ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 174

Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020)

Method of Controlling the Movement of an Anthropomorphic Manipulator in the Working Area With Dynamic Obstacle

Vyacheslav Petrenko
Institute of Information Technology and Telecommunications
North-Caucasus Federal University
Stavropol, Russia
vip.petrenko@gmail.com

Fariza Tebueva*
Institute of Information Technology and Telecommunications
North-Caucasus Federal University
Stavropol, Russia
fariza.teb@gmail.com

Sergey Ryabtsev
Institute of Information Technology and Telecommunications
North-Caucasus Federal University
Stavropol, Russia
losde5530@gmail.com

Mikhail Gurchinsky
Institute of Information Technology and Telecommunications
North-Caucasus Federal University
Stavropol, Russia
gurcmikhail@yandex.ru

***Abstract*—Currently, the rapidly developing direction of anthropomorphic robotics attracts great interest of developers. This is due to the need to perform routine, harmful and hazardous types of work without direct human intervention, which is the key to ensuring the safety of the tasks performed. The article discusses the issue of optimizing the trajectory of the manipulator when performing operations in the working area with an obstacle. To achieve this goal, an algorithm for controlling the movement of an anthropomorphic manipulator in a working area with dynamic obstacles is proposed using deep learning technology with amplification of a convolutional artificial neural network based on the DQN learning algorithm. This algorithm is more scalable than peers because it can be used for a wide variety of path planning problems in both deterministic and non-deterministic environments. The results of modeling the operation of a manipulator with seven rotational degrees of mobility in the working area with a typical obstacle in the form of a sphere are presented. The presented simulation results demonstrate the effectiveness of the proposed method and the need for its further development.**

Keywords—anthropomorphic manipulator, dynamic environment, dynamic obstacles, machine learning, convolutional neural networks

I. INTRODUCTION

The most urgent task of robotics today is replacing a person with a robot when doing monotonous hard work (industrial robots), or replacing a person when performing operations in conditions dangerous to life and health (extreme robotics). One of the main areas of extreme robotics is service anthropomorphic robots (SAR). The most widely used are copying and autonomous control methods for SAR [1]. With copy control, control signals are remotely generated by the operator. This control method allows you to secure the execution of operations for a person, but still requires his full involvement. An alternative is autonomous SAR (ASAR), used in the field of catering, hotel business and in everyday life. Target operations are performed by ASAR using an anthropomorphic manipulator. Management of an anthropomorphic manipulator includes tasks such as path planning, solving direct and / or inverse kinematics and dynamics problems.

When performing a series of targeted operations in robotics, the methodological apparatus of training with reinforcements has established itself. Training methods are used in robotics for a large number of tasks, such as the planning of the path of mobile robots [2], the planning of the movements of autonomous underwater vehicles [3], adaptive planning of the path of space manipulators in an unknown environment [4, 5], the planning of the trajectory of unmanned aerial vehicles [6], analysis of motion trajectories in the exoskeleton of the lower extremities [7]. Thus, the development of deep learning methods is an urgent task for control tasks in robotics.

The aim of this work is to develop a method for controlling the movement of an anthropomorphic manipulator in the working area with a dynamic obstacle based on deep training with reinforcement. Known methods for planning the path of an anthropomorphic manipulator in the working area with an obstacle, based on solving problems of kinematics and dynamics [8–10], based on an analytical approach. However, the most promising are methods for solving these problems using such an artificial intelligence method as deep learning, due to the following advantages:

- ability to parallelize information processing;
- the possibility of self-learning, ie create generalizations;
- the ability to solve problems with unknown patterns;
- noise immunity in input data;
- the ability to adapt to environmental changes;
- the possibility of potentially ultra-high performance and fault tolerance in the hardware implementation of a neural network.

In this paper, an attempt is made to use the methodological apparatus of training with reinforcement to

control the movement of an anthropomorphic manipulator in the working area with dynamic obstacles. The solution of the problems of controlling the movement of an anthropomorphic manipulator in the working area using deep learning technologies with reinforcement are considered in articles [11–14]. In [11], an approach is considered that uses, as a model that describes trajectories, interpolation by splines. Model parameters are set using the reinforcement learning algorithm. The article deals with trajectory tracking and trajectory planning based on a fuzzy output system, and the RL algorithm is used to plan the path to prevent collisions with obstacles [12]. The study [13] considers an asynchronous version of the deep learning algorithm with reinforcement for solving complex tasks performed by manipulators. In [14], a method for planning the movement of robotic manipulators based on reinforced learning is presented, the task of planning movement is considered as a Markov decision-making process.

