Task and Spatial Planning by the Cognitive Agent with Human-like Knowledge Representation *

Aitygulov Ermek[1], Gleb Kiselev[2,3], and Aleksandr I. Panov[1,3]

[1] Moscow Institute of Physics and Technology, Moscow, Russia
aytygulov@phystech.edu, panov.ai@mipt.ru
[2] National Research University Higher School of Economics, Moscow, Russia
kiselev@isa.ru
[3] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia

Abstract. The paper considers the task of simultaneous learning and planning actions for moving a cognitive agent in two-dimensional space. Planning is carried out by an agent who uses an anthropic way of knowledge representation that allows him to build transparent and understood planes, which is especially important in case of human-machine interaction. Learning actions to manipulate objects is carried out through reinforcement learning and demonstrates the possibilities of replenishing the agent's procedural knowledge. The presented approach was demonstrated in an experiment in the Gazebo simulation environment.

Keywords: cognitive agent· sign· sign-based world model· human-like knowledge representation· behavior planning· pseudo-physical logic· reinforcement learning· spatial planning· task planning.

1 Introduction

One of the main tasks researchers are facing with in the field of robotics and artificial intelligence is the task of ensuring the effective interaction of robots and people in collaborative scenarios, i.e. when a person and a machine perform joint actions in a shared environment. To solve this problem the questions arise of arranging the operation the robotic system in such a way that its actions are transparent, predictable and quickly interpretable by a person, in other words, it is necessary that the robot's behavior be human-like in cases of human-machine interaction becomes especially urgent. One of the directions of scientific research aimed at solving this issue is the direction for the development of cognitive

* The results concerning models of sign components and planning algorithms (Sections 4.1 and 4.2) were obtained under the support of the Russian Science Foundation (project No. 16-11-00048), and the results on reinforcement learning for manipulator (Sections 4.3 and 5) were obtained under the support of the Russian Foundation for Basic Research (project No. 17-29-07079).

agents, i.e. such intelligent agents who would learn and plan their actions using approaches based on cognitive models of human behavior [1, 2].

In this paper, we consider the task of developing a cognitive agent that plans to move in space and actions to manipulate objects using the so-called sign-based world model [4–6]. This way of knowledge representation about the environment, the agent himself and other participants in joint activities is based on the psychological theories of Leontyev's activity [9] and Vygotsky's cultural-historical approach [10], which ensures his simple interpretation by human. In this paper, the agents world model is spatial procedural and declarative knowledge that use pseudo-physical logic [3], created with the use of psychological data on the human-like spatial reasoning. Spatial knowledge constructed by analyzing the map using egocentric coordinates allows maintaining the agent's autonomy regardless of the state of the "center", and various levels of map representation reduce the requirements for its computing resources.

The agent's actions planning, carried out within the world model, is also psychologically plausible. We leave out the details of the reactive functions [7, 8] and the algorithms for recognizing the objects of the surrounding space significant for the agent [11]. The presented in the paper algorithm of spatialMAP planning is hierarchical and abstracts from the details of the implementation of an action, solving the task of creating a sequence of abstract agent actions (moving, rotating, picking up an object), which will lead to the set goal. At each iteration of the plan execution, the planner can be restarted, which makes it possible to make a more detailed plan for implementing the abstract action.

The world model of a cognitive agent can be replenished through learning. In this paper, complex actions to move objects are constructed through reinforcement learning through the TRPO algorithm [12], which allows to optimize the strategies of choosing smaller actions with guaranteed monotonous improvement. The constructed functions, the control over which is transmitted every time after obtaining the appropriate prescription from the planning algorithm, allow to interact with different kinds of objects without classifying the methods of interactions and having only an abstract description of the required state at the end of the action. After the successful completion of the learning algorithm and the performance of the action, the action is saved as an experience and re-learning is no longer required.

The cognitive agent described in this paper is able to function in real environments, which is experimentally confirmed in simulations in Gazebo. Also, work is underway to implement experiments in real conditions with a robotic system that includes a platform allowing the movement of the agent, an arm similar to the one for the Turtlebot 2, as well as the camera, lidar and other sensors.

