besides, we adopt the following values for the parameters of the system: $H_{\mathrm{C}} = 1.0$, $\varepsilon = 0.0005$, $A = 200$, $\delta = 0.005$, $b = 1$.

And, finally, the key constant for integration, the time discrete, d*t*. We choose it at ~0.005.

The main computational load in the described numerical scheme is due to a large number of equations (gc) and (ws). It is assumed that for the successful predictions N_{GC} should be in the order of tens and hundreds of thousands. This fact forces for searches of the circumstances, which could acselerate the calculations in this numerical scheme.

For this purpose we are trying to make use of relations between the sampling interval of the observable parameters X_i and the integration time discrete. If the former is substantially less, than the latter, the variables can be divided into the "fast" and "slow" ones.

Indeed, for Eq. (gc) the input function $\Sigma_j U^{ij}_{\mathrm{mf}}(t)$ is a stepwise with the time step, equal to the time step of the observed values. Thus, solutions of this equation will look like it is presented at Fig. 7.

At the intervals of constancy of the input variables, there are analytical expressions for the output of granule cells. The expression for the right-hand part of (ws) contains piece-wise constant function $\chi(\tau)$. Consequently, there could be chosen such time intervals, for which solutions for $w_i(t)$ also could be expressed analytically.

This means that numerical expressions (gc-n) and (ws-n) might be updated not at each time step of integration, but only in moments of changes of values of the input variables and of the function $\chi(t)$. On this basis the numerical scheme for the integration of the system of equations might be changed.

Let the last time moment, when the value of an input variable or the value of $\chi(t)$ has changed be *T*. We will use the following auxiliary variables:

$$Ak(t), \quad Bk(t), \quad Ck(t), \quad Dk(t), \quad Ek(t).$$

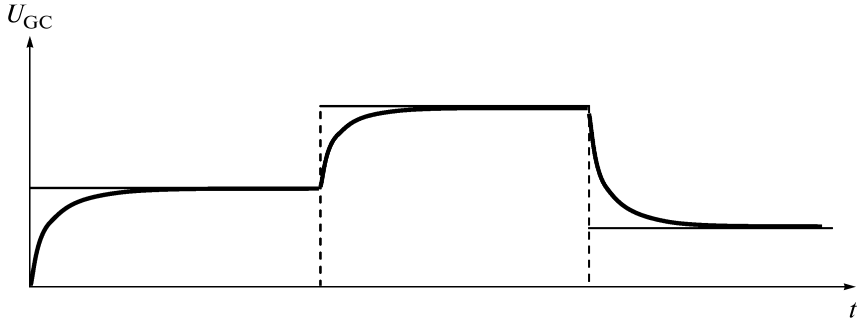

Fig. 7. The signal at the output of granule cells.

$$Ak(0) = 0, \quad Bk(0) = 0, \quad Ck(0) = 0, \quad Dk(0) = 0, \quad Ek(0) = 0.$$

If $t = 0$ or $t = T$, then

$$\mathrm{I}_{\mathrm{PC}}(t) = \Sigma_i U^i_{\mathrm{GC}}(t) ww^i(t); \tag{ipc-q}$$

otherwise

$$\begin{aligned}\mathrm{I}_{\mathrm{PC}}(t) = {} & Ak(T) + Bk(T)\exp((T-t)/T_{\mathrm{GC}}) + Ck(T)(t-T) + Dk(T)((t-T)\exp((T-t)/T_{\mathrm{GC}}) \\ & + (1 - \exp((T-t)/T_{\mathrm{GC}})T_{\mathrm{GC}}) + Ek(T)\exp((T-t)/T_{\mathrm{GC}})(1 - \exp((T-t)/T_{\mathrm{GC}})T_{\mathrm{GC}};\end{aligned} \tag{ipc-qm}$$

$$U_{\mathrm{PC}}(t + \mathrm{d}t) = U_{\mathrm{PC}}(t) + (I_{\mathrm{PC}}(t) - U_{\mathrm{PC}}(t))\mathrm{d}t/T_{\mathrm{PC}}; \tag{pc-q}$$

$$V_{\mathrm{PC}}(t) = \mathrm{SG}(U_{\mathrm{PC}}(t)); \tag{vpc-q}$$

$$U_{\mathrm{NC}}(t + \mathrm{d}t) = U_{\mathrm{NC}}(t) + (V_{\mathrm{PC}}(t) - U_{NC}(t))\mathrm{d}t/T_{\mathrm{NC}}; \tag{nc-q}$$

