Project “Animat Brain”: Designing the Animat Control System on the Base of the Functional Systems Theory

Vladimir G. Red’ko[1], Oleg P. Mosalov[2], Danil V. Prokhorov[3], Konstantin V. Anokhin[4], Mikhail S. Burtsev[5], Alexander I. Manolov[2], Valentin A. Nepomnyashchikh[6]

[1] Institute of Optical Neural Technologies, Russian Academy of Sciences,
Vavilova Str., 44/2, Moscow, 119333, Russia
vgredko@gmail.com

[2] Moscow Institute of Physics and Technologies,
Institutsky per., 9, Dolgoprudny, Moscow region, 141700, Russia
olegmos_@mail.ru, paraslonic@yandex.ru

[3] Toyota Technical Center in Ann Arbor, MI, USA
dvprokhorov@gmail.com

[4] P.K. Anokhin Research and Development Institute of Normal Physiology, Russian Academy of Medical Sciences, Mokhovaya Str., 11/4, Moscow, 103009, Russia
k_anokhin@yahoo.com

[5] M.V. Keldysh Institute for Applied Mathematics, Russian Academy of Sciences,
Miusskaya Sq., 4, Moscow, 125047, Russia
mbur@narod.ru

[6] I.D. Papanin Institute for Biology of Inland Waters, Russian Academy of Sciences,
Borok, Yaroslavskaja obl., 152742, Russia
nepom@ibiw.yaroslavl.ru

Abstract. The paper describes our attempts to design an animat control system (the Animat Brain) on the basis of the Petr K. Anokhin's theory of functional systems. This biological theory proposes general schemes and regulatory principles of animal purposeful adaptive behavior. The Animat Brain is aimed at controlling adaptive behavior of an animat that has several natural needs (energy replenishment, safety, reproduction). The animat control system consists of a set of hierarchically linked functional systems and enables predictive and purposeful behavior. The paper describes: 1) the first version of the Animat Brain that is based on adaptive critic designs (ACDs), 2) results of computer simulations of these ACDs, 3) the second version of the Animat Brain, in which every functional system consists of two neural networks: controller and model. The controllers are intended to form chains of actions and the models are intended to predict futures events.

1 Introduction

In this paper we outline our attempts to design an animat control system (the Animat Brain) on the basis of the biological theory of functional systems. This theory was proposed and developed in the period 1930-1970s by Russian neurophysiologist

Petr K. Anokhin [1] and provides general schemes and regulatory principles of purposeful adaptive behavior in biological organisms.

In the first version of the Animat Brain, we use the reinforcement learning approach [2], namely, we propose animat control system on the base of adaptive critic designs (ACDs) [3-5]. ACDs are intelligent schemes that can be used as control systems of self-learning agents.

In order to analyze the possibilities to use an ACD as a functional system (FS), we simulated evolution of population of agents that have the ACD based control system. We revealed several interesting features of the ACD that are due to interaction between learning and evolution in agent populations. In particular, we revealed that ACD operation can be evolutionary unstable. In order to overcome this problem, we propose the second (more biologically plausible) version of the Animat Brain architecture, in which a FS consists of two neural network (NN) blocks: the model and the controller. The model NN is dedicated to predict future states; the controller NN is intended to form animat actions.

The paper is organized as follows. Section 2 outlines Anokhin's theory of functional systems. The first version of the Animat Brain architecture that is based on ACD is described shortly in Section 3. Section 4 is an overview of simulations on evolving population of ACD based agents. Section 5 describes the second version of the Animat Brain. Section 6 concludes the paper.

2 Anokhin's Theory of Functional Systems

The project "Animat Brain" is based on neurophysiological theory of functional systems [1]. Functional systems were put forward by Petr K. Anokhin in the 1930s as an alternative to the predominant concept of reflexes. Contrary to reflexes, the endpoints of functional systems are not actions themselves but adaptive results of these actions. Initiation of each behavior is preceded by the stage of afferent synthesis (Fig. 1). It involves integration of neural information from a) dominant motivation (e.g., hunger), b) environment (including contextual and conditioned stimuli), and c) memory (including innate knowledge and individual experience). The afferent synthesis ends with decision making, which results in selection of a particular action.