The paper is organized as follows. Part 2 presents the formulation of the problem of planning the movements of a cognitive agent, and briefly describes the algorithm for the operation of the cognitive agent. Part 3 provides an overview of modern methods of planning movements using pseudo-physical logics, as well as a comparison of reinforcement learning algorithms. Part 4 provides a detailed implementation of the planning and reinforcement learning algorithm. Part 5 contains a description of the experiments performed.

2 Problem Statement

The goal driven behavior of the cognitive agent is realized through an iterative procedure, which consists of 3 basic steps:

1. Agents learning.
2. Planning actions to achieve the target situation.
3. Plan implementation in the environment.

Agents learning is based on the reinforcement learning approach, for the implementation of which the algorithm TRPO is used. Learning takes place in a synthetic environment, which is a minimalistic model of the environment, containing only the information necessary for learning. Reinforcement learning is a machine learning tool that allows an agent to develop the desired behavior strategy based on the environmental response. This method uses a system of penalties and rewards for the actions of the agent, which allows you to take into account the experience of previous interactions. To describe the activity of a cognitive agent, a probability distribution $\pi(o|s)$ is used that characterizes the probability of an agent choosing an action o in a state s. Probability distribution π is called a strategy: $\pi(o|s) = P(o_t = o|s_t = s)$.

The agent, following the strategy, applies the actions and passes from the state to the state, receiving for it a reward r, which can be either positive or negative.

As an evaluation of the strategy, a value $\eta(\pi)$ is considered that is the mathematical expectation of the discounted remuneration for the whole session: $\eta(\pi) = E_\pi[\sum\limits_{t=0}^{\infty} \gamma^t r(s_t)]$.

The TRPO algorithm described in this paper uses a surrogate function, the maximization of which, with the right choice of step, entails optimizing the value $\eta(\pi)$. Combining with the algorithm Natural policy gradient [13] greatly improves the work of the algorithm.

In the case of using as a goal situation for reinforcement learning some sub-goal in the overall task of planning, the result is a sequence of actions (strategy) to achieve this sub-goal. After the formation of such meta-actions for the sub-goals follows the process of planning actions. The plan P to achieve a set of facts G (the target state of a cognitive agent) is a sequence of pairs , where $a_0...a_N$ is the set of actions of the agent, and $\sum_0, ..., \sum_N$ a set of states such that $G \subseteq \sum_N$. The plan P describes the process of solving the planning problem.

The planning problem consists of a description of the initial situation S, the final situation F, and the planning domain $D = \langle T, R, \mathrm{Pr}, A\rangle$. The description of the situation S in the spatial planning case we are considering contains the initial coordinates of the objects on the agent map, the boundaries of the map, as well as a description of the agent's state (its direction and the state of the manipulator).

The situation description F contains the final coordinates of the objects, the agent and the map constraints. Planning domain includes description of object types, description of roles R (abstract classes, for example “block?x”, “direction-start”, “region?y”), description of predicates and actions. Predicates express relationships between objects (predicate “ontable”), agent status (“manipulator empty”, “agent direction”) and spatial logic of the problem (predicates “close”, “close”, “far”). The predicates of spatial logic are interrelated in such a way that the predicate description for any distance, except for the “close” distance, consists of predicates of smaller distances to intermediate objects. Actions $\forall a \in A, a = \langle n, Cnd, Eff \rangle$ have the form, where n- the name of the action, Cnd- the facts describing the condition for the applicability of the action, but Eff - the facts that are actualized as a result of the application of the action.

The implementation of the plan is carried out in the Gazebo simulation environment, where a step-by-step execution of the plan takes place and the knowledge about the agent's capabilities obtained using the TRPO algorithm is used.

3 Related Works

The spatial representation of the planning task requires the cognitive agent to know the function of estimating distances, the possibility of representing and manipulating spatial quantities in its own world model. Most of the approaches that have been developed in this area can be divided into three areas: spatial-network approaches, approaches based on biologically plausible representation of the environment and approaches based on the psychological representation of knowledge.

Spatial-network approaches [14, 15] do not require knowledge of the environment used by cognitive agents, but are among the most common approaches in robotics when using mobile non-intelligent systems. The description of their activities is reduced to the partition of the map of the area into cells and the transitions of agents through these cells. The advantage of these approaches over the others is the speed of building an action plan, as inadequacies can be distinguished inapplicability in real conditions with a previously unknown or partially known map of the environment.