$$V_{\mathrm{NC}}(t) = \mathrm{SG}(U_{\mathrm{NC}}(t)); \tag{vnc-q}$$

$$\text{if } U_{\mathrm{CFC}}(t) < H_{\mathrm{C}}, \text{ then} \tag{cfc-q}$$

$$U_{\mathrm{CFC}}(t + \mathrm{d}t) = U_{\mathrm{CFC}}(t) + (y_m(t) + V_{\mathrm{NC}}(t) - U_{\mathrm{CFC}}(t))\mathrm{d}t/T_{\mathrm{CFC}},$$

$$\tau(t + \mathrm{d}t) = \tau(t) + \mathrm{d}t;$$

otherwise

$$U_{\mathrm{CFC}}(t + \mathrm{d}t) = 0,$$

$$\tau(t + \mathrm{d}t) = 0;$$

$$\text{step} = 1;$$

$$\text{if } \chi(\tau(t + \mathrm{d}t)) > \chi(\tau(t))$$

$$\text{step} = 1;$$

if $t + \mathrm{d}t = t^k$, i.e. $t + \mathrm{d}t$ is the moment of fixation of the observed parameters,

step = 1; if step = 1

$$U^i_{\mathrm{GC}}(t) = \Sigma_j V^{ij}_{\mathrm{mf}}(T+0) + (U^i_{\mathrm{GC}}(T))\exp((T-t)/T_{\mathrm{GC}}), \tag{qc-q}$$

$$\text{for } i = 1, \ldots, N_{\mathrm{GC}};$$

$$\begin{aligned}w^i(t) = {} & w^i(T) + \varepsilon\chi(T+0)\Sigma_j V^{ij}_{\mathrm{mf}}(T+0)(t-T) + \varepsilon\chi(T+0)(U^i_{\mathrm{GC}}(T) \\ & - \Sigma_j V^{ij}_{\mathrm{mf}}(T+0)(1 - \exp((T-t)/T_{\mathrm{GC}}))T_{\mathrm{GC}};\end{aligned} \tag{ws-q}$$

for $i = 1, \ldots, N_{\mathrm{GC}}$;

$$Ak(t) = \Sigma_i w^i(t) U^i_{\mathrm{GC}}(t);$$

$$Bk(t) = \Sigma_i w^i(t)(U^i_{\mathrm{GC}}(t) - \Sigma_j V^{ij}_{\mathrm{mf}}(t+0));$$

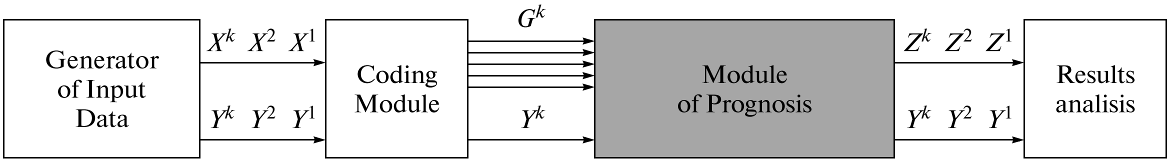

Fig. 8. Scheme of the test bench.

$$Ck(t) = \varepsilon\chi(t+0)\Sigma_i U^i_{\mathrm{GC}}(t) U^i_{\mathrm{GC}}(t);$$

$$Dk(t) = \varepsilon\chi(t+0)\Sigma_i U^i_{\mathrm{GC}}(t)(U^i_{\mathrm{GC}}(t) - \Sigma_j V^{ij}_{\mathrm{mf}}(t+0));$$

$$Ek(t) = \varepsilon\chi(t+0)\Sigma_i (U^i_{\mathrm{GC}}(t) - \Sigma_j V^{ij}_{\mathrm{mf}}(t+0))(U^i_{\mathrm{GC}}(t) - \Sigma_j V^{ij}_{\mathrm{mf}}(t+0));$$

$$T = t + \mathrm{d}t;$$

$$z_m(t) = H_{\mathrm{C}} - U_{\mathrm{PC}}.$$

Thus altered computational scheme enables to avoid multiple calculations and summing of signals at the Purkinje cell, which constitute the bulk of computational load for system simulations. The proposed scheme assumes the calculations in fact being performed only in the moments of changes in the input parameters and in the moments of liana cell excitations (cf. expression (cfc-q)). This scheme enables substantial acceleration of the computations and its effective parallelization.

5. ANALYSIS OF THE MODEL

On the basis of the described computational scheme, we have implemented the experimental test bench for analysis and tuning the parameters range of the model of the prognosis module (Fig. 8).

