

Computer Simulation of a Group of Mobile Robots in Special Software Tools

Vadim D. Raskatov ¹ and Vasily I. Rubtsov ^{1*}

¹Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. The problem of controlling a group of special robots for application in the high-risk conditions (for example, for fire extinguishing, search for hazardous substances, or elimination of consequences of emergencies) is solved. According to the task, robots operate in the industrial and urban environment, i.e., the movement of robots is planar. Various strategies of group interaction between robots are analyzed; a literature review is conducted. Due to the task specifics, a decentralized management strategy has been selected. For solving the problem, a movement algorithm of a group of mobile robots with a subsequent visual check of the resulting algorithm operation is proposed. Implementation of the algorithm's work is performed in the MATLAB environment. For preliminary algorithm verification and modeling of various dangerous cases, a model of robot group using the Mobile Robotics Simulation Toolbox of MATLAB was made. For a more detailed and accurate analysis of the real robot's behavior, a model in the ROS package has been developed. That model interacts with the developed algorithm through the ROS Toolbox package of MATLAB. This software can be used for the training of engineering staff in the field of mathematical modeling, group control, and multi-agent systems. As a result of verification, the program and the model successfully solve the task. Yet, the model in the ROS package requires greater computational power because of the large number of complex calculations.

1 Introduction

Robots are used in many fields of human activity, such as industrial production, zones of radioactive and chemical contamination, warfare conditions, and space exploration [1-5]. Mobile robots are taking an increasingly important role in the tasks handled by various departments, such as Emergencies Ministry, FSB, and others. Usage of robots allows us to solve problems of search and withdrawal of different hazardous substances, to perform complex technological operations, including those dangerous for people with minimal operator participation [1], [2-6].

In various researches, it is proved that the application of single robots can appear ineffective in solving complex problems. For such issues, it is required to use groups of cooperating robots [1], [2].

* Corresponding author: rubtsov@mail.ru

It is caused by if using robots in the group, the acting radius increases due to the distribution of robots in all operating zone. The number of possible executed operations can be increased by providing robots in a group with various equipment. Moreover, if one robot goes out of order, the rest can distribute tasks among themselves, then the task assigned to the group can be successfully fulfilled [5-7].

From the economic point of view, the serial production of robots for group work is more profitable than single production [1], [7-8].

However, the task of controlling a whole group of robots is more complicated than controlling a single one. Thus, the theory of group control of robots is at the stage of development and active research [1], [7-9].

2 Research objective

The objective of this work is:

- Development and testing of visual mathematical models of a group of mobile robots operating in an urban industrial environment;

To achieve the goal, it is necessary to solve the following **problems**:

- selecting a strategy for group management;
- selecting algorithms and their implementation for the motion task in the industrial and urban environment;

- creating a model of a group of mobile robots in the Simscape Multibody package with the additional Mobile Robotics Simulation Toolbox for mobile robots;

- creating a model of a group of mobile robots in ROS while interacting with the developed robot control system via ROS Toolbox add for MATLAB.

This work is conducted within grant No. 16-29-04178 of the Russian Foundation for Fundamental Research, "Development of algorithms for combining and jointly processing navigation and visual information in a group of heterogeneous robots."

Assume that the operating area for mobile robots is obtained (e.g., with a UAV) (Fig. 1), and that the robot's initial locations are known, as well as the location of the target points for the robot to reach. There are no sudden changes in the operating zone, and the surface is flat. Such an environment is called industrial-city. The group management task with the optimization criteria as the shortest distance to each target point (i.e., each robot selects the nearest target) should be solved. The motion is planar, controlled by two coordinates and an angle. The robots move without sliding. Target points and the environment are stationary [1].

An example of the task implementation is the functioning of a group of mobile robots in a manufacturing facility or urban environment in case of emergency response, fire extinguishing, etc. Task provides characteristics of the operating area and the location of the target points (e.g., they were obtained by other reconnaissance robots).