The biological approach is typical for tasks that do not require the agent's conceptual general knowledge of the environment. In most cases, the agent is not intelligent, but is able to make simple deductive assumptions about changes in the environment. In [16], a model based on studies of rat's brain activity is described [17–19]. The model describes a hierarchical environment represented by maps of different scales. Planning of movement takes into account all possible goals of achieving the goal, which requires a large amount of resources for calculating possible changes in activity, taking into account the dynamics of environmental changes. This problem was partially addressed in [20, 21], which led to the creation of the RatSLAM system, which allowed the agent to travel long distances in real terrain.

Within the framework of a psychologically plausible approach to the issue of agent action planning, problems associated with the incompleteness and inaccuracy of the description of the environment are considered. To solve the tasks set, a wide range of ways of representing the agent's knowledge is used, many of which allow approximating knowledge of the environment up to the level required for action planning. In [22, 23], an approach is considered in which the spatial model is perceived by an artificial agent as a set of the most likely actions in the current position of the agent, which approximates the representation of the spatial relationships of the artificial agent to the representation that is used by human.

In this paper we describe an approach that takes into account the merits of the hierarchical representation of the map by the agent, the possible incompleteness of knowledge about the objects of the map and the dynamics of its change. Sign-based knowledge representation formalism allows an agent to cooperate with other cognitive agents [24] and create a plan consisting of actions based on the pseudo-physical logic of the spatial relationships of the location of objects on the map. The approach uses not only actions to move the agent, but also actions that manipulate the surrounding objects. For this, the capabilities of the reinforcement learning algorithm were used. The reinforcement learning algorithms can be conditionally divided into two groups: based on the choice of strategy by maximizing the value function and based on the search for an optimal strategy in the strategy space [29]. Examples of reinforcement learning algorithms based on utility maximization are the algorithms described in [25, 26]. The algorithm of Q-learning [27] for the robotic system manipulator was applied, the reward depended on the distance from the part of the manipulator responsible for capture the target. The space of actions was discrete. An example of the application of the first approach in the continuum of action is the work [28]. To make a decision the agent trained according to the method from the first group compares the value of the utility function of each action, and in spite of the fact that this approach makes the algorithms flexible in application to various tasks, in some problems it is inefficient. Algorithms of the second group change the strategy directly without spending time on evaluating all actions. TRPO [12], used in this work, which allows to work in the continuum of action space, belongs to the second group and in the search for strategy changes parameters only in a certain neighborhood, therefore it converges along a smoother trajectory.

4 Synthesis of Behavior of Cognitive Agent

4.1 Human-like Knowledge Representation

A sign representation of the agent's knowledge was proposed in [4, 30]. The sign is a tuple of four components $s = \langle n, p, m, a \rangle$, where n - is the component of the name, p - the component of the image,m - the component of the significance, a- the component of the personal meaning. Signs can mediate both elementary objects and complex actions. The same semantic networks describe the components of the sign, whose nodes are special structures called causal matrices [6]. Causal matrices are structured sets of references to other signs and elementary features. Each of the sign components corresponds to a certain type of information, for example, the sign image component describes the process of object recognition and categorization. The significance component represents the agent's knowledge of the environment, and the component of the personal meaning describes the agent's preferences and the nature of his activity. The name component allows you to make a naming process, i.e. link the remaining components into a single logical structure.

The main task of the spatialMAP algorithm described in this paper is the synthesis of the plan for moving the agent with the sign-based world model on the map and the agent's implementation of the interaction with the objects that are located on it. In the agent's world model, the map is displayed using the signs of cells and regions [24], which are assigned to the agent in advance based on pseudo-physical logic. At the recognition stage, the agent divides it into 9 regions, the size of which depends only on the characteristics of the card itself, and associates them with the signs "Region-0" - "Region-8". Next, the agent looks at the region in which it is located. If no objects are present in the region, the agent connects the "Cell" and "Cell-4" signs with this area and builds around it a focus of attention that describes the current situation consisting of 9 cells. If there are any objects in the region, the agent recursively divides the region into 9 parts until a segment of the map containing only agent is formed. After this, the focus formation procedure described above is followed. Next, causal matrices are formed on a network of values for the "Location" and "Contain" signs (see 1), which describe the location of all regions and cells relative to the cell with the agent, as well as the objects that are in them.

