*Advances in Intelligent Systems Research, volume 174*
**Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020)**

# A Robotic Complex Control Method Based on Deep Reinforcement Learning of Recurrent Neural Networks for Automatic Harvesting of Greenhouse Crops

Vyacheslav Petrenko
*Academic department of Organizationand Technology of Information Protection*
*North-Caucasus Federal University*
Stavropol, Russia
vip.petrenko@gmail.com

Fariza Tebueva*
*Academic department of Applied Mathematics and Computer Security*
*North-Caucasus Federal University*
Stavropol, Russia
fariza.teb@gmail.com

Mikhail Gurchinsky
*Academic department of Applied Mathematics and Computer Security*
*North-Caucasus Federal University*
Stavropol, Russia
gurcmikhail@yandex.ru

VladimirAntonov
*Academic department of Applied Mathematics and Computer Security*
*North-Caucasus Federal University*
Stavropol, Russia
ant.vl.02@gmail.com

***Abstract*—The modern development of technology determines the feasibility of the transition in agriculture from manual labor to automatic production. One of the promising areas is the automation of growing vegetable crops in greenhouse complexes. Necessary factors for intensive plant growth and unfavorable for human health, such as high temperature and humidity, as well as an atmosphere saturated with chemicals, make the task of robotizing agricultural operations urgent in this area. The method for controlling a robotic complex for automatic fruit collection in greenhouse complexes is proposed. Work in greenhouse complexes is characterized as non-deterministic and with partial observability of the environment; therefore, the deep recurrent neural network DRQN was used as the basis for the method of controlling the robotic complex. Deep learning with reinforcement was used for optimizing its weights. The presented simulation results demonstrate the efficiency of the proposed method and the need for its further development.**

***Keywords—deep reinforcement learning, recurrent neural networks, recurrent Q-networks, automation and robotics, decision- making***

## I. INTRODUCTION

Currently, there is a steady increase in the pace of automation of industry and production. The reason for this is that the cultivation and collection of vegetable plants in greenhouses is no longer possible without the introduction of automatic control systems, which should be aimed at creating and maintaining optimal conditions for the productive functioning of greenhouse complexes. To obtain a high yield, it is necessary to provide plants with timely watering, a certain amount of sunlight, favorable climatic parameters of the environment, fertilizers and nutrients. In this case, the optimal technological process should exclude the human factor [1]. On the other hand, the efficient and timely collection of greenhouse crops using robotic devices will reduce the company's costs for the wages of employees whose work occurs in adverse conditions for human life and health.

The advantages of using a mobile manipulation robotic complex for the automatic collection of tomatoes in greenhouse complexes are:

- increasing production process safety;
- loweringoperatingcosts;
- improving the efficiency and quality of the production process;
- reduction in the number of staff;
- increasing the rate of unmanned intellectual production to attract additional investment and subsidies.

Mobile manipulation robotic complexes belong to the category of service robots that can be used in the domestic sphere, when performing emergency rescue operations and research missions. Examples of the use of service robots are harvesting in agriculture, delivering food and reception of orders at catering establishments, caring for sick and disabled people, cleaning rooms, performing emergency and rescue operations (to monitor the emergency zone and eliminate the consequences of emergency situations), moon exploration and Mars, demining, military operations (fire support, delivery of supplies and medicines, transportation of the wounded, reconnaissance), diagnostics and maintenance of pipelines [2–4].

In addition to the tasks of low-level control, pattern recognition, fruit capture and navigation, an important task

Copyright © 2020 The Authors. Published by Atlantis Press B.V.
This is an open access article distributed under the CC BY-NC 4.0 license -http://creativecommons.org/licenses/by-nc/4.0/. 

of controlling a robotic complex for collecting greenhouse crops is the task of path planning [5]. The verbal statement of the problem of planning the path of the robotic complexmovement during the collection of greenhouse crops has the following form: it is necessary to find the path of movement in the operating environment from the initial position to the final one, in case there are no collisions with obstacles. However, this task turns out to be very difficult, since the robot is an articulated design. Therefore the movement of one of the links has a direct effect on the other links [6], and is important because it determines the safety of the robot's movement, ensuring the achievement of the goal without collisions with other objects the environment.

