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

Method for Reconfiguring the Kinematic Structure of a Mechatronic-Modular Robot in Non-Deterministic Conditions

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

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

Andrey Pavlov
Institute of Information Technologies and Telecommunications
North Caucasus Federal University
Stavropol, Russia
losde5530@gmail.com

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

***Abstract*—Modular robots, consisting of many identical modules, are one of the most difficult areas of robotics. Each newly added element changes the shape and capabilities of the end device, for example, adds functionality or allows the robot to move in new planes. The reconfiguration of the kinematic structure is a sequence of movements of each robot module from the initial position of the initial configuration to the final position of the desired configuration. The paper considers a method for reconfiguration the kinematic structure of a mechatronic-modular robot using reinforcement learning. The proposed method will be built on the basis of a learning algorithm, where the information for training will be the actions taken and the "reward" is a value characterizing the quality of the robot's completion of the target task. The purpose of the training is to build a control algorithm that maximizes the total reward for a certain period of time. The effectiveness of the learning algorithm was tested by computer simulation of a robot, consisting of 5, 10 and 15 modules, in the formation of the target configuration.**

Keywords—modular robotic, reinforcement learning, path planning, multi-agent systems, Q-learning

I. INTRODUCTION

Currently, the huge interest of developers of robotic systems is attracted by the intensively developing unmanned technologies, the concept of which is to use robots to perform routine, harmful and hazardous types of work without direct human participation, which is the key to ensuring safety and high efficiency in solving tasks [1]. Active research is being carried out in the field of artificial intelligence and the development of multifunctional modular robots (MMR) [2-7]. MMRs are robotic systems consisting of many homogeneous or heterogeneous modules that interact with each other and exchange information to perform the target operation. MMR modules have the ability to move relative to each other and connect / disconnect through special connectors, creating various configurations [2, 3]. The peculiarity of the modular design provides the versatility and high flexibility of the robot when performing the assigned task, which is the reason for the extensive field of application of the MMR. One of the main areas of application is the operation of MMRs during emergency and search and rescue operations, the specificity of which is non-deterministic environmental conditions.

At the moment, modern MMRs are not able to completely replace a person when performing complex tasks in a non-deterministic environment. The inability of such robots to completely replace humans is due to the low speed of performing target operations in offline mode. The reasons for the low speed of performing MMR the target operation are the difficulty in formalizing the task of decentralized control of the robot modules with a significant limitation of the software and hardware components of the MMR modules. An increase in the speed of performing target operations, and, as a result, in the efficiency of MMR functioning is possible due to the development and modification of mathematical methods and algorithms for adaptive reconfiguration of the kinematic structure of MMR that minimize the time to complete the target operation.

A significant part of the time during the functioning of the magnetic resonance is spent on the reconfiguration of the kinematic structure to overcome the obstacles or environmental restrictions that arise when performing the target operation [3]. It is possible to reduce the time required to change the configuration of MMR modules in accordance with the operating conditions by developing methods and algorithms for adaptive reconfiguration of the kinematic structure of MMR in a non-deterministic environment with obstacles, which is an actual task in modern applied mathematics and robotics.

The creation of a modular robot with an adaptive (reconfigurable) kinematic structure today is one of the most promising directions in robotics. Modular robots are created using modules of the same type, which are combined into an integral structure [6]. The combination of modules of the same type allows you to build mechanisms completely different in their structure. This provides significant advantages in comparison with classical mobile robots: higher reliability and overcoming obstacles of various complexity [7]. Modular robotics combines all the latest

advances in robotics, mechatronics and control theory. The most important area of application of modular robots is extreme robotics, the main tasks of which are to create and implement complexes used in extreme situations.

The modularity of the MMR design and, as a consequence, the ability to reconfigure the kinematic structure is a feature of modern modular robots. Modules are the basic elements of a robot, and the resulting configuration directly affects the efficiency of performing targeted operations. To form a robot configuration corresponding to the problem being solved and environmental conditions, a set of rules is required to plan the best path for moving modules from the initial configuration A to the target configuration B without collisions. Reconfiguration the structure of a robot is a problem from the NP class, since the complexity of solving the problem grows exponentially with an increase in the number of modules [4].