A specific neural module, acceptor of the action result, is being formed before the action itself. The acceptor stores an anticipatory model of the required result of a goal-directed action. Such model is based on a distributed neural assembly that includes various parameters (i.e., proprioreceptive, visual, auditory, olfactory) of the expected result. It should be noted that performance of the acceptor of the action result is similar to sensorial anticipations in modern theories of anticipatory adaptive behavior [6]. Execution of every action is accompanied by a backward afferentation. If parameters of the actual result are different from the predicted parameters stored in the acceptor of action result, a new afferent synthesis is initiated. In this case, all operations of the functional system are repeated until the final desired result is achieved.

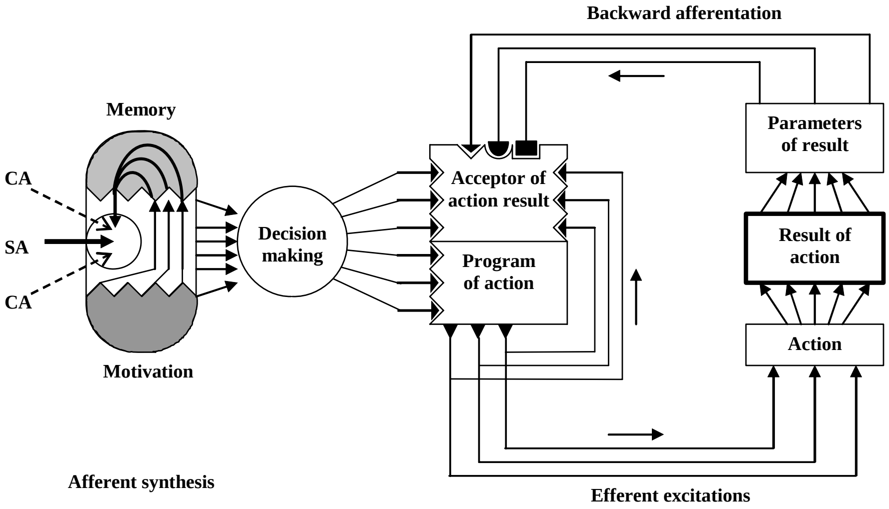

Fig. 1. General architecture of a functional system. SA is starting afferentation, CA is contextual afferentation. Operation of the functional system includes: 1) preparation for decision making (afferent synthesis), 2) decision making (selection of an action), 3) prognosis of the action result (generation of acceptor of action result), 4) backward afferentation (comparison between the result of action and the prognosis)

3 First Version of Animat Brain Architecture: ACD Based Hierarchical Control System

This section includes a short description of the first version of the Animat Brain architecture [7] that is based on a simple scheme of ACD. We suppose that animat control system is a set of hierarchically organized formal FSs; any FS is an ACD. The formal FS includes the following important features of its biological prototype: a) prognosis of the action result, b) decision making, c) comparison of the prognosis and the result, and d) correction of prognosis mechanism.

The considered ACD consists of two NNs: model NN and critic NN and serves to select one from several actions. For example, for movement control the actions can be move forward, turn left, turn right. The animat in any moment *t* should select one of these actions, $t = 0,1,2,\ldots$

The ACD operation is intended to maximize utility function $U(t)$ [2]:

$$U(t) = \sum_{j=0}^{\infty} \gamma^j r(t+j), \qquad t = 1,2,... \qquad (\mathbf{1})$$

where $r(t)$ is a particular reinforcement (reward, $r(t) > 0$, or punishment, $r(t) < 0$) obtained by the ACD at the moment *t*, and γ is the discount factor ($0 < \gamma < 1$).

The role of the model is to make predictions of the next state for all possible actions a_i, $i = 1,2,\dots,n_a$.

The critic is intended to estimate the state value function $V(\mathbf{S})$ [2] for the current state $\mathbf{S}(t)$ and next state predictions $\mathbf{S}^{\mathbf{pr}}{}_i(t+1)$ for all possible actions. The values V are estimates of the utility function U. Actions are selected on the base of these estimations by means of ε-greedy rule [2] that enables preference of actions corresponding to large values $V(\mathbf{S}^{\mathbf{pr}}{}_i(t+1))$.

The model is learnt by means of back-propagation algorithm [8]; the critic is learnt by means of temporal-difference method [2].

More detailed description of the ACD operation is given in the next section for the particular type of the adaptive critic.

The animat control system is a hierarchically linked set of FSs. The highest level of the hierarchy corresponds to the main animat needs such as energy replenishment, safety, reproduction. Lower levels correspond to tactical goals and sub-goals of behavior. It is supposed that any time moment only one FS is active.