The work [7] provides an overview of the path planning methods of autonomous mobile robots. Path planning methods for autonomous moving objects can be divided into graph-analytical methods based on representations of the configuration space, methods based on Voronoi diagrams [8–9], methods based on graph theory [10] and intelligent methods involving the use of fuzzy logic [11 ], heuristics [12] and artificial neural networks [13]. In [14], a path planning method based on a strategy for updating learning rules was proposed.

The complexity of the application of these methods is due to the non-determinism and partial observability of the external environment of the robotic complex. Ensuring full observability of the greenhouse complex is a complex and resource-consuming task. The use of a robotic complex of some model of the greenhouse complex is more effective. The model of the greenhouse complex should take into account the patency map of various sections of the greenhouse complex, microclimatic conditions, the function of the fruits ripeness over time and microclimatic conditions, the current distribution of the fruits, and much more. The construction of such a model using analytical methods is difficult and expensive. A promising solution to this problem is the use of recurrent neural networks that can detect hidden models of the external environment. An additional advantage of this approach is the possibility of using a single apparatus of neural networks to solve other problems facing the robotic complex for collecting greenhouse crops. It includes pattern recognition and fruit capture. The use of a single apparatus of neural networks allows in the future the construction of a monolithic architecture of a control system for a robotic complex.

## II. STATEMENT OF THE PROBLEM

The aim of this work is to increase the level of automation of the fruit-picking process in greenhouse complexes by developing a control method for a robotic complex for picking fruit based on learning reinforced by deep recurrent neural networks (RNN).

Reinforced learning methods are teaching methods of a certain model (or agent) by trial and error. This approach to training formalizes the methodology of rewards or fines imposed on the trained model in order to accordingly increase or decrease the probability of specific results in the future. The cycle of interaction of the agent with the environment is illustrated in Figure 1.

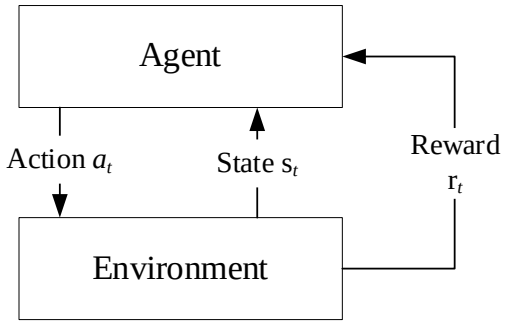


Fig. 1. The cycle of interaction between the agent model and the environment

The main objects of the experiment in reinforcement learning are the agent and the environment [15]. The environment is usually called the system or world which interacts with the agent. At each iteration of training, the agent operates with his observation of the parameters of a certain state of this world (possibly incomplete) and then decides what action to take. The environment also varies depending on the actions of the agent. However , it can also change independently. In addition, the agent receives some response from the environment, called a reward. The reward is a numerical value indicating how good or bad the current state of the environment is for the agent. The goal of the agent is to maximize the final total reward.

Let us describe the formalized scheme of the teaching method. Deep learning with reinforcement of the RNN is used to solve problems posed as a partially observable Markov decision-making process [16], [17]:

$$(S, A, R, T, \Omega, O), \quad (1)$$

where:

- $S$ – is a set of all states, $s_t \in S$ – э is the state of the agent at time $t$;
- $A$ –set of all possible actions, $a_t \in A$ – is an agent action committed at a point in time $t$;
- $R: S \times A \times S \to R$ – is a reward function
- $R(s_t, a_t, s_{t+1})$; this reward becomes known to the agent after reaching the state $s_{t+1}$;
- $T : S \times A \times S \to [0,1]$ – transition function where $T(s, a, s')$ – probability of transition to state $s'$from state$s$after performing an action $a$;
- $\Omega$ – set of possible observations $o \in \Omega$by the state agent $S$;
- $O: S \times \Omega \to [0,1]$ – probability of obtaining the observed state $o$, if the environment is in a state$s$.

