Pilot communication protocols for group of mobile robots in USAR scenarios

Artyom Pashkin
Laboratory of Intelligent Robotic Systems (LIRS)
Intelligent Robotics Department, ITIS
Kazan Federal University
Kazan, Russia
artyom.pashkin@gmail.com

Roman Lavrenov
Laboratory of Intelligent Robotic Systems (LIRS)
Intelligent Robotics Department, ITIS
Kazan Federal University
Kazan, Russia
lavrenov@it.kfu.ru

Aufar Zakiev
Laboratory of Intelligent Robotic Systems (LIRS)
Intelligent Robotics Department, ITIS
Kazan Federal University
Kazan, Russia
zaufar@it.kfu.ru

Mikhail Svinin
College of Information Science and Engineering, Ritsumeikan University
Kyoto, Japan
svinin@fc.ritsumei.ac.jp

***Abstract*—This paper describes process of development of communication protocol for group of mobile robots in Urban Search and Rescue (USAR) operations based on wireless technology and Robot Operating System framework. During this work cases of mobile robot's usage in USAR as well as modern worldwide researches in this area were studied and it was found out that one of the main issues with mobile robot usage in such situation is communication problem. Disasters typically destroy communication infrastructure and rescuers forced to use workarounds to be able to still use robots. Based on these studies goal for the paper has been set – to develop communication protocol that allows robots to communicate and exchange data with minimal human interference. To achieve this goal multiple types of communication technologies were analyzed and Bluetooth was chosen based on qualities most suited for this task. After that list of commands needed for USAR was combined and three of them (Follow, Stop, MoveAround) were chosen to be implemented during this research. Testing environment was created in simulator Gazebo with three TurtleBot robots to verify execution of developed command. Finally, commands and command handler were implemented and tested on robots in created testing environment by using trial scenarios.**

Keywords—USAR, communication protocol, mobile robots, ROS, Bluetooth.

I. Introduction and USAR researchs review

During Urban Search and Rescue (USAR) operations rescuers may face zones, which cannot be accessed by humans, because of hazardous environment or some physical limitations. For such cases mobile robots controlled remotely by human operator are being used [1]. However, these robots have their own limitations and one of them is a need for being constantly connected with operator, which not always may be achieved for various reasons: communication interferences, thick walls, large distance [2].

Several papers about USAR were reviewed to understand how robotics can help rescuers in such operations and in which direction moves modern researches. One of them is work about USAR simulator [3]. It details about problems with traditional ways of data exchange, such as damage to communication lines, malfunctions in systems of emergency population warning. It also looks into most important USAR system functions: work with information of different types, long-term planning, reliability.

Another article highlights key aspects of interaction between humans and robots in USAR operations [4]. It details that a typical scenario is an unmanned ground or aerial vehicle helps rescuers with exploration of disaster site. Operations like this may continue for days and authors of paper say that their project helps to integrate all experience accumulated over several sorties within a mission into a single world-view. They made it possible by using ontology and agent-based framework which unites team-members through usage of shared knowledge base. Authors of paper [5] want to use shared knowledge base as well so that robots could contextually share information with humans by using augmented situation awareness. The work of Bartlett and Cooke [6] also highlights the idea of human and robot cooperation and compares intelligent robot with remotely controlled in performance, situation awareness, trust and workload.

Paper [7] describes development of new approach for cooperation of multiple robots. This approach based on hierarchical reinforcement learning (HRL). It allows robots to cooperatively learn to explore and identify victims and effectively allocate task between members of group.

In another research [8] author develops framework to effectively work with groups of heterogeneous robots. It helps to choose most suitable robots for new tasks and reform groups if one of the robots was lost or a new one was encountered.

Entire chapter in "Springer Handbook of Robotics" dedicated to disaster robotics [9]. Authors of works in this chapter say that over the course of time people started to use robots not only for life-saving response, but for recovery activities as well, such as re-establishing normal operations in community. Chapter details various types of robots used in many cases of disastrous events. One of the parts of the work highlights communication problems, because disasters typically destroy communication infrastructure. Because of

978-1-7281-3021-7/19/$31.00 ©2019 IEEE
DOI 10.1109/DeSE.2019.00017

that rescuers usually have to use either long wires or repeaters for wireless connection. In many cases robots were lost due to these issues. Most of the solutions for communication problems reduced to how to extend coverage zone for mobile robots so that human operator could still control them. In this paper it was decided to go in different direction which is automation of robots and reduction of required supervision from human operators.