The generator of the input data simulates behavior of the systems to study as a sequence of vectors of parameters $\{X^k\}$ and characteristics $\{Y^k\}$. The particular way of generation is connected with concrete problems. It could be periodic or random sequence, alternately, the data can be collected and stored from the real dynamic systems. Then, the data are coded, according to the described above rules of coding at mossy fibers and are transferred to the module of prognosis. In the prognosis module, at each time step, the prognosis, Z, is calculated, using the accumulated values of the granule cells synaptic weights. The prognosis is compared to the real values of the characteristics, Y, and the synaptic weights are updated. For the analysis of the prognosis results, the system input and output data are visualized. This test bench has been used for model tests and example practical problems. Here are the problems, which we have tested in computational experiments:

—to try simple examples of dynamical systems for testing the abilities of the model, i.e. to see, if it is capable of learning to short-time prognosis of behavior of dynamical systems;

—to pick up the values of the parameters of the model, for which the prognosis is most effective;

—to evaluate time, needed for the model to deal with problems of high dimensions.

The test was organized in the following way. First, we tuned the bench parameters, in particular, the dimensions of input and output, the number of granule cell inputs to Purkinje cells, time constants for all the modeled cells, the parameters of synaptic modifications, and threshold of the liana cell. Besides, for each of the input variables, we choose the range of the variables and the number intervals in the range (i.e. the number of mossy fibers, which were selected for those variables.

For each problem we have several sessions of computations. First sessions were connected with “learning of the system”. Then we performed the evaluation of the results of the learning. When the results of the latter were not satisfactory, we continued the learning process. If the calculations took too much time, we modified the values of the parameters for the problem (the number of inputs, constants of the neural model of parameters of the coding). The results analysis was performed by comparison of plots of the really observed data with the plot of the results of the prognosis.

The session of calculation in the learning mode was defined by the total modeling time in the session, by the duration period, which was repeated for learning, by the time step in changes of the input variables, the time step for the output data and the time discrete for integration. In general, for “good learning”, the sequence of input signals should be repeated many times, the more, the better.

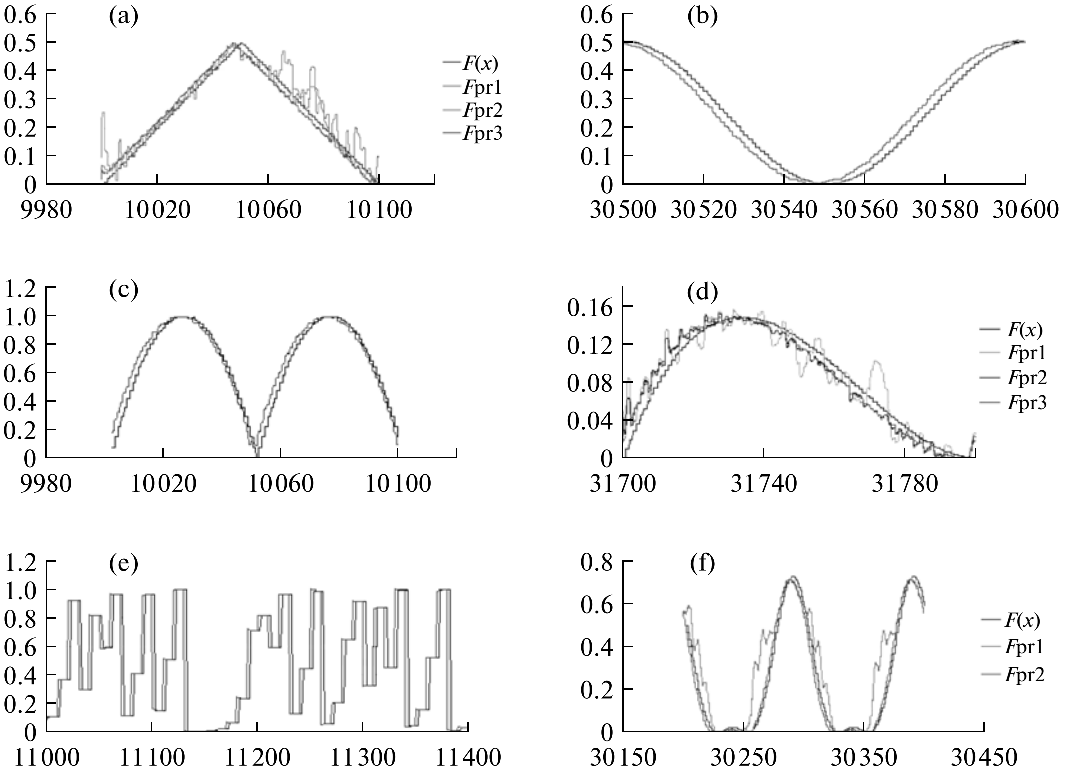

Fig. 9. Results of the tests of the prognosis module.

