Chapter 33
Modeling Tracks and Controller for Servosila Engineer Robot

Ilya Moskvin and Roman Lavrenov

Abstract Modeling is an important part of solving any problem in robotics. Models let our opportunity to make mistakes that could break real robot and learn on these mistakes causing no damage to the robot. This paper presents a creation of model for crawler robot “Servosila Engineer.” In this model, tracks were replaced by big count of small wheels to get construction physically close to the track but with simple realization for simulation. To test efficiency of model was created three groups of controllers: for ground tracks wheels, flippers and flippers tracks wheels. This model takes into account mistakes made creating previous version and has opportunity to be used in teleoperation mode. To create controller, we tried some base types of controllers represented by Gazebo plugin. Created model and controller are integrated with Gazebo simulation system and ROS framework and will be used in our future researches.

Keywords Modeling · Tracked robot · Crawler robot · ROS · Gazebo · Mobile robot · Servosila Engineer

33.1 Introduction

Nowadays, robots are used in many areas of human activity, including industrial manufacturing [1] and goods handling [2], medicine [3] and entertainment [4], space [5] and rescue [6]. In search and rescue robotics robots perform tasks of mapping [7], explore unreachable debris [8] and dangerous areas [9], support in people search and extraction [10]. To successfully perform these tasks, we need to solve a number

I. Moskvin (✉) · R. Lavrenov
Laboratory of Intelligent Robotic Systems (LIRS), Intelligent Robotics Department, Higher Institute for Information Technology and Intelligent Systems, Kazan Federal University, Kazan, Russia
e-mail: moskvin.ilya@yandex.ru

R. Lavrenov
e-mail: lavrenov.roman@gmail.com

A. Ronzhin and V. Shishlakov (eds.), *Proceedings of 14th International Conference on Electromechanics and Robotics “Zavalishin’s Readings”*, Smart Innovation, Systems and Technologies 154, https://doi.org/10.1007/978-981-13-9267-2_33

of sub-tasks, e.g., path planning [11], robot balance prediction and control on debris [12], human and scene recognition [13, 14], etc. To run experiments in real-world environments under realistic urban search and rescue scenario conditions using real robots is dangerous both for a human operator and for a robot itself. Therefore, in order to test research findings prior to integrating those into real search and rescue robotic systems, researchers typically prefer to verify them in a plausible simulation, where a realistic model of a robot could demonstrate similar to real-world behavior.

For rescue robotics, if a task requires using a ground mobile robot on urban debris, the optimal decision is to employ tracked robots, which provide significantly better capabilities of obstacle surmounting and adaptability on rough terrain than other types of unmanned ground vehicles [15].

Research that is presented in this paper was carried out as part of a path planning task on random step environment (RSE) for Servosila Engineer robot, which was created by Russian company "Servosila". RSE is an artificial environment that simulates typical debris of urban search and rescue operations [12, 16]. The setup of our RSE is described in more details in Sect. 33.4. To solve the path planning task, a robot must be able to overcome certain types of obstacles and identify impassable parts of terrain. In particular, to overcome some complicate shaped obstacles, tracked robots might exploit additional front or back flippers [17].

This paper concentrates on modeling of Servosila Engineer rescue robot. The modeling of this robot had been initially tackled by replacing tracks with closed structured in terms of physics construction [18]. The modeling concentrated on tracks modeling only and in order to speed up the modeling process, the authors had not properly modeled Servosila Engineer robot but had adapted a similar to Husky robot [19] model structure. The authors had used wheels with a diameter that was equal to the length of Servosila Engineer robot track's height and a single additional pair of wheels that simulated tracks of static flippers of the robot.

In our current research in order to realistically simulate the robot locomotion effect, we used a large number of small (comparatively to the dimensions of the robot) wheels. This approach provided a reasonable approximation of a tracked locomotion in terms of physical interaction between a track and its underlying terrain as well as a contact area of the tracks with the surface. We created a model of Servosila Engineer robot with movable flippers, which is able to overcome small obstacles while moving through RSE. The developed model simulates real-robot behavior significantly better than the initial model, which had been presented in [18].