This article proposes a method for controlling the movement of an anthropomorphic manipulator based on deep learning with reinforcement of a convolutional artificial neural network based on the DQN learning algorithm [7]. The convolutional neural network is used to generate control signals for the drives of the anthropomorphic manipulator in order to achieve the target position and avoid collisions with dynamic obstacles.

II. MATERIALS AND METHODS

The ASAR manipulator is considered as a control object, the kinematic scheme of which is similar to the kinematic scheme of a human hand [15, 16] and is shown in Figure 1. Similar manipulators are used in AR-600, FEDOR robots and are considered in detail in [17–21]. This manipulator has 7 rotational degrees of mobility. The axes of degrees of mobility 1–3 intersect at the center of the shoulder joint, the axis of degree of mobility 4 passes through the elbow joint, the axes of degrees of mobility 5–7 intersect at the carpal joint.

The position and orientation of the coordinate systems associated with the links are shown in figure 2, the Denavit-Hartenberg parameters are shown in table 2. Known values for the manipulator are the lengths of the shoulder, elbow and wrist parts – l_{B1-B2}, l_{B2-B3}, l_{B3-B4}, respectively. View length data l_{Bi-Bj} correspond to the distance between the nodal points B_i and B_j kinematic model.

The input signals for the control object are efforts $\mathbf{T}(t) = \{\tau_i(t), i = 1,..,7\}$, where τ_i – the forces developed in the drives of the anthropomorphic manipulator, i – mobility degree number. The output signals of the control object are the cartesian coordinates of the radius vectors of the shoulder joint $\boldsymbol{A_0}$, elbow joint $\boldsymbol{A_1}$, wrist joint $\boldsymbol{A_2}$ and center of grasp $\boldsymbol{A_3}$, figure 1.

The goal of management is to achieve the center of gripping $\boldsymbol{A_3}$ anthropomorphic manipulator of some target point $\boldsymbol{p_{target}}$ subject to the absence of collisions with dynamic obstacles. Dynamic obstacles in this work are approximated by a set of one sphere centered at a point $\boldsymbol{O_j}$ and radius r_j, where $j = \overline{1, m}$.