For the beginning, for the analysis of the learning abilities of the system we used the simplest mathematical dependencies, like $Y = F(X)$. Figure 9 shows the results of some test problems.

At the Fig. 9:

(a) $Y(t + \Delta) = F(X) = 0.5X(t)$,

where $X(t) = t/50$, if $t \le 50$, and

$X(t) = t/50$, if $t > 50$.

Fpr 1 – $N_{\text{GC}} = 1000$, $T_{\text{Learn}} = 10000$,

Fpr 2 – $N_{\text{GC}} = 1500$, $T_{\text{Learn}} = 10000$,

Fpr 3 – $N_{\text{GC}} = 2000$, $T_{\text{Learn}} = 10000$.

(b) $Y(t + \Delta) = 0.25X(t)$,

where $X(t) = \cos(t\pi/50) + 1.0$.

$N_{\text{GC}} = 2000$, $T_{\text{Learn}} = 30000$.

(c) $Y(t + \Delta) = 4X(t)(1 - X(t))$,

where $X(t) = t/50$, if $t \le 50$ and

$X(t) = (50 - t)/50$, if $t > 50$.

$N_{\text{GC}} = 1000$, $T_{\text{Learn}} = 10000$.

(d) $Y(t + \Delta) = X(t)(X(t) - 1)(X(t) - 1)$,

where $X(t) = t/100$, $0 < X < 1$,

Fpr 1 – $N_{\text{GC}} = 1000$, $T_{\text{Learn}} = 10000$,

Fpr 2 – $N_{\text{GC}} = 1500$, $T_{\text{Learn}} = 11000$,

Fpr 3 – $N_{\text{GC}} = 2000$, $T_{\text{Learn}} = 31700$.

(e) $Y(t + \Delta) = X(t)$

where $X(0) = 0.1$, and

$X(t) = 4X(t - \delta)(1 - X(t - \delta))$, for $t > 0$, where δ is the time discrete of input changes.

The latter is the classical “dynamic chaos” problem, which is hard to predict by standard methods. In our case, a significant role played influence of the border conditions for coding of the variable X in the regions, close to $X = 0$ and $X = 1$. To avoid these complexities we have extended the coding interval to the segment [–0.1, 1.1].

f) $Y(t + \Delta) = 0.25X_1(t)X_2(t)$,

where $X_1(t) = \cos(t\pi/50 + \pi/2) + 1$,

$X_2(t) = \cos(t\pi/50) + 1$.

At Fig. 9 the thin solid line shows the original functional dependencies, thick line shows plots for the successful prognosis and pale line shows deficient prognosis results (due to the insufficient learning time or small number of the granule cells, N_{GC}). We have failed to find parameters for satisfactory prognosis of the test problem (f). However, the problem could be solved in the slightly modified form:

f*) $Y(t + \Delta) = 0.25X_1(t)X_2(t)$,

$X_1(t) = \cos(t\pi/50 + \pi/2) + 1$,

$X_2(t) = \cos(t\pi/50) + 1$,

$X_3(t) = t \times 50$.

We have introduced here a subsidiary variable $X_3(t)$, which seemingly had no relations to the function $Y(t + \Delta) = X_1(t)X_2(t)$. But that helped us to solve the problem. The computational results for the such redundant task are notable (block Fpr 2 at Fig. 9). They show that there could be no “excessive” information in searches for solution of prognosis problems, even in cases, when at first glance the additional information seems to be irrelevant to the task. That information might indirectly help the prognosis system to identify the state of whole plant.

The tests have demonstrated that the exact values of the parameters T_{gc}, T_{pc}, T_{nc}, T_{cpc} are not very critical for the results. These values must be just inside the reasonable ranges, which are determined by the values of the time steps in integration, the time step of input parameters changes, etc. The most critical parameters are the number Purkinje cell inputs and the parameters of synaptic modification H_C, ε, A, δ, b.

The modeling of the test problems has shown that the modeling performance rate is determined, first of all by the pair of main parameters: (1) the time step of input parameters changes, and (2) the number of granule cells, N_{GC}. For example, for the time step of input parameters changes of 1 second, and the number of granule cells, $N_{GC} = 10000$, the ratio of the real time to the modeling time was about 100 : 1, practically not depending on the concrete modeling task.

We have used also the modeling test bench for the real world problems of prognosis. One of them was the prognosis of the flows distribution in the transport network.

The simplified scheme of this problem looks as described below. Let the communication network look like the graph of Fig. 10.

