Navigation stack for the crawler robot Servosila Engineer

Alexandra Dobrokvashina
Laboratory of Intelligent Robotic Systems (LIRS)
Intelligent Robotics Department
Institute of Information Technology and Intelligent Systems
Kazan Federal University
Kazan, Russia
AlSDobrokvashina@stud.kpfu.ru

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

Tatyana Tsoy
Laboratory of Intelligent Robotic Systems (LIRS)
Intelligent Robotics Department
Institute of Information Technology and Intelligent Systems
Kazan Federal University
Kazan, Russia
tt@it.kfu.ru

Edgar A. Martinez-Garcia
Laboratorio de Robotica
Institute Engineering and Technology
Universidad Autonoma de Ciudad Juarez
Juarez, Mexico
edmartin@uacj.mx

Yang Bai
Information Science and Engineering Department
Ritsumeikan University
Kyoto, Japan
yangbai@fc.ritsumei.ac.jp

***Abstract*—Navigation is one of the most critical problems in all areas of mobile robotics, including urban search and rescue robotics. Autonomous navigation tasks appear almost in every robotic project. A preliminary validation on a new navigation algorithm could be efficiently performed within a simulator, which allows constructing any type of an environment and simulate different operational conditions for a robot. Therefore a development of an appropriate simulated robot model and its navigation capabilities within a simulator are recently in a focus of many research teams. This article presents a number of improvements to the Gazebo simulator model of the crawler robot Servosila Engineer and a newly developed navigation stack. The navigation stack was validated with the robot's model in the Gazebo and with a real robot in a laboratory environment.**

***Index Terms*—crawler robot, Servosila Engineer, USAR, urban search and rescue, 3D model, navigation, Gazebo, ROS**

I. Introduction

The simulated environment became very popular for tests algorithms and makes experiments with robots. Simulators such as Gazebo [1], [2] or Webots [3] allow recreating different surfaces and conditions including things that are strongly hard to recreate in real life [4], [5]. For example simulations could be useful for testing algorithms or explore possible behaviour of the robot in extreme situations and in urban search and rescue (USAR) projects [6]–[8]. Simulation experiments also could save a robot from the destruction of itself or something or someone else by a simple unseen mistake in code [9].

This paper is mostly about the robot Servosila Engineer and its simulation model (Fig. 1). This robot was created by the Russian company Servosila [10]. Its design allows the robot to work in environments that could be dangerous for human. Insulated and waterproof shell could be very useful in bad weather conditions. Servosila Engineer has several on-board sensors. These are four cameras, an IMU sensor and a laser scanner that is placed on the head of the robot. Cameras of the robot include one stereo pair, zoom camera and one rear camera [11], [12]. Also, a robot equipped with a bright lantern. It could be useful for the operator to have more information about the surroundings of the robot without external light in the environment.

Servosila Engineer became a good education platform. Manipulator with end-effector, cameras, laser sensor and tracks with pair of active flippers open a big amount of possible projects for it [13]. The problem is that Servosila Engineer does not have its own simulation model. It makes impossible to present some experiments. Especially experiments that possible could harm the robot, for example, autonomous going through uneven terrain.

The creation of a physically reliable simulation model for Servosila Engineer is one of the most important goals for now. In [14] presented the last results of creating this model. They are fixed visual and collision meshes, simulated crawlers and a working manipulator with a moving end-effector (Fig. 2).

2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA) | 978-1-6654-2248-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIEA51954.2021.9516393

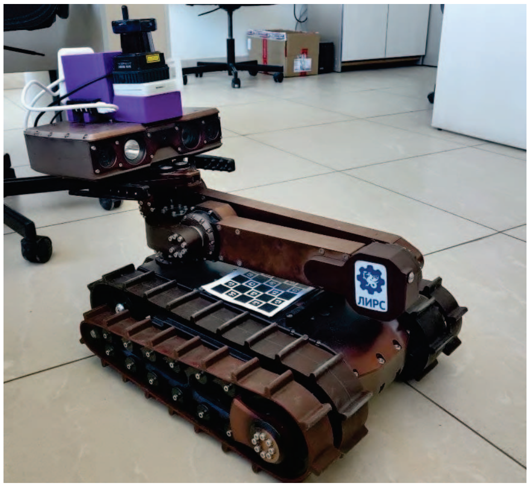

Fig. 1. Crawler robot Servosila Engineer.