The rest of the paper is organized as follows. Section 33.2 presents a literature review of the previous research in the area. Section 35.3 deals with the system setup, including modeling and controller setup. Section 33.4 demonstrates our experimental work within Gazebo simulator and finally, we conclude in Sect. 33.5.

33.2 Literature Review

In previous paper about creating the model of Servosila Engineer robot, the authors attempted to use imaginary wheels instead of tracks. Gazebo plugin that gives ability to use tracks made of plates. They used model based on "Hector" vehicle model with addition of four imaginary wheels on each side, one of these wheels was on flipper. Flippers were static because they do not affect navigation task performance.

To make robot movable, they used hector navigation package. Diameter of imaginary wheels was equal to track height. Therefore, this model of robot has three surface contact points from each track and two additional contact points from flippers. Tracks have solid surface contact area, so these wheels were not enough to simulate track motions. "Hector" navigation involves moving along flat terrain and flippers was in static state, so this model is not intended to overcome obstacles. Making wheels smaller and increasing count of wheels in conjunction with creating custom controller must bring model close to real robot.

Xie [20] described experiments with crawler robot SHU-I. It is a six-track robot with four independent swig arms. Authors analyzed its capability of stairs climbing. In addition, the relationship between channel-crossing capability and structural parameters of the six-track robot were analyzed. They established a mathematical model, deduced the optimum surmounting capability, and checked it in simulation. In the paper, they described sequence of motions and kinematics calculations for this robot to overcome stairs. They tested channel-crossing ability of six-track robot with four independent swig arms and tested robot with stairs of different stage height.

SHU-I differs from "Engineer," but, however, if eliminating motions of back pair of arms, we can apply result motions and calculations to "Engineer" taking into account difference between flippers and arm's length. Unfortunately, authors did not described the details of model creating but mentioned that used one motor for front and one motor for backtracks.

Biggest part of the paper is devoted to calculations of center of mass of the robot movement through the climbing. Therefore, this paper was not very useful for modeling research, but it will be very useful for path planning calculations on RSE.

Interesting example of crawler robot is described in [21]. Tracks of this robot placed on some kind of feet, so they are on the same line and this robot cannot rotate staying on same place. This paper described the modelling solution for crawler robot and solving the problem of its pose error. Authors calculate pose error model consisting of position error and angle error. Unfortunately, authors did not describe the process of modeling of robot.

Robot Aladdin presented in [22] has similar structure as SHU-I. Authors use USARSim to check if this robot can explore the environment overcomes some simple obstacles. They made experiments with different levels of autonomy of the robot. As we can find from figures model of the robot for simulation is very simple, it has four wheels to move and use flippers only to lean on the surface and move next. So, abilities of this model are very limited and if it loses contact between surface and all four wheels, it will lose the moving ability.

Some simulations of vehicles with non-deformable tracks are described in [23]. Authors divide tracks by materials and opportunity to be deformed.

Authors distinguish four types of track models chain-track deformable tracks, track approximated by four wheels, track made by plates with grousers and non-deformable track model. They compared all these models in six use cases: computation speed, plausibility on flat surfaces, plausibility on rough terrain, non-deformable tracks, and deformable tracks and grousers. Realization of each type of tracks differed only in the representation of track, so all models had same parameters. Therefore, they created four models: model based on contact surface motion that made motions like CSM, wheels instead of tracks—model had eight wheels, four on each side. This model has close representation to our model, but our case has bigger contact surface conditioned by smaller size and bigger count of wheels. In addition, wheels of their robot are velocity controlled synchronized on each side by speed. Next model type is subdivision to plates—it is the model with belt subdivided into 10 or 2 cm plates. Next is no friction model, and it moves just using forces applied to center of mass of the robot.

Authors' tested models in cases of straight move, rotation, circular moving, back and forth, overcoming ramp, staircase, stand on staircase and pallet.

The research [24] showed that the simulation of tracks using plates is good in all cases instead of case when you need height computation speed. Other models showed good results in the speed of simulation and plausibility on flat surfaces. Model with wheels, which is similar to our "Engineer" model, provided good plausibility and was computationally fast. However, it suffers from unrealistic slippage and stands on staircase scenario. The results show that presented contact surface motion is one of the fastest ones and showed good results in all scenarios besides circular path.