The flows of material substances enter the input nodes. That can be flows of transporting vehicles, of people or of finances. In the inner nodes the flows are redistributed and continue movement from node to node, coming finally to the output nodes. The flow redistribution in the inner nodes obey certain rules and conditions, determined by the concrete state of the network parameters, such as load of communication routes, time of day and transportation control signals. It is important to have the prognosis of the state of the network. In particular, we should be able to predict the flows at the output of the network, as dependent on the input flows, measurements of the inner flows in certain control checkpoints, the load capacity of the routes, and other parameters. We have analyzed several versions of the transportation networks. Figure 11 shows results of prognosis of several output flows with random input flows and fixed schemes of flows distribution at the inner nodes.

Thick white lines show the real flows, while black lines show prognosis of the flow after long learning sessions. The coincidence of prognosis and real data is reasonably good. The thin white line shows the activity of the liana cell.

Figure 12 shows results of solution of one more prognosis tasks. These are the data on electrical energy consumption of a region, depending on the time of the day and date. In principle, the consumption rate depends on the time and other factors, such as the current weather, state of the local economics, etc. However, in our case the set of available data was rather restricted. We actually used real data on energy consumption in one Russian town for two years; the average daily temperature was also available. The data was separated into the learning and control pools. In the learning mode the first pool of data has been presented to the system many times. Afterwards, the data of the second pool were fed to the system input. Figure 12a shows the initial learning phase of the computations. Black line presents the learning data and gray line—the predictions of the network. Figure 12b demonstrates performance of the system after many learning

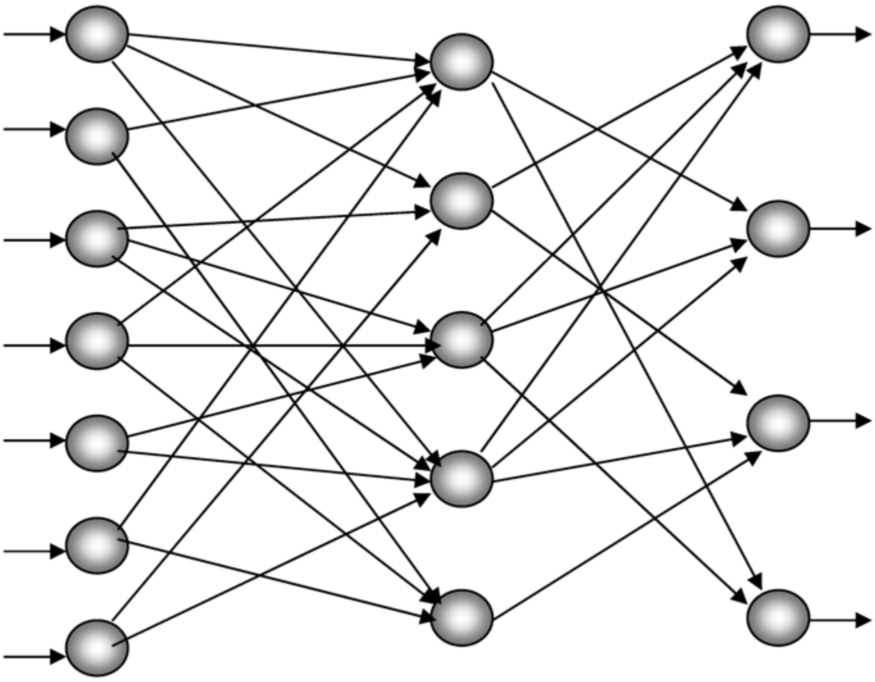

Fig. 10. The scheme of the transport network.