When training with reinforcement, the method makes a random choice of actions. According to the rule $\pi(o, a) = P(a|o)$ – is the probability of a choice by method $a$ in observable condition $o$.The purpos e of the training is to determine an action selection strategy $\pi : \Omega \times A \to [0,1]$, which maximizes the total reward. Cost function $V_\pi(o_t)$ for the strategy $\pi$ is defined as the amount of discounted rewards that can be obtained starting from the observed state $o_t$ in accordance with the strategy $\pi$:

$$V_\pi(o_t) = R(o_t, a_t) + \gamma R(o, a_{t+1}) + \\ +\gamma^2 R(o_{t+2}, a_{t+2}) + \ ... = \sum_{i=0}^{\infty} \gamma^i R(o_{t+i+1}, a_{t+i+1}), \quad (2)$$

where $R(o_t, a_t)$ – the reward received from the transition from th eobserved state $o_t$ into observable state $o_{t+1}$, and$0 \leq \gamma \leq 1$ – the discount factor, providing a decrease in the significance of a more distant reward. The optimal cost function $V^*(a)$ is determined by the following formula:

$$V^*(a) = \max_\pi(V_\pi(a)). \quad (3)$$

The optimal cost function $V^*(a)$ satisfies Bellman's equality:

$$V^*(o) = \max_\pi\big(R(o, a)\big) + \gamma V^*(o'), \quad (4)$$

where $o'$ – the observed state that the agent goes into after selecting an action $a$ in the observed state $o$. Thus, the task is to determine the optimal policy $\pi^*(o) : \Omega \rightarrow$ A, which allows to maximize the total agent reward:

$$\pi^*(o) = \operatorname*{argmax}_a\big(R(o, a)\big) + \gamma V^*(o'). \quad (5)$$

## III. Methods

### A. Limitations

This article proposes a method for controlling a robotic complex that provides autonomous bypass of the greenhouse complex in order to collect fruits of a given ripeness. However, issues such as the search for tomatoes with the help of technical vision, the capture of fruits and further manipulation with them are not considered. Methods for performing these tasks are proposed in articles [18–23].

The greenhouse complex is consideredas a discrete space — a field with the size $m \times n$ which each fruit cell contains a fruitplant. On the fruit plant there is a certain number of *k* fruits, taken equal to 1 to simplify the environment (in a more complex case, the value can be increased, as a result of that the robot will need to decide whether to collect all the fruits from the plant, or only a part of them that will fit in a container for transportation). The limitation in the problem is the carrying capacity $w = 2$ of the mobile manipulation robotic complex. As a result of that, after collecting the fruits from two cells, the robotic complex is forced to return to the starting point for shipping the fruits and recharging if necessary. The speed of the work is taken constant. An additional limitation is the partially observed external environment, that is, a complete map of the greenhouse complex is initially absent. Observations are formed on the basis of studies of neighboring cells in the vicinity of the first rank.

Accordingly, to estimate the number of fruits in the cell and decide on the collection of fruits or return to the base can only be in the neighboring cell from the observed one. Environments that provide an agent with an incomplete set of data as observations are called partially observable Markov decision-making processes. Methods for solving such problems are more complicated than in completely observable environments, however, this limitation of the model most closely matches the real work of the robotic complex when harvesting tomatoes in the greenhouse complex. An illustration of the partially observed medium is shown in Figure 2. A blue cell corresponds to a robotic complex, white cells correspond to plants without fruits, green cells correspond to plants with fruits, red cells correspond to obstacles. The observed part of the external environment for the robotic complex is indicated in yellow.

In the process of moving through the greenhouse complex, the robotic complex can detect fruits and determine the degree of ripeness. The aim of the robotic complex is to collect as many fruits as possible in the shortest possible time. Given a fixed speed, this goal corresponds to minimizing the path of movement of the robotic complex. The following actions are available to the robotic complex - moving along the space connected graph, as well as collecting fruits. This task has some similarities with the traveling salesman problem, however, the main difference in this problem statement is the ability to visit the graph vertices again due to the limitations listed above.