33.3 Literature Review

33.3.1 Model Creation

Russian "Servosila" company creates mobile ground robot "Engineer" pictured on Fig. 33.1. It was designed for search and rescue operations in unreachable or dangerous for human terrains. "Engineer" is dust- and waterproof, and it is capable of working in hard weather conditions. Tracks and flippers make it able to overcome obstacles resulting from disasters. Weight of full-equipped robot is near 16 kg. The robot is able to penetrate into damaged buildings and explore it floor by floor climbing stairs with the help of flippers. Head of the robot is equipped with robotic arm that can move objects. Set of sensors includes laser scanner, stereo vision system (two cameras in front side of the head and one on the backside) and block of inertial sensors.

3D model of robot pictured on Fig. 33.2 was created for Gazebo simulation system integrated with ROS. Model contains base, two flippers, and four big wheels, three

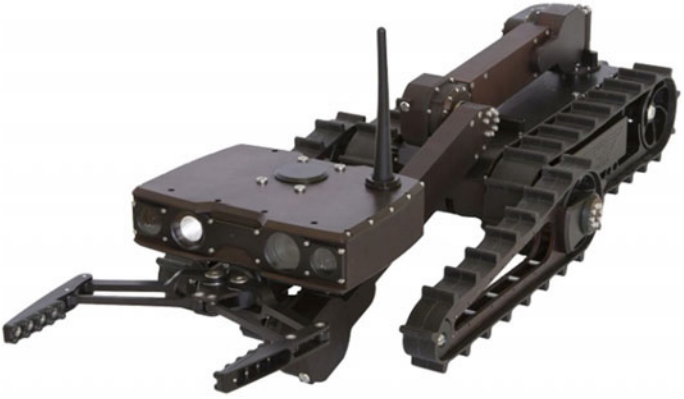

Fig. 33.1 Robot Servosila Engineer

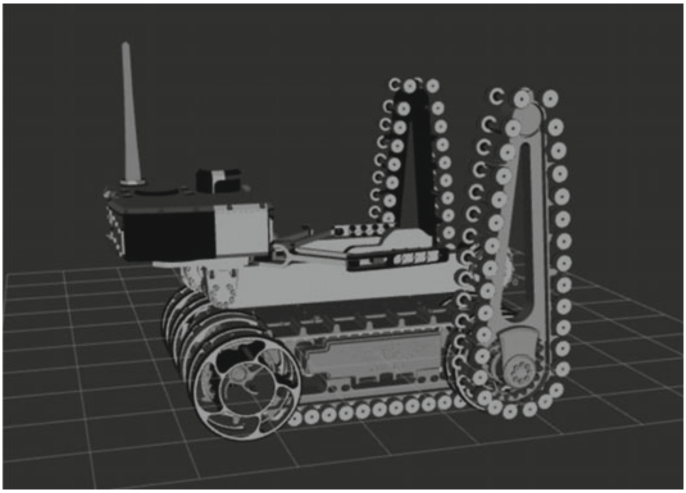

Fig. 33.2 Model of Servosila Engineer

parts of head construction, 11 small wheels instead of tracks on each side and 16 wheels instead of flipper tracks.

We also created a view of this model which you can see on Fig. 33.3 with alpha 0.5 to be able to see all components of the robot and motions of all parts while we were testing the model.

To simulate moving using tracks, we replaced each track including tracks on flippers by 11 wheels for ground track and 16 wheels for flipper tracks. Large number and small size of wheels let us get a surface area close to surface area of tracks. We created all parts of the robot using CAD models provided by the creators of the robot. “Servosila” company also gives us information about weight and inertia of