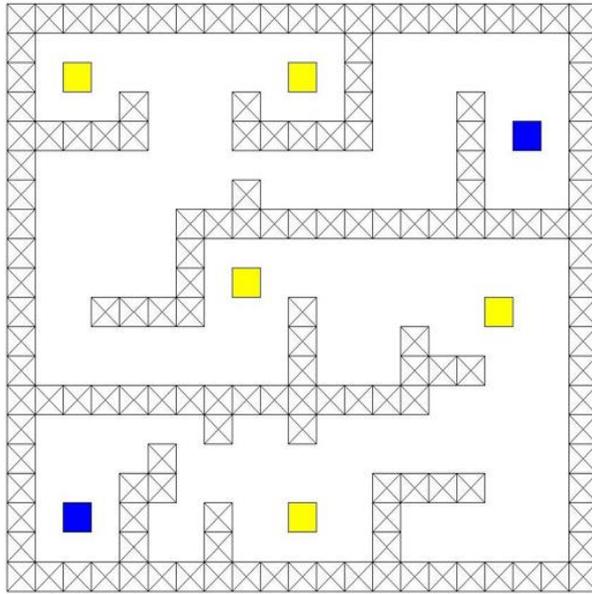


Fig. 1. Example of an operating area for a group of robots. Blue dots represent the locations of the robots, and yellow dots are the target.

3 Description of the main parts of the program to the task performance

Based on a literature review for a group of mobile robots moving in a complex environment, we decided to implement decentralized collective management of the group. When operating under the set conditions, one of the robots may be disabled, which is not critical for the selected control strategy [10-12]. Also, the task solved by each robot will be relatively simple, due to only optimizing its actions, and providing sufficient intellectual level of each robot is not difficult [12].

For solving the problem, we implemented the program in MATLAB for further visualization in Simscape Multibody and ROS. The program's algorithm is shown in Fig. 2 [1].

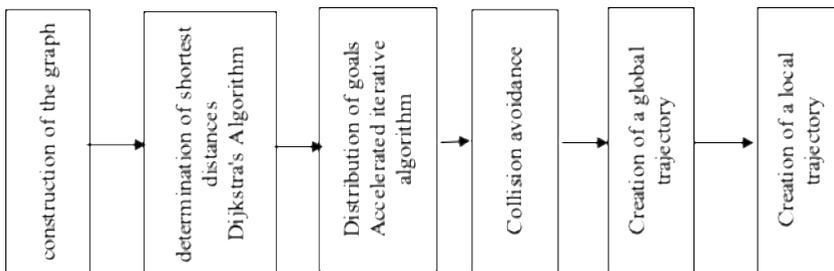


Fig. 2. Program's algorithm for solving the set problem.

3.1 Area discretization

Area is divided into square samples, i.e., the operating zone is a grid or square matrix whose values contain information about the state of the corresponding square in the area [13].

3.2 Graph construction

It is known that the shortest trajectory on a plane with the obstacles is a fractured trajectory passing through obstacle corners [3,14]. Therefore, according to the known area plan, its model can be formed as a graph whose edges correspond to all straightways of motion and are weighted by their length, and the tops - by convex corners of obstacles. The initial and final location of the mobile robot should be defined on the graph. The points corresponding to the graph tops should be spaced away from the obstacles regarding the overall dimensions of the control object [15]. The problem of building a global trajectory is to find on the global graph the shortest path between the tops corresponding to the initial and final positions.

The tops of the graph are all free cells of the discrete area obtained earlier, and its edges are the possible directions of movement between them.

3.3 Determining the shortest distances

That subtask is solved using the Dijkstra algorithm. The Dijkstra algorithm is a graph algorithm that finds the shortest paths from one of the tops of the graph to all the others. The algorithm operates only for the graphs without negative weight ribs [16].

The search will be conducted for the first top (the robot's base point). And the shortest distances of interest will be the distances to the target points.

3.4 Distribution of targets for robots

The solution of this subtask based on an accelerated algorithm of collective goal distribution [1]. This algorithm is developed using an iterative procedure of collective action optimization.

3.5 Construction of a global trajectory of each robot in a group

A model for building a global trajectory for each robot based on specified control points (graph tops) of the route (Fig. 3) [18-19] was created in Simulink.

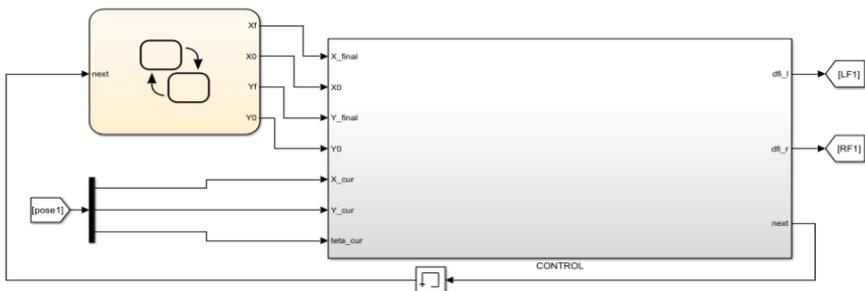


Fig. 3. Scheme for building a global trajectory via Simulink.