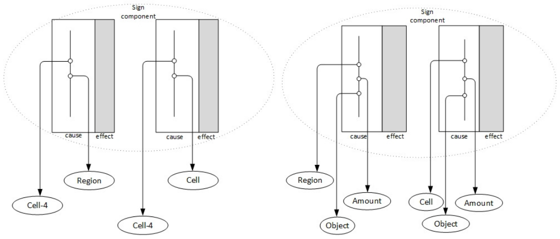

Fig. 1. Causal matrices of the the "Location" and "Contain" signs.

After this follows the process of formation of causal matrices of the initial and final situations and map that are required for the synthesis of the action plan (the process is shown in Algorithm 1). Matrices of situations consist of references to signs describing the relationship at the focus of attention of the agent, consisting of cells and the agent itself (its direction and the state of the manipulator). The map matrices describe the status of the task map on a more abstract level and contain references to the regions signs.

4.2 SpatialMAP algorithm

The process of plan synthesis is implemented using the spatialMAP algorithm and allows you to build an action plan from the initial to the finish situations of the planning task. The input to the algorithm is given a description of the situations and the planning domain that is required to refine the predicates and actions applicable in the present task.

```
1  $T_{agent} := GROUND(map, struct)$
2  $Plan := MAP\_SEARCH(T_{agent})$
3  Function MAP_SEARCH($z_{sit-cur}$,$z_{sit-goal}$,$z_{map-cur}$,$z_{map-goal}$,$plan$,$i$):
4      if $i > i_{\max}$ then
5          return $\emptyset$
6      end
7      $z_{sit-cur}, z_{map-cur} = Z^a_{sit-start}, Z^a_{map-start}$
8      $z_{sit-goal}, z_{map-goal} = Z^a_{sit-goal}, Z^a_{map-goal}$
9      $Act_{chains} = getsitsigns\,(z_{sit-cur})$
10     for $chain$ $in$ $Act_{chains}$ do
11         $A_{signif}| = abstract\_actions\,(chain)$
12     end
13     for $z_{signif}$ $in$ $A_{signif}$ do
14         $Ch| = generate\_actions\,(z_{signif})$
15         $A_{apl} = activity(Ch, z_{sit-cur})$
16     end
17     $A_{checked} = metacheck(A_{apl}, z_{sit-cur}, z_{sit-goal}, z_{map-cur}, z_{map-goal})$
18     for $A$ $in$ $A_{checked}$ do
19         $z_{sit-cur+1}, z_{map-cur+1} = Sit\,(z_{sit-cur}, z_{map-cur}, A)$
20         $plan.append(A, z_{sit-cur})$
21         if $z_{sit-goal} \in z_{sit-cur+1}$ $and$ $z_{map-goal} \in z_{map-cur+1}$ then
22             $F_{plans}.append\,(plan)$
23         end
24         else
25             $Plans := MAP\_SEARCH$
                 $(z_{sit-cur+1}, z_{sit-goal}, z_{map-cur+1}, z_{map-goal}, plan, i+1)$
26         end
27     end
```

Algorithm 1: Process of plan synthesis by cognitive agent

The process of plan synthesis consists of two main stages: the stage of replenishing the agent's world model with new signs based on the planning and learning task (step 1) and the recursive search phase (steps 2-27). The recursive search phase begins with the comparison of the current recursion step with the maximum possible (steps 4-6), if the step is less than the maximum, then the matrices of the present and target situations and the map should be obtained (steps 7-8). Next, in step 9 chains of causal matrices of signs are formed, which enter the present planning situation (in the first step of the recursion, the matrix of the initial situation). In steps 10-12, a process is underway to search for matrices of abstract (not specified within the framework of the present task) actions. For each matrix of actions found, a process of its refinement takes place on the set of matrices of signs activated in this task (steps 13 - 14). At step 15, a process of selecting the appropriate actions in the present situation occurs. Then, at step 17, among all the remaining actions, those whose application will create the situation most similar to the target one are selected. After this, in steps 19-20 the plan is replenished with the selected action and a new situation is created from the effects of the action and the signs entering the present situation. In steps 21-23, the activation of the matrices of the target situation and the map by the agent is checked, if the matrices of the target situation and the cards were activated, then the algorithm ends, if not, then in step 25 a recursive call of the plan search function takes place with an increase in the number of iterations by 1. After the planning process is over, the shortest one is selected from all the plans that have been planned and the process of its execution begins. A plan is a list of tuples that consist of actions and states. Each state include coordinates, and direction of the agent after the action is performed.