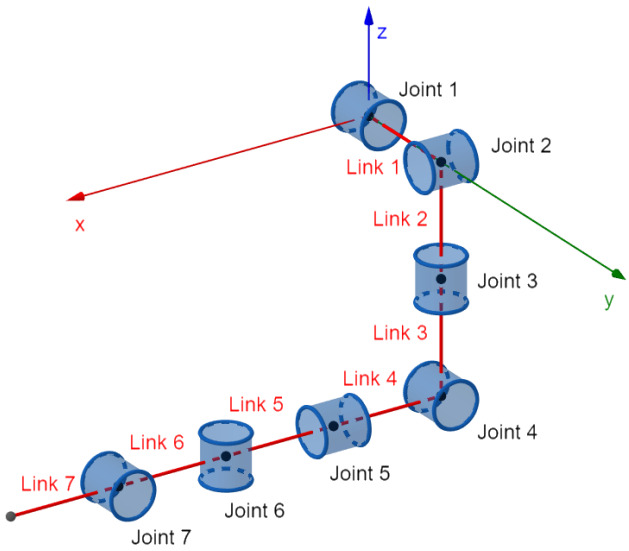

Figure 1 – Kinematic diagram of an anthropomorphic manipulator

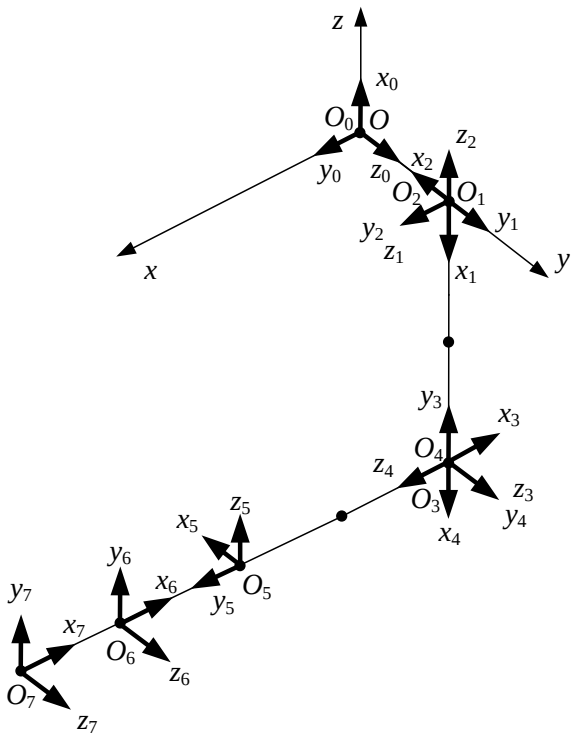

Figure 2 – Link-related coordinate systems

We accept the following notation.

${}^{i}\mathbf{T_j}$ – homogeneous transformation matrix from the ith coordinate system to i, $i < j$, compiled in accordance with the presentation of Denavit-Hartenberg:

$$ {}^{i}\boldsymbol{T_j} = \prod_{k=i+1}^{j} {}^{i-1}\boldsymbol{A_i}, i < j, $$

$$ {}^{i-1}\boldsymbol{A_i} = \boldsymbol{T_{z,\theta}}(\theta_i)\boldsymbol{T_{z,d}}(d_i)\boldsymbol{T_{x.a}}(a_i)\boldsymbol{T_{x,\alpha}}(\alpha_i), $$

$$\boldsymbol{T}_{z,\theta}(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 & 0 \\ sin(\theta) & cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{T}_{z,d}(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{T}_{x.a}(a) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{T}_{x,\alpha}(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) & 0 \\ 0 & sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where ${}^{\mathbf{i-1}}\mathbf{A_i}$ – homogeneous complex transformation matrix for adjacent coordinate systems;

$\mathbf{T_{z,\theta}(\theta)}$ – homogeneous matrix of elementary rotation about an axis z on the corner θ;

$\mathbf{T_{z,d}(d)}$ – homogeneous matrix of elementary shift along the z axis by a distance d;

$\mathbf{T_{x.a}(a)}$ – homogeneous matrix of elementary shift along the x axis by a distance a;

$\mathbf{T_{x,\alpha}(\alpha)}$ – homogeneous matrix of elementary rotation about the x axis by an angle α.

$\mathbf{T_i}$ – matrix of transformation into a global coordinate system from the ith coordinate system. This matrix can be found by the following formula:

$$\boldsymbol{T_i} = \boldsymbol{T_0}\ {}^{\mathbf{0}}\boldsymbol{T_i}, i > 0,$$

$$\mathbf{T_0} = \mathbf{T_x}(-90°)\mathbf{T_z}(-90°),$$

$$\boldsymbol{T'_i} = \boldsymbol{T'_0}\ {}^{\mathbf{0}}\boldsymbol{T'_i}, i > 0,$$

$$\boldsymbol{T'_0} = \boldsymbol{T_{B1-C1}}\boldsymbol{T_x}(-90°)\boldsymbol{T_z}(-90°),$$

$$\boldsymbol{T_{B1C1}} = \begin{pmatrix} 1 & 0 & 0 & B_1C_{1\,x} \\ 0 & 1 & 0 & B_1C_{1\,y} \\ 0 & 0 & 1 & B_1C_{1\,z} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

where $\mathbf{K_j^i}$ – uniform point radius vector $\mathbf{K_j}$ in the coordinate system;

$\mathbf{K_j}$ – uniform point radius vector $\mathbf{K_j}$ in the global coordinate system;

$K_{j\,x}^i$, $K_{j\,y}^i$, $K_{j\,z}^i$ – projection radius vector $\mathbf{K_j^i}$ on axis x, y, z i-in the coordinate system;

$K_{j\,x}$, $K_{j\,y}$, $K_{j\,z}$ – projection radius vector $\mathbf{K_j^i}$ on axis x, y, z global coordinate system.

The range of possible angles for each articulation of the manipulator shown in figure 1 is shown in table 1.