### B. A deep learning method with RNN reinforcement for controlling a robotic complex

From the point of view of reinforcement learning, the robotic complex is a trained agent. As an approximator of the function that converts the input data of the agent to the output, RNN is used. For RNN learning, the classical Q-learning method can be used [24]. However, in this case, the agent's observations will have a high dimension, which will complicate the convergence of the method and will require considerable time for training. For this reason, it was proposed to use the approximation function [14], which allows the method to be applied to larger problems, even when the state space is continuous. An artificial neural network [25] was chosen as an approximator, as implemented in the DQN method in [26]. The main task of the agent is to bypass all the vertices of the field with the need to periodically return to the base. That means that the memory buffer can be used to store a sequence of examined vertices, and on their basis a decision will be made on planning the movement to a new vertex. Thus, this fact necessitates the use of the Deep reccurent Q-network learning method [17], where an LSTM layer [27] in a neural network is added in front of a fully-connected layer (fully-connected layer).

Each cell of the observed state of the environment is encoded with one color pixel and is fed to the RNN input. With each new episode of training, the agent randomly appears at the initial vertex marked in gray on the field map (Fig. 2). In the process of functioning, the agent has 3 possible options: moving forward, backward, left and right. For each move, the agent receives a conditional penalty $r_m$, reflecting the cost of moving. When an agent reaches a plant cell with fruits painted in green, the fruit is automatically collected, while the cell in the environment ceases to be marked as containing fruits and becomes empty - painted in white. The agent receives a reward $r_f$ for collecting the fruit.. Red color indicates obstacle cells. For getting into a cage with an obstacle, i.e. a collision, the agent receives a fine $r_o$. When the agent reaches the capacity limit *w*its color becomes darker (black), which signals the RNN that it is necessary to return to the initial peak to receive a reward. Upon reaching the initial peak *B* with a full load, the agent receives a reward $r_g$ , and the episode is considered completed. An example of the greenhouse complex map is shown in Figure 2. The part of the environment observed by the agent is marked in yellow.

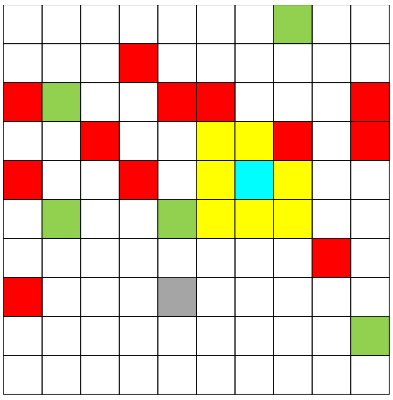

Fig. 2. An example of the environment state

The architecture of the used RNN is presented in Figure 3. RNN is used to evaluate the $Q$- value, which reflects the maximum cumulative remuneration received by the agent in the future when the action $a$ is performed in the observed state $o$. When the RNN is initialized, its elements are assigned a random weight value, which determines a random value $Q$ in the initial state. During the training process, the RNN weights are optimized in such a way as to ensure a change in the output values according to the following equation:

$$Q(o,a) := Q(o,a) + \alpha\left(r + \gamma \max_{a'} Q(o',a') - Q(o,a)\right), \quad (6)$$

where $Q(o,a)$ – potential value of action choice $a$ in partially observable state $o$, $r$ – remuneration for an action $a$in partially observable state$o$, $\alpha$ – iterative rate of change of value $Q$ in the learning process (learning rate).