Goal of this work is to develop protocol, which uses wireless communications and allows to unite number of robots in a group for them to cooperate with each other with minimal human interaction or completely autonomously.

One of the main parts of the protocol is a communication link that allows connecting mobile robots into a group and organizing them by using topology. This technology should be wireless and energy saving because robots are mobile and have limited battery capacity. In this paper multiple communication technologies were analyzed and compared to choose one that suits the most for the goal of this work.

The paper goes on with the list of commands which are needed for USAR and can be implemented in this protocol. From list three commands were chosen to be implemented during this work: Follow, Stop and MoveAround.

Selected environment for verification of the protocol is Robot Operating System (ROS) and simulator is Gazebo. Testing environment that consists of map and three robots were created for Gazebo. Map is a small office and robots are TurtleBots made by Willow Garage.

II. Selection of Wireless Technology

Protocol will be tested on robots that is in possession of Laboratory of Intelligent Robotic Systems (LIRS). Because of that it was decided to collect data about these robots, i.e. sensors and communication links. Results are on Table 1.

TABLE I. SENSORS AND COMMUNICATIONS ON LIRS ROBOTS

	Bluetooth	*Wi-Fi*	*Sensors*
PAL Robotics PMB 2	4.0	a/b/g/n/ac	Laser Hokuyo URG-04LX-UG1. IMU
Servosila Engineer	USB slot	Unknown version	Laser Hokuyo. Camera x4
Robotis OP 2	USB slot	n	IMU. Camera
Robotis OP 3	4.1	ac	IMU + Magnetometer. Camera Logitech C920 HD pro
DJI Phantom 4	Unknown version	Unknown version	Camera (Access through Manifold ROS pkg)
Artik	4.1	a/b/g/n/ac	Camera slot

Most of the robots have Wi-Fi and Bluetooth modules, or USB slot, which can be used to add needed communication module.

Various technologies were reviewed among wireless ones from which 4 most popular solutions were chosen and then were compared against each other: Bluetooth, Ultra-Wide Band (UWB), ZigBee and Wi-Fi. Results of comparison shown on Table 2 [10]. Most significant metrics for us highlighted with bold font.

As a result, Bluetooth was chosen. Its speed less than Wi-Fi and UWB, but it's enough for our purposes. Power consumption is more efficient on Bluetooth and ZigBee, but ZigBee's speed may be not enough if we decide to send some raw data like image from camera with 5Hz rate.

Topologies for communication network were researched as well. Default topology for Bluetooth is Piconet with 7 slaves connected to single master. Scatternet used to expand this network by connecting multiple Piconets, but most fitting for the task topology is Mesh Network. Mesh Network takes into consideration that devices in it may be not static, but mobile. Multiple implementations of such topology were found with described algorithms for organizing network, balancing, data transmitting [11] [12] [13].

Preferable version of Bluetooth is 5.0, because it has built-in mesh topology, which is very convenient for this work, but currently modules with version 4.2 are more common, cheaper and easier to found.

III. Testing Environment

Testing environment was created before protocol development with purpose to ease development and integration process.

As the main framework for development, Robot Operating System (ROS) was chosen [14]. ROS helps with development of software for different robots and comes with basic functional, such as movement and messaging [15].

TABLE II. COMPARISOIN OF WIRELESS COMMUNICATION TECHNOLOGIES

Standard	*Bluetooth*	*UWB*	*Zigbee*	*Wi-Fi*
IEEE spec	802.15.1	802.15.3a	802.15.4	802.11a/b/g
Frequency band	2.4GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 Ghz
Max signal rate	**1Mb/s**	**110Mb/s**	**250kb/s**	**54Mb/s**
Nominal range (meters)	**10**	**10**	**10-100**	**100**
Channel bandwidth	1MHZ	500MHz-7.5GHz	0.3/0.6 MHz; 2 MHz	22MHz
Modulation type	GFSK	BPSK, QPSK	BPSK(+ASK),	BPSK, QPSK COFDM, CCK, MQAM
Spreading	FHSS	DS-UWB, MB-OFDM	DSSS	DSSS, CCK, OFDM
Coexistence mechanism	Adaptive freq. hopping	Adaptive freq. hopping	Dynamic freq. selection	Dynamic freq. selection, transmit power control (802.11h)
Basic cell	**Piconet**	**Piconet**	**Star**	**BSS**
Extension of the basic cell	**Scatternet**	**Peer-peer**	**Cluster treemesh**	**ESS**
Max number of cell nodes	**8**	**8**	**> 65000**	**2007**
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

Version of ROS in this paper is Kinetic Kame, which was installed on Ubuntu 16.04.