The control tasks of modular robots associated with the formation and reconfiguration of the kinematic structure are given in [5-20]. The listed works propose a methodological approach to the development of methods and algorithms for reconfiguration the structure of robots with a mechatronic-modular design. The development is based on the following methods: system analysis, methods of mathematical analysis, theory of evolutionary algorithms, theory of distributed computing systems, theory of automata, operations research, decision theory, probability theory, set theory, theory of constructing models of complex systems, graph theory, simulation. However, at the current stage of development of modular robotics, the application of this methodology for the control of MMR in non-deterministic conditions becomes ineffective. This can be explained by the increasing complexity of tasks that are solved by robots of this class, which inevitably leads to an increase in the number of control indicators and the complication of control tasks for MMR modules.

The task of reconfiguring the kinematic structure of MMR can be considered as a variant of the general problem of planning and controlling the movement of the robot, which has been studied in robotics for many decades [9–12]. However, this problem differs from traditional approaches in that the connectivity of the modules (or the topology of the kinematic structure of the modular robot) also change if necessary. Moreover, the task is complicated by the presence of an unlimited number of modules and, accordingly, an excessive degree of robot mobility, which requires analysis of multidimensional data. Standard methods of motion planning, which work well with several dimensions and provide a solution, are ineffective with an excessive number of degrees of mobility MMR, which configuration space can be very complex due to the topology of the robot.

Planning the trajectory of the MMR modules during reconfiguration can be simplified by sampling. Then, reconfiguration planning comes down to finding a sequence of discrete movements in which the modules perform only one movement at a time [13].

Currently, the development of methods and algorithms for reconfiguring modular robots has gained a certain distribution using methods of artificial intelligence (AI), which is reflected in the work of many researchers [14-20]. In the listed works, the following methods were used: genetic algorithms, artificial neural networks, cellular automata, particle swarm optimization. An analysis of the results showed that the use of such methods allows the MMR to adapt most flexibly to real environmental conditions, forming models that are fully adequate to the task, which distinguishes them from the background of completely formal systems. Reconfiguration methods that operate using AI methods not only implement standard adaptive control methods, but also offer their own algorithmic approaches to a number of problems, which solution is difficult due to informality [21].

Recently, control systems for autonomous robots based on artificial neural networks (ANNs) have gained great popularity. ANN-based control systems have the following advantages:

- ability to parallelize information processing;
- the possibility of self-learning, i.e. 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 modern robotics, decomposition methods are widely used, the main idea of which is to split a global task into a group of subtasks. In [22], an algorithm was proposed for planning the movement of a robot based on the decomposition of the target task into a sequence of subtasks. The authors of [23] developed an algorithm for distributing tasks between global and local level controllers when controlling a robot. In [24], ANNs are used in the implementation of reinforced learning to reduce the dimensionality of the state space, which increases the speed and efficiency of training a group of robots. However, the disadvantages of ANN include high computational complexity and low learning speed in general, since the greatest efficiency is achieved by increasing the number of layers of neurons (the so-called deep neural networks), which also determines the specific requirements for a computing device [17].

The task of increasing the efficiency of the functioning of the MMR, in particular, the speed of performing target operations, in a non-deterministic environment is very acute and still has not found its final solution in most applied problems. Obviously, the further development of modular robotics is possible only on the basis of an integrated approach, which will use the whole variety of existing methods and controls for modular robotics. Thus, the future of intelligent control of modular robotics lies in combining traditional control with the potential capabilities and prospects of using systems based on the use of AI methods, in particular, based on reinforcement learning.

II. PROBLEM FORMULATION

The aim of this work is to increase the level of automation of intelligent agents of a mechatronic-modular robot during reconfiguration of the kinematic structure in non-deterministic conditions through the use of reinforcement learning. The level of automation of intelligent MMR agents in this work is measured by the

number of automated actions that ensure the functioning of the robot with minimal operator involvement. In reinforcement learning, there is an agent interacting with the environment by taking actions. The environment gives a reward for these actions, and the agent continues to take them. Figure 1 shows the reinforcement learning scheme.

