Modelling a Crawler Robot Using Wheels as Pseudo-Tracks: Model Complexity vs Performance

Ilya Moskvin, Roman Lavrenov, Evgeni Magid
Laboratory of Intelligent Robotic Systems,
Intelligent Robotics Department,
Kazan Federal University,
Kazan, Russia
e-mail: magid@it.kfu.ru

Mikhail Svinin
Robot Dynamics and Control Laboratory,
College of Information Science and Engineering,
Ritsumeikan University,
Kyoto, Japan
e-mail: svinin@fc.ritsumei.ac.jp

***Abstract*—Crawler mobile robots have a large variety of applications, including urban search and rescue. Before new algorithms and concepts could be integrated into a real robot control system and tested in field experiments, extensive virtual simulations should be performed in order to verify them carefully. This requires a proper modelling of a robot in a virtual environment, but a specific construction of crawler robot crawlers complicates the modelling. In this paper, in order to provide a high level of physical interaction similarity of a crawler with a supporting surface we model crawlers of mobile robot Servosila Engineer with a set of virtual wheels and study a relationship between a model complexity and a simulation performance. The model complexity is reflected by a number of virtual wheels that approximate each crawler of the robot. Verification experiments were performed in ROS/Gazebo simulator.**

Keywords-ROS; Gazebo; modelling; mobile robot; crawler robot; simulation performance; model complexity; virtual wheels

I. Introduction

Crawler mobile robots have a large variety of applications. During urban search and rescue operations, crawler robots are employed for victims' and survivors' search within dangerous areas and debris while avoiding risks of injuring for human rescue teams. Since a high level of mobility is a natural feature of crawler robots, such robots are often preferred within urban search and rescue tasks [1], [2].

Before new algorithms and concepts could be integrated into a real robot control system and tested in field experiments, extensive virtual simulations should be performed in order to verify them carefully [3]. Testing the algorithms in virtual environments significantly reduces the cost of mistakes and speeds up development processes. However, this requires the proper modelling of a robot in a virtual environment [4]. But a specific construction of crawler robot crawlers significantly complicates the modelling. The modelling difficulties are caused by a necessity to implement compound physics of a crawler motion and interaction with a supporting surface, which is hard to precisely predict due to uneven contact surface, varying friction and other issues [5].

By now, the most frequently used approach of crawler modelling suggests approximating each crawler with a set of well-synchronized virtual wheels. Even though this approach introduces limitations of overcoming some obstacles and reduces adherence of a track with a supporting surface, we might attempt minimizing these negative effects by increasing the number of wheels while simultaneously decreasing their diameter. An infinite number of wheels with an infinitely small diameter would ideally approximate a continuous contact area of a crawler and its supporting surface, but unfortunately, such solution is not feasible within modern simulations.

In this paper, we attempt to find a feasible trade-off between model complexity (and thus the level of approximation) that is reflected by a number of virtual wheels being employed and a performance of such model within a simulator. The performance is measured in terms of robot ability to traverse an environment, real-time that is required to perform a particular locomotion task and real-time factor (RTF) of the simulator. RTF could be considered as a criterion of time efficiency of a simulation, which is measured as a ratio of the simulation time to a real-time for a particular task; e.g., if it takes 10 seconds of a real-time to compute 1 second of simulation time, RTF is equal to 0.1. Thus, efficient simulations are characterized by higher RTF[6].

In our research, we used Russian mobile crawler robot Servosila Engineer (Fig. 1), which is described in more details in Section III. We model the robot in Gazebo simulator, which is integrated with Robot operating system (ROS), and analyze the influence of virtual wheels number on the simulation performance. In simulated experiments, we vary the number of virtual wheels from six to thirty with an incremental step of four, while measuring execution time, RTF and robot ability to traverse an environment.

II. Existing Approaches

A large variety of different constructions of mobile crawler robots exist. Typical tracked robots, which are employed in commerce, military, governmental and private services, could be categorized into 3 major groups depending on configuration of tracks: fixed double tracked robots (e.g., [7]); four-track robots that have two main crawlers and two sub-crawlers (e.g., [8]) and six-track robots with four sub-crawlers (e.g., [9],[10]).