Table 1. Range of possible angles of articulations of the manipulator

Joint	Minimum angle	Maximum angle
Joint 1	-90	90
Joint 2	-20	90
Joint 3	-45	45
Joint 4	-70	90
Joint 5	-90	90
Joint 6	-45	45
Joint 7	-70	70

Table 2 – Denavit-Hartenberg parameters for the kinematic model of the operator's hand

Joint	$\boldsymbol{\theta_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{a_i}$	$\boldsymbol{d_i}$
1	180°	90°	0	0
2	−90°	90°	0	0
3	−90°	90°	0	$-l_{B1-\mathrm{B2}}$
4	−90°	90°	0	0
5	−90°	90°	0	$l_{B2-\mathrm{B3}}$
6	−90°	90°	0	0
7	0°	0°	$-l_{B3-B4}$	0

According to position and orientation i link in the coordinate system associated with $(i-1$ the link is described by four parameters: $a_i, d_i, \alpha_i, \theta_i$,where a_i – distance between the intersection of the axis z_{i-1} with axis x_i and the start i coordinate systems counted along the axis x_i, i.e. the shortest distance between axles z_{i-1} and z_i; d_i – distance between the intersection of the axis z_{i-1} with axis x_i and the beginning $(i-1)$ coordinate systems counted along the axis z_{i-1}; α_i – angle to rotate the axis z_{i-1} around axis x_i, so that it becomes aligned with the axis z_i (the sign is determined in accordance with the rule of the right hand); θ_i – angle to rotate the axis x_{i-1} around axis z_{i-1}, so that it becomes aligned with the axis $x_i x_i$ (the sign is determined in accordance with the rule of the right hand). Thus, the kinematic structure and position of the AP manipulator are described by four vectors a, d, α, θ . Vectors a, d, α are constant for the selected kinematic scheme of the manipulator AR. The vector of generalized coordinates that uniquely determines the position of the AP manipulator is the vector of rotation angles $\boldsymbol{\theta}$.

As a methodological apparatus, deep learning with reinforcement was used. The classic reinforcement learning scheme is shown in figure 3.

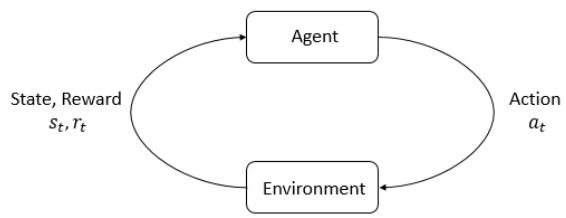

Figure 3 - Classical reinforcement training scheme

The main objects of the experiment in reinforcement learning are the agent and the environment. The environment is usually called the system or world with which the agent interacts. At each iteration of the algorithm, 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.

The environment (Enviroment) is an anthropomorphic manipulator operating in the operating environment. Agent includes an approximator that describes the decision-making policy and the algorithm for training the approximator. A convolutional artificial neural network is used as an approximator. As an approximator training algorithm, the Deep Q-Network (DQN) method was used [7].

Partial-learning algorithms try to find a strategy that ascribes actions to the states of the environment, one of which can be chosen by the agent in these states. An environment is usually formulated as a Markov decision-making process (MSS) with a finite set of states, and in this sense, reinforcement learning algorithms are closely related to dynamic programming. The probabilities of winnings and state transitions in MPNRs are usually random variables, but stationary in the framework of the problem.

Formally, the simplest learning model with reinforcement consists of:

- sets of states of environment S;
- sets of actions A;
- sets of real-valued scalar “wins” (rewards) R.

At each moment of time t, the agent makes a decision on choosing a certain action a_t based on the current state of the environment s_t using some policy μ:

$$a_t = \mu(s_t).$$

Into the state space s_t set of rotation angles included $\boldsymbol{\Theta} = \{\theta_i, i = \overline{1,n}\}$ in kinematic pairs of an anthropomorphic manipulator, as well as information on obstacles, including the coordinates of their centers $\boldsymbol{O_j}$, radii r_j, and speeds V_i, where $j = \overline{1,m}$:

$$s_t = < \boldsymbol{\Theta}, \boldsymbol{O}, \boldsymbol{r}, \boldsymbol{V} >.$$

The a_t action space includes the space of possible values of rotation angles in the drives of the anthropomorphic manipulator:

$$a_t = \boldsymbol{\tau_t},$$

$$\tau_{i,min} \leq \tau_{i,t} \leq \tau_{i,max}, i = \overline{1,n}.$$

Reinforcement learning algorithm adjusts weights based on reward value r_t to transition between states s_t and s_{t+1} when committing an action a_t.