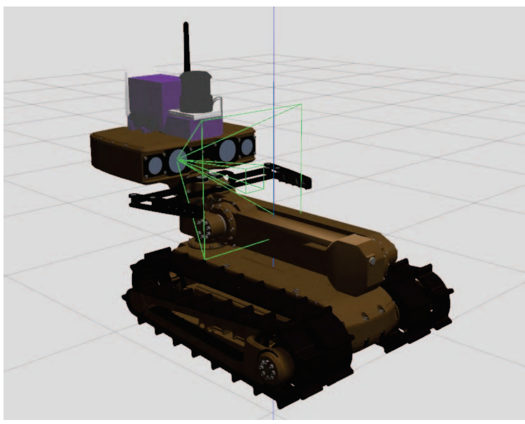

Fig. 2. Simulation model of crawler robot Servosila Engineer.

This article will present the next steps of the work on the simulation model of the Russian crawler robot Servosila Engineer in ROS/Gazebo. The main point here is the navigation stack. It was added to the created simulation model and tuned with customized parameters. There were also found several problems with the model during experiments. So they were fixed for better working.

II. Navigation stack

It was decided to use the same navigation stack for the simulation model as on the real Servosila Engineer. It uses a hector mapping algorithm for simultaneous localization and mapping (SLAM) purposes. Navigation also includes the DWA algorithm as the local planner and base_ROS_planner as global [15]. Earlier same stack was used in real-life experiments with this robot and it shows good result. It was also additionally tuned for simulation model because of the more complicated structure of the project.

As already been mentioned simulation model was created in ROS/Gazebo environment [16]. Navigation on the real robot was also created with the help of the ROS. The reason is that ROS proposed a good solution for the realisation of every part of the navigation stack [17]. Huge amount of already existing SLAM and path-planning algorithms and instruments for building any kind of navigation stack with them [18], [19]. Flexible parameters could fit most of the robots [20], [21]. Move_base is the one of the those instruments. It used for building of navigation stack in the ROS environment. Move_base is a ROS-node that connect local and global planners, topics with maps and controllers. It gives the robot opportunity to move along the sent goal. Right setting and tuning of parameters will make the way to the goal maximally clean and optimized.

Another part of the navigation stack is mapping. In this solution, we use hector mapping for cartography. It is a SLAM (simultaneous localization and mapping) approach. It successfully used with modern LIDAR systems and provides an estimated scan rate 2D pose [22]. It has good correlation with most of the behaviours needed in real-world scenarios.

Tuning of the navigation stack was done simultaneously on the simulation model and the real robot. That allows to check the behaviour of the model and compare it to the behaviour of the real robot. As the result of this experiment, it was seen that there still some issues in the model. But still the navigation stack was successfully installed.

Tuning of parameters was done using [23]. It gives very useful instructions. It also illustrated correlations between most of the parameters and trajectory that robot builds. That makes tuning much easier than doing it with just the names of the parameters known. In tables I- II could be found a piece of parameters that were changed during setting up navigation stack.

TABLE I
Final parameters of the costmaps

Costmap parameter name	**Value**
Footprint	**[[-0.36, -0.20], [-0.36, 0.20], [0.26, 0.20], [0.26, -0.20]]**
Inflation radius	**0.6**
Cost Scaling factor	**1.2**
Obstacle range	**3.5**
Inflation radius	**0.6**

In Figures 3 and 4 could be found illustration from experiments with tuning navigation stack [24]. First one shows the robot itself in a room which should be discovered. The second picture shows created map. It presented with RViz visualization tool [25].

Parameters for navigation stack mostly common for real robot and for the simulation model [26]. That was the main reason for choosing the same navigation stack for both of them. It was mostly about frames to provide proper work of the

TABLE II
FINAL PARAMETERS OF THE LOCAL PLANNER

Planner parameter name	Value
Holonomic Robot	false
Yaw goal tolerance	0.3
XY goal tolerance	0.3
Sim time	3.0
DWA	true
Latch XY goal tolerance	false
Recovery behavior enabled	false
Clearing rotation allowed	false

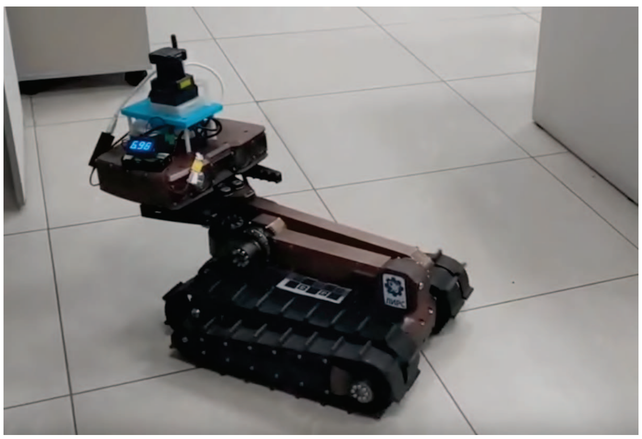

Fig. 3. Experiments with tuning navigation stack on real Servosila Engineer (robot in room).