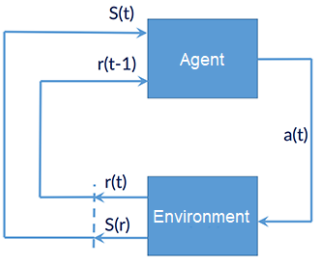

Fig. 1. Reinforcement learning scheme

Using reinforcement learning algorithms, developers can find a strategy that ascribes actions to the states of the environment, one of which the agent can select in these states. The environment is usually formulated as a Markov decision-making process (MDMP) 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 MDMP 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).

At an arbitrary instant of time t the agent is characterized by the state $s \in S$ and the set of possible actions $A(s_t)$. Choosing the action $a \in A(s_t)$, it goes into the state s_{t+1} and receives the gain r_t. Based on this interaction with the environment, the agent should develop a strategy $\pi: S \to A$ that maximizes the value of $R = r_0 + r_1 + \cdots + r_n$ in the case of a MDMP having a terminal state, or the value:

$$R = \sum \gamma^t r_t, \quad (1)$$

where $0 \le \gamma \le 1$ – discount factor for upcoming winnings.

Thus, the task of reconfiguration the kinematic structure of the MMR can be represented as a sequence of actions leading each module of the MMR from the current position to the target one. The mathematical formulation of the problem is as follows: it is necessary to determine the set of actions $a_1, a_2, \ldots, a_n$ for each MMR module to form the desired configuration with the known proper position $q_i(x_i, y_i)$ and the given configuration P with the maximum possible reward R.

III. Methods

The paper considers one of the approaches in reinforcement learning using the direct parameterization $\pi(s, a)$. There are other approaches [25] that use the state quality and action quality functions. The considered approach is characterized by simplicity in development with discrete spaces of states and actions. Below we consider one of the algorithms of this class - Q-learning. This algorithm uses the Q-function to determine the optimal policy, which arguments are the state of the observed environment and the selected action. This allows iterative way to build a Q-function and thereby find the optimal control policy. The expression for updating the Q-function has the following form:

$$Q(s_t, a_t) \leftarrow r_t + \alpha[r_{t+1} + \gamma \max Q(s_{t+1}, a)], \quad (2)$$

where a – action selected at time t from the set of all possible actions A.

Estimates of the Q-function are stored in a 2-dimensional table, which inputs are state and action. In systems using Q-learning, expression (2) is usually combined with the temporal difference method (TD (λ)), which was proposed in [26]. With the parameter of the method of the time difference λ equal to zero, only the current and subsequent value of the predicted Q-values are involved in the update, therefore, in this case the method is called one-step Q-learning. The expression for one-step Q-learning is as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max Q(s_t, a) - Q(s_t, a_t)]. \quad (3)$$

In this case, the sought function of the value of the action Q, directly approaches the optimal function of the value of the action Q^*, regardless of the strategy used. The strategy determines which state-action pairs are adjusted and visited, however, to ensure convergence, it is only necessary that all pairs continue to be adjusted during the operation of the algorithm. The Q-learning algorithm is shown in Figure 2.

Initialize $Q(s, a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each episode):
    Find a by s, using a strategy derived from Q
    Perform action a, find r, s'
    $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max Q(s_t, a) - Q(s_t, a_t)]$
    $s \leftarrow s'$,
    Until s goes into a terminal state

Fig. 2. Q-Learning Algorithm

As a simplified model of MMR, a regular lattice was chosen in which one robot module (agent) occupies one cell. Each agent at time t has 5 options: step up, step right, step down, step left, stay in place.

In order to reduce the size of the Q-table, agent observations were presented in vector form. So, the current position of the agent relative to the target cell is presented in vector form q_i', calculated by the formula (4):

$$V = q_i - q_{Target},$$

$$\begin{cases} x_i' = 1, if\ x_V > 0, \\ x_i' = -1, if\ x_V < 0, \\ x_i' = 0, if\ x_V = 0, \\ y_i' = 1, if\ y_V > 0, \\ y_i' = -1, if\ y_V < 0, \\ y_i' = 0, if\ y_V = 0, \end{cases} \quad (4)$$

In addition to the agent's own position on the regular lattice, the observation includes the encoded value $n = (0, 1, \ldots, 15)$ of the first-order neighborhood of the cell in which the agent is located for the presence of neighbors.Thus, agent observations can be described by the tuple $< q_i', \mathrm{n} >$. These transformations allow the agent to clearly determine the direction in which it is necessary to move to achieve the goal, taking into account the presence of obstacles (other agents) in neighboring cells. In addition, the Q-table size was 144 rows and 720 possible values of the Q-function estimate.