One should also distinguish two large groups of tracked vehicles with regard to the flexibility of a crawler shape: a fixed shape and variable geometry tracked vehicles. Fixed shape type includes non-variable geometry robots, which always keep a constant shape and thus cannot climb

978-1-7281-6784-8/20/$31.00 ©2020 IEEE

sufficiently high steps even if good traction between a crawler and a supporting surface could be achieved (technically, the step should not exceed a half of a track-driving wheel diameter). This construction has high reliability and allows a simpler control and path planning due to the lower number of degrees of freedom (DoF). On the contrary, variable geometry tracked type robots are able to modify their geometry in order to change their center of mass and climb higher obstacles. For example, to modify geometry, a robot could use its sub-crawlers (flippers) or change the shape of its main crawlers (tracks) [11]. In [11] the authors drive their robot upstairs with and without flippers in order to demonstrate that the use of the flippers makes motion safer and easier.

An example of the most commonly used fixed shape crawler robot model is Hector Darmstadt robot [12]. Authors in [8] used a similar to Hector robot model approach in order to create a simplified model of Servosila Engineer robot. The simplified model had static flippers and four invisible pseudo-wheels on each side of the robot that were used to approximate main tracks of the robot.

An interesting example of variable geometry tracked vehicle is described in [13]. The authors created a transformable wheel–track robot with a tail rod, which is used for relocation of the robot's center of gravity. The robot works in a wheel mode on a flat terrain, which allowed for higher velocities and energy saving. If a necessity to climb stairs arises, the robot switches to a track mode.

In [10] authors presented a six-track robot (two main crawlers, four sub-crawlers) that surmounts obstacles by calculating sub-crawlers' required positions from obstacle's height and elevation angle. Several types of obstacles including stairs and channels of different heights were used in Matlab simulation. Calisi et.al [14] simulated Alladin six-track robot in USARSim simulator [15], which was a popular modelling tool before ROS and Gazebo turned into the mainstream of robotics. The paper suggested rules for the front and rear flippers behavior in the presence of obstacles, which were verified both in simulation and with a real robot. Another approach for autonomous flipper control modelling in ROS was suggested in [16], where the robot autonomously learned a flipper control strategy for obstacle traversal without any prior knowledge.

Pecka et.al [17] presented a technique that allows for both computationally fast and sufficiently plausible simulation of vehicles with non-deformable tracks. They compared models with chain-like deformable tracks, non-deformable tracks, approximated by four wheels tracks and tracks that were constructed from plates with grousers.

III. SERVOSILA ENGINEER ROBOT AND ITS GAZEBO MODEL

Servosila Engineer robot (Fig. 1) was designed for search and rescue operations in dangerous and unreachable for human environments, including typical urban search and rescue scenarios. It is dust-proof and waterproof, and is capable of working in harsh weather conditions. Its main crawlers and additional front sub-crawlers (flippers) allow overcoming obstacles that may appear as a result of natural or human-made disasters. The weight of a fully equipped robot is about 16 kg; Table I describes weights of all parts of the robot that were used for its modelling in Gazebo simulator. The robot is equipped with a powerful 4-DoF manipulator that can grasp and move objects, e.g., a standard fire extinguisher. Sensors, which are located inside the robot head, include a laser range finder, inertial sensors, a stereo vision system (two frontal cameras) and a single zoom camera in the front part of the head and a single back-looking camera [18]. Table II describes the characteristics of Servosila Engineer robot onboard system.

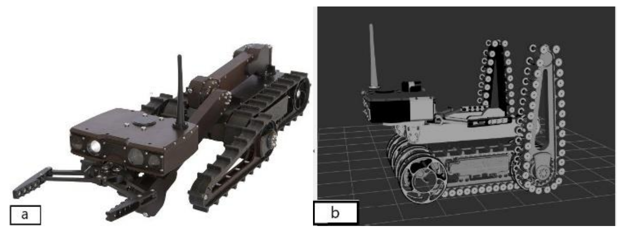

Figure 1. Servosila Engineer robot (a), its model in Gazebo simulator (b).

TABLE I. WEIGHT CHARACTERISTICS OF SERVOSILA ENGINEER ROBOT

Equipment	Weight parameter
Robot chassis with two main reversible tracks, two traction motors and motor control electronics	8.8 kg
On-board control and power system	2.1kg
Sealed connector for external payloads or external computer	0.1 kg
LiFePo battery	3.7 kg
Power supply for a standard robot battery (with a cable)	1.6 kg

The basic model of the robot in Gazebo simulator contains a base, two frontal flippers, four large wheels (that are used to rotate rubber tracks of the real robot), three parts of the robot head construction, and a set of small wheels on each side of the base that model the main crawlers of the robot. These wheels are used to approximate the crawlers behavior. To describe the configuration of the model we used URDF format. Each part of the model was created using mesh files, which were kindly provided by the robot manufacturer, Servosila company. The track wheels, the flipper wheels and the flippers of the model are movable while the head is static.