$$Plan := [(a_1, S_1), (a_2, S_2), (a_3, S_3)]$$

The plan is sent to the agent in the Gazebo environment sequentially, the agent after the execution of each of the actions returns the result of execution. If the result is positive, the next step is sent, otherwise there is replanning process.

The next step describes the process of generating personal meanings (actions), obtained with the reinforcement learning algorithm.

4.3 Learning of sub-plans

To describe the agent's interaction with the environment, the Markov decision-making process (S, O, P, r, γ) is used, where S a set of states, O set of actions, $P : S \times O \times S \rightarrow [0, 1]$ transition probability distribution, reward function and γ discounting factor. In this paper, the action space is continual, so a multidimensional normal distribution $N(\mu, \sum)$ is used to determine the strategy π, where μ and $\sum$ are specified by the neural network. Thus, the strategy π is parametrized by the weights of the θ neural network, and all functions of π are functions of θ.

The function $\eta(\theta)$, which is the evaluation of the strategy π_θ, is replaced by the following surrogate function, which links the two strategies:

$$L_\theta(\widetilde{\theta}) = \eta(\theta) + E_{\pi_\theta} \frac{\pi_{\widetilde{\theta}}(o|s)}{\pi_\theta(o|s)} (Q_\theta(s, o) - V_\theta(s))$$

where Q_θ and V_θ are the value functions of an action and a state and are defined as follows:

$$Q_\theta(\widetilde{s_t}, \widetilde{o_t}) = E_{\pi_\theta}(\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}) | s_t = \widetilde{s_t}, o_t = \widetilde{o_t})$$

$$V_\theta(\widetilde{s_t}) = E_{\pi_\theta}(\sum_{l=0}^{\infty} \gamma^l r(s_{t+l})|s_t = \widetilde{s_t})$$

Optimization L_θ with $\widetilde{\theta}$ by restriction to the average Kullback-Leibler distance entails an increase in the initial function $\eta(\theta)$. To search the optimal direction problem, the natural policy gradient method is used, which uses linear approximation L and quadratic approximation $\overline{D}_{KL}$: for $\frac{1}{2}(\theta_{old} - \theta)^T K(\theta_{old})(\theta_{old} - \theta) \leq \delta$, where $K(\theta_{old}) = \varDelta_\theta \overline{D}_{KL}^{\theta_{old}}$. Update rule:

$$\theta_{new} = \theta_{old} + \alpha K(\theta_{old})^{-1} \nabla_\theta L(\theta)|_{\theta=\theta_{old}}$$

The value $K(\theta_{old})^{-1} \nabla_\theta L(\theta)|_{\theta=\theta_{old}}$ is the solution of the equation $K(\theta_{old})x = \nabla_\theta L(\theta)|_{\theta=\theta_{old}}$with respect to x, the value α is selected by linear search for a maximum L with constraints $\overline{D}_{KL}^{\theta_{old}}(\theta_{old}, \theta) \leq \delta$.

5 Experiments in simulator

As part of the demonstration of the procedure for synthesizing the behavior of a cognitive agent, an experiment was conducted to move the robotic agent Turtlebot 2 in Gazebo to the table where a small block was placed and the block was picked up by the agent's manipulator. The plan consisted of a list of actions, including "move", "rotate" and "pick-up" actions. The process was organized using a client-server architecture, where the client was a spatialMAP planner on a remote machine that sent a message using the services of the ROS operating system to the server. Messages are about the goal move point in case the "move" action was activated, about changing the direction of the agent when activating the action "rotate" and the activation of the "pick-up" action. When the "pick-up" action was activated, the script obtained using the TRPO algorithm started working. Agent's scheme of the environment is presented in 2.