For this task, we will use Simscape Multibody, which allows conducting simulation with physical laws in the environment [19].

On the upper part of the figure placed a robot's control system. On the input of block Chart from the Stateflow library enters the coordinates of points through which the robot should pass. The matrix of these coordinates is a result of operating the group control task program and is a regular variable matrix in MATLAB variable space. This block outputs the coordinates of the point the robot should reach. The CONTROL subsystem implements the primary control law of the robot, which is the calculation of the angle robot should turn to, and the distance robot should pass. These values are compared to the current values of angle and distance, and the control system ensures the matching of these values by issuing corresponding control actions on the model of the robot's wheels. After the pairing, outputs the next signal, which generates in the Chart block coordinates of the next point unless it is the endpoint of the matrix and trajectory.

3.6 Creating a local trajectory for each robot in the group

The global trajectory is only a recommended future trajectory and cannot be directly transferred to the executive system. The movement along the global trajectory should be done considering the situation in the local area surrounding the robot, which usually coincides with the vision system view area [1].

4 Model realization using Mobile Robotics Toolbox

4.1 Model description

A special Mobile Robotics Simulation Toolbox [20] was developed in MATLAB to simulate work with tasks for mobile robots. The Toolbox contains specialized blocks for solving direct and inverse tasks of kinematics, as well as for modeling motion with different sensors. These units are represented various types of mobile robots, such as differential robots, robots with steering wheels, and Omni-wheels.

The obtained model as the blocks in the package is shown in Fig. 4. To simplify the task, a differential robot has been selected to increase the speed of simulation.

The Model contains blocks of the kinematic model of differential robot, distance sensor block (lidar), and two visualization blocks - on the Figure graphic, and for visualization in Simscape Multibody.

To ensure group interaction, an additional Chart block may be required, which will contain the coordinates of all robots in the group. According to the algorithm, the block will regulate group interaction. For example, when one robot goes out of order, its coordinates either do not change for a long time, or there is no signal. Then the nearest working robot to the target point of the broken one adds this target point to the matrix, calculating the route to it.

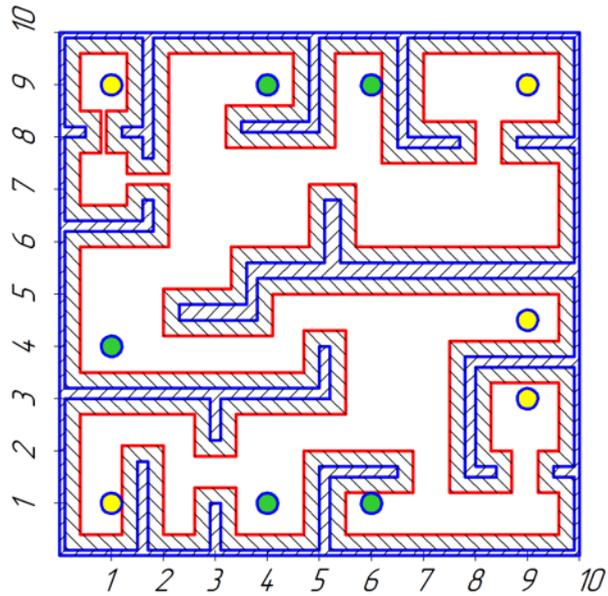


Fig. 5. An example of a group control task. The robot moves from yellow dots - base points, to green dots - target points; shaded area - impassable zone.

Fig. 6 shows the initial and final state of the Simulink model. You can see that mobile robots have moved from the start to the endpoint.