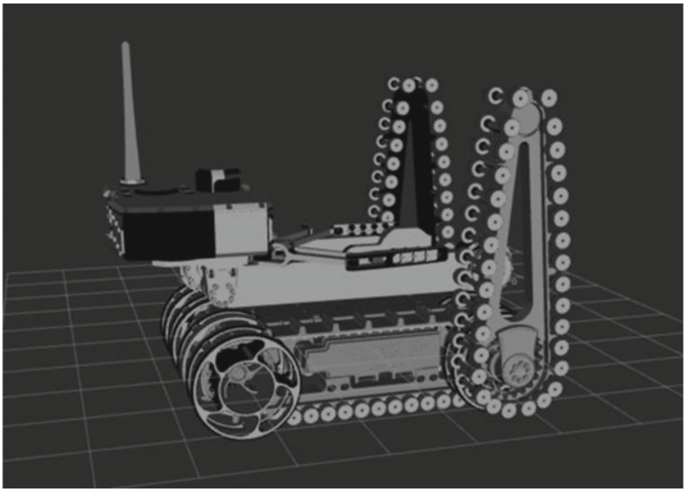

Fig. 33.3 Model of robot where all parts are visible

main parts of the robot: base flippers, tracks, two parts of robotic arm and head. Weights of wheels on tracks were calculated using (33.1):

$$m_\mathrm{w} = \frac{m_\mathrm{t}}{\mathrm{wheel}}\mathrm{count}, \tag{33.1}$$

where m_w—mass of wheel, m_t—mass of track, wheel count is 11 for track and 16 for flipper. Inertia of wheels was set as default. Received model is close enough to make experiments in simulation and be sure that there will be not a big difference with experiments with real robot.

33.3.2 Controller Setup

After we created model of robot, we need to move it in simulation. As simulation environment, we choose Gazebo so to create controller for track wheels plugin *gazebo_ros_control* was used.

This plugin gives the ability to use one of the three types of PID controller for each joint of the robot or fourth type for some count of wheels.

Main goal of controller is to calculate inverse kinematics for robot motions. We needed to take ability to move robot with given speed in given direction and circle motions around given point. To reach this goal, we tried to use four different types of base controllers given by *gazebo_ros_control* plugin.

We tried to separately control each wheel and create custom program for synchronization. Firstly, joint effort controller was used. After excessing limit of speed equal to 5 m/s effort was too high and robot started to stay on back wheels and then fall to its had. Then, we tried to use other types of PID controllers. We could not normally control the robot model because of wrong values of PID parameters and low transmission reduction parameter.

Then, we tried to use differential drive controller. It synchronizes track wheels and flipper wheels and set their speed using values from *cmd_vel* ROS topic.

To use differential drive controller, we created configuration file, where we set parameters of wheels such as diameter of wheel, distance between left and right side. To use this controller, all synchronized wheels must be same size. For flipper motion, we used joint position controller because we need to put flippers in needed position instead of moving with given speed what we need when we control wheels. All controllers' work scheme is presented on Fig. 33.4.

To calculate the motion of robot using track wheels controller, the following formulas (33.2)–(33.4) were used. For translation:

$$a = \frac{l}{r}, \tag{33.2}$$

where l is the distance, which we need to move, a is an angle that the wheel turns, and r is wheel radius. For rotation around robot center:

$$a = \varphi * \frac{d}{2r}, \tag{33.3}$$

where d is distance between each left–right pair of wheels. For rotation around side of the robot:

$$\varphi = a * \frac{r}{d}. \tag{33.4}$$

To calculate motion around center of rotation, formulas (33.5) and (33.6) were used:

$$R = \frac{\frac{l}{2(v_r + v_l)}}{v_r - v_l}; \tag{33.5}$$

$$\omega = \frac{(v_r - v_l)}{l}, \tag{33.6}$$

where R is the distance between center of rotation and midpoint between wheels. l is distance between wheels and ω is angular speed of robot around center of rotation.

Controller was used to create teleoperation package, which gives us ability to move robot in teleoperation mode. Using this package, we can set linear and angular speed of the robot and move flippers.