Some FS actions are commands for effectors (real actions); another type of actions is control commands. Control commands are intended to transfer activity from one FS to another; these commands can be delivered from high levels to low levels and returned back.

The more detailed description of ACD based architecture of the Animat Brain is given in [7].

In order to analyze what could be features of the ACDs in such architecture, we simulated evolution of population of simple agents that have the ACD based control system [9]. The main results of simulations are described in the next section.

4. Simulation of Evolving Population of ACD Based Agents

4.1 Description of the Adaptive Agent Model

Agent Task. We consider an example of very simple agents, namely, agent-brokers and analyze adaptive features of these agents. In order to check the stability features of agents with respect to random variations of ACD neural network synaptic weights, we investigate evolutionary processes in populations of ACD agents.

We suppose that any agent-broker has a capital $C(t)$ that is distributed into cash and stocks. The fraction of stocks in the net capital of the agent is equal to $u(t)$. The environment is determined by the time series $X(t)$, $t = 1,2,\dots$, where $X(t)$ is the stock price at the moment t. The goal of the agent is to increase its capital $C(t)$ by changing the value $u(t)$. The capital dynamics is [10]:

$$C(t+1) = C(t)\,[1 + u(t+1)\,\Delta X(t+1) \,/\, X(t)]\,, \qquad (\mathbf{2})$$

where $\Delta X(t+1) = X(t+1) - X(t)$. For convenience, we use the logarithmic scale for the agent resource, $R(t) = \log C(t)$ [11]. The current agent reward $r(t)$ is defined by the expression: $r(t) = R(t+1) - R(t)$:

$$r(t) = \log\,[1 + u(t+1)\,\Delta X(t+1) \,/\, X(t)]\,. \qquad \textbf{(3)}$$

For simplicity, we assume that the variable $u(t)$ takes only two values, 0 or 1. The value $u(t+1)$ characterizes two possible agent actions: 1) transform all capital into cash, $u(t+1) = 0$; 2) transform all capital into stock, $u(t+1) = 1$.

Agent Control System. The agent control system is an ACD that consists of two NNs: model and critic (see Fig. 2). Assuming $|\Delta X| << |X|$, we set that the ACD state $\mathbf{S}(t)$ at moment t is characterized by two values, $\Delta X(t)$ and $u(t)$: $\mathbf{S}(t) = \{\Delta X(t), u(t)\}$.

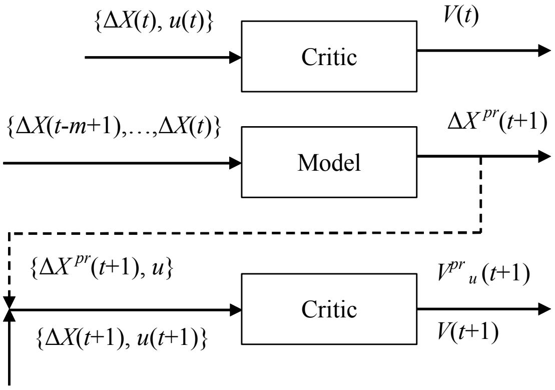

Fig. 2. The scheme of the ACD. The model predicts changes of the time series. The critic (the same NN is shown in two consecutive time moments) forms the state value function $V(\mathbf{S})$ for the current state $\mathbf{S}(t) = \{\Delta X(t), u(t)\}$, the next state $\mathbf{S}(t+1) = \{\Delta X(t+1), u(t+1)\}$, and its predictions $\mathbf{S}^{\mathbf{pr}}{}_{\mathrm{u}}(t+1) = \{\Delta X^{pr}(t+1), u\}$ for two possible actions, $u = 0$ or $u = 1$

The model predicts changes of the stock time series. The model output $\Delta X^{pr}(t+1)$ is based on m previous values of ΔX: $\Delta X(t\text{-}m+1),\ldots,\Delta X(t)$, which are used as the model inputs. The model is implemented as a multilayer perceptron (MLP) with one hidden layer of tanh nodes and linear output. The critic is intended to estimate the state value function $V(\mathbf{S})$. The values V approximate the utility function U in (1) for given states. The critic is also a MLP of the same structure as the model.

ACD operation is as follows. At any moment t, the following operations are performed:

1) The model predicts the next change of the time series $\Delta X^{pr}(t+1)$.