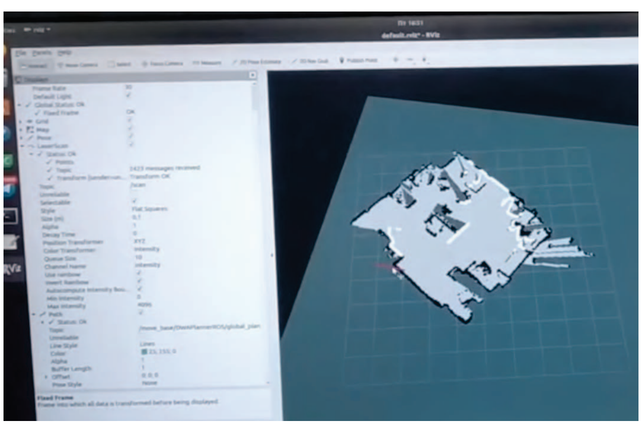

Fig. 4. Experiments with tuning navigation stack on real Servosila Engineer (map of the room in RViz).

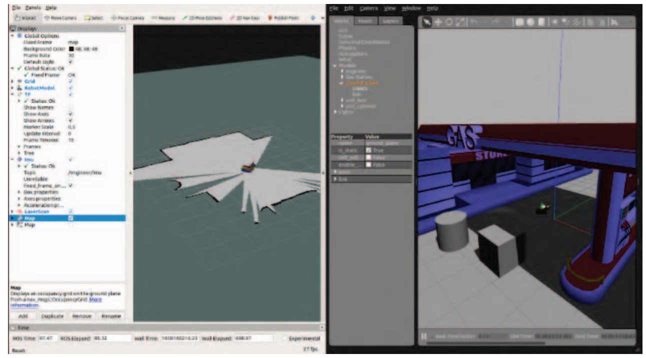

Fig. 5. Working navigation stack building map in simulation experiments.

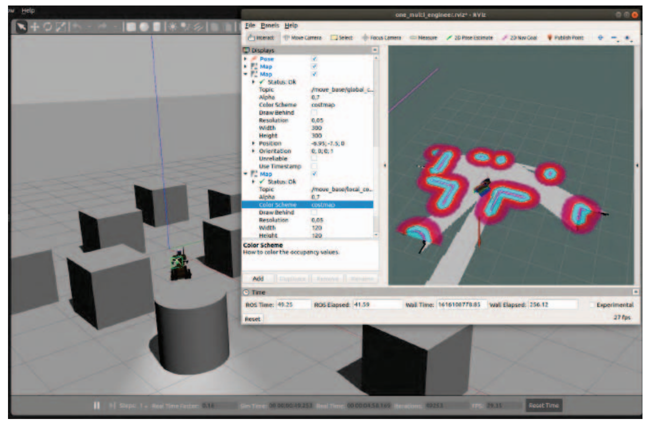

Fig. 6. Working navigation stack planing path in simulation experiments.

navigation stack. Integration with ROS in a real robot means having transforms only for the center of the base and laser scanner. At the same moment simulation model has much more frames starting from the footprint of a base to frame for each of the invisible wheels, that are used for simulating crawlers.

Several problems were found through setting up navigation stack on simulation model. More information about them will be presented in the next section. Finally, we came to the fine working navigation stack. An example of its work presented in Fig. 5 and Fig. 6. There could be found successfully created maps for different environments, and in Fig. 6 could be found a planned path. This path builds automatically by navigation stack. The goal is setting by operator.

III. MODEL IMPROVEMENTS

We found some mistakes in the model during the navigation experiments with the simulation model. The inertial parameters was the most critical of them. Gazebo simulator requires to give certain information for every link of the robot. One of these parameters is inertia. In the Gazebo simulator it is needed to toggle Inertial point in the menu of View to see inertial links of the robot. The main links that were done earlier have right inertia, but there were some problems with the end-effector, base and simulation wheels. And if the inertia of end-effector and simulation wheels do not affect navigation, the inertia of the base does. The inertia of the base of the robot was rotated along y-axes for about 90 degrees for the some reasons. That makes the robot stuck on the floor while moving. And that became a reason for wheels to seems to slip on the terrain. ”Slippering” wheels brokes odometry and that possibly could ruin all navigation and cause such problems as illustrated in Fig. 7. Here could be found the result of the wrong inertia declaration and mistakes in static mapping for robot frames.