TABLE II. SERVOSILA ENGINEER ROBOT ONBOARD SYSTEM CHARACTERISTICS

System	Parameter
Processor	Intel i5, 4th generation
RAM	4GB
SSD	32 GB
OpenCL technology	supported
Radio channel	WiFi

To move the model in Gazebo simulation we created a controller, which simulates differential driving abilities. It contains base controllers for each virtual wheel of the crawlers and the flippers. The controllers of the flippers are based on ROS Gazebo control plug-in [19]. To apply this plug-in for the model we created configuration files for the virtual wheels' controllers and the flippers controllers, and added transmission tags to URDF file of the model [20]. The high- level controller synchronizes all virtual wheels and allows to control them simultaneously with the flippers. More details about the robot modelling could be found in [21].

Virtual experiments were run in Gazebo simulator, which allows adding correctly created models to any scene while providing physics-based interaction of the model and the scene that approximates real-world physics. Yet, if the model has a lot of details and is hard to calculate in real-time, Gazebo slows down the simulation time. This ratio between real-time and simulation time, which is tracked by the RTF indicator, indicates the model complexity. For the virtual experiments, we created a patch of a map with stairs to test the ability of different (in a sense of a virtual wheels number) models to negotiate with traversable obstacles [22].

IV. Comparative Analysis of Models

A. Metrics

Virtual experiments in Gazebo are run to compare the applicability of the models with regard to their complexity (a number of wheels) and performance within a simulation, which is based on RTF. Gazebo RTF is used as a parameter that indicates the model complexity ***C***.

We vary the model complexity by increasing a number of virtual wheels that approximate the main crawlers. This directly influences the robot capability to traverse low-height obstacles (including stairs); we refer this as "penetration capability" of the robot ***P***. In particular, in simulations this parameter corresponds to the inverse of simulation time ***T*** that is required in order to complete a benchmark stairs climbing task by different robot models. Value ***t*** corresponds to real world time that was used for a benchmark stairs climbing task by a corresponding robot model. We set penetration capability ***P*** to the inverse of the magnitude of ***T***, which corresponds to an analogue of a normalized speed of a benchmark stairs traversal task. Thus, the faster the model traverses the benchmark stairs - the higher is its penetration capability ***P***. The relationship between the aforementioned parameters is reflected by the following equation:

$$P = \frac{1}{T} = C * t \tag{1}$$

B. Virtual Experiments

Virtual experiments were performed with seven models with 6 (Fig. 2a), 10 (Fig. 2b), 14 (Fig. 2c), 18 (Fig. 2d), 22 (Fig. 3a), 26 (Fig. 3b), and 30 (Fig. 3c) virtual wheels per crawler. While the original model contained also virtual wheels that approximate the front flippers (Fig. 1), we removed these virtual flipper wheels for the models that were used in the comparative analysis since the flippers were always raised up during locomotion and thus did not influence virtual locomotion tests. This allowed saving some computational resources of the simulator without obtaining side effects on the results of the tests. The models differ only in the number of virtual wheels per crawler. When we increase the number of wheels - we decrease the wheel diameter and weight respectively so that to keep a total weight of the crawler (i.e., the total weight of approximating wheels) and locations all of virtual wheels within the crawler location (i.e., between the large wheels that are used to rotate the rubber track or the real robot).

Each model was initially placed in front of virtual stairs (Fig. 4) and a timer was started. The stairs are modelled in a such way that every stair has a different height, which gradually increases. The model was driven up the stairs using a controller while the front flippers were held upwards (at a constant angle) during the entire experiment. If a particular model failed to climb up the stairs, the test was marked as unsuccessful and variable ***T*** was set to zero. For successful tests both ***T*** and ***t*** variable obtained particular values. Moreover, since we log every test result, a model failure at a particular stair informs us about a maximal height of an obstacle, which could be traversed by the particular model.