Simulator for environment is Gazebo, because it's comes with ROS and allows to create effective simulation of mobile robots [16]. Gazebo can simulate indoor and outdoor spaces [17], functional models of robots, sensors, cameras, etc. One of the main advantages of Gazebo is that it is already pre-installed with ROS and fully integrated with it, which means that software created for robot's simulation models can be run on real ones.

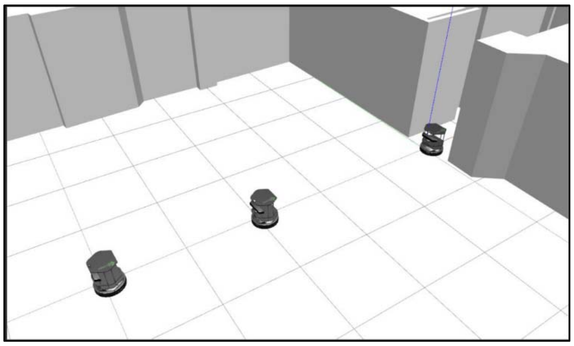

Fig. 1. Window of Gazebo simulator with testing environment

Testing environment for protocol consists of map “Small_indoor_scenario” that imitates office, and 3 models of robot TurtleBot, created by “Willow Garage” company [18].

Environment was created with help of official ROS documentation by creating ROS Launch files that start simulation of the map, spawn robots on it and launch all needed modules, such as movement and navigation. Fig. 1 shows windows of Gazebo simulator with 3 launched robots.

For movement robot uses Navigation stack. It's a ROS package that takes robot's data such as odometry, sensors streams, like laser scanners, and goal and outputs velocity commands that are sent to mobile base. To work properly it needs tree of robot's transformations which consists of information about how separate parts of the robot located with regards to each other. This tree and all navigation part were provided by robot's manufacturer because it's an open source project.

In future it's planned to launch different robots. At first would be used Husky, Russian robot Servosila Engineer, Hector Quadrotor.

IV. PROTOCOL DEVELOPMENT

Table 3 with list of commands that can be used in protocol was created. This list consists of basic interaction operations, that later can be combined for more complex commands. Example for such complex operation is command LandOnTop which allows to land flying robot to ground one.

A. Command Follow

First implemented command was Follow. Robot-1 (Leader) sends command Follow to Robot-2 (Follower) as message Command with type COMMAND_FOLLOW in ROS topic Robot-2/comm_protocol/command, after that Leader starts to send its position to Follower and Follower moves to specified position. Position represented by Frame which consists of position and orientation of robot's coordinate frame with regards to world coordinate system. Diagram for command Follow is shown on Fig. 2.

Type of message that contains coordinates is a geometry_msgs/PoseStamped, which is a ROS built-in type. It consists of coordinate system with regards to world coordinate system. Robot-1 sends this message to ROS topic Robot-2/comm_protocol/follow.

TABLE III. COMMANDS FOR PROTOCOL

Command	***Target***	***Parameters***	***Description***
Follow	Any	Frame	Follow sender.
ShowCam	Robot with camera	Frame, repeat timeout, repeat count	Send image of the specified position from camera. Repeat, Period – send once or periodically, period of repeats.
ShowScan	Robot with laser scanner		Same as ShowCam, but with scan.
LandOnTop	Flying and ground robots	Frame	Flying robots lands on ground one.
Goto	Any	Frame	Move to specified position
Stop	Any		Stop command that currently running
ForwardMsg	Any	Target, message	Forward message to target.
Return	Any		Return to “Base”
MoveAround	Any	Radius	Move around sender with specified radius.