2) The critic estimates the state value function for the current state $V(t) = V(\mathbf{S}(t))$ and the predicted states for both possible actions $V^{\mathrm{pr}}{}_{u}(t+1) = V(\mathbf{S}^{\mathbf{pr}}{}_{u}(t+1))$, where $\mathbf{S}^{\mathbf{pr}}{}_{u}(t+1) = \{\Delta X^{pr}(t+1), u\}$; $u = u(t+1)$, $u = 0$ corresponds to the action “transform all capital into cash”, $u = 1$ corresponds to the action “transform all capital into stock”.

3) The ε-greedy rule is applied [2]: with the probability $1 - \varepsilon$ the action corresponding to the maximum value $V^{pr}{}_{u}(t+1)$ is selected, and the alternative action is selected with the probability ε $(0 < \varepsilon << 1)$.

4) The selected action ($u(t+1) = 0$ or $u(t+1) = 1$) is carried out. The transition to the next time moment $t+1$ occurs. The current reward $r(t)$ is calculated in accordance with

(3) and received by ACD. The value $\Delta X(t+1)$ is observed and compared with its prediction $\Delta X^{pr}(t+1)$. The NN weights of the model are adjusted to minimize the prediction error using the error backpropagation [8] with α_M as the model learning rate.

5) The critic computes $V(t+1)$. The temporal-difference error is calculated [2]:

$$\delta(t) = r(t) + \gamma V(t+1) - V(t) . \qquad (\mathbf{4})$$

6) The weights of the critic NN are adjusted to minimize the temporal-difference error (4) using its backpropagation and the gradient descent with α_C as the critic learning rate. Such learning is dedicated to increase accuracy of approximation of utility function (1) for given states by means of the critic NN.

Scheme of Evolution. We consider an evolving population that consists of n agents. Each agent has a resource $R(t)$ that changes in accordance with values of agent rewards: $R(t+1) = R(t) + r(t)$, where $r(t)$ is calculated in (3). At the beginning of any generation, initial resource of all agents is equal to zero.

Evolution passes through a number of generations, $n_g = 1,2,\ldots$ The duration of each generation is T time steps. At the end of each generation, the agent having the maximum resource $R_{max}(n_g)$ is determined. This best agent gives birth to n children that constitute a new (n_g+1)-th generation. The initial synaptic weights of both NNs (the model and the critic) form the agent genome $\mathbf{G}$. The genome $\mathbf{G}$ does not change during agent life; it is transferred (with small mutations) from the parent to offsprings. Temporary synaptic weights of the NNs $\mathbf{W}$ are changed during agent life via learning. At the beginning of (n_g+1)-th generation, we set for each newborn agent $\mathbf{G}(n_g+1) = \mathbf{G}_{\text{best}}(n_g)$ + **mutations,** $\mathbf{W}_0(n_g+1) = \mathbf{G}(n_g+1)$. A normally distributed random value with zero mean and standard deviation P_{mut} is added to each synaptic weight at mutations.

4.2 Results of Simulations

General Characteristics of Evolving Agent Population. The described model was investigated by means of computer simulations. We used two examples of model time series:

1) sinusoid:

$$X(t) = 0.5[1 + \sin(2\pi t/20)] + 1 , \qquad (\mathbf{5})$$

2) stochastic time series from [10, Example 2]:

$$X(t) = \exp[p(t)/1200] , \quad p(t) = p(t\text{-}1) + \beta(t\text{-}1) + k\,\lambda(t) , \quad \beta(t) = \alpha\beta(t\text{-}1) + \gamma(t) , \qquad (\mathbf{6})$$

where $\lambda(t)$ and $\gamma(t)$ are two random normal processes with zero mean and unit variance, $\alpha = 0.9$, $k = 0.3$.

The parameters of simulation were as follows. For all simulations we set: $\gamma = 0.9$, $\varepsilon = 0.05$, $P_{mut} = 0.1$; $m = 10$, $\alpha_M = \alpha_C = 0.01$, $N_{hM} = N_{hC} = 10$, where N_{hM} and N_{hC} are numbers of hidden neurons of the model and critic. Parameters n and T were set to different values, depending on the simulation, as specified below.

We analyze the following cases: 1) case L (pure learning); in this case we consider a single agent that learns by means of temporal difference method; 2) case E (pure evolution), i.e., evolving population without learning; 3) case LE, i.e., learning combined with evolution, as described above.