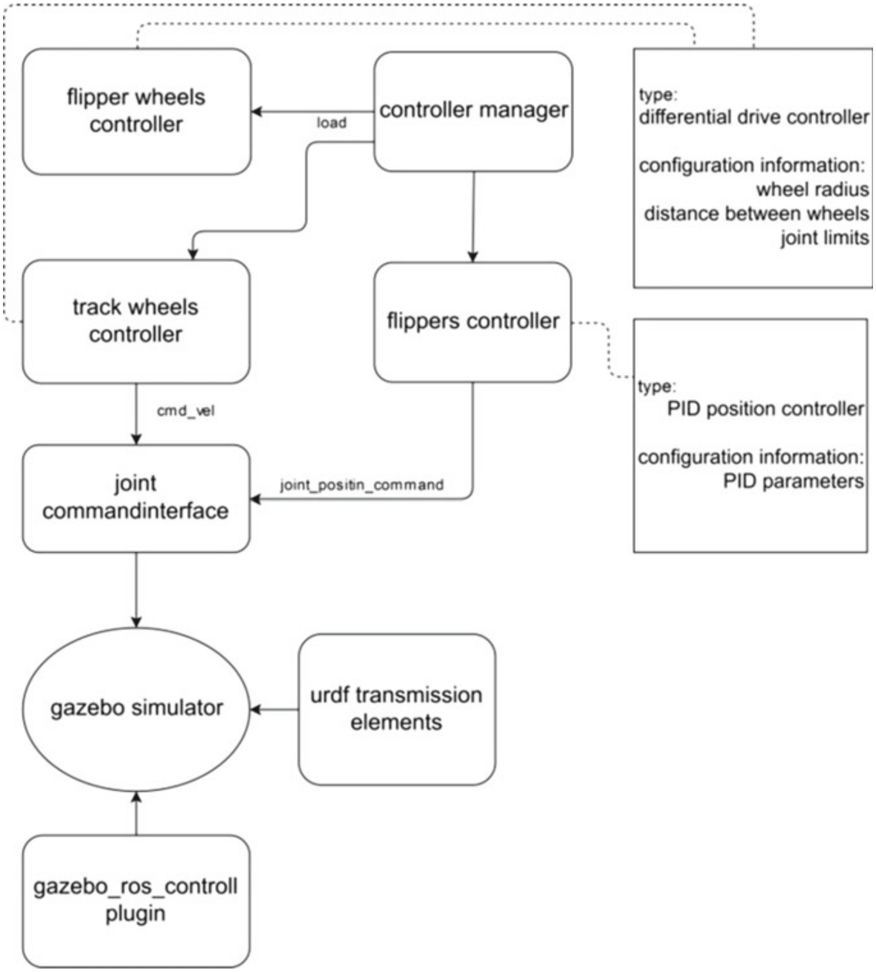

Fig. 33.4 Controller work scheme

33.4 Experiments

Random step environment (RSE) is built from wooden blocks, and in our laboratory, we construct it from blocks of 100 mm length and width with a varying height of 100, 200, 300, or 400 mm. Current small pilot patch full environment has 2 m width and 4 m length. This model was integrated into Gazebo simulation system. First, we put it into flat terrain to test if the robot model can move at all using controller and teleoperation package. You can see robot in Gazebo on flat terrain on Fig. 33.5. Then, we created simple model of RSE and tried to move robot over it.