To implement the TRPO algorithm, two environments were created: a synthetic learning environment and a framework for applying the algorithm to Gazebo. Two environments have the same space of states and actions. To describe the agent's interaction with them, an example of the manipulator's grip of an object on the table is given below.

3 shows the model of a manipulator in a synthetic environment in the two-dimensional case. Points 1-4 are joints of manipulators. The action is to change the angle in one of them (in 3D, rotation around the vertical axis is added). Point B- the target point at which the agent should move point 4.

The remuneration system works as follows: if, as a result of the action, the length of the vector $\overrightarrow{4B}$ has decreased, then the agent receives a reward in the amount $\left|\overrightarrow{4B}\right|$, if not changed, then it is fined 5, and if increased, is fined 2 $\left|\overrightarrow{4B}\right|$. The state of the agent is a sequence $(\beta_1, \beta_2, \beta_3, \alpha_1, \alpha_2, \alpha_3, \overrightarrow{4B}, \overrightarrow{3B}, \overrightarrow{2B})$ (in 3D it is added α_4), where β_i are the angles in joints and α_i are the angles between the following vectors: $\alpha_1 = (\widehat{\overrightarrow{14}, \overrightarrow{1B}})$, $\alpha_2 = (\widehat{\overrightarrow{24}, \overrightarrow{2B}})$, $\alpha_3 = (\widehat{\overrightarrow{34}, \overrightarrow{3B}})$. In such a state

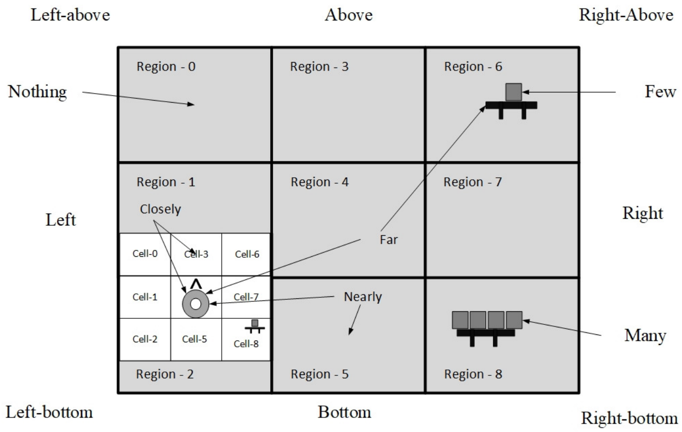

Fig. 2. Scheme of cognitive agent's spatial representations.

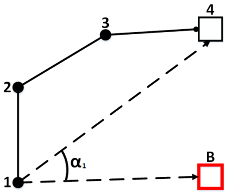

Fig. 3. Model manipulator in a synthetic environment.

space, the inequality $\alpha_1 \leq 0$ means that the goal point B is below the vector $\overrightarrow{14}$ and it is necessary to make a turn in the joint 1 by the corresponding angle. This representation of the position of the manipulator relative to the goal point makes the strategy πless dependent on position B. Because the space of states and actions for the two media are identical, the neural network trained in a synthetic environment can be used in an environment interacting with Gazebo.

6 Conclusion

The paper presents an original approach to the synthesis of cognitive agent behavior, which is realized through the interaction of a reinforcement learning approach and a planning algorithm based on a psychologically plausible way of representing knowledge. A scheme of such interaction is proposed for robotic platforms with a manipulator, and an example of the work of this approach in the task of moving a platform in space and manipulating it with external objects is demonstrated. In future works, it is planned to disclose the interaction of centralized planning algorithms for agent coalitions and reinforcement learning methods, allowing the interaction of agents with the environment in real conditions.