IV. Results

In order to test the proposed method, a software implementation of the algorithm in the Python programming language was performed. During the simulation, a computer was used with the following characteristics: Intel Core i7-8550U 1.8GHz processor, 8Gb RAM. The parameters of the reinforcement learning algorithm for the reconfiguration of the MMR are presented in Table 1. Epsilon is a coefficient that determines the probability of choosing an action that has not been previously investigated. Learning factor - a coefficient that determines how much an agent trusts new information. The value of the discount factor determines how quickly the agent will receive a reward for the actions performed.

TABLE I. Reinforcement Learning Algorithm Parameters

Parameter	Value
Field size	10
Number of agents	5/10/15
Number of episodes	50000
The number of steps in each episode	200
Epsilon	0,99
Learning rate	0,3
Discount factor	0,95
Single Agent Reward	10000
Penalty for collision with other agents	-10000
Penalty for movement	-40

The target configuration is specified by a set of coordinates of each module randomly at each iteration. An example of the initial position of the modules of the target configuration for the MMR, consisting of 5 modules, is presented in Figure 3.

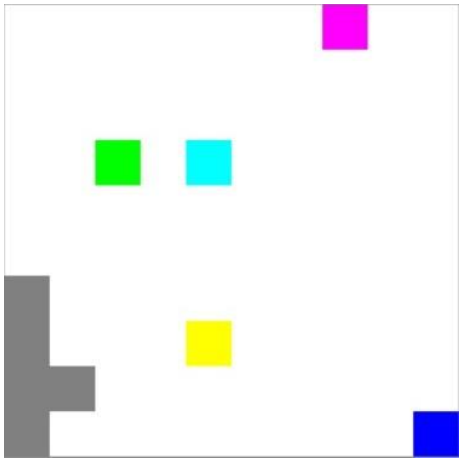

Fig. 3. An example of the initial position of the MMR modules. (*cells shaded in gray indicate the positions of the target configuration; colored cells are agents*)

During the training of 5 agents, the average value of remuneration was 2103232.2. At the same time, the average number of episodes in which agents successfully took positions according to the target configuration was 641, which can be considered an acceptable indicator of MMR functioning compared to the results obtained with an increase in the number of agents. The learning outcomes are shown in Figure 4.

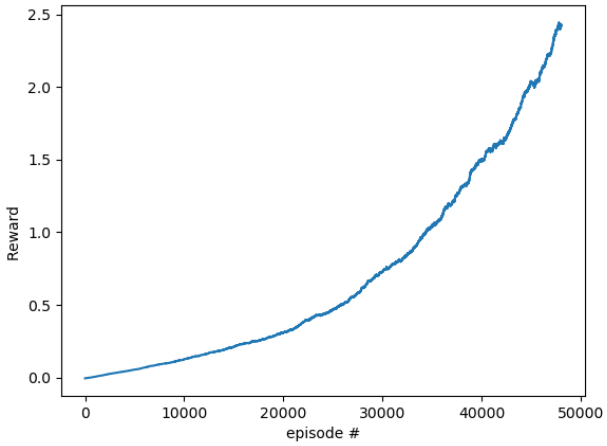

Fig. 4. Graph of the learning outcomes of 5 agents

To test the scalability of the Q-learning algorithm, an experiment was conducted with 10 and 15 agents, relevant examples are presented in Figures 5 and 6.