C. Analysis

According to the results of virtual experiments increasing a count of wheels rises a complexity of a model and overcoming ability of model as shown in Fig. 5-6.

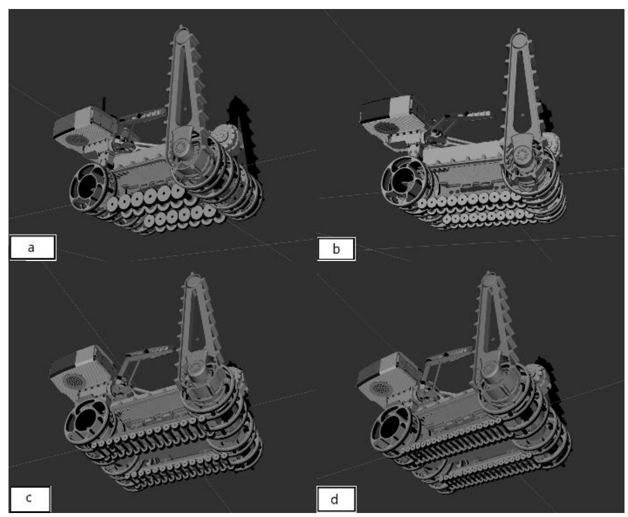

Figure 2. Robot models with 6 wheels (a), 10 wheels (b), 14 wheels (c) and 18 wheels (d).

Time of an experiment ***t*** increases with a number of wheels. When the number of wheels goes over 18, ***t*** increases significantly faster (Fig. 5, left image). This happens due to a necessity of changing speed and acceleration limits in order to keep correct locomotion of the robot, and thus smaller limits increase the time of the experiment. Naturally, increasing the number of wheels causes the increase of model complexity, which is immediately reflected in calculations time and RTF ***C*** (Fig. 6). The simulation time of the experiment ***T*** (Fig. 5 right image) is directly obtained by multiplication of ***t*** and RTF ***C***. The decrease of penetration parameter ***P*** with a number of wheels increase does not exceed five percent until the number of wheels is less than 18. When the number of wheels goes over 18, ***P*** decreases drastically (about three times), which is also explained by the necessity of changing speed and acceleration limits.

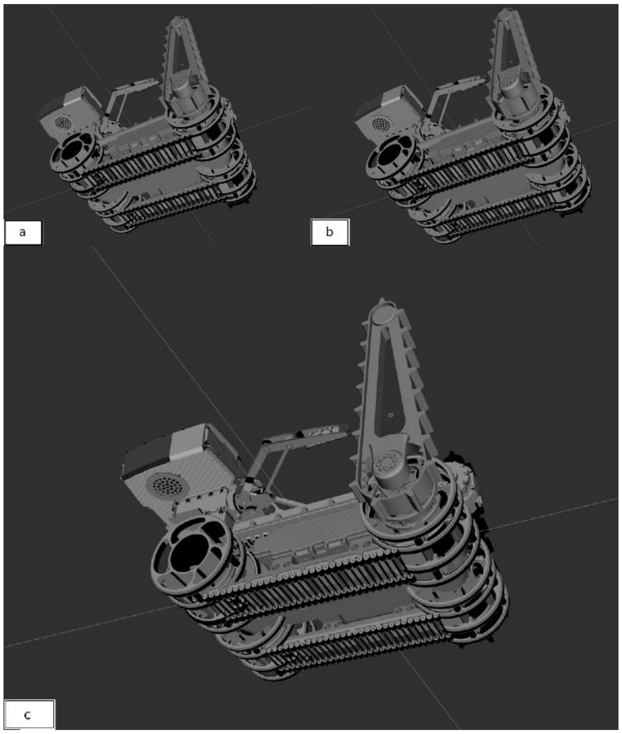

Figure 3. Robot models with 22 wheels (a), 26 wheels (b), 30 wheels (c).

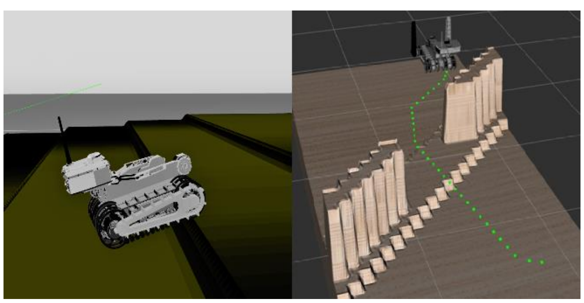

Figure 4. Stair climbing test in Gazebo simulator; note that stairs have different height (left) and an experiment of path planning (right).