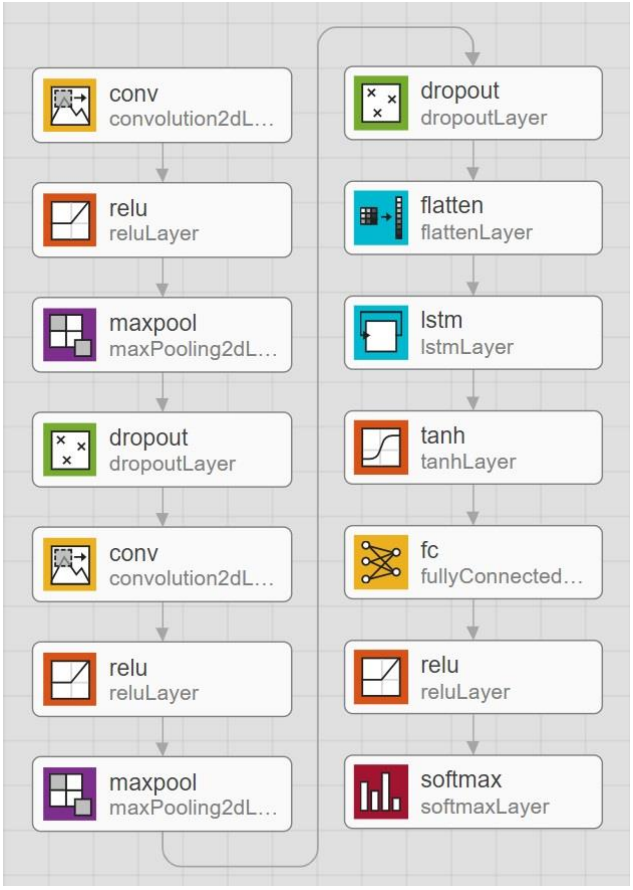


Fig. 3. The architecture of the used recurrent neural network

To ensure a balance in the learning process between an agent exploring new observable states and using accumulated knowledge in the form of $Q$-values, the agent does not always choose the action with the highest $Q$-value. An indicator $0 \le \epsilon \le 1$ is introduced into the learning process, reflecting the relative importance of obtaining new experience and using the accumulated one. At each step, a random variable $0 \le k \le 1$ is generated. If $k > \epsilon$, the agent performs the action $a$ with the highest $Q$-value, otherwise it performs a random action $a$. In the learning process, the value $\epsilon$ decreases from a certain initial value to zero, which means a complete transition from exploring the space of possible states to using the accumulated experience.

## IV. RESULTS

To test the proposed method, the design, training and modeling of the RNN and the greenhouse complex were carried out in the Python programming language using the SciPy library of scientific calculations and the Tensorflow framework. The code was placed in the repository [28]. During the simulation, a computer was used with the following characteristics: Intel Core i7-8550U 1.8GHz processor, 8Gb RAM. The simulation parameters are presented in table 1. The parameter values are selected experimentally to optimize the learning speed.

TABLE I. SIMULATIONOPTIONS

| Parameter | Value |
|---|---|
| Fieldsize$m \times n$ | $10 \times 10$ |
| The number of cells with fruit plants | 2-10 |
| The number of cells with obstacles | 5-20 |
| Numberoftrainingepisodes | 20000 |
| The maximum number of steps in each episode | 200 |
| Initialvalue$\epsilon$ | 0,99 |
| Learning rate$\alpha$ | 0,3 |
| Discountfactor$\gamma$ | 0,99 |
| The Agent reward $r_g$ | 1000 |
| Agent reward $r_f$ for fruit collection | 250 |
| Penalty $r_o$for collision with an obstacle | -300 |
| Penalty $r_m$ for each move | -10 |

In each training episode, from 2 to 10 fruit plants, as well as from 5 to 20 obstacles, are randomly placed on the field. Also, the positions of the agent and the initial vertex are randomly set, which initially coincide, since the robotic complex must begin and finish work at this initial vertex.

Agent training results were obtained using the TensorBoard library and are presented in Figures 4-7. On these graphs, the abscissa axis indicates the number of episodes of model training, and the ordinate axis indicates the value of a specific indicator of the training process. Figure 4 shows the accuracy of RNN training, which is a measure of the accuracy of approximating $Q$-values using RNN. The established accuracy value between 0.7 and 0.8 indicates that the structure of the used RNN can be improved.