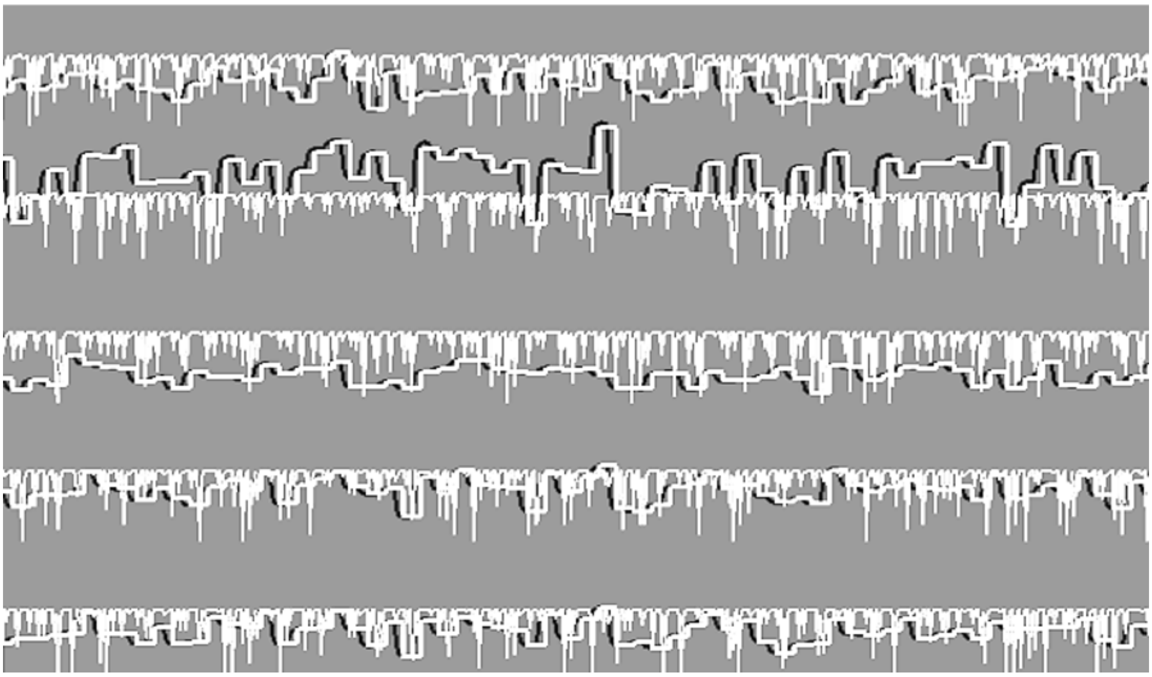

Fig. 11. Results of prognosis of flows in the transportation network.

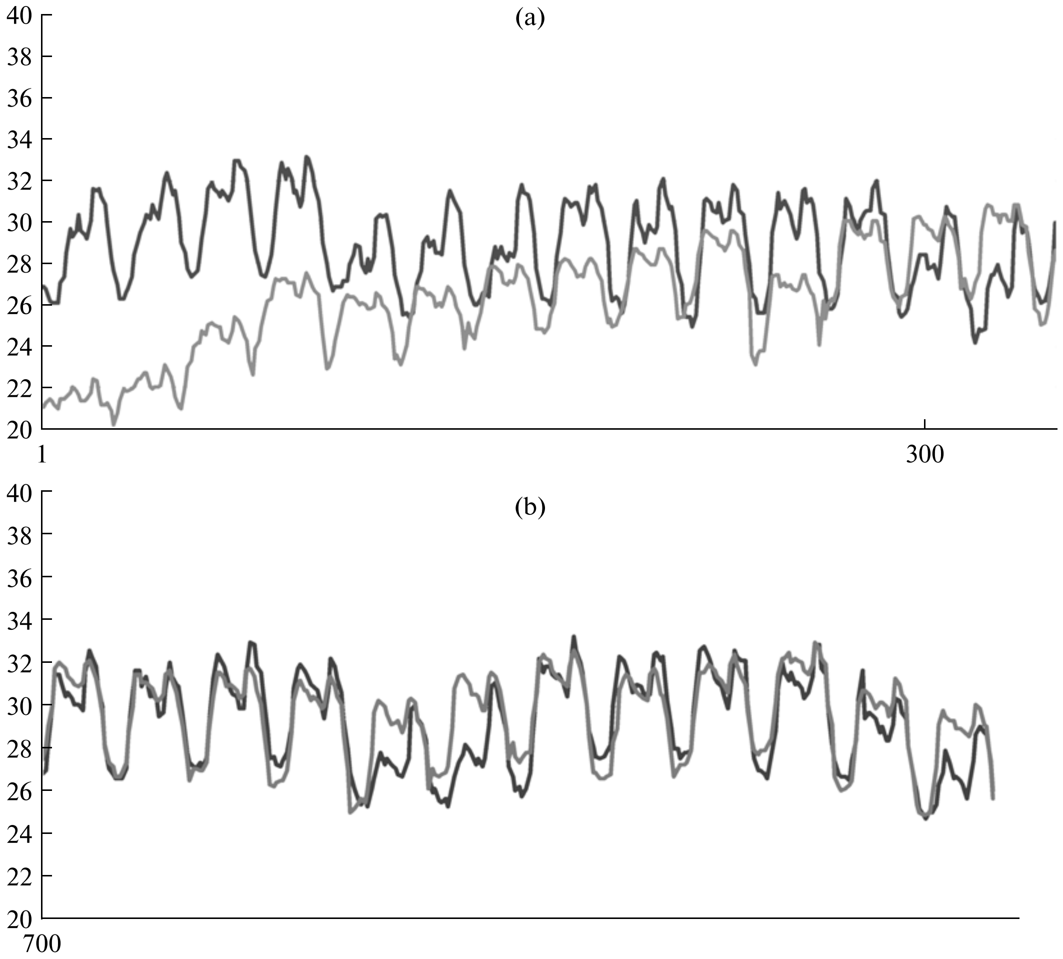

Fig. 12. Prognosis of the electric energy consumption.

cycles. The deflection of prognosis from the original data is 5%, on average, and 8% at the extreme discordance.