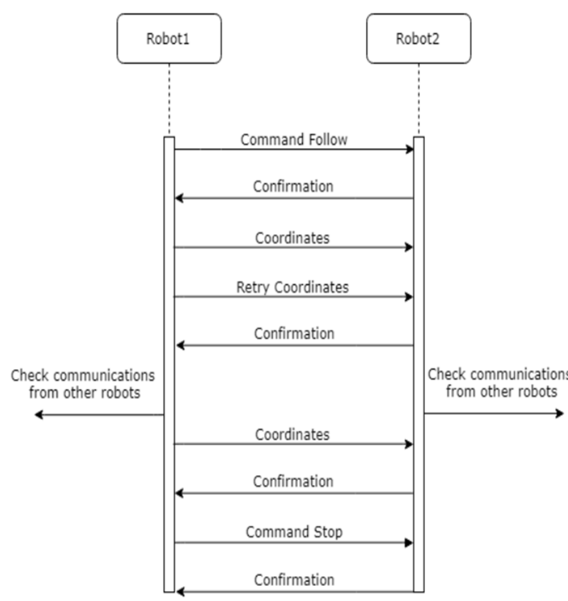

Fig. 2. Diagram for Follow command

Goal for robot is set by sending PoseStamped message to Navigation stack with help of Actionlib [19]. Actionlib is a

ROS package that provides easy-to-use interface and returns feedback from navigation commands.

After every message from leader, it is expected that follower sends Confirmation message to ROS topic Robot-1/comm_protocol/command as message Command with type COMMAND_CONFIRM. If message receive was not acknowledged protocol retries two more times and then stop command execution.

Periodically robot pauses command execution and checks incoming messages for command with higher priority. Priority system itself is not implemented yet.

For optimization of network usage timeout for message sending was added. Also, leader or sender will stop command execution in case of disconnecting and failing to reconnect.

Special scenario was created for testing purpose, which launches from Linux terminal by user. It forces one of the robots to send Follow command to another one. Example of command execution is shown on Fig. 3. Arrows show direction of movement. On upper half Robot-1 already passed through doorway. On lower half Robot-1 moved a little further and Robot-2 followed him through doorway as well.

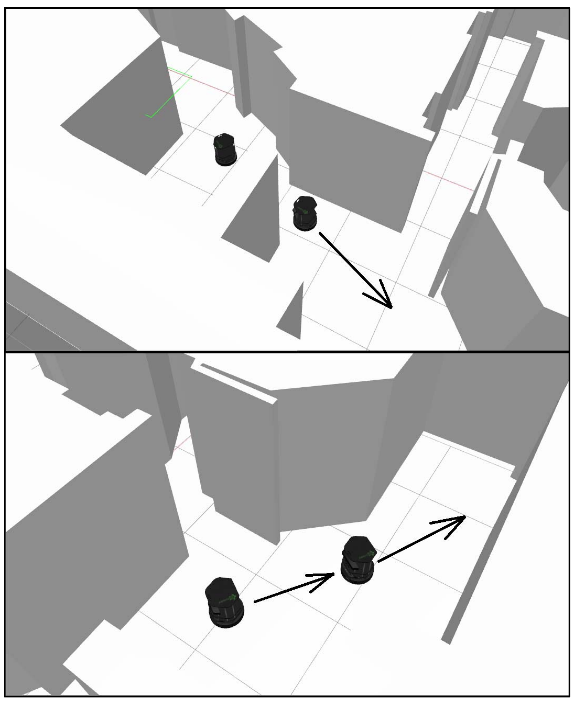

Fig. 3. Example of Follow command in Gazebo simulator

B. *Command Stop*

In order to stop execution of command was created command Stop. After receiving this command robot will stop running any command and stop movement to goal if there was one.

Protocol wait for command confirmation and retry two more times in case failure same as in Follow command scenario. Diagram for command Stop is shown on Fig. 4.

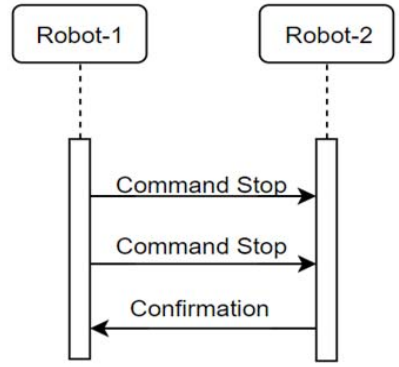

Fig. 4. Diagram for Stop command

C. *Command MoveAround*

This command is used to force receiver of command to move around sender with specified radius. Result of this command execution is on Fig. 5

