Recursive sugoals discovery based on the Functional Systems Theory

Evgenii E. VITYAEV[a,b,1], Alexander V. DEMIN[c]
[a] *Sobolev Institute of Mathematics SB RAS,*
630090, Novosibirsk, Russia. E-mail: vityaev@math.nsc.ru
[b] *Novosibirsk State University, Novosibirsk*
[c] *A.P.Ershov Institute of Informatics Systems SB RAS,*
630090, Novosibirsk, Russia

Abstract. The paper presents a model of adaptive behavior of an aminat based on the physiological functional system theory and Probabilistic Dynamic Logic. The main distinction of this model is the possibility for automatic generation of new sub-goals, which allows us to solve more complex multi-level tasks. Animat has been created on the basis of this model, and an experiment has been carried out in order to train it and to compare it with the existing approaches based on reinforcement learning. The results of comparison have shown that the proposed model learns and acts efficiently.

Keywords. Animat, Control System, Probabilistic Dynamic Logic

Introduction

In this work we applied Probabilistic Dynamic Logic models [1][2][3] for the goal-seeking behavior models, investigated in the Functional Systems Theory [4][5] for developing the adaptive animat. We describe a general scheme of functional systems, training algorithm and automatic subgoals formation. On the basis of the proposed model, an elementary animat and its environment have been implemented in a computer program. Using this program, we carried out experiment in the animat learning and made test comparison with some existing approaches.

1. A model of a functional system operation

Functional systems theory. The model we propose is based on the theory of functional systems, developed in 1930-70s by the famous Soviet neurophysiologist P.K. Anokhin [4][5]. According to this theory, a functional system, that achieves some results beneficial for an organism (for example, need - satisfaction), is considered to be a unit of this organism's activity.

The initial stage of a behavioral act of any complexity is *afferent synthesis* that includes synthesis of motivation, memory and information about the environment. As a

[1] Corresponding Author. This work is partially supported by the Russian Science Foundation grant #11-07-00560-a and Integration projects #47, 111, 119 of the SB RAS.

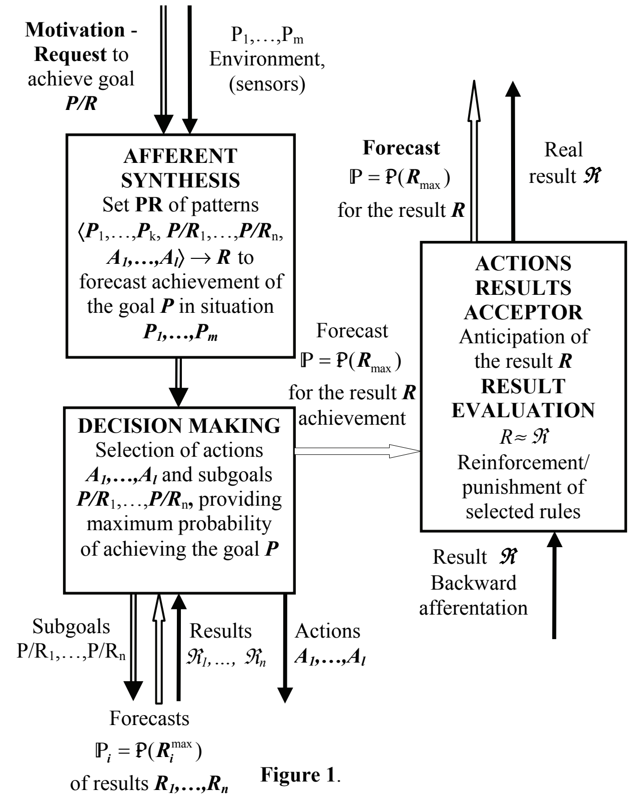

Figure 1.

result of the afferent synthesis, all possible ways of achieving the goal in this situation are evoked from the memory. At the stage of *decision-making*, only one particular way of action is selected according to the initial need. To provide the achievement of the results, an *actions results acceptor* is being created beforehand which is a model of the parameters of the expected (predicted) results. Then actions are followed by signals about achievement of the results, called backward afferentation, which would perfectly comply with the properties of the actions results acceptor.

Functional system model. Figure 1 shows the model [6][7] of a functional system based on [4][5]. Let us think that a goal ***P/R*** is set to the functional system that consists in reaching the result ***R***. This is done in the form of a request to the functional system to achieve the goal ***P/R***. Information about the environment is also supplied in the form of stimulus (sensors) $\boldsymbol{P_1},...,\boldsymbol{P_m}$.