6. CONCLUSIONS

We have elaborated mathematical model of the module of data prognosis, based on the cerebellar modeling. In fact, we have implemented the numerical scheme of data prognosis, based on the model of cerebellum, which can be applied in real time for the high-dimensional problems including the neural-type coding of the input information. The "modeling bench" has enabled detailed studies of the system for several classes of signals. Our test computations have demonstrated that the cerebellar-based module of data prognosis might an effective and practical tool for multidimensional predictions.

APPENDIX

CONTINUOS DATA REPRESENTATION IN MULTI-NEURONAL SYSTEMS

Since the beginning of the XX-th century it is known (Adrian, 1923, cited in [9]) that the value (intensity) of external signals is coded in neural system by the frequency of impulses. However, one neural element can code signals in a limited range. In cases, when a wider range is needed (say, for control purposes), several neural elements, working in parallel, might be used. The work of such groups of neural elements has been analyzed in the theory of simple neural pools. Work of simple neural pools can be complicated by the synchronicity of neuronal firing. On the other hand, switching between synchronous and asynchronous states of neuronal pools might implement a specific neuronal computational function (MAE—"many are equal" of John Hopfield). To avoid some of synchronization problems multineuronal systems might need certain non-zero “tonic” value of random noise in the system. This property of complex control systems is known as stochastic resonance.

Another way of representation of continuos values has been first implemented in artificial neuronal systems by Teuvo Kohonen (described in [9]). The method is known as Kohonen’s self-organizing feature maps. The idea of this representation is to make an automatic mapping of one- two- or three-dimensional sets of sensory signals onto corresponding one- two- or three-dimensional chains of neurons in the neural network. A possibility to apply Kohonen’s feature maps to the real neural systems is doubtful due to the fact that there are no few-dimensional neural chains in real neural systems.

The case might be amended if one would consider mapping not to the single neurons, but to the spatially local compactly activated neural groups, which can be stable under certain suggestions on the interneuronal connections geometry (a “Mexican hat” connections rule), as has been demonstrated by Sunichi Amari (references in [9]). However, this type of neural coding is extremely wasteful in the number of neurons.

An effective possibility of coding of continuous signals in multineuronal systems has been first presented in paper [7], which have used the Hebb-Hopfield method for neuronal interconnections matrix formation [11] for quasi-continuous sets of neural states. Here, for the first time has been described the neural network, the set of stable states of which (the set of attractors) is a “continuous” one-dimensional set of states of the type “snake-in-a-box”. The number of different states in such continuous attractor is about $2^{0.1N}$, i.e. it grows exponentially with the neural network size. The mapping of the range of changes of a continuous one-dimensional variable onto the continuous one-dimensional attractor of a neural network can be regarded as the neural representation of that variable. The continuous attractors in neural networks might be not only one-dimensional, but might be two-dimensional, three-dimensional, or have any small (compared to the number of neurons in the neural network, N) number of dimensions The total number of effectively different points in all the continuous attractors of one neural network is bound by the pointed above value of $2^{0.1N}$. The continuos attractors of neural networks, called also bump attractors, are the most common representation of continuos variables in mutineuronal live systems [12]. However, it should be also beard in mind that variable values read-out from one neural network to another, can be modulated by general brain oscillations in alpha and theta ranges [10, 22].

ACKNOWLEDGMENTS

The work was sponsored by the Russian Foundation of Basic Research grant no. 10-07-00206, and the grant of Presidium of the Russian Academy of Sciences (The Program “Basic Sciences for Medicine”).

REFERENCES

1. Baestaens, D.E., Van den Bergh, W.M., and Wood, D., *Neural Network. Solution for Trading in Financial Markets*, Pitman Publishing, 1995.

1 2. Barto, A.G., Fag, A.H., Sitkoff, N., and Houk, J.C., A Cerebellar Model of Timing and Prediction in the Control of Reaching, *Neural Computation*, 1999, vol. 11, pp. 565–594.