References

1. Laird, J.E .: The Soar Cognitive Architecture. MIT Press (2012)
2. Sun, R., Hlie, S. .: Psychologically realistic cognitive agents: taking human cognition seriously. J. Exp. Theor. Artif. Intell. 25, 65-92 (2012).
3. Pospelov, D.A., Osipov, G.S.: Knowledge in semiotic models. In: Proceedings of the Second Workshop on Applied Semiotics, Seventh International Conference on Artificial Intelligence and Information-Control Systems of Robots (AIICSR97). pp. 112. , Bratislava (1997).
4. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior control as a function of consciousness. I. World model and goal setting. J. Comput. Syst. Sci. Int. 53, 517529 (2014).
5. Osipov, G.S., Panov, A.I., Chudova, N. V.: Behavior Control as a Function of Consciousness. II. Synthesis of a Behavior Plan. J. Comput. Syst. Sci. Int. 54, 882896 (2015).
6. Panov, A.I.: Behavior Planning of Intelligent Agent with Sign World Model. Biol. Inspired Cogn. Archit. 19, 2131 (2017).
7. Emelyanov, S., Makarov, D., Panov, A.I., Yakovlev, K.: Multilayer cognitive architecture for UAV control. Cogn. Syst. Res. 39, 5872 (2016).
8. Brooks, R.A., Intelligence without representation, Artificial Intelligence 47 (1991), 139159.
9. Leontyev, A.N.: The Development of Mind. Erythros Press and Media, Kettering (2009).
10. Vygotsky, L.S.: Thought and Language. MIT Press (1986).
11. Siagian, C., Itti, L.: Biologically-Inspired Robotics Vision Monte-Carlo Localization in the Outdoor Environment. In: IEEE International Conference on Intelligent Robots and Systems. pp. 17231730 (2007).
12. Schulman, S. Levine, P. Moritz, M. Jordan, P. Abbeel. Trust region policy optimization. 2015.
13. Sham Kakade. A natural policy gradient. 2002.
14. Daniel, K., Nash, A., Koenig, S., Felner, A. (2010). Theta*: Any-angle path planning on grids. Journal of Artificial Intelligence Research, 39, 533579.
15. Palacios, J. C., Olayo, M. G., Cruz, G. J., Chvez, J. A. (2012). Thin film composites of polyallylamine-silver. Superficies y Vacio.
16. Erdem, U. M., Hasselmo, M. E. (2014). A biologically inspired hierarchical goal directed navigation model. Journal of Physiology Paris, 108(1), 2837.
17. Morris, R.G.M., Garrud, P., Rawlins, J.N.P., OKeefe, J., 1982. Place navigation impaired in rats with hippocampal lesions. Nature 297 (5868), 681683.
18. Steele, R.J., Morris, R.G.M., 1999. Delay-dependent impairment of a matching-to-place task with chronic and intrahippocampal infusion of the NMDA-antagonist D-AP5. Hippocampus 9 (2), 118136.
19. Steffenach, H.-A., Witter, M., Moser, M.-B., Moser, E.I., 2005. Spatial memory in the rat requires the dorsolateral band of the entorhinal cortex. Neuron 45 (2), 301 313.
20. Milford, M., Wyeth, G. (2010). Persistent navigation and mapping using a biologically inspired slam system. International Journal of Robotics Research, 29(9), 11311153.
21. Milford, M., Schulz, R. (2014). Principles of goal-directed spatial robot navigation in biomimetic models. Philosophical Transactions of the Royal Society B: Biological Sciences, 369(1655), 2013048420130484.

22. Epstein, S. L., Aroor, A., Sklar, E. I., Parsons, S. (2013). Navigation with Learned Spatial Affordances, 16.
23. Epstein, S. L., Aroor, A., Evanusa, M., Sklar, E. I., Parsons, S. (2015). Spatial abstraction for autonomous robot navigation. Cognitive Processing, 16, 215219.
24. Kiselev, G.A., Panov, A.I.: Sign-based Approach to the Task of Role Distribution in the Coalition of Cognitive Agents. SPIIRAS Proc. 161187 (2018).
25. Albers, A., Yan, W., Frietsch, M.. Application of Reinforcement Learning for a 2-DOF Robot Arm Control. November 2009
26. Stephen James, Edward Johns. 3D Simulation for Robot Arm Control with Deep Q-Learning. 2016.
27. C.J.C.H. Watkins. Learning from delayed rewards. 1989.
28. Shixiang Gu, and Ethan Holly, Timothy Lillicrap, Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy update. 2016.
29. Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. 1999.
30. Osipov, G.S.: Sign-based representation and word model of actor. In: Yager, R., Sgurev, V., Hadjiski, M., and Jotsov, V. (eds.) 2016 IEEE 8th International Conference on Intelligent Systems (IS). pp. 2226. IEEE (2016).