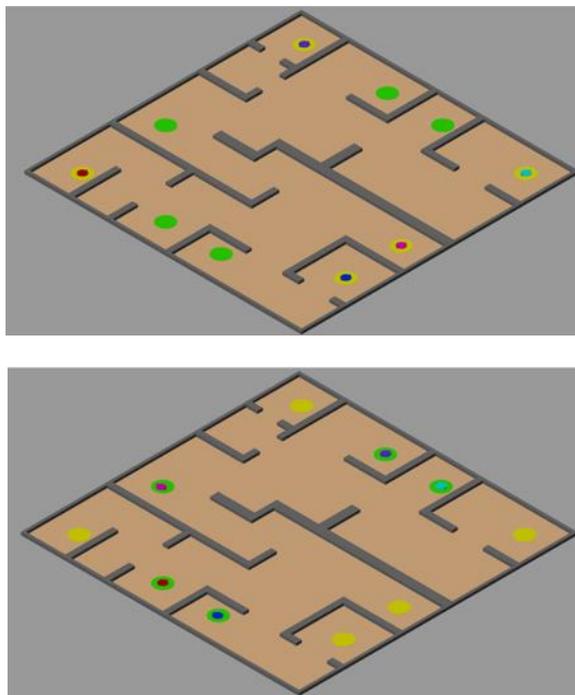


Fig. 6. The initial and final position of mobile robots.

Given example shows that the task of group control has been successfully solved, in particular at the given initial position and target points, mobile robots have performed movement between them, as shown by the model in Simscape Multibody.

For further development of engineering courses, it is possible to use the developed model for other numbers of robots and different types of mobile robots (for example, a robot with steering wheels or with Omni-wheels).

5 Model implementation using ROS Toolbox and ROS

5.1 Model description

Model development is carried out in ROS on Linux using the Gazebo visualization package. The work is using interaction with the previously developed program in MATLAB. Interaction is provided with the kit for MATLAB ROS Toolbox [21].

Robot models in ROS are initially described in the *.urdf file, where described blocks: visual (visual characteristics), collision (friction and object interaction area), inertial (inertial characteristics of the robot). Combinations of the robot's components are carried out through blocks joint. The file containing all the mentioned components is described in xml language [22-23].

Initial visualization of the resulting robot based on the urdf file is performed in rviz package. It allows to verify the physical model and to set up interaction with different sensors (camera, lidar, gyroscope, etc.). Examples of robots in the rviz package are shown in Fig.7.

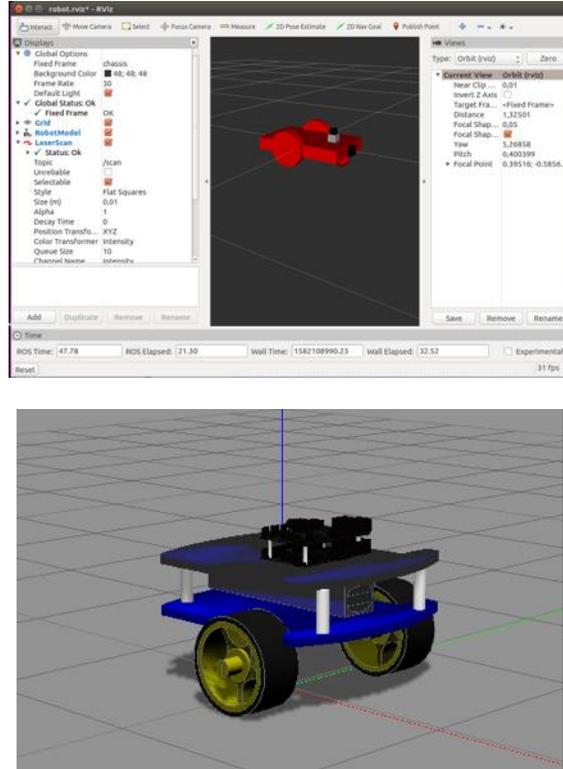


Fig. 7. Models of mobile robots in the rviz package.

The rviz package is only a means of visualizing the model, not a means of simulating or motion modeling. For these purposes, the standard gazebo package is required.

To run all necessary files in gazebo, launch files are required. They describe all the required data to create a working space. They also specify the type of map, the necessity to visualize sensors' actions, the location and orientation of the robot, and various other options for setting up [23]. An example of a launch file for one robot is shown on Fig. 8.

An example of placing a robot in the gazebo is shown in Fig. 9.

For interaction with the obtained model, ROS uses the addition of MATLAB ROS Toolbox. ROS is forming data of positioning in the /state_robot and transferring them to MATLAB. Then from the model to Simulink, data about necessary movements are transferred to the /cmd_vel, which is perceived by the gazebo robot model. A scheme in MATLAB is shown in Fig. 10.