We compare the agent resource values attained during 200 time steps for these three cases of adaptation. For the cases E and LE, we set T = 200 (T is generation duration) and record the maximal value of agent resource in a population $R_{max}(n_g)$ at the end of each generation. For the case L, we have just one agent whose resource is reset $R(T(n_g\text{-}1)+1) = 0$ after the passing of every T = 200 time steps; the index n_g is incremented by one after every T time steps, i.e., $R_{max}(n_g) = R(Tn_g)$.

The plots R_{max} vs. n_g for the sinusoid (5) are shown in Fig. 3. In order to exclude the decrease of the value $R_{max}(n_g)$ due to the random choice of actions when applying the ε-greedy rule for the cases LE and L, we set $\varepsilon = 0$ after $n_g = 100$ for the case LE and after $n_g = 2000$ for the case L.

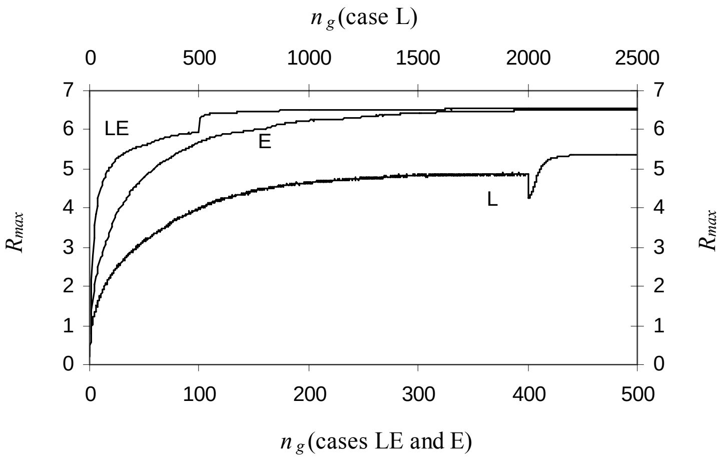

Fig. 3. The plots of $R_{max}(n_g)$ for the sinusoid (5). The curves LE, E and L correspond to the cases of learning combined with evolution, pure evolution and pure learning, respectively. Each point of the plots represents the average over 1000 simulations. For LE and E cases n = 10, T = 200. See the text for details

According to (3), there is obvious optimal policy of behavior for our simple agents: transform all capital into stock/cash when stock price rises/falls. Analysis of agent behavior demonstrates that both pure evolution (the case E) and learning combined with evolution (the case LE) are able to find the optimal policy. With this policy, the agent attains asymptotic value R_{max} = 6.5 (see Fig. 3). For the pure learning (the case L) the optimal policy is not found, the asymptotic value of R_{max} is only 5.4. Analysis reveals that the pure learning is able to find only the following satisfactory policy. The agent buys stocks when stock price rises (or falls by a small amount) and sells stocks when stock price falls significantly – the agent obviously prefers to keep the capital in stocks.

However, searching for the optimal policy by means of pure evolution is slower than when combining learning with evolution, as becomes apparent when examining the curves E and LE in Fig. 3. So, while learning in our model is not optimal by itself, it helps evolution to find better policies faster.

Baldwin Effect. The role of learning in evolving agent populations can be observed as the Baldwin effect [12,13], or the genetic assimilation of initially learned features during Darwinian evolution. This effect is found in number of computer experiments, one of which is shown in Fig. 4.

We examine how the best agent resource $R_{max}(t)$ changes during the first five generations for the sinusoid time series (5). Fig. 4 shows that during the early generations (generations 1 and 2), any significant increase of the agent resource begins only after a lag of 100 to 300 time steps. The best agent optimizes its policy by learning. Subsequently, the best agents find an advantageous policy faster and faster. By the fifth generation, a newborn agent "knows" a decent policy because it is encoded in its genome **G**, and the learning does not improve the policy significantly. Thus, Fig. 4 demonstrates that the initially learned policy becomes inherited (the Baldwin effect).