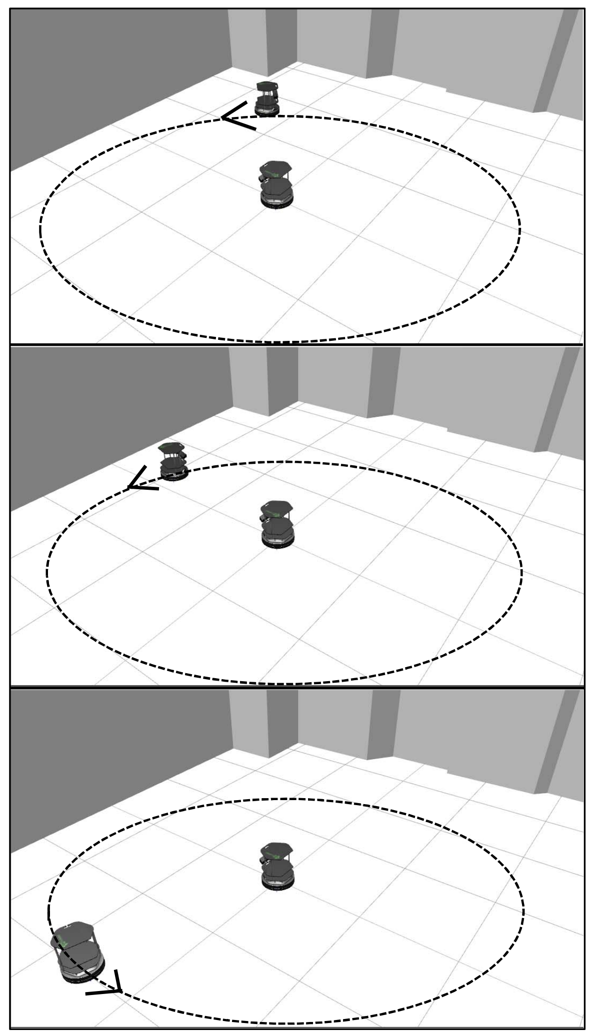

Fig. 5. Example of MoveAround command in Gazebo simulator

To make round trajectory axis X is divided to number of iterations (currently number is 20). For each next movement goal, Y is calculated from equation of circle, but orientation is also needed. For that we want to position robot same as tangent. Angle is calculated and transformed to quaternion, because it's the format needed by Navigation Stack [20].

For testing of this command was created special scenario similar to Command Follow. It shows window of Gazebo simulator with robots executing command MoveAround. Approximate trajectory of Robot-2 shown with dashed line. It's a circle with center in position of Robot-1. Direction of movement is anticlockwise and it's indicated by arrow. Diagram for command MoveAround is shown on Fig. 6.

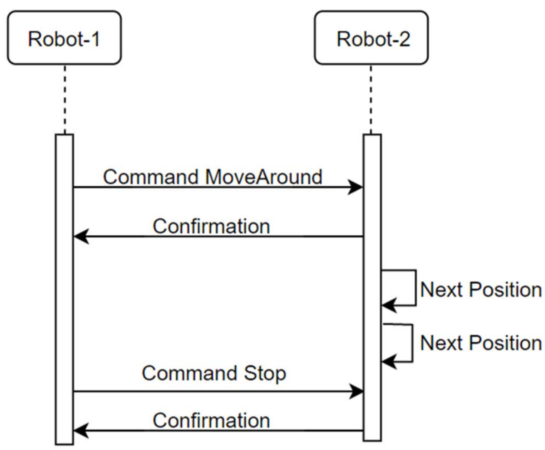

Fig. 6. Diagram for MoveAround command

V. CONCLUSIONS

Goal of this paper is creation of protocol, that gives allows robots to interact with each other almost or completely autonomously by using commands that would be useful in USAR operations. As a result of this paper: multiple wireless technologies were compared and Bluetooth was chosen, testing environment in Gazebo was created, list of commands for protocol was created and 3 of them (Follow, Stop and MoveAround) were implemented and tested in simulator.

In future, every command in list will be implemented, more commands will be designed by combining basic ones, Bluetooth usage will be integrated and protocol will be tested on real robots. In addition, the protocol will be tested on the previously developed [21, 22] multi-criteria path-planning algorithm.

ACKNOWLEDGEMENTS

The reported study was funded by the Russian Foundation for Basic Research (RFBR) according to the research project No. 19-58-70002.