For further work with engineering courses, it is possible to create other types of mobile robots by upgrading urdf files, launch files, and models of attachments.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <launch>
3
4   <arg name="world" default="empty"/>
5   <arg name="paused" default="false"/>
6   <arg name="use_sim_time" default="true"/>
7   <arg name="gui" default="true"/>
8   <arg name="headless" default="false"/>
9   <arg name="debug" default="false"/>
10
11 <include file="$ (find robot_description)/launch/robot_view.launch" />
12
13 <include file="$ (find gazebo_ros)/launch/empty_world.launch">
14   <arg name="paused" value="$ (arg paused)"/>
15   <arg name="world_name" value="$ (find robot_gazebo)/world/empty.world"/>
16   <!-- arg name="world_name" value="$ (find robot_gazebo)/world/turtlebot_playground.world"/-->
17   <!-- arg name="use_sim_time" value="$ (arg use_sim_time)"/>
18   <arg name="gui" value="$ (arg gui)"/>
19   <arg name="headless" value="$ (arg headless)"/>
20   <arg name="debug" value="$ (arg debug)"/>
21 </include>
22
23 <!-- start robot state publisher -->
24 <!-- node pkg="robot_state_publisher" type="robot_state_publisher" name="robot_state_publisher" output="screen" >
25   <param name="publish_frequency" type="double" value="50.0" />
26 </node-->
27
28 <-- param name="robot_description" command="$ (find xacro)/xacro.py '$ (find robot_description)/urdf/robot.xacro'"/>
29 <node name="mybot_spawn" pkg="gazebo_ros" type="spawn_model" output="screen"
30   args="-urdf -param robot_description -x 1 -y 1 -z 0.3 -Y 0 -model mybot" />
31
32 <!-- param name="robot_description2" command="$ (find xacro)/xacro.py '$ (find robot_description)/urdf/robot2.xacro'"/>
33 <node name="mybot_spawn1" pkg="gazebo_ros" type="spawn_model" output="screen"
34   args="-urdf -param robot_description2 -x 1 -y 5 -z 0.3 -Y 0.785 -model mybot1" /-->
35
36
37 </launch>
```

Fig. 8. Launch file for robot's model in gazebo.

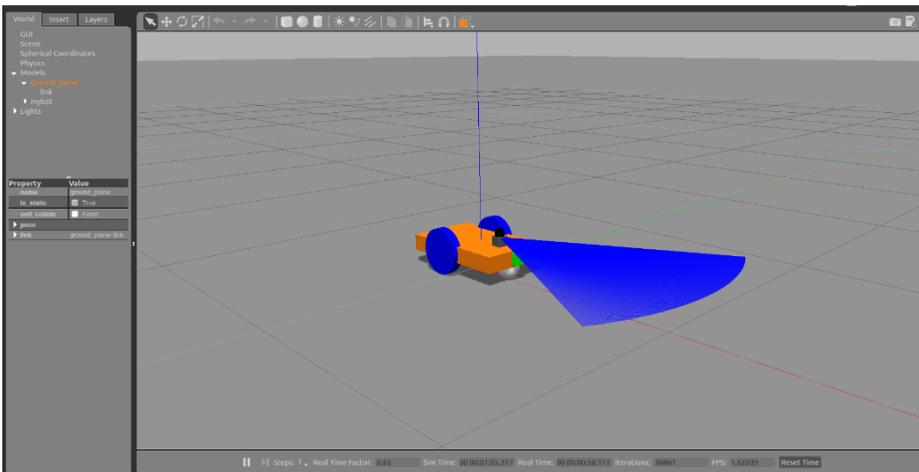


Fig. 9. Robot's model in gazebo.

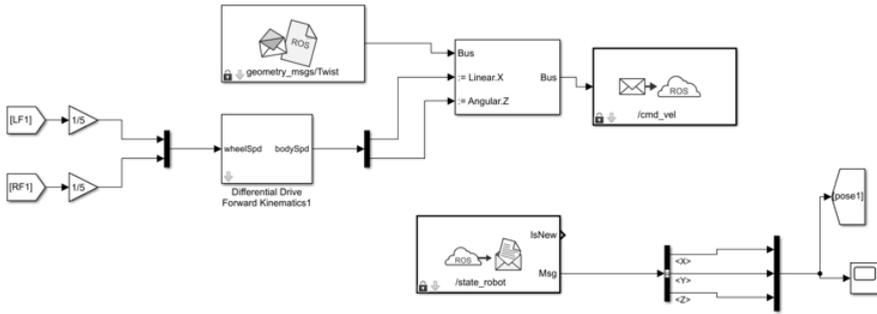


Fig. 10. MATLAB interaction scheme in ROS Toolbox package.

5.2 Example of the resulting model operation

The model is in the final stage of debugging. The general algorithm of motion and interaction between MATLAB and ROS is executed, and the full visual part of walls and marks of initial and endpoints will be completed in the future.

An example of the location of 3 robots for motion is shown in Fig. 11.

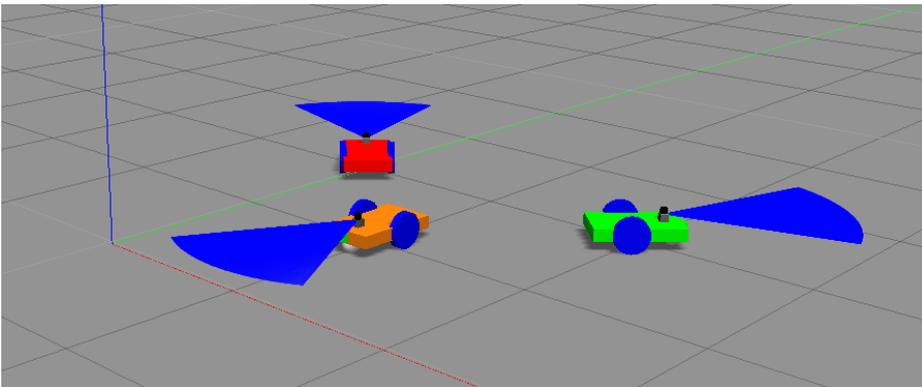


Fig. 11. Model of three robots in the gazebo at the beginning of the movement.

6 Conclusion

This work conducted a study of group management algorithms, and for the task was chosen decentralized management strategy.

For verification of the correctness of the group control task, a mathematical model of a group of robots in the industrial city environment was constructed for visual evaluation.