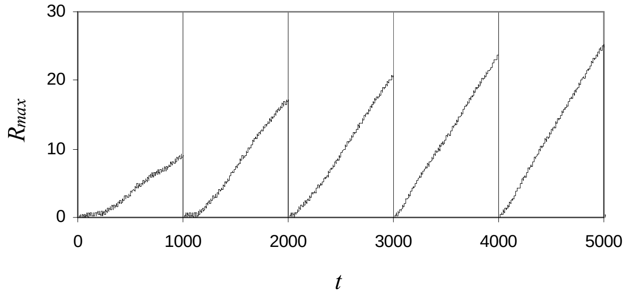

Fig. 4. The plots of the resource $R_{max}(t)$ of the best agent in the population for the first five generations. This is the case of learning combined with evolution on the sinusoid time series; n = 10, T = 1000. The ends of generations are shown by vertical lines. During the early generations (generations 1 and 2), there is an obvious delay in the increase of the agent resource. An advantageous policy is found only after some learning period during first 100 to 300 time steps. By the fifth generation, the rapid increase of the resource begins at the start of the generation, demonstrating that the advantageous policy has become inherited

Peculiarities of Model Prediction. ACD control system includes a model NN for predicting the next change $\Delta X(t+1)$ of the time series (Fig. 2). We analyzed the operation of the model NN in evolving agent population and revealed a very interesting feature. The model NN can produce incorrect predictions, yet the agent can still use these predictions to select appropriate actions. For example, Fig. 5 demonstrates that the predictions $\Delta X^{pr}(t+1)$ approximately corresponds to real changes, but differs from $\Delta X(t+1)$ in both sign and scale. Detailed analysis of selected actions shows that these predictions ensure actually optimal agent policy.

We believe that these peculiarities of model NN performance are mainly due to the dominant role of evolution over learning for the optimization of agent control systems and evolutionary modification of ACD operation mode. The ACD operation does not correspond to "correct" mode described in subsection "Agent Control System", however this operation is useful. Such a modification of operation mode seems to favor agent control systems that are evolutionary stable.

We can note that the observed spontaneous amplification of ΔX^{pr} by the model NN seems to be helpful to achieve stable operation of the critic NN because the real values $\Delta X(t+1)$ are too small (on the order of 0.001). For the cases E and LE we observed similar amplification of values $\Delta X^{pr}(t+1)$ as compared with real values $\Delta X(t+1)$ in all simulations for given set of parameters. The reverse in sign of $\Delta X^{pr}(t+1)$ with respect to $\Delta X(t+1)$ was observed in approximately 50% of computer experiments.

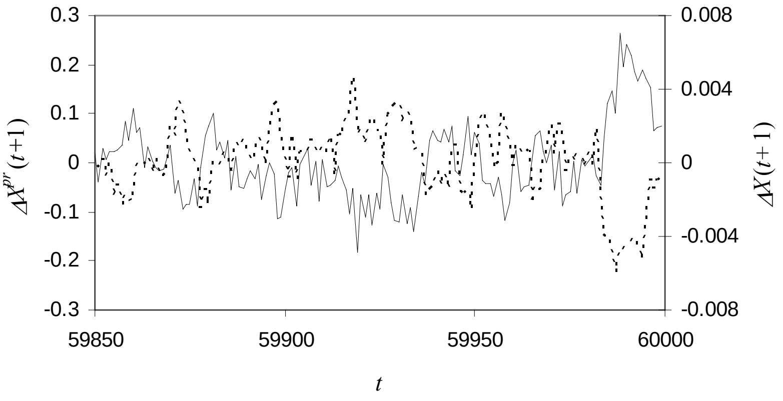

Fig. 5. The plots of predicted $\Delta X^{pr}(t+1)$ (dotted line) and real values $\Delta X(t+1)$ (solid line). This is the case of learning combined with evolution on the stochastic time series; $n = 10$, $T = 200$. The curves $\Delta X^{pr}(t+1)$ and $\Delta X(t+1)$ differ in both scale and sign

Comparison with Searching Behavior of Simple Animals. For the case of evolution alone, an interesting type of behavior is observed in the first stages of evolution. The agent has a rough policy that reflects only general features of changing environment (Fig. 6). The agent buys/sells stocks when the stock rises/falls significantly, and it ignores small and short-term variations of the stock price. There exists inertia in switching between two tactics of behavior (sell stocks and buy stocks). This inertial behavior is very similar to foraging tactics in some animals, e.g. caddis fly larvae [14]; it helps an animal to react adaptively to only general large-scale patterns in environment.