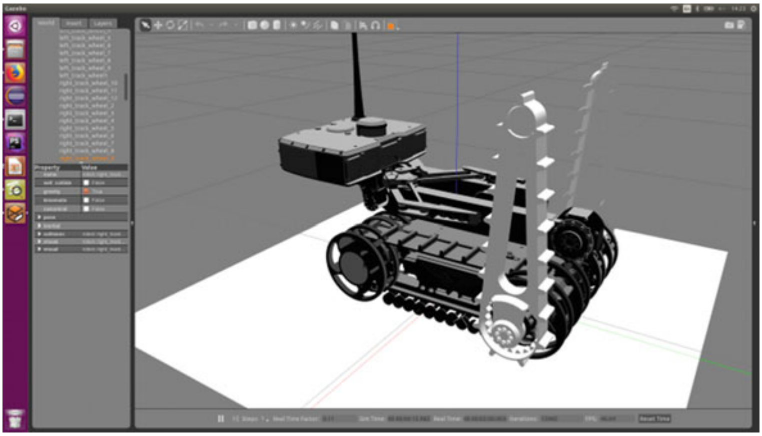

Fig. 33.5 Servosila Engineer in Gazebo simulation

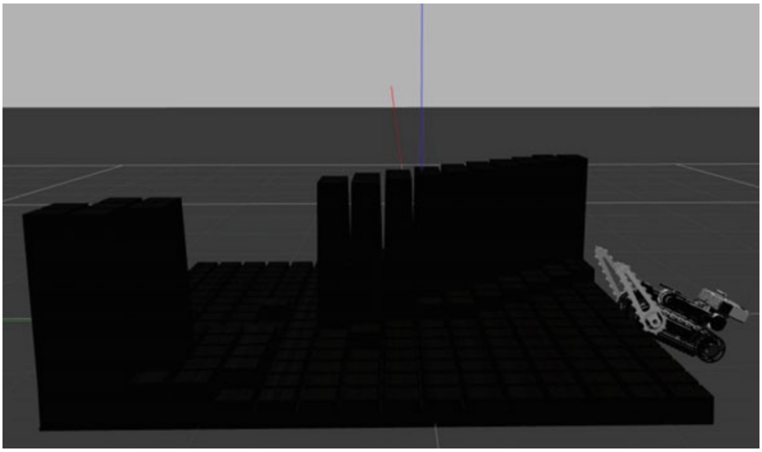

Fig. 33.6 Servosila Engineer climbs RSE in Gazebo simulation

Experiment on RSE showed that this model can overcome obstacles with or without using flippers depending on the size of obstacles as you can see on Figs. 33.6 and 33.7.

33.5 Conclusions and Future Work

To be able to simulate the behavior of the robot "Engineer" and hold experiments with this robot model of "Engineer" was created. This model built of parts given by

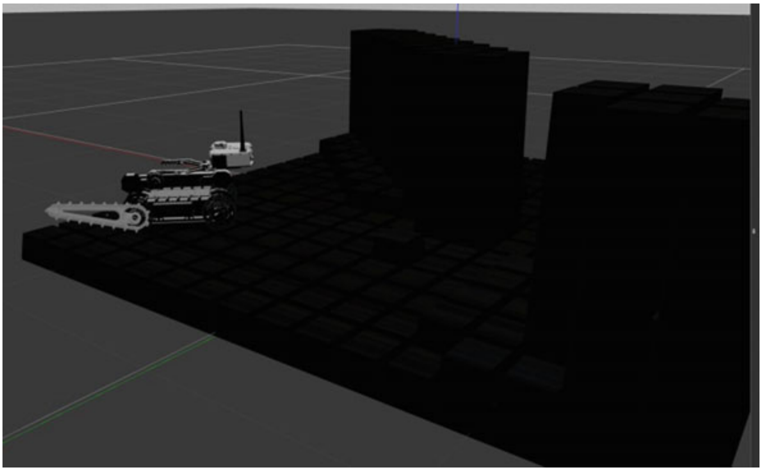

Fig. 33.7 Servosila Engineer on RSE in Gazebo simulation in the end of experiment

creators of the robot, so it is close to real robot. To simulate motions using track, we replaced it with big count of wheels.

We plan to modify the modeled robot so that it can be controlled from the remote control program [25] we developed, just like a real robot.

To analyze the ability to replace tracks with small wheels keeping high level of simulation plausibility controller for model of robot "Engineer" was created.

Simulation of tracks using a big count of small wheels shows that this kind of model is close to real track and gives robot ability to overcome rough terrain obstacles. Compared to the previous attempt to create model of "Engineer," this one has movable flippers and shows good results in simulation. In future, we will use this model to solve the task of path planning on RSE and check our algorithms on real robot. In addition, we will try to use bigger count of smaller wheels instead of tracks and make head of the robot movable to be able to change center of mass (CoM) of the model. It can help to keep balance in conditions that bring fall of the robot for now.

Acknowledgements This work was supported by the Russian Foundation for Basic Research (RFBR), project ID 19-58-70002.