The first model was compiled in MATLAB with Mobile Robotics Simulation Toolbox. It shows high accuracy and speed of modeling and is entirely suitable for the task.

The second model is made in ROS, which connects via ROS Toolbox with a previously written program in MATLAB. This model considers a larger number of possible physical features and is more realistic. But when operating, it shows low modeling speed. The model is very complex and needs further improvement.

The received program complex with modifications for specific tasks can be used for the visual decision-making process for a group management task in the industrial and city

environment at the research of various objects of interest, and also for liquidation of consequences of multiple accidents in the conditions of the severe danger for the person.

Also, based on this material, it is possible to create a large number of training courses, advanced courses, etc. This course will be useful for specialists in the field of robotics, automatic control systems, simulation of complex technical systems, etc. Modernization is simple, and the number of possible areas is vast. Such courses do not require specialized software, except MATLAB and ROS package on Linux.

References

1. V.D. Raskatov, V.I. Rubtsov, *Simulation MATLAB motion of the mobile robot's group when working in the industrial-urban environment*, AIP Conference Proceedings 2195 (2019)
2. V. Serebrenny, M. Shereuzhev, I. Metasov, *Approaches to the robotization of agricultural mobile machines*, MATEC Web of Conferences 2018, **Vol. 161**, Art.no 3014 (2018)
3. D. Vail, M. Veloso, *Dynamic multi-robot coordination*, Multi-Robot Systems: From Swarms to Intelligent Automata, Eds.: A. Schultz, L. Parker, and F. Schneider, Kluwer Academic Publishers, V. II., pp. 87–100 (2003)
4. Vladimir Serebrennyj, Andrey Boshlyakov, Alexander Ogorodnik, *Current Control in the Drives of Dexterous Robot Grippers*, Third International Conference on Interactive Collaborative Robotics, ICR 2018, Leipzig, Germany, September 18–22, pp. 239-248 (2018)
5. D. Hladek, J. Vascak, P. Sincak, *Multi-robot control system for pursuit-evasion problem*, Journal of electrical engineering, **Vol. 60**, No. 3, pp. 143-148 (2009)
6. I.A. Kalyaev, A.R. Gajduk, S.G. Kapustyan, *Modeli i algoritmy kolektivnogo upravleniya v gruppah robotov*[Collective control models and algorithms in groups of robots], Moscow: Fizmatlit (2009)
7. Y. Liu, J. Yang, Y. Zheng, Z. Wu, M. Yao, *Multi-Robot Coordination in Complex Environment with Task and Communication Constraints*, Intern. J. of Advanced Robotic Systems, **Vol. 10**, pp. 1-14 (2013)
8. D.A. Beloglazov, A.R. Gajduk, *Gruppovoe upravlenie podvizhnymi ob"ektami v neopredelennyh sredah*[Group control of moving objects in uncertain environments], edited by V.H. Pshihopova, Moscow: Fizmatlit (2015)