Conclusion from ACD Simulations. Thus, the investigation of simple ACD based agents demonstrates that there are certain difficulties in designing evolutionary stable NN animat control systems using ACD. The main problem is: evolution reorganizes ACD performance, namely, evolutionary stable agent control systems can be found, but such control systems do not operate as correct ACD (e.g. see subsection "Peculiarities of Model Prediction"). In order to overcome these difficulties, in the next section we propose another structure of functional systems which is more biologically plausible.

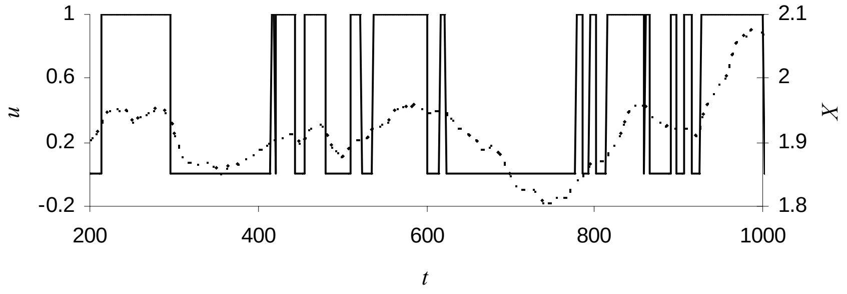

Fig. 6. Time dependence of action selection *u*(*t*) for the best agent in the population (solid line) during the first stages of evolution (without learning). Time series *X*(*t*) is also shown (dashed line). n = 100, T = 200. The agent has a rough policy that reflects only general features of changing environment

5. Second Version of Animat Brain Architecture

This version of Animat Brain also includes a set of NN based formal functional systems (FSs). As in previous version, we suppose that the highest level of animat control system hierarchy corresponds to the main animat needs (energy replenishment, safety, reproduction). FSs of lower levels correspond to tactical goals and sub-goals of behavior and have no rigid hierarchy. It is supposed that at any time moment, only one FS is active, in which the current action is formed. The animat receives reinforcements (rewards and punishments) which are related to animat needs.

Each FS consists of two NNs: the controller and the model. At any time moment t (t = 1,2,…), the operation of the active FS can be described as follows. The state vector **S**(*t*) characterizing the current external and internal environment is fed to the FS input. The controller forms the action **A**(*t*) in accordance with given state **S**(*t*), i.e. the controller forms the mapping **S**(*t*) -> **A**(*t*). Some actions **A**(*t*) are commands onto effectors (actual actions), another actions are activation commands for other FSs. The model predicts the next state for given vectors **S**(*t*) and **A**(*t*), i.e. the model forms the mapping {**S**(*t*), **A**(*t*)} -> $\mathbf{S}^{\mathbf{pr}}(t+1)$. The mappings **S**(*t*) -> **A**(*t*) and {**S**(*t*), **A**(*t*)} -> $\mathbf{S}^{\mathbf{pr}}(t+1)$ are stored in NN synaptic weights. The activation commands are delivered from one FS to others in accordance with connectivity matrix C_{ij} , the value C_{ij} characterizes the probability that j-th FS is activated by i-th FS.

It is supposed that there are primary and secondary repertoires of behaviors. The primary repertoire is formed by evolution: there is a population of animats and synaptic weights of controller and model NNs, connectivity matrix C_{ij} , as well as a set of FSs are adjusted during evolutionary processes similar to those of described in the previous section.

The secondary repertoire of behavior is formed by learning. There are two regimes of learning: 1) the extraordinary mode and 2) the fine tuning mode.

The extraordinary mode is a rough search of behavior that is adequate to the current situation. This mode comes, if the predicted state $\mathbf{S}^{\mathbf{pr}}(t+1)$ in the active FS strongly differs from the real state $\mathbf{S}(t+1)$. In terms of the functional system theory (Section 2, Fig. 1), large difference between $\mathbf{S}^{\mathbf{pr}}(t+1)$ and $\mathbf{S}(t+1)$ means that parameters of result differ essentially from parameters stored in acceptor of action results.

In the extraordinary mode, a random search for new behaviors takes place; namely, the connectivity matrix C_{ij} is substantially changed, new FSs are randomly generated and selected. This mode is similar to neural group selection in the Edelman's theory of Neural Darwinism [15].