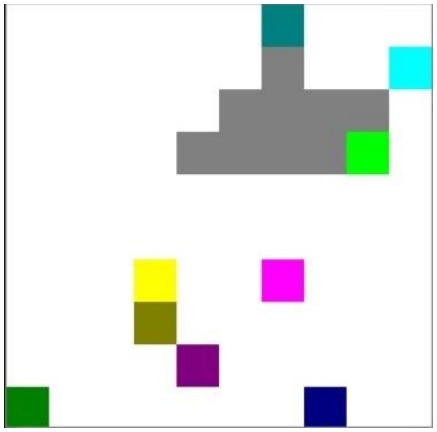

Fig. 5. Example of reinforcement training episode scaling up to 10 agents

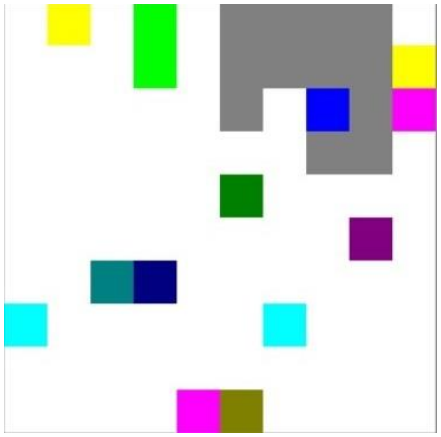

Fig. 6. Example of reinforcement training episode scaling up to 15 agents

The learning outcomes of agents are shown in Figures 7 and 8.

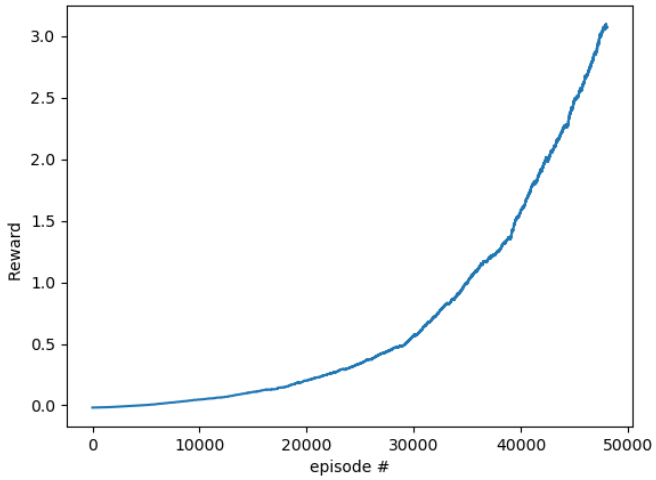

Fig. 7. Reinforcement Learning Outcome with 10 Agent Scaling

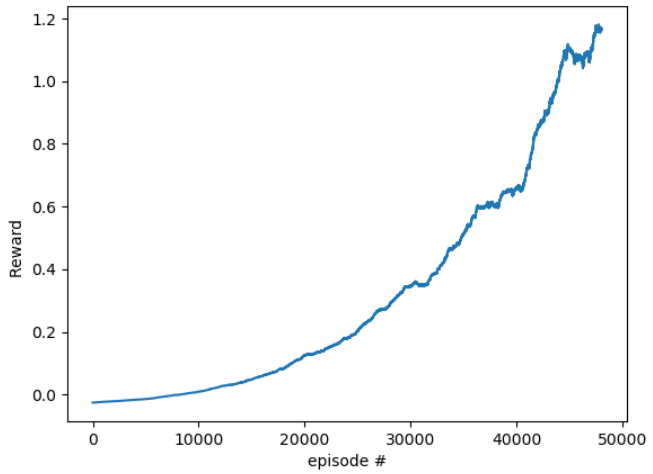

Fig. 8. Reinforcement Learning Outcome with Scaling of 15 Agents

For clarity, the outcomes obtained for training 5, 10 and 15 agents are summarized in Table 2.

TABLE II. AGENT LEARNING OUTCOMES

Parameter	Number of agents		
	5	***10***	***15***
Average remuneration	$2.1 * 10^7$	$2.7 * 10^7$	$1.1 * 10^7$
Number of Successfully Completed Episodes	704	32	0
The number of episodes without a solution	395	119	6
Number of episodes completed by agent collisions	964	1849	1994

According to the data presented in Table 2, with an increase in the number of agents, the number of successful training episodes sharply decreases due to an increase in the number of collisions between agents. In addition, during the training process, problematic cases were discovered, for example, when the target configuration is located in the corners of the field, and the first agents who reached the target positions can block the way for other agents. Based on the foregoing, we can conclude that Q-training is effective for a small group of robots without the possibility of scaling.