The remuneration consists of the following values:

1) The cost of displacement proportional to the modulus of the change in generalized coordinates when changing position.

2) Penalty for collision of a manipulator with an obstacle. The collision of the manipulator is considered to have occurred if the manipulator has come closer to a distance less than a certain threshold value D. To calculate the values of the distances between the links of the manipulators, the modified method proposed in [5] was used.

3) The accuracy of positioning the grip of the manipulator, expressed as the distance between the coordinates of the target point and the actual coordinates of the grip.

In order to test the proposed method, a software implementation of the algorithm was performed in the Python programming language. During the simulation, a computer was used with the following characteristics: Intel Core i7-6500U 2.5GHz processor, 16Gb RAM.

The structure of the neural network model used is shown in Figure 4. This model consists of one input layer and 4 fully connected layers, each of which, in turn, consists of 1024 neurons. The model input is the target position in the three-dimensional coordinate system, and the output is the required rotation angle of each joint to achieve the target position.

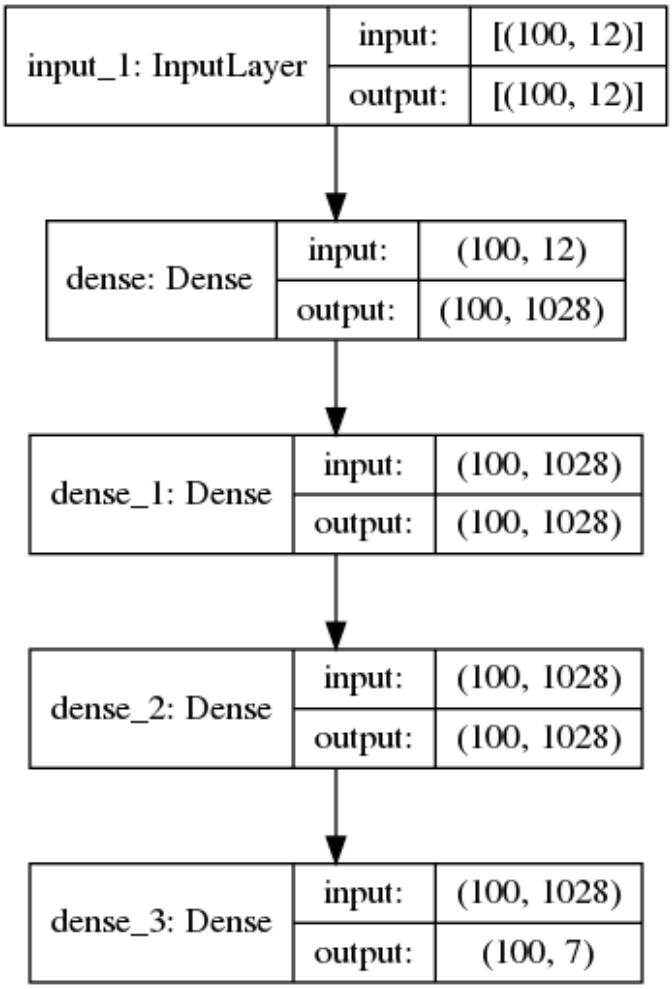

Figure 4 – Structure of the neural network model

Figure 5 shows an example of visualization of the learning process. The pyplot library was used to build the surrounding space; the implementation of the manipulator itself was performed using the pybotics library.

The learning process is divided into 5 eras. Each era consists of 1,000,000 episodes. For each episode, before the start of training, a set of input data corresponding to it is generated. From the generated data, 90,000 sets are separated for testing the results. The learning outcomes graph is shown in Figures 6 and 7.