In the fine tuning mode, learning is adjustment of NN synaptic weights in the FS that is active at the current moment of time and in the FSs that were active in some previous steps of time. As synaptic weights are updated in those NNs, which were active in previous time steps, this learning mode allows forming chains of consecutive actions. Synaptic weights of model NNs are modified to minimize prediction errors (e.g. by means of error back-propagation [8]). Synaptic weights of controller NNs are adjusted by Hebbian-like rule: the synaptic weights in controllers are modified to make the mappings $\mathbf{S}(t)$ -> $\mathbf{A}(t)$ more strong/weak at positive/negative reinforcements.

We began computer simulations of a simple particular variant of the second version of the Animat Brain assuming two animat needs (safety and energy replenishment) and simple cellular environment.

6. Conclusion

Thus, we reviewed Animat Brain architectures that are based on the functional systems (FSs) theory. In the first version of the Animat Brain we tried to use the reinforcement learning approach [2], namely we used adaptive critic design (ACD) based FSs. However, simulation of ACD agents demonstrates that correct ACD operation can be evolutionary unstable: evolution reorganizes ACD operation in some sophisticated manner. So, now we are developing more biologically plausible Animat Brain architecture, which is based on the FS that consists of the model NN and the controller NN. The controller NNs are intended to form chains of actions and the model NNs are intended to predict future events. In the case of unexpected events, considerable learning takes place and animat behavior is reorganized. We intend to find conditions in which predictions of future events (formed by model NNs) and generations of chains actions (formed by controller NNs) are consistent with each other.

Acknowledgments

This work is supported in part by the Russian Foundation for Basic Research, Grant No 04-01-00179 and the Program of the Presidium of the Russian Academy of Science “Mathematical Modeling and Intelligent Systems”, Project No 16.2.45.

References

1. Anokhin, P.K.: Biology and Neurophysiology of the Conditioned Reflex and Its Role in Adaptive Behavior. Pergamon, Oxford (1974)
2. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA: A Bradford Book (1998)
3. Widrow B., Gupta, N., Maitra, S.: Punish/Reward: Learning with a Critic in Adaptive Threshold Systems. IEEE Transactions on Systems, Man and Cybernetics. 3 (1973) 455-465
4. Werbos, P.J.: Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence. General Systems Yearbook. 22 (1977) 25-38
5. Prokhorov D.V., Wunsch, D.C.: Adaptive critic designs. IEEE Trans. Neural Networks. 8 (1997) 997-1007
6. Butz, M.V., Sigaud, O., Gérard, P.: Internal Models and Anticipations in Adaptive Learning Systems. In Butz, M. V., Sigaud, O., Gérard, P. (eds.): Anticipatory Behavior in Adaptive Learning Systems. Springer-Verlag, Berlin (2003) 86-109
7. Red'ko V.G., Prokhorov D.V., Burtsev M.S.: Theory of Functional Systems, Adaptive Critics and Neural Networks. In Proc. International Joint Conference on Neural Networks (IJCNN 2004), Budapest (2004) 1787-1792
8. Rumelhart D.E., Hinton G.E., Williams R.G.: Learning Representation by Back-Propagating Error. Nature. 323 (1986) 533-536
9. Red’ko V.G., Mosalov O.P., Prokhorov D.V.: A Model of Evolution and Learning. Neural Networks. 18 (2005) 738-745
10. Prokhorov, D., Puskorius, G., Feldkamp L.: Dynamical Neural Networks for Control. In Kolen J., Kremer S. (eds.): A Field Guide to Dynamical Recurrent Networks. IEEE Press, New York (2001) 257-289
11. Moody, J., Wu, L., Liao, Y., Saffel, M.: Performance Function and Reinforcement Learning for Trading Systems and Portfolios. Journal of Forecasting. 17 (1998) 441-470
12. Baldwin, J.M.: A New Factor in Evolution. American Naturalist. 30 (1896) 441-451
13. Turney P., Whitley D., Anderson R. (eds.): Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect. Special Issue of Evolutionary Computation on the Baldwin Effect. 4 (1996)
14. Nepomnyashchikh, V.A.: Selection Behaviour in Caddis Fly Larvae. In Pfeifer R. et al (eds.): From Animals to Animats 5: Proceedings of the Fifth International Conference of the Society for Adaptive Behavior. MIT Press, Cambridge, MA (1998) 155-160
15. Edelman, G.M.: Neural Darwinism: The Theory of Neuronal Group Selection. Oxford University Press, Oxford 1989