In the process of afferent synthesis, all information related to achievement of the goal ***P*** is evoked from the memory by the motivation excitement. This information is stored as a set of rules $\boldsymbol{PR} = \{ \langle \boldsymbol{P_1},...,\boldsymbol{P_k},(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n,\boldsymbol{A_1},...,\boldsymbol{A_l} \rangle \to \boldsymbol{R} \}$, where $\boldsymbol{P_1},...,\boldsymbol{P_k}$ – properties of a situation; $(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n$ – subgoals that should achieve the results $\boldsymbol{R_1},...,\boldsymbol{R_n}$ in order to achieve the goal $\boldsymbol{P}$; $\boldsymbol{A_1},...,\boldsymbol{A_l}$ – actions, that must be fulfilled to achieve the result ***R***. Note that only those rules are evoked, in which the situation properties $\boldsymbol{P_1},...,\boldsymbol{P_k}$ are included in the situation description $\boldsymbol{P_1},...,\boldsymbol{P_m}$.

To achieve the subgoals, we send the corresponding requests *recursively* down the hierarchy (in figure 1 it is indicated by the double arrow pointing down). These requests activate all information related to achievement of these subgoals in the lower level functional systems, which, in turn, may require achievement of the other goals at even lower levels, and so on. If some subgoal cannot be achieved in this situation (there are no rules predicting its achievement in this situation), then denial is received in reply to the request and the corresponding rule is excluded from the consideration. If all requested subgoals of the rule R = ($\langle \boldsymbol{P_1},...,\boldsymbol{P_k},(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n,\boldsymbol{A_1},...,\boldsymbol{A_l}\rangle \to \boldsymbol{R}$) are achieved, then it passes to the decision-making block as a possible way of the goal achievement.

In the decision-making block all obtained rules are assigned the probability of the goal $\boldsymbol{P}$ achievement. The estimated probability of the rule

R = ($\langle \boldsymbol{P_1},...,\boldsymbol{P_k},(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n,\boldsymbol{A_1},...,\boldsymbol{A_l}\rangle \to \boldsymbol{R}$) is:

$\mathrm{P}(\boldsymbol{R}) = \mathrm{P}(\boldsymbol{R} \mid \boldsymbol{P_1},...,\boldsymbol{P_k},\boldsymbol{R_1},...,\boldsymbol{R_n},\boldsymbol{A_1},...,\boldsymbol{A_l}) \cdot \mathrm{P}(\boldsymbol{P_1}) \cdot ... \cdot \mathrm{P}(\boldsymbol{P_n})$, where

$\mathrm{P}(\boldsymbol{R} \mid \boldsymbol{P_1},...,\boldsymbol{P_k},\boldsymbol{R_1},...,\boldsymbol{R_n},\boldsymbol{A_1},...,\boldsymbol{A_l})$ - conditional probability of the rule R,

$\mathbb{P}(\boldsymbol{P_1}),...,\mathbb{P}(\boldsymbol{P_n})$ – probabilities of subgoals $\boldsymbol{P_1},...,\boldsymbol{P_n}$ achievement.

As decision in the decision-making block the rule $\boldsymbol{R}_{\max}$ with maximum estimation of the probability $\mathrm{P}(\boldsymbol{R}_{\max})$ is selected as the way of the goal $\boldsymbol{P}$ achievement. The probability $\mathbb{P}(\boldsymbol{P}) = \mathrm{P}(\boldsymbol{R}_{\max})$ performs the forecasted probability of the goal $\boldsymbol{P}$ achievement.

Forecast $\mathrm{P}(\boldsymbol{R}_{\max})$ (indicated by the white arrow) for the anticipated result $\boldsymbol{R}$ is sent to the actions results acceptor block. Forecasts for the subresults $\boldsymbol{R_1},...,\boldsymbol{R_n}$ in all lower level functional subsystems are also sent to the corresponding actions results acceptors. Forecasts $\mathbb{P}(\boldsymbol{P_1}),...,\mathbb{P}(\boldsymbol{P_n})$ of all subresults $\boldsymbol{R_1},...,\boldsymbol{R_n}$ for all subgoals are being sent from the functional subsisitems to the decision making block of the higher level system (also indicated by the white arrow).

The anticipated results $\boldsymbol{R}$ and $\boldsymbol{R_1},\dots,\boldsymbol{R_n}$ for the goal ***P/R*** and subgoals $(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n$ perform the actions results acceptor. As demonstrated in a lot of physiological experiments [4][5] the real receptors are activated for the control of corresponding results achievement. Actions results acceptor controls not only the final result achievement (the satisfaction of some need) but also all hierarchy and sequence of intermediate results that must be achieved for the final goal achievement.