Even though 18 virtual wheels model seems reasonably good for approximating a crawler with regard to the afore-mentioned parameters, virtual experiments demonstrated that for all controlled wheels proper acceleration and jerk upper limits should be set strictly. Setting wrong upper limits causes a model spin around its center of mass since all wheels create torques in a single direction. Empirical approach to limits selection demonstrated that if a number of virtual wheels per crawler exceeds 14 the proper acceleration limits are too small, which prevents an effective use of such models in Gazebo simulator.

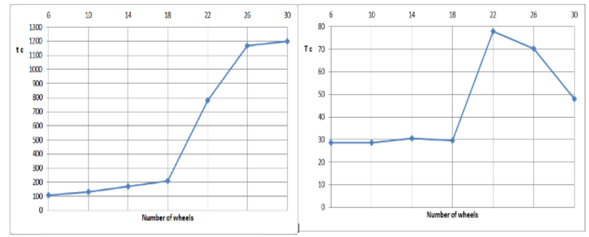

Figure 5. Time of an experiment (***t***) as a function of virtual wheels number (left) and time with regard to RTF (right).

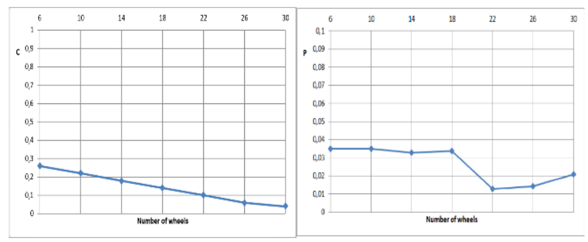

Figure 6. RTF (***C***) dependence on virtual wheels number (left) and the penetration capability of the robot (***P***) in stair climbing task as a function of virtual wheels number (right).

V. Conclusions

In this paper, in order to provide a high level of physical interaction similarity of a crawler with a supporting surface, we modelled crawlers of mobile robot Servosila Engineer with a set of virtual wheels and study a relationship between model complexity and simulation performance. The performance was measured in terms of robot ability to traverse an environment, real-time that is required to perform a particular locomotion task and real-time factor (RTF).

We used Russian mobile crawler robot Servosila Engineer, which was modelled in ROS-based Gazebo simulator. In simulated experiments, we varied a number of virtual wheels from six to thirty with an incremental step of four. To evaluate the model’s mobility for environment traversal the robot was teleoperated to climb stairs. The wheel number increase naturally increased the number of calculations, while the dependence of robot ability to traverse an environment had a non-linear behavior. Verification experiments in Gazebo simulator demonstrated that the optimal number of wheels for Servosila Engineer model was 18 virtual wheels per single crawler. For 18 virtual wheels, even though RTF factor was two times worse than for 6-wheel approximation, all other benchmark criteria were within an acceptable level. Further increase of the number of virtual wheels significantly increases the physical similarity between a simulated interaction of a model with a supporting surface and the behavior of the robot in real-world environments, but this significantly decreases model performance within the simulator and makes simulation infeasible.

A virtual empirical approach to acceleration and jerk upper limits selection of controlled wheels demonstrated that if a number of virtual wheels per crawler exceeds 14, the corresponding limits become too small, which prevents an effective use of such models in Gazebo simulator. Therefore, it is recommended to keep the number of virtual wheels between 14 and 18 per crawler. Next, we plan to use this 14-wheel per crawler model for path planning algorithms verification within a random step environment in Gazebo simulator.

ACKNOWLEDGMENT

This work was supported by the Russian Foundation for Basic Research (RFBR), project ID 19-58-70002, and by the Japan Science and Technology Agency, the JST Strategic International Collaborative Research Program, Project No. 18065977. This work was partially supported by the research grant of Kazan Federal University.

REFERENCES

[1] K. Ito, H. Maruyama, Semi-autonomous serially connected multi-crawler robot for search and rescue. Advanced Robotics, vol.30(7), 2016, pp. 489-503.

[2] E. R. Stepanova, M. Heyde, A. Kitson, T. Schiphorst, B. E. Riecke, Gathering and applying guidelines for mobile robot design for urban search and rescue application. Int. Conf. on Human-Computer Interaction, Springer, 2017, pp. 562-581.