The fundamental solution of this problem for an arbitrary greenhouse complex requires the development of a method of structural-parametric synthesis of RNN for specific characteristics of the greenhouse complex. The results of changes in agent remuneration are presented in Figures 5-7. So, Figure 5 shows a graph of the change in the maximum value of the reward received by the agent. Figure 6 shows a graph of the change in the minimum value of the reward received by the agent. Figure 7 shows the average agent reward per episode. The settled value of the reward for the episode means the end of the agent's training, and the positiveness of the established value means the absence of collisions with obstacles and the task of collecting fruit.


Fig. 4. RNN training accuracy

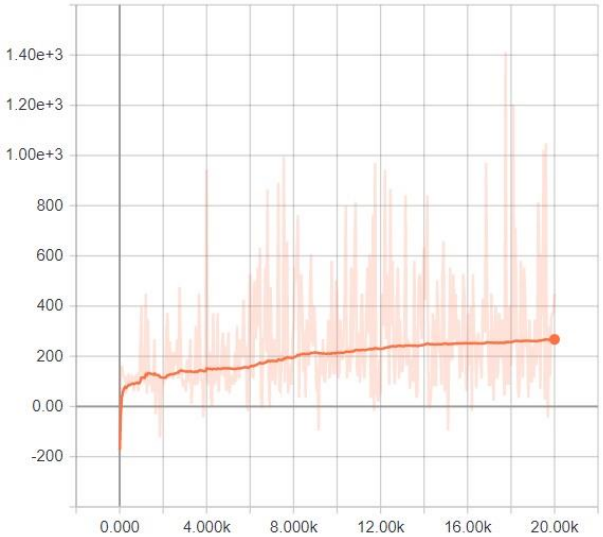


Fig. 5. Schedule for changing the maximum agent reward per episode

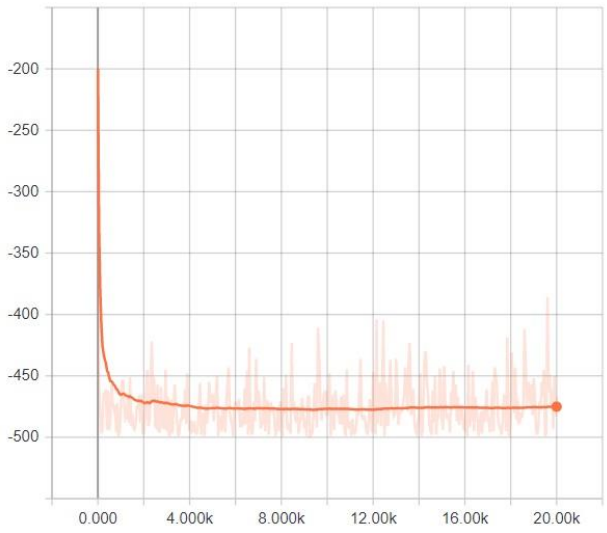


Fig. 6. Schedule for changing the minimum agent reward per episode

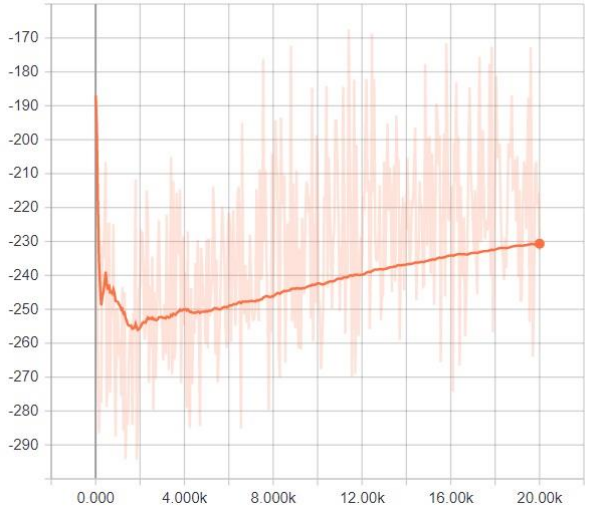


Fig. 7. Schedule change of the average agent reward per episode

Summarizing the obtained results, it can be argued that the developed method is suitable for the automatic collection of greenhouse crops, however, some adjustments are required to integrate the method into the robotic complex. In particular, refinement of the used model of the greenhouse complex is required. Firstly, in the current model, the distribution of obstacles and fruits at the beginning of each episode is generated independently. Although in real greenhouse complexes some of the obstacles are static, so their position should not change from episode to episode. Also, the location of uncollected fruits should not change from episode to episode. Secondly, it is necessary to include ripeness of fruits in the observed state of the environment. According to the current model, only ripe fruits can be present in the field cell, i.e., the ripening process is not taken into account. In this regard, for the agent, the fruits appear in the cells randomly , and the model of the environment hidden in the agent cannot predict their appearance. If we include in the observed state of the environment the degree of ripeness of the fruits in the field cell, on the basis of this, we can train the agent to predict the appearance of ripe fruits

in new cells and take this predictive value into account to maximize reward when choosing the direction of movement.

In addition, it is necessary to develop a method of structural-parametric synthesis to optimize the structure of the RNN for the characteristics of specific greenhouse complexes.