9. Sergey Vorotnikov, Konstantin Ermishin, Anaid Nazarova, Arkady Yuschenko, *Multi-agent Robotic Systems in Collaborative Robotics*, A. Ronzhin et al. (eds): ICR 2018, LNAI 11097, Springer, pp. 270-279 (2018)
10. S.A. Stoeter, P.E. Rybski, M.D. Erickson, M. Wyman, M. Gini, D.F. Hougen, N.A. Papanikolopoulos, *Robot Team for Exploration and Surveillance: Design and Architecture*, Proc. of the Intern. Conf. on Intelligent Autonomous Systems 6, Venice, Italy, July 2000, pp. 767–774 (2000)
11. A.V. Timofeev, *Intelligent Control and Operations Research for Multi-Agent Robot Systems*, International Autonomous Systems: Intern. Scientific Issue, Karlsruhe–Ufa: USATU, pp. 119–124 (1998)
12. T. Kamada, K. Oikawa, *AMADEUS: A Mobile, Autonomous Decentralized Utility System for Indoor Transportation*, IEEE Intern. Conf. on Robotics and Automation, Leuven, Belgium, May 16–20, **Vol. 4**, pp. 2229–2236 (1998)

13. M. Nathan, M. Zavlanos, V. Kumar, G. Pappas, *Distributed multi-robot task assignment and formation control*, Proc. of the IEEE Intern. Conf. on Robotics and Automation, pp. 128-133 (2008)
14. V.P. Noskov, V.I. Rubtsov, I.V. Rubtsov, *Matematicheskie modeli dvizheniya i sistemy tekhnicheskogo zreniya mobil'nyh robototekhnicheskikh kompleksov: uchebnoe posobie*[Mathematical models of motion and technical vision systems for mobile robotic complexes: study guide], Moscow: Izdatel'stvo MGTU im. N.E. Baumanova[Publishing BMSTU], (2015)
15. Konstantin Mashkov, Vasily Rubtsov, Ivan Rubtsov, *Development of robotics technologies in agriculture*, MATEC Web Conf. 224 05004 (2018). DOI: 10.1051/mateconf/201822405004
16. Kadhim Oleiwi, R. Al-Jarrah, H. Roth, B. Kazem, *Integrated Motion Planning and Control for Multi Objectives Optimization and Multi Robots Navigation*, IFAC-PapersOnLine 48, **Vol. 10**, pp. 99-104(2015)
17. V. Serebrenny, A. Boshliakov, G. Ovsiankin, *Active stabilization in robotic vision systems*, MATEC Web of Conferences 2018, **Vol. 161**, Art.no 3019 (2018)
18. S.A. Stoeter, I.T. Burt, N. Papanikolopoulos, *Scout Robot Motion Model*, Proc. of the IEEE Intern. Conf. on Robotics and Automation, Taipei, Taiwan, May (2003)
19. Description of the Simscape Multibody package. Available at: <https://www.mathworks.com/products/simmechanics.html>. (last accessed 2020/02/20)
20. Description of the Mobile Robotics Simulation Toolbox. URL: <https://www.mathworks.com/matlabcentral/fileexchange//66586-mobile-robotics-simulation-toolbox> (last accessed 2020/02/20).
21. Description of the ROS Toolbox. URL: <https://www.mathworks.com/products/ros.html> (last accessed 2020/02/20).
22. S. Zabihifar, A. Yuschenko, *Hybrid force/position control of a collaborative parallel robot using adaptive neural network*, Lecture Notes in Computer Science 2018, **Vol. 11097 LNAI**, pp. 280 – 290 (2018). DOI: 10.1007/978-3-319-99582-3_29
23. K-Y. Tu, M-T. Chang Chien, *Formation Control of the Multi-Robot Team's Behaviors Based on Decentralized Strategies*, Proc. of the American Control Conference, pp. 1-5 (2006)