[3] Vokhmintsev, A., Timchenko, M., & Yakovlev, K. Simultaneous localization and mapping in unknown environment using dynamic matching of images and registration of point clouds. In 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). IEEE, 2016, pp. 1-6.

[4] Ronzhin, A., Saveliev, A., Basov, O., & Solyonyj, S. Conceptual model of cyberphysical environment based on collaborative work of distributed means and mobile robots. In International Conference on Interactive Collaborative Robotics. Springer, Cham. 2016, pp. 32-39.

[5] D. Bereznikov, A. Zakiev Network Failure Detection and Autonomous Return for PMB-2 Mobile Robot. International Conference on Artificial Life and Robotics (ICAROB 2020), pp. 444-447.

[6] G. Vasilyev, A. Sagitov, L. Gavrilova, Su K.-L., T. Tsoy, Walking algorithm for ROBOTIS OP3 humanoid robot with force sensors. The 12th International Conference on the Developments in eSystems Engineering (Kazan, Russia; 7-10 October 2019)

[7] E. Magid, T. Tsubouchi, E. Koyanagi, T. Yoshida, Static balance for rescue robot navigation: Losing balance on purpose within random step environment. International Conference on Intelligent Robots and Systems, 2010, pp. 349-356.

[8] I. Mavrin, R. Lavrenov, M. Svinin, S. Sorokin, E. Magid, Remote control library and GUI development for Russian crawler robot Servosila Engineer. MATEC Web of Conferences, EDP Sciences, 2018, v. 161, p. 03016.

[9] K. Nagatani, S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, Y. Hada, Redesign of rescue mobile robot Quince. International Symposium on Safety, Security, and Rescue Robotics, 2011, pp. 13-18.

[10] S. Xie , S. Bao, B. Zou , H. Pu , J. Luo, J. Gu, The Research on Obstacle-surmounting Capability of Six-track Robot with Four Swing Arms. Int. Conf. on Robotics and Biomimetics (ROBIO) 2013, pp. 2441-2445.

[11] J. L. Paillat, Variable Geometry Tracked Vehicle, description,model and behavior. Mechatronics 2008.

[12] S. Kohlbrecher, C. Rose, D. Koert, P. Manns, F. Kunz, B. Wartusch, K. Daun, A. Stumpf, O. Stryk, RoboCup Rescue 2016 Team Description PaperHector Darmstadt

[13] X. Gao, D. Cui, W. Guo, Dynamics and stability analysis on stairs climbing of wheel–track mobile robot. International Journal of Advanced Robotic Systems, 2017.

[14] D. Calisi, D. Nardi, K. Ohno, S. Tadokoro, A semi-autonomous tracked robot system for rescue missions. SICE Annual Conference, 2008, pp. 2066-2069.

[15] S. Balakirsky, C. Scrapper, S. Carpin, M. Lewis, USARSim: providing a framework for multi-robot performance evaluation. In Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, 2006, pp. 98-102.

[16] M. Pecka, V. Salansky, K. Zimmermann, T. Svoboda, Autonomous flipper control with safety constraints. Int. Conf. on Intelligent Robots and Systems, 2016, pp. 2889-2894.

[17] M. Pecka, K. Zimmermann, T. Svoboda, Fast Simulation of Vehicles with Non-deformable Tracks. International Conference on Intelligent Robots and Systems, 2017.

[18] Sokolov, M., Bulichev, O., & Afanasyev, I. Analysis of ROS-based Visual and Lidar Odometry for a Teleoperated Crawler-type Robot in Indoor Environment. In ICINCO (2), 2017, pp. 316-321.

[19] Qian, W., Xia, Z., Xiong, J., Gan, Y., Guo, Y., Weng, S., ... & Zhang, J. Manipulation task simulation using ROS and Gazebo. In 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). IEEE, 2014, pp. 2594-2598.

[20] Yousuf, A., Lehman, W., Mustafa, M. A., & Hayder, M. M. Introducing kinematics with robot operating system (ROS). In 122nd ASEE Annual Conference and Exposition. 2015, pp. 1-18.

[21] I. Moskvin, R. Lavrenov, Modeling Tracks and Controller for Servosila Engineer Robot. In Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings", Springer, Singapore, 2020, pp. 411-422.

[22] 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). IEEE, 2017, pp. 190-195.