Construction of multi-agent mobile robots control system in the problem of persecution with using a modified reinforcement learning method based on neural networks

M L Patkin and G N Rogachev
Samara State Technical University, Samara, Russia
E-mail patkinm@gmail.com and gogachev@mail.ru

Abstract. A method for constructing a multi-agent control system for mobile robots based on training with reinforcement using deep neural networks is considered. Synthesis of the management system is proposed to be carried out with reinforcement training and the modified Actor-Critic method, in which the Actor module is divided into Action Actor and Communication Actor in order to simultaneously manage mobile robots and communicate with partners. Communication is carried out by sending partners at each step a vector of real numbers that are added to the observation vector and affect the behaviour. Functions of Actors and Critic are approximated by deep neural networks. The Critics value function is trained by using the TD-error method and the Actor's function by using DDPG. The Communication Actor's neural network is trained through gradients received from partner agents. An environment in which a cooperative multi-agent interaction is present was developed, computer simulation of the application of this method in the control problem of two robots pursuing two goals was carried out.

1. Introduction
Complicating technical systems and tightening the requirements for the quality of their management lead to the need for continuous improvement of control algorithms. Methods of intellectual controlling and making decisions have been significantly improved recently due to the use of artificial neural networks.

Training with reinforcement is a very powerful method of solving problems of managing complex systems, the main advantage of which is that it does not require a prior knowledge of the control object. This knowledge arises in the process of interaction of the managing agent with the control object. In tasks that are solved in a multi-agent way the complexity of management is determined not only by the complexity of the management objects, but also by the level of agent interaction, which opens up almost limitless scope for the improvement of managing agents.

In this paper, we propose an environment for simulating multi-agent cooperative interactions and a method for educating agents to develop and understand messages about how to achieve the goal.

2. The task of optimal control of a group of robots
Let us consider the problem of optimal control [1] of a group of robots in the setting of cooperative partially observable Markov game [2]. Model of control objects is set:

\[ x^i = f^i(s^i,a^i), \]  

(1)
where \( x' \in \mathbb{R}^n \); \( s' = [s'_1...s'_{n_i}] \) - state vector of robot \( i \); \( a_i \in A_i \subseteq \mathbb{R}^{m_i} \), \( a' = [a'_1...a'_{n_i}] \) - vector of available actions for each robots; \( A_i \) - compact set; \( i = 1,...,N \); \( N \) - number of robots.

Initial conditions:

\[
x'(0) = x^{(0)}, i = 1,...,N.
\]  

Group of robots aspire to achieve the set of goals. The number of goals equals to the number of robots in a group. Goals is the objects which are set by:

\[
g^i = f^i_k(P_{\text{norm}}),
\]  

where \( P_{\text{norm}} \) represents normal probability distribution.

Terminal conditions:

\[
\phi_k^i(x'(t_g), g^i(t_g)) = 0, k = 1,...,J_i, l_i \leq n_i,
\]  

where

\[
t_g = \begin{cases} t, \text{if } t < t_{\text{term}} \text{ and } \phi_k^i(x'(t), g^i(t)) = 0, k = 1,...,J_i, i = 1,...,N \\ t_{\text{term}}, \text{if } t = t_{\text{term}}, \end{cases}
\]  

\( t_{\text{term}} \) - robots control time limit.

Constraint on the state:

\[
\alpha_k(x'(t), g^i(t)) \leq 0, k = 1,...,r.
\]  

From (4) it is clear that for the robot \( i \) on the terminal condition will only depend on the state of robot \( i \) and the state of the goal \( i \) and certainly on global conditions such as (5) and (6). Simply speaking robot \( i \) is to catch the goal \( i \).

For each robot it is necessary to find such a set of actions \([a_0,...,a^i_{g}]\) that \( \phi_k^i(x'(t_g), g^i(t_g)) = 0 \).

3. Multi-agent Actor-Critic method for cooperative environment

The classic Actor-Critic method assumes that the policy function of the Actor will be formed based on value function from the Critic [3, 4]. The function of the Critic is developed according to the TD algorithm [5, 6]. In multi-agents cases (figure 1), especially in the case of POMDP, cooperation between agents begins to have a strong impact on the quality of the agents' work, because in observation \( i \) information that is more important for robot \( j \) can be found. For such situations the modification of the Actor-Critic algorithm was made.

The main idea of the modified Actor-Critic method for the case of interacting multi-agent systems in the division of the Actor into the Action Actor and the Communication Actor (figure 2). The Action Actor, as in the classic method, is responsible for the development of control actions, and the Communication Actor generates a message for the partners, which leads to more efficient management of the multi-agent system.