After decision is made and actions results acceptor activated, real actions stat to be realized. This actions lead to the real results $\mathfrak{R}$ and $\mathfrak{R}_1,\dots,\mathfrak{R}_n$, that were received thought corresponding backward afferentation [4][5]. This results are compared with anticipated results $\boldsymbol{R}$ and $\boldsymbol{R_1},\dots,\boldsymbol{R_n}$. If forecasts coincide with the real results (with the given degree of accuracy), then rules selected in the decision-making blocks are reinforced, otherwise punished. Reinforcement/punishment consists in changing the probabilistic estimations of these rules, which consists not in simple rising/decreasing of the conditional probabilities, as in reinforcement learning, but in conditional reflex modeling by the semantic probabilistic inference [8][9].

Subgoals discovering. Initially animat has a hierarchy of functional systems given a priori. In the process of learning, animat may automatically discover new subgoals and generate the corresponding functional systems.

According to J. Gibson [10] (1) *elements are organized hierarchically*, small elements are *hierarchically bild-in* large elements, and (2) elements *occur* in different places of the world. Results of actions are well predicted only in the frame of one holistic element. Hence, the most perfect predictions are also organized hierarchically in accordance with the following principal:

Principle of hierarchical prediction: (1) predictions are always performed in the frame of one holistic element; (2) predictions between different elements performed in the *hierarchically* upper level systems, where these different elements belong to one holistic element.

Subgoals are such results of sequences of actions (in perception or movement) that belong to one element. To divide sequences of actions (and its results) into parts, each of which belongs to one element, we propose the following definition of subgoals. Subgoals are such results of some sequence of actions that (1) significantly increase the probability of higher-level goals achievement and (2) actions following these sequence can't be defined unambiguously (may belong to other elements) [6].

In order to identify subgoals, the set of rules $\boldsymbol{PR}$ of some functional system is analyzed. For the rule R $= (\langle \boldsymbol{P_1},...,\boldsymbol{P_k},(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n,\boldsymbol{A_1},...,\boldsymbol{A_l}\rangle \to \boldsymbol{R})$ we define as $\boldsymbol{Sen}(\boldsymbol{R}) = \{\boldsymbol{P_1},...,\boldsymbol{P_k}\}$ the set of sensors, as $\boldsymbol{Act}(\boldsymbol{R}) = \{\boldsymbol{A_1},...,\boldsymbol{A_l}\}$ the set of actions and as $\boldsymbol{Sub}(\boldsymbol{R}) = \{(\boldsymbol{P}/\boldsymbol{R})_1,...,(\boldsymbol{P}/\boldsymbol{R})_n\}$ the set of subgoals. In accordance with the functional systems theory [4][5], the anticipated result $\boldsymbol{R}$ is not only a final result (the need satisfaction), but also all intermediate results that are achieved by actions from the set $\boldsymbol{Act}(\boldsymbol{R}) = \{\boldsymbol{A_1},...,\boldsymbol{A_l}\}$. So, the result $\boldsymbol{R}$ is a consequence of more detailed results $\boldsymbol{R} = \{\boldsymbol{R^1},...,\boldsymbol{R^t}\}$.

Subrule R_{sub} = $(\langle \boldsymbol{P_1^{new}},...,\boldsymbol{P_{k'}^{new}},(\boldsymbol{P}/\boldsymbol{R})_1^{new},...,(\boldsymbol{P}/\boldsymbol{R})_{n'}^{new},\boldsymbol{A_1^{new}},...,\boldsymbol{A_{l'}^{new}}\rangle \to \boldsymbol{R^{new}})$, of the rule R (we right $R_{sub} \lhd R$) produces a new subgoal $\boldsymbol{P^{new}}/\boldsymbol{R^{new}}$ and corresponding functional system, if the following conditions are satisfied:

1) $\boldsymbol{Sen}(\boldsymbol{R_{sub}}) \subset \boldsymbol{Sen}(\boldsymbol{R})$, $\boldsymbol{Sub}(\boldsymbol{R_{sub}}) \subset \boldsymbol{Sub}(\boldsymbol{R})$ $\boldsymbol{Act}(\boldsymbol{R_{sub}}) \subset \boldsymbol{Act}(\boldsymbol{R})$,
   $\boldsymbol{R^{new}} = \{\boldsymbol{R_1^{new}},...,\boldsymbol{R_s^{new}}\} \subset \{\boldsymbol{R^1},...,\boldsymbol{R^t}\} = \boldsymbol{R}$;
2) any other rule, R' $\in$ $\boldsymbol{PR}$ that does not contain a subrule R_{sub} (the relation $R_{sub} \lhd R'$ is not true) has less value of the estimated probability $\mathrm{P}(\boldsymbol{R}') < \mathrm{P}(\boldsymbol{R})$. This means that achievement of the result $\boldsymbol{R^{new}}$ considerably increases the estimated probability $\mathrm{P}(\boldsymbol{R})$ of the rule R;
3) sequence of actions $\{\boldsymbol{A_1^{new}},...,\boldsymbol{A_{l'}^{new}}\}$ is maximal – it can't be extended without losing the value of estimated probability $\mathrm{P}(\boldsymbol{R_{sub}})$ of the rule R_{sub}.

The subrule R_{sub} produces a new functional system at the lower level that evoked by the new subgoal $\boldsymbol{P^{new}}/\boldsymbol{R^{new}}$ of the reduced (relative to R) rule

R_{red} = $\langle \boldsymbol{Sen(R)},\boldsymbol{Sub(R)} \setminus \boldsymbol{Sub(R_{sub})},\boldsymbol{P^{new}}/\boldsymbol{R^{new}},\boldsymbol{Act(R)} \setminus \boldsymbol{Act(R_{sub})}\rangle \to \boldsymbol{R}$

A new functional system has a goal $\boldsymbol{P^{new}}/\boldsymbol{R^{new}}$, that may be achieved by the rule R_{sub}. Thus, for each functional system, its set of rules $\boldsymbol{PR}$ is analyzed and new subgoals are identified. For any newly-generated functional systems a set of rules $\boldsymbol{PR}$, including at least a rule R_{sub} , is discovered, using the semantic probabilistic inference.

2. Experiment

A virtual world and an animat were modeled, and the main goal for the animat was to pick up "food". The animat's world is a square field divided into 25×25 cells, which are of the following type: empty cells, obstacles, food and pill. Obstacles are located only along the virtual world's perimeter thus forming its natural borders. The animat may move across the field and perform 3 types of actions: to step one cell forward, to turn left, to turn right.

Some amount of food is randomly distributed over the field. To pick up food, the animat has to turn to the cell containing food. In this case it is assumed that it "eats" the food, the cell is cleaned and new "food" object appears at random in another place of the field. Pills, like food, are randomly distributed over the field. Before eating food, the animat has first to find, pick and keep a pill. When it eats food, the pill disappears and, to eat the next portion of food, it again has to find and pick up a pill, and so on. The pill is picked, when the animat has to step to the cell containing the pill. However, if the animat has one pill, it can't pick any more pills until it uses it to eat food. When the animat picks a pill, the cell is cleaned and a new pill appears in another place of the field at random. The amounts of pills and food were kept equal to 100.

The animat has ten sensors. Eight of them are located around the animat: "in front of-to the left", "in front of", "in front of-to the right", "to the left", "to the right", "behind-to the left", "behind", "behind-to the right"; one sensor is in its center "center" and one is "pill availability" that informs the animat about pill availability and gets values "yes" or "no". Each sensor informs the animat about the type of the cell.

In order to evaluate efficiency of the control system we made test comparison with systems built on the basis of reinforcement learning [1]. For comparison, we selected two control systems built on the basis of Q-Learning. The essence of this algorithm is in consequent refinement of estimates for the total reward value $\boldsymbol{Q}(\boldsymbol{s}_t, \boldsymbol{A}_t)$, that obtained after performing an action A_t in a situation s_t:

$$\boldsymbol{Q}^{(i+1)}(\boldsymbol{s}_t, \boldsymbol{A}_t) = \boldsymbol{Q}^{(i)}(\boldsymbol{s}_t, \boldsymbol{A}_t) + \alpha(\boldsymbol{r}_t + \gamma \max_A \boldsymbol{Q}^{(i)}(\boldsymbol{s}_{t+1}, \boldsymbol{A}) - \boldsymbol{Q}^{(i)}(\boldsymbol{s}_t, \boldsymbol{A}_t)).$$

The first system is Q-Lookup Table, which is based on the table that contains Q-values for all possible situations and actions. The second system is Q-Neural Net, which uses an approximation of the function $\boldsymbol{Q}(\boldsymbol{s}_t, \boldsymbol{A}_t)$ using neural networks. In this case each possible action A_i uses a separate neural network NN_i.