3. Cherkassky, V. and Muller, F., *Learning from Data: Concepts, Theory, and Methods*, New York: Wiley, 1998, p. 600.

4. Courchesne, E. and Allen, G., Prediction and Preparation, Fundamental Functions of the Cerebellum, by Eric and Greg Allen, *Learning Memory*, NY: Cold Spring Harbor Laboratory Press, 1997, vol. 4, no. 1, pp. 1–35.

5. Dunin-Barkowski, W.L., Analysis of Output of All Purkinje Cells Controlled by One Climbing Fiber Cell, *Neurocomputing*, 2002, vols. 44–46, pp. 391–400.

6. Dunin-Barkowski, W.L., Theory of the Cerebellum, in *Lectures on Neuroinformatics*, Moscow: MEPHI, 2010, pp. 14–48 [in Russian].

7. Dunin-Barkowski, W.L. and Osovets, N.B., Hebb-Hopfield Neural Networks Based on One-Dimensional Sets of Neurons States, *Neural Processing Lett.*, 1995, vol. 2, no. 4, pp. 28–31.

8. Gorban, A.N., Dunin-Barkowski, W.L., Kirdin, A.N., Mirkes, E.M., Novokhodko, A.Yu., Rossiev, D.A., Terekhov, S.A., Senashova, M.Yu., and Tsargorodtsev, V.G., *Neuroinformatics*, Novosibirsk: Nauka, 1998, p. 296 [in Russian].

9. Haykin, S., *Neural Networks and Learning Machines*, 3rd Ed., NY: Prentice Hall/Pearson, 2008, p. 850.

10. Hoffmann, L.C. and Berry, S.D., Cerebellar Theta Oscillations are Synchronized During Hippocampal Theta-Contingent Trace Conditioning, *Proc. Natl. Acad. Sci.*, 2009, vol. 106, pp. 21371–21376.

11. Hopfield, J.J., Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Acad. Sci. USA*, 1982, vol. 79, pp. 2554–2558.

12. Hopfield, J.J., Neurodynamics of Mental Exploration, *Proc. Nat. Acad. Sci. USA*, 2010, vol. 107, pp. 1648–1653.

13. Hopfield, J.J. and Brody, C.D., What is Movement? Transient Synchrony as a Collective Mechanism for Spatiotemporal Integration, *Proc. Nat. Acad. Sci. USA*, 2001, vol. 98, pp. 1282–1287.

14. Ito, M., Cerebellar Long-Term Depression: Characterization, Signal Transduction, and Functional Roles, *Physiol. Rev.*, 2001, vol. 81, no. 3, pp. 1143–1195.

15. Ito, M., Control of Mental Activities by Internal Models in the Cerebellum, *Nature Rev. Neurosci.*, 2008, vol. 9, pp. 304–313.

16. Kawato, M., Imamizu, H., Miyauchi, S., Tamada, T., Sasaki, Y., Takino, R., Putz, B., and Yoshioka, T., Human Cerebellar Activity Reflecting an Acquired Internal Model of a New Tool, *Nature*, 2000, vol. 403, pp. 192–195.

17. Lutherer, L.O., Everse, S.J., and Williams, J.L., Neurons of the Rostral Fastigial Nucleus are Responsive to Cardiovascular and Respiratory Challenges, *J. Autonom. Nerv. Syst.*, 1989, vol. 27, pp. 101–112.

18. Mauk, M. and Donegan, N.H., A Model of Pavlovian Eyelid Conditioning Based on the Synaptic Organization of the Cerebellum, *Learning Memory*, 1997, vol. 3, pp. 130–158.

19. Panchin, Y.V., Cellular Mechanism for the Temperature Sensitive Spatial Orientation in Clione, *Neuroreport*, 1997, vol. 8, no. 15, pp. 3345–3348.

20. Spoelstra, J., Schweighofer, N., and Arbib, M., Cerebellar Learning of Accurate Predictive Control for Fast-reaching Movements, *Biol. Cybernetics*, 2000, vol. 82, pp. 321–333.

21. Widiputra, H., Pears, R., Serguleva, A., and Kasabov, N., Dynamic Interaction Networks in Modelling and Predicting the Behaviour of Multiple Interactive Stock Markets, Intelligent Systems in Accounting, *Finance Management*, 2009, vol. 16, pp. 189–205.

22. Wikgren, L., Nokia, M.S., and Penttonen, M., Hippocampo–Cerebellar Theta Band Phase Synchrony in Rabbits, *Neurosci.*, 2010, vol. 165, no. 4, pp. 1538–1545.

SPELL: 1. ok