## V. CONCLUSION

This paper presents a control method for a robotic complex for the automatic collection of greenhouse crops, which use will increase the level of automation of the fruit collection process in greenhouse complexes. The method is based on deep learning supported by recurrent neural networks. The simulation results confirmed the efficiency of the developed method. Experimental studies of the physical model of the robotic complex can be carried out only after integration into the control system of the robotic complex of lower-level methods that are responsible for finding the fruit using technical vision, capturing the fruit and further manipulating the object, as well as after finalizing the model of the environment used in training. Research on these issues will be addressed in future work.

## REFERENCES

[1] V. V. Dvorny, V. S. Kostenko, and V. A. Kvartnikov, 'The use of solar hybrid installations for power supply of a "smart" greenhouse at agricultural enterprises in the Krasnodar Territory', *Agricultural innovation*, pp. 115–117, 2016.

[2] I. M. Kutlubaev, A. A. Bogdanov, N. V. Novoseltsev, M. V. Krasnobaev, and O. A. Saprykin, 'Control system of the anthropomorphous robot for work on the low-altitude earth orbit', *Int. J. Pharm. Technol.*, vol. 8, no. 3, pp. 18193–18199, 2016.

[3] V. Petrenko, S. Ryabtsev, F. Tebueva, O. Trofimuk, and M. Gurchinsky, 'Development of the Structure of the Upper-Limb Exoskeleton for Operator's Motion Capture With Master-Slave Control', *Atl. Press Ser. Adv. Intell. Syst. Res.*, 2019, doi: https://doi.org/10.2991/itids-19.2019.28.

[4] G. R. Shakhmametova and N. I. Yusupova, 'Intelligent Technologies Integration in the Task of Unaccented Trajectories Search in Robotics', *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 538–543, Jan. 2018, doi: 10.1016/j.ifacol.2018.11.270.

[5] 'Liu W. Path Planning Methods in an Environment with Obstacles (A Review). Mathematics and Mathematical Modeling. 2018;(1):15-58. (In Russ.) https://doi.org/10.24108/mathm.0118.0000098.' .

[6] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, 'Path planning and trajectory planning algorithms: A general overview', in *Mechanisms and Machine Science*, vol. 29, Kluwer Academic Publishers, 2015, pp. 3–27.

[7] N. Sariff and N. Buniyamin, 'An overview of autonomous mobile robot path planning algorithms', in *SCOReD 2006 - Proceedings of 2006 4th Student Conference on Research and Development 'Towards Enhancing Research Excellence in the Region'*, 2006, pp. 183–188, doi: 10.1109/SCORED.2006.4339335.

[8] F. Peralta *et al.*, 'Development of a Simulator for the Study of Path Planning of An Autonomous Surface Vehicle in Lake Environments', in *IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON 2019*, 2019, doi: 10.1109/CHILECON47746.2019.8987711.

[9] 'Garrido, Santiago & Moreno, Luis. (2015). Mobile Robot Path Planning using Voronoi Diagram and Fast Marching. 10.4018/978-1-4666-8693-9.ch003.'