REFERENCES

[1] Magid E. et al. Artificial intelligence based framework for robotic search and rescue operations conducted jointly by international teams, in Proceedings of 14th Int. Conf. on Electromechanics and Robotics "Zavalishin's Readings", ER(ZR) – 2019, – Vol. 154, DOI 10.1007/978-981-13-9267-2

[2] Alishev, N., Lavrenov, R., Hsia, K. H., Su, K. L., & Magid, E. (2018, September). Network Failure Detection and Autonomous Return Algorithms for a Crawler Mobile Robot Navigation. In 2018 11th International Conference on Developments in eSystems Engineering (DeSE) – 2018. - pp. 169-174.

[3] Tadokoro S. et al. The robocup-rescue project: A robotic approach to the disaster mitigation problem //Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. – IEEE, 2000. – vol. 4. – pp. 4089-4094.

[4] de Greeff J. et al. Human-robot teamwork in USAR environments: the TRADR project //Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts. – ACM, 2015. – pp. 151-152.

[5] Chandra R., Rocha R. P. Knowledge-based Framework for Human-Robots Collaborative Context Awareness in USAR Missions //2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC). – IEEE, 2016. – pp. 335-340.

[6] Bartlett C. E., Cooke N. J. Human-robot teaming in urban search and rescue //Proceedings of the Human Factors and Ergonomics Society Annual Meeting. – Sage CA: Los Angeles, CA : SAGE Publications, 2015. – Vol. 59. – №. 1. – pp. 250-254.

[7] Liu Y., Nejat G. Multirobot cooperative learning for semiautonomous control in urban search and rescue applications //Journal of Field Robotics. – 2016. – Vol. 33. – №. 4. – pp. 512-536.

[8] Gunn T., Anderson J. Dynamic heterogeneous team formation for robotic urban search and rescue //Journal of Computer and System Sciences. – 2015. – Vol. 81. – №. 3. – pp. 553-567.

[9] Murphy R. R., Tadokoro S., Kleiner A. Disaster robotics //Springer Handbook of Robotics. – Springer, Cham, 2016. – pp. 1577-1604.

[10] Lee J. S. et al. A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi //Industrial electronics society. – 2007. – Vol. 5. – pp. 46-51.

[11] A Mesh Network for Mobile Devices using Bluetooth Low Energy Sirur S. et al. A mesh network for mobile devices using Bluetooth low energy //SENSORS, 2015 IEEE. – IEEE, 2015. – pp. 1-4.

[12] Ok D. et al. Self-organizing mesh topology formation in Internet of things with heterogeneous devices //Networks and Communications (EuCNC), 2017 European Conference on. – IEEE, 2017. – pp. 1-5.

[13] Zhang B. et al. BLE mesh: a practical mesh networking development framework for public safety communications //Tsinghua Science and Technology. – 2018. – vol. 23. – №. 3. – pp. 333-346.

[14] Quigley M. et al. ROS: an open-source Robot Operating System //ICRA workshop on open source software. – 2009. – Vol. 3. – №. 3.2. – pp. 5.

[15] Afanasyev, I., Sagitov, A., & Magid, E. ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In International Conference on Advanced Concepts for Intelligent Vision Systems – 2015. – pp. 273-283.

[16] Takaya K. et al. Simulation environment for mobile robots testing using ROS and Gazebo //2016 20th International Conference on System Theory, Control and Computing (ICSTCC). – IEEE, 2016. – pp. 96-101

[17] Lavrenov, R., Zakiev, A., & Magid, E. Automatic mapping and filtering tool: From a sensor-based occupancy grid to a 3D Gazebo octomap. In 2017 International Conference on Mechanical, System and Control Engineering (ICMSC) – 2017. - pp. 190-195.

[18] Garage W. Turtlebot //URL: http://turtlebot.com/turtlebot2

[19] Marder-Eppstein E., Pradeep V. ROS actionlib package documentation //URL: http://www.ros.org/wiki/actionlib

[20] Bokovoy, A., Fomin, M., Yakovlev, K.S. Implementation of the Pathfinding System for Autonomous Navigation of Mobile Ground Robot. Int. Conf. "Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems", Tampere, Finland – 2018. – pp. 72-78.

[21] Lavrenov, R., Matsuno, F., & Magid, E. Modified spline-based navigation: guaranteed safety for obstacle avoidance. In International Conference on Interactive Collaborative Robotics – 2017. – pp. 123-133.

[22] Lavrenov, R., Magid, E.,, Matsuno, F., Svinin, M. M., Suthakorn, J. (2019). Development and implementation of spline-based path planning algorithm in ROS/Gazebo environment. *Trudy SPIIRAN*, – 2019 – Vol.*18*(1) – pp.57-84.