V. CONCLUSION

Despite the inconsistency of the results, Q-training has proven effective in training one agent. It is assumed that the values of the Q-table during the training process by several agents are repeatedly rewritten and may deviate from optimal ones due to the lack of an approximation function. Verification of this assumption can be carried out by using artificial neural networks instead of the Q-table to form the optimal agent policy, which will be the subject of research in further works.

Thus, this article presents a method for reconfiguration the kinematic structure of a mechatronic-modular robot in non-deterministic conditions based on reinforcement learning. The method was based on the Q-learning algorithm, which allows to effectively train the agent in discrete spaces of states and actions. The results of modeling the developed method for a robot consisting of 5, 10 and 15 modules are presented. The results obtained indicate the need to modify the Q-learning algorithm to be able to scale the system and increase the autonomy of the intelligent agents of the mechatronic-modular robot during reconfiguration of the kinematic structure in non-deterministic conditions.

ACKNOWLEDGMENT

This research is financially supported by the Fund for assistance to the development of small forms of enterprises in the scientific and technical sphere under the Grant agreement 13684GU/2018 from 01 April 2019. The research topic: “Development of a reconfigurable robotic complex with an adaptive kinematic structure based on small-sized spherical robots for use in emergency situations”. Work on the project is carried out at the North-Caucasus Federal University (NCFU).

REFERENCES

[1] Decree of the President of the Russian Federation “On the Strategy for Scientific and Technological Development of the Russian Federation” dated December 01, 2016 No. 642 // Meeting of the legislation of the Russian Federation. - item 20.

[2] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian, «Modular self-reconfigurable robot systems [grand challenges of robotics]», Robotics Automation Magazine, IEEE, vol. 14, no. 1, pp. 43-52, March 2007.

[3] Stoy, K., Brandt, D., & Christensen, D. J. (2010). Self-reconfigurable robots: an introduction. Cambridge, MA: MIT Press.

[4] Gorbenko A.A., Popov V.Y. Programming for modular reconfigurable robots. Programming and Computer Software, 2012; 38: 13-23.DOI: 10.1134/S0361768812010033.

[5] Petrenko, V.I. Tebueva F.B., Pavlov A.S., Antonov V.O., Kochanov M.S. Path Planning Method in the Formation of the Configuration of a Multifunctional Modular Robot Using a Swarm Control Strategy // 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019), Advances in Intelligent Systems Research. 2019. Vol-166. P. 165-170. DOI: https://doi.org/10.2991/itids-19.2019.30.

[6] Mezenceva O.S., Petrenko V.I., Zhilina E., Pavlov A.S., Apurin A.A. Developing a concept of available multi-functional modular robot for education and research // CEUR Workshop Proceedings SLET 2019 - Proceedings of the International Scientific Conference Innovative Approaches to the Application of Digital Technologies in Education and Research. 2019.

[7] Petrenko, V., Tebueva, F., Pavlov, A., Gurchinsky, M. The method of the kinematic structure reconfiguration of a multifunctional modular

robot based on the greedy algorithm // 12th International Conference on Developments in eSystems Engineering (DeSE), Kazan, Russia, 2019, pp. 42-47, doi: 10.1109/DeSE.2019.00018.

[8] N. Yusupova, O. Smetanina, A. Agadullina and E. Rassadnikova, "The development of ontologies to support the decisions in production systems management," 2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC), Vladivostok, 2017, pp. 188-193, doi: 10.1109/RPC.2017.8168096.

[9] Kovács, G., Yusupova, N., Smetanina, O., Rassadnikova, E.Methods and algorithms to solve the vehicle routing problem with time windows and further conditions(2018) Pollack Periodica, 13 (1), pp. 65-76.DOI: 10.1556/606.2018.13.1.6.

[10] Kutlubaev, I.M., Zhydenko, I.G., Bogdanov, A.A.Basic concepts of power anthropomorphic grippers construction and calculation (2016) 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2016 - Proceedings, №7910963. DOI: 10.1109/ICIEAM.2016.7910963.