References

1. Kohrt, C., et al.: An online robot trajectory planning and programming support system for industrial use. Robot. Comput. Integr. Manufact. **29**(1), 71–79 (2013)
2. Tsarouchi, P., Michalos, G., Makris, S., Chryssolouris, G.: Vision system for robotic handling of randomly placed objects. Procedia CIRP **9**, 61–66 (2013)
3. Sagitov, A., Tsoy, T., Li, H., Magid, E.: Automated open wound suturing: detection and planning algorithm. J. Robot. Network. Artif. Life **5**(2), 144–148 (2018)

4. Fujita, M., Kitano, H.: Development of an autonomous quadruped robot for robot entertainment. Auton. Robots **5**(1), 7–18 (1998)
5. Chirikjian, G.S., Zhou, Y., Suthakorn, J.: Self-replicating robots for lunar development. IEEE/ASME Trans. Mechatron. **7**(4), 462–472 (2002)
6. Motienko, A.I., Ronzhin, A.L., Pavljuk, N.A.: The modern development of rescue robots, opportunities and principles of their application. Sci. Bull. NSTU **60**, 147–165 (2015)
7. Ohno, K., et al.: Semi-autonomous control system of rescue crawler robot having flippers for getting over unknown-steps. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 3012–3018 (2007)
8. Wong, C., Yang, E., Yan, X. T., Gu, D.: An overview of robotics and autonomous systems for harsh environments. In: 2017 23rd International Conference on Automation and Computing (ICAC), pp. 1–6 (2017)
9. Nagatani, K., et al.: Multirobot exploration for search and rescue missions: a report on map building in RoboCupRescue 2009. J. Field Robot. **28**(3), 373–387 (2011)
10. Harbers, M., et al: Exploring the ethical landscape of robot-assisted search and rescue. In: A World with Robots, pp. 93–107 (2017)
11. Magid, E., Lavrenov, R., Afanasyev, I.: Voronoi-based trajectory optimization for UGV path planning. In: 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), IEEE, pp. 383–387 (2016)
12. Magid, E., Tsubouchi, T.: Static balance for rescue robot navigation-translation motion discretization issue within random step environment. In: 7th Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, pp. 415–422 (2010)
13. Rivlin, G.F.E., Shimshoni, I., Soldea, O.: Efficient search and verification for function based classification from real range images. Comput. Vis. Image Underst. **105**(3), 200–217 (2007)
14. Ronzhin, A.L., Vatamanyuk, I.V., Ronzhin, A.L., Železnŷ, M.: Mathematical methods to estimate image blur and recognize faces in the system of automatic conference participant registration. Autom. Remote Control **76**(11), 2011–2020 (2015)
15. Magid, E., Tsubouchi, T., Koyanagi, E., Yoshida, T.: Static balance for rescue robot navigation: losing balance on purpose within random step environment. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 349–356 (2010)
16. Jacoff, A., Downs, A., Virts, A., Messina, E.: Stepfield pallets: repeatable terrain for evaluating robot mobility. In Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, pp. 29–34 (2008)
17. Okada, Y.: Shared autonomy system for tracked vehicles to traverse rough terrain based on continuous three-dimensional terrain scanning. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 357–362 (2010)
18. Sokolov, M., et al. Modelling a crawler-type UGV for urban search and rescue in Gazebo environment. In: International Conference on Artificial Life and Robotics, pp. 360–363 (2016)
19. Akai, N.: Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 1–6 (2017)
20. Xie, S.: The research on obstacle-surmounting capability of six-track robot with four swing arms. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2441–2445 (2013)
21. Sheng, W., Chen, H., Xi, N.: Navigating a miniature crawler robot for engineered structure inspection. IEEE Trans. Autom. Sci. Eng. **5**(2), 368–373 (2008)
22. Calisi, D., Nardi, D., Ohno, K., Tadokoro, S.: A semi-autonomous tracked robot system for rescue missions. In: 2008 SICE Annual Conference, pp. 2066–2069 (2008)
23. Pecka, M., Zimmermann, K., Svoboda, T.: Fast simulation of vehicles with non-deformable tracks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6414–6419 (2017)

24. Dissanayake, M., Sattar, T.P., Howlader, O., Pinson, I., Gan, T.H.: Tracked-wheel crawler robot for vertically aligned mooring chain climbing design, simulation and validation of a climbing robot for mooring chains. In: 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), pp. 1–6 (2017)
25. Mavrin, I., Lavrenov, R., Svinin, M., Sorokin, S., Magid, E. Remote control library and GUI development for Russian crawler robot Servosila Engineer. In: MATEC Web of Conferences, EDP Sciences, vol. 161, p. 03016 (2018)