In our experiment the animat needs at first to find a pill, and then to find food. The purpose of the experiment was to demonstrate the possibility of automatic generation of subgoals during the goal-seeking behavior.

At the beginning, animat has only a basic functional system, and its goal is to achieve the situation when, at the same time, a pill is available and the central sensor indicates that food is found. The corresponding goal-predicate is: ("center" = "food" and "available pill" = yes).

At each test started during our experiment the animat did stably identify a new subgoal described by the subgoal-predicate $\boldsymbol{P_1}$ = ("available pill" = yes) and created the corresponding functional system. The model was working in the following way. When the animat had no pill, the rule $\boldsymbol{P_1} \rightarrow \boldsymbol{P}$ was launched as the most probable one. It passed the control function to a lower-level functional system that performed the pill search. And when the animat had a pill, rules with a higher probability were launched in the basic functional system and food was found.

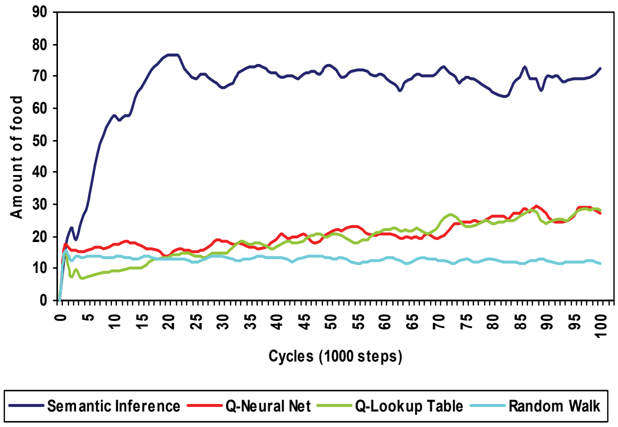

Figure 2. Amounts of food collected by animats with subgoals.

The results of this experiment are shown in fig. 2. The diagram shows the average values for every control system based on 20 tests. In each test the animat was given 100,000 steps, and had to learn how to accomplish the given task. It follows from the diagram, that control system based on the functional systems model surpasses the reinforcement learning systems both in the speed of learning and in the quality of its functioning. The control system Q-Lookup Table performs a poor learning ability and unstable functioning. The control system Q-Neural Net in some cases is capable to learn to react to all sensor data correctly when the training period is increased up to 300,000 – 500,000 steps. The control system Q-Lookup Table could not reach optimal behavior even after 500,000 steps. Open source code of the animat control system is available at address http://www.math.nsc.ru/AP/ScientificDiscovery/PDF/animat.exe.

References

[1] E.E. Vityaev, B.Y. Kovalerchuk, L. Perlovsky, S.O. Smerdov, Probabilistic Dynamic Logic of Phenomena and Cognition // WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain, pp. 3361-3366

[2] Perlovsky, L.I. (2007). Modeling Field Theory of Higher Cognitive Functions. Chapter in Artificial Cognition Systems, Eds. A. Loula, R. Gudwin, J. Queiroz. Idea Group, Hershey, PA, pp.64-105.

[3] Perlovsky, L.I. (2010). Neural Mechanisms of the Mind, Aristotle, Zadeh, & fMRI, IEEE Trans. Neural Networks, 21(5), 718-33.

[4] P.K. Anokhin, Biology and neurophysiology of the conditioned reflex and its role in adaptive behavior. Oxford etc.: Pergamon press, 1974.- 574p.

[5] P.K. Anokhin, Functional system.- In: Dictionary of behavioral science. New York etc., 1973, p.153.

[6] A.V. Demin, E.E. Vityaev, Logical model of adaptive control system // Neuroinformatics, v3(1), 2008, p.79-107. (in Russian).

[7] E.E. Vityaev, Brain principals of the functional systems theory and information theory of emotions // Neuroinformatics, v3(1), 2008, p.25-78. (in Russian).

[8] E.E. Vityaev, The logic of prediction // Proceedings of the 9th Asian Logic Conference (August 16-19, Novosibirsk, Russia), World Scientific Publishers, 2006 pp.263-276

[9] E.E. Vityaev, B.Y. Kovalerchuk, Empirical Theories Discovery based on the Measurement Theory. Mind and Machine, v.14, #4, 551-573, 2004.

[10] Gibson, J.J. (1979). The Ecological Approach to Visual Perception. Boston: Houghton Mifflin. (1986)

[11] Sutton R. and Barto A., Reinforcement Learning: An Introduction. – Cambridge: MIT Press, 1998.