[11] 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.

[12] Lynch KM, Park FC. 2017. Modern Robotics. Cambridge, UK: Cambridge Univ. Press.

[13] Liu, J., Zhang, X., & Hao, G. (2016). Survey on research and development of reconfigurable modular robots. Advances in Mechanical Engineering. DOI:https://doi.org/10.1177/1687814016659597.

[14] Ababsa, Tarek & DJEDI, NourEddine & Duthen, Yves. (2017). Genetic Programming-based Self-Reconfiguration Planning for Metamorphic Robot. International Journal of Automation and Computing. DOI:10.1007/s11633-016-1049-4.

[15] Baca, José & Dasgupta, Raj & Hossain, S.G.M. & Nelson, Carl. (2013). Modular robot locomotion based on a distributed fuzzy controller: The combination of modred's basic module motions. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems. 4302-4307. DOI:10.1109/IROS.2013.6696973.

[16] Dong, Bo & Zhou, Fan & Liu, Keping & Li, Yuanchun. (2017). Torque sensorless decentralized neuro-optimal control for modular and reconfigurable robots with uncertain environments. Neurocomputing. DOI:282. 10.1016/j.neucom.2017.12.012.

[17] Guettas, Chourouk & Foudil, Cherif & , Thomas Breton & Duthen, Yves. (2014). Cooperative Co ‐ evolution of Configuration and Control for Modular Robots. International Conference on Multimedia Computing and Systems -Proceedings. DOI:10.1109/ICMCS.2014.6911138.

[18] Li, Yan & Lu, Zengpeng & Zhou, Fan & Dong, Bo & Liu, Keping & Li, Yuanchun. (2019). Decentralized Trajectory Tracking Control for Modular and Reconfigurable Robots With Torque Sensor: Adaptive Terminal Sliding Control-Based Approach. Journal of Dynamic Systems, Measurement, and Control. 141. DOI:10.1115/1.4042550.

[19] Yeom, Kiwon. (2015). Morphological approach for autonomous and adaptive system: The construction of three-dimensional artificial model based on self-reconfigurable modular agents. Neurocomputing. 148. 100–111. DOI:10.1016/j.neucom.2012.12.082.

[20] Brunete, A., Ranganath, A., Segovia, S., de Frutos, J. P., Hernando, M., & Gambao, E. (2017). Current trends in reconfigurable modular robots design. International Journal of Advanced Robotic Systems. DOI:https://doi.org/10.1177/1729881417710457.

[21] Zhu, Yanhe & Dongyang, Bie & Wang, Xiaolu & Zhang, Yu & Jin, Hongzhe & Zhao, Jie. (2016). A distributed and parallel control mechanism for self-reconfiguration of modular robots using L-systems and cellular automata. Journal of Parallel and Distributed Computing. 102. DOI:10.1016/j.jpdc.2016.11.016.

[22] Zhang Q., Fan C.X. Motion planning of robot on the basis of task decomposition and speed distribution // Huanan Ligong Daxue Xuebao/Journal South China Univ. Technol. (Natural Sci. South China University of Technology, 2016. T. 44, № 3. C. 44–50.DOI: 10.3969/j.issn.1000-565X.2016.03.007.

[23] Yasuda G. Distributed Controller Design for Cooperative Robot Systems Based on Hierarchical Task Decomposition // Int. J. Humanoid Robot. World Scientific Publishing Co. Pte Ltd, 2017. T. 14, № 2.DOI: 10.1142/S0219843617500177.

[24] Kawano H. Hierarchical sub-task decomposition for reinforcement learning of multi-robot delivery mission // Proceedings - IEEE International Conference on Robotics and Automation. 2013. C. 828–835.DOI: 10.1109/ICRA.2013.6630669.

[25] Sutton R., Barto A. Reinforcement learning: An Introduction. Cambridge, MA: MIT Press, 1998. 322 pp.

[26] R.S. Learning to predict by the methods of temporal differences. Mach Learn 3, 9–44 (1988). https://doi.org/10.1007/BF00115009.