[10] J. Yu, 'Average case constant factor time and distance optimal multi-robot path planning in well-connected environments', *Auton. Robots*, Mar. 2019, doi: 10.1007/s10514-019-09858-z.

[11] M. A. Soliman, A. T. Azar, M. A. Saleh, and H. H. Ammar, 'Path Planning Control for 3-Omni Fighting Robot Using PID and Fuzzy Logic Controller', in *Advances in Intelligent Systems and Computing*, 2020, vol. 921, pp. 442–452, doi: 10.1007/978-3-030-14118-9_45.

[12] Q. Wu *et al.*, 'Real-time dynamic path planning of mobile robots: A novel hybrid heuristic optimization algorithm', *Sensors (Switzerland)*, vol. 20, no. 1, Jan. 2020, doi: 10.3390/s20010188.

[13] R. Fareh, M. Baziyad, M. H. Rahman, T. Rabie, and M. Bettayeb, 'Investigating Reduced Path Planning Strategy for Differential Wheeled Mobile Robot', *Robotica*, vol. 38, no. 2, pp. 235–255, Feb. 2020, doi: 10.1017/S0263574719000572.

[14] H. van Hasselt, 'Reinforcement learning in continuous state and action spaces', in *Adaptation, Learning, and Optimization*, vol. 12, Springer Verlag, 2012, pp. 207–251.

[15] 'Josh Achiam Part 1: Key Concepts in RL — Spinning Up documentation. URL: https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*. MIT Press, 1998.

[17] M. Hausknecht and P. Stone, 'Deep recurrent q-learning for partially observable MDPs', in *AAAI Fall Symposium - Technical Report*, 2015.

[18] 'Arad, B, Balendonck, J, Barth, R, et al. Development of a sweet pepper harvesting robot. J Field Robotics. 2020; 1– 13. https://doi.org/10.1002/rob.21937'. .

[19] 'Cardenas-Weber, M., Stroshine, R. L., Haghighi, K., & Edan, Y. (1991). Melon material properties and finite element analysis of melon compression with application to robot gripping. Transactions of the ASAE, 34(3), 920-0929.' .

[20] 'Bac, C.W., van Henten, E.J., Hemming, J. and Edan, Y. (2014), Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. J. Field Robotics, 31: 888-911. doi:10.1002/rob.21525'. .

[21] V. Petrenko *et al.*, 'Analysis of the effectiveness path planning methods and algorithm for the anthropomorphic robot manipulator', in *2019 International Siberian Conference on Control and Communications, SIBCON 2019 - Proceedings*, 2019, doi: 10.1109/SIBCON.2019.8729657.

[22] V. I. Petrenko, F. B. Tebueva, M. M. Gurchinsky, V. O. Antonov, and J. A. Shutova, 'Solution of the dynamics inverse problem with the copying control of an anthropomorphic manipulator based on the predictive estimate of the operator's hand movement using the updated Brown method', in *IOP Conference Series: Materials Science and Engineering*, 2018, doi: 10.1088/1757-899X/450/4/042013.

[23] V. I. Petrenko, F. B. Tebueva, M. M. Gyrchinsky, V. O. Antonov, and J. A. Shutova, 'The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only', in *IOP Conference*

*Series: Materials Science and Engineering*, 2018, doi: 10.1088/1757-899X/450/4/042016.

[24] R. S. Sutton, 'Learning to predict by the methods of temporal differences', *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, Aug. 1988, doi: 10.1007/bf00115009.

[25] C. Tesau and G. Tesau, 'Temporal Difference Learning and TD-Gammon', *Commun. ACM*, vol. 38, no. 3, pp. 58–68, Mar. 1995, doi: 10.1145/203330.203343.

[26] V. Mnih *et al.*, 'Human-level control through deep reinforcement learning', *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[27] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory', *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[28] A. Juliani, 'DeepRL-Agents'. [Online]. Available: https://github.com/awjuliani/DeepRL-Agents/. [Accessed: 01-Apr-2020].