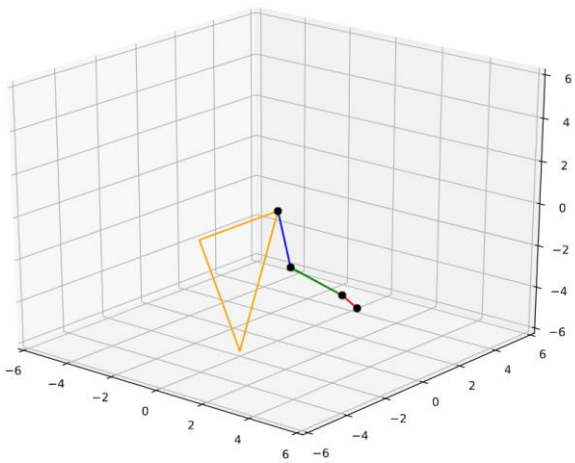

Figure 5 - An example of visualization of the learning process

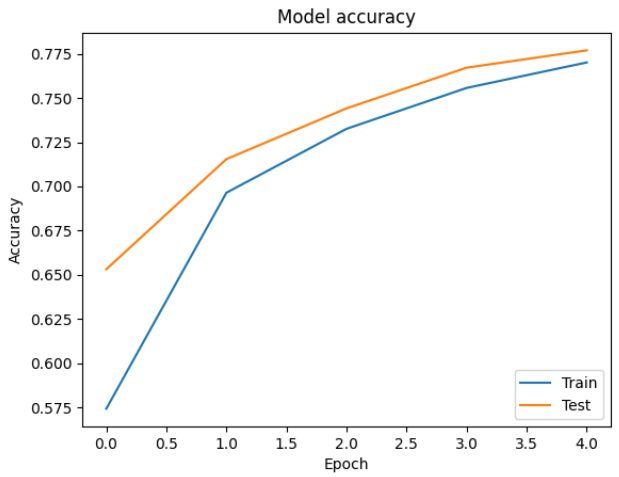

Figure 6 - Accuracy of placement of the manipulator for the training and test data set

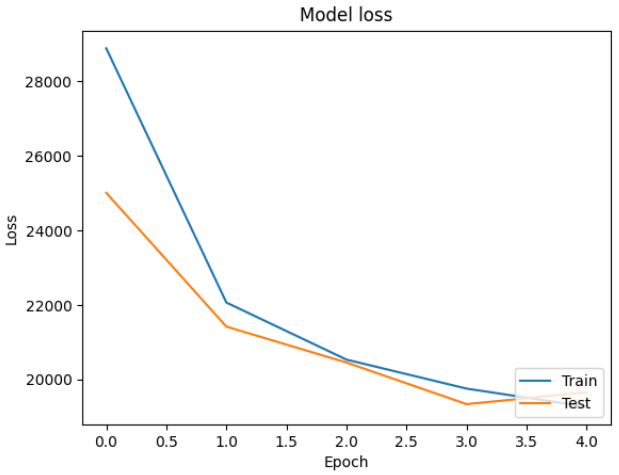

Figure 7 - Average penalty for training and test data sets

The experimental results are presented in Fig. 6-7. These figures show, respectively, the accuracy of the placement of the manipulator and the average penalty for training and test data sets. The data shows the accuracy of positioning of the manipulator increasing with training time to its peak of 77.5%, which is not a satisfactory result, but shows the possibility of using this method in this area.

III. CONCLUSION

As a result of applying the method, a sufficiently high accuracy of positioning of the manipulator was achieved, however, there are also cases in which the manipulator did not reach the target position in the allotted time. This performance can be explained by the fact that the sizes and movements of obstacles were randomly generated, which could lead to situations in which the collision of an anthropomorphic manipulator with an obstacle was impossible. This problem and the interpretation of the results obtained require more in-depth study.

REFERENCES

[1] Copying manipulator setting device, № RU169864U1, published 20.04.2020 in newsletter. № 11

[2] Gao P. et al. A global path planning algorithm for robots using reinforcement learning // IEEE International Conference on Robotics and Biomimetics, ROBIO 2019. Institute of Electrical and Electronics Engineers Inc., 2019. P. 1693–1698.

[3] Sun Y. et al. Mapless Motion Planning System for an Autonomous Underwater Vehicle Using Policy Gradient-based Deep Reinforcement Learning // J. Intell. Robot. Syst. Theory Appl. Springer, 2019. Vol. 96, № 3–4. P. 591–601.

[4] Li T. et al. A path planning algorithm for space manipulator based on q-learning // Proceedings of 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference, ITAIC 2019. Institute of Electrical and Electronics Engineers Inc., 2019. P. 1566–1571.

[5] Qu C. et al. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning // Appl. Soft Comput. J. Elsevier Ltd, 2020. Vol. 89.