![Figure 1. Multi-agent reinforcement learning.](image)

![Figure 2. Scheme of modified Actor-Critic](image)
Let us consider how the modified method works. On the step $t-1$, Agent $i$ generates a message $m_i$ that is added to the observation vectors $o_i$ of the other agents (figure 3). On the basis of this, the remaining agents on the step $t$ generate the control action $a_i$. In the process of learning the neural networks of the Actors, the propagation of gradients [7] that comes from the end of the neural network to the beginning, comes from the neural network of the Action Actor to the neural network of the Communication Actor. Under the action of gradients, the Communication Actor trains to generate such messages that would lead to a reduction in the Action Actor errors.

![Diagram](image)

**Figure 3.** Modified Actor-Critic method.

### 4. Experiments

The environment is a field of 1000x1000 pixels, in which there are two robots and two targets (the environment code is based on https://github.com/harvitronix/reinforcement-learning-car). The objectives of robots: for the robot 1 it is to catch the target 1, for the robot 2 it is to understand the robot 2. The robots are equipped with device that indicate self position and position of the target, as well as with 3 sonars placed under different angles (figure 4). The goal of the robots: robot 1 is to catch target 1, robot 2 - goal 2. The target is considered caught if the robot approached it closer than 60 pixels (between the centers).

The robot is also able to "hear" what the partner says to it. The state of the robot is a vector of 9 numbers: 3 numbers for the sonars readings, 4 numbers for the position of the robot and the goal, and 2 numbers for the messages from the partner.

The robot is controlled by selecting one of three commands: "left", "forward", "right". The main idea of the experiment is that robot 1 observes goal 2, while it is supposed to catch target 1, and vice versa. This feature allows you to find out how well agents can start communicating. Robots should learn by means of a communication channel to convey information about where the target is to the partner. Experiments were conducted for static purposes and for purposes that move with variable speed and in a random direction.
The figure 4 shows green and red circles - agents, white lines emanating from them - sonars, blue and yellow circles - targets. Table 1 shows architectures of neural networks of Actors and Critic where $N_a$ - number of actions and $N_m$ length of message vector.

**Table 1.** Architectures of Actors and Critic.

|   | Action Actor | Communication Actor | Critic     |
|---|--------------|---------------------|------------|
| 1 | Fullyconnected: 256 | Fullyconnected: 256 | Fullyconnected: 256 |
| 2 | Dropout: 20% | Dropout: 20% | Dropout: 20% |
| 3 | RELU | RELU | RELU |
| 4 | Fullyconnected: 256 | Fullyconnected: 256 | Fullyconnected: 256 |
| 5 | Dropout: 20% | Dropout: 20% | Dropout: 20% |
| 6 | RELU | RELU | RELU |
| 7 | Fullyconnected: $N_a$ | Fullyconnected: $N_m$ | Fullyconnected: 1 |

**Table 2.** Experiments results. Sum of agents reward for 500 runnings.

|               | Static goals | Dynamic goals |
|---------------|--------------|---------------|
| No communication | -3314        | -16777        |
| Communication  | 37624        | 26797         |

Figures 5-8 show the results of trained agents. As it can be seen, the capture of targets was made from different positions and not in an optimal way. This non-optimality can be explained by the fact that the messages transmitted by partners did not always reflect how to act.

Figure 4. Visualisation of environment.

However, as it is shown in table 2, the result of the system trained by using the communication channel exceeds the result for the system without the communication. In this table the cells contain the values of the total reinforcement for 500 starts for two agents.
5. Conclusion
A modified neural network of the Actor-Critic method for controlling a multi-agent system in the form of interacting mobile robots is proposed. Similar work is currently underway [8, 9], which shows the promise of such an approach. In the future work on improving the quality of the learning algorithm is planned.

Acknowledgments
The reported study was funded by RFBR and Samara Region according to the research project № 17-48-630410.

References
[1] Rogachev G, Patkin M, and Rogachev N 2017 ITM Web of Conferences 10 02006.
[2] Witten I H 1977 Information and control 34(4) 286.
[3] Konda V R and Tsitsiklis J N. 2000 Actor-critic algorithms Advances in neural information processing systems.
[4] Bhatnagar S et al. 2009 Automatica 45(11) 2471.
[5] Tesauro G 1995 Communications of the ACM 38 58.
[6] Sutton R S, and Barto A G 1998 Reinforcement learning: An introduction (Cambridge: MIT press).
[7] Silver D et al. 2014 "Deterministic policy gradient algorithms." Proceedings of the 31st International Conference on Machine Learning (ICML-14).
[8] Gupta J K, Egorov M and Kochenderfer M 2017 "Cooperative multi-agent control using deep reinforcement learning." International Conference on Autonomous Agents and Multiagent Systems. Springer, Cham.
[9] Lowe R et al. 2017 "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments." arXiv preprint arXiv:1706.02275