[6] Vyacheslav Petrenko, Fariza Tebueva, Vladimir Antonov, Nikolay Untewsky, Mikhail Gurchinsky. Energy-Efficient Path Planning: Designed Software Implementation // Proceedings of the 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019). Series:Advances in Intelligent Systems Research. doi: https://doi.org/10.2991/itids-19.2019.30

[7] Yuan Y. et al. DMP-Based Motion Generation for a Walking Exoskeleton Robot Using Reinforcement Learning // IEEE Trans. Ind. Electron. Institute of Electrical and Electronics Engineers Inc., 2020. Vol. 67, № 5. P. 3830–3839.

[8] Petrenko, V., Tebueva, F., Gyrchinsky, M., Antonov, V., Shutova, J. The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only // IX International Multidisciplinary Scientific and Research Conference Modern Issues in Science and Technology / Workshop Advanced Technologies in Aerospace, Mechanical and Automation Engineering. IOP Conference Series: Materials Science and Engineering (2018). doi: 10.1088/1757-899X/450/4/042016

[9] Petrenko, V.I., Tebueva, F.B., Gurchinsky, M.M., Antonov, V.O., Shutova, J.A. 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 // IX International Multidisciplinary Scientific and Research Conference Modern Issues in Science and Technology / Workshop Advanced Technologies in Aerospace. IOP Conference Series: Materials Science and Engineering (2018). doi: 10.1088/1757-899X/450/4/042013

[10] Petrenko V., Tebueva F., Antonov, V., Gurchinskiy, M., Pavlov, A., Sergey, R, Shutova,Y. Analysis of the Effectiveness Path Planning Methods and Algorithm for the Anthropomorphic Robot Manipulator// International Siberian Conference on Control and Communications, SIBCON 2019 - Proceedings (2019), IEEE. doi: 10.1109/SIBCON.2019.8729657Petrenko V.I. et al. The method of forming a geometric solution of the inverse kinematics problem for chains with kinematic pairs of rotational type only // IOP Conference Series: Materials Science and Engineering. Institute of Physics Publishing, 2018. Vol. 450, № 4.

[11] Koshmanova N.P., Trifonov D.S., Pavlovsky V.E.. Controlling the manipulator with reinforcement learning // Nonlinear dynamics. 2012. Т. 8. № 4 (Mobile robots). 2012. Vol. 4. P. 689–704.

[12] Wen S. et al. Q-learning trajectory planning based on Takagi–Sugeno fuzzy parallel distributed compensation structure of humanoid manipulator // Int. J. Adv. Robot. Syst. SAGE Publications Inc., 2019. Vol. 16, № 1.

[13] Gu S. et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates // Proc. - IEEE Int. Conf. Robot. Autom. 2017. P. 3389–3396.

[14] Kim M. et al. Motion Planning of Robot Manipulators for a Smoother Path Using a Twin Delayed Deep Deterministic Policy Gradient with Hindsight Experience Replay // Appl. Sci. MDPI AG, 2020. Vol. 10, № 2. P. 575.

[15] Copy Manipulator, № RU135956U1, published 27.12.2013 in newsletter № 36

[16] Copy Manipulator, № RU146552U1, published 10.10.2014 in newsletter № 28

[17] Kutlubaev I.M. et al. Control system of the anthropomorphous robot for work on the low-altitude earth orbit // Int. J. Pharm. Technol. International Journal of Pharmacy and Technology, 2016. Vol. 8, № 3. P. 18193–18199.

[18] Kutlubaev I.M., Zhydenko I.G., Bogdanov A.A. Basic concepts of power anthropomorphic grippers construction and calculation // 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2016 - Proceedings. Institute of Electrical and Electronics Engineers Inc., 2016.

[19] Petrenko, V.I., Tebueva, F.B., Gurchinsky, M.M., Antonov, V.O., Pavlov, A.S. Predictive assessment of operator's hand trajectory with the copying type of control for solution of the inverse dynamic problem. SPIIRAS Proc. 18, 123–147 (2019). DOI: 10.15622/sp.18.1.123-147.

[20] G. Kovács, D. Bogdanova, N. Yussupova, and M. Boyko, "Informatics tools, AI models and methods used for automatic analysis of customer satisfaction," *Stud. Informatics Control*, vol. 24, no. 3, pp. 261–270, 2015, doi: 10.24846/v24i3y201503.

[21] N. Yussupova and D. Rizvanov, "Decision-Making Support in Resource Management in Manufacturing Scheduling," *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 544–547, 2018, doi: 10.1016/j.ifacol.2018.11.269.