It took some time to realize that problem was not in wheels but inertia. But realizing that problem gets us to the solution. The inertia of the robot was maximally fixed and now it looks as in Fig. 8. For example, here is fixed inertial block with inertia matrix of the base:

```
<xacro:property name="mass" value="20"/>
<xacro:property name="length" value="0.45"/>
<xacro:property name="width" value="0.15"/>
<xacro:property name="height" value="0.1"/>
<inertial>
<mass value="${mass}"/>
<origin xyz="0 0 -0.015" rpy="1.57 0 1.57"/>
<inertia
    ixx="${mass/12*(height**2 + length**2)}"
        ixy="0.0" ixz="0.0"
    iyy="${mass/12*(width**2 + length**2)}"
        iyz="0.0"
    izz="${mass/12*(width**2 + height**2)}"/>
</inertial>
```

There are still some problems with simulation wheels. They appear definitely because of their size and count of such wheels. Simulation wheel need to be very small for the simulation which means small inertia. The problem is that small inertia could cause robots destruction in the simulation environment. It happens because of problems with Gazebo physics. And that happens whenever wheels are added with ”right” inertia. We are still working on that problem.

The problem of the ”slippering” wheels was successfully resolved at least for now. So navigation stack works fine. There also was another improvement. It came from a change of stand for the laser scanner on the real robot. So for having the actual model of the robot in simulation it was proposed to change the model of the stand in simulation either. In Fig. 9 could be found a new model of the stand for laser scanner.

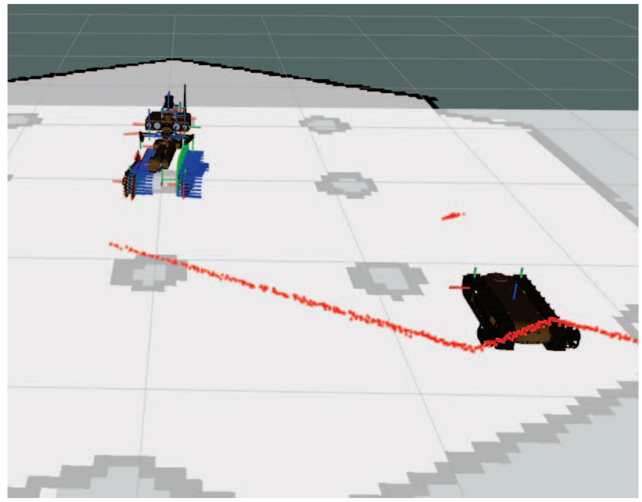

Fig. 7. Problem with visualisation of robot in RViz.

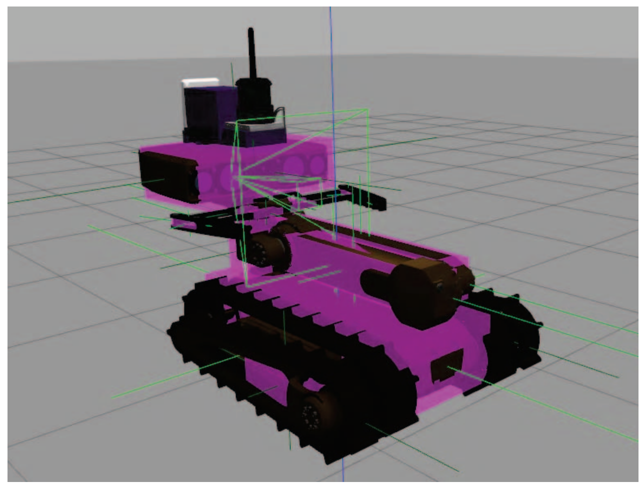

Fig. 8. Inertia of the model simulation model (inertial modules marked purple).

IV. Future work

Now, when the navigation stack is settled and works properly our next goal is to move the robot through uneven terrains. For such experiments in real-life, we have a random step environment (RSE) field which successfully used for that [27]. In Fig. 10 could be found example of such environment. As experiments with the model should be maximally right and reproducible in real life so it was decided to create a simulation model for RSE. The proposed model should be easily rebuilt as a real one by adjusting the parameters of the model. One of the possible configuration of the created model illustrated in Fig. 11. The created model builds RSE using given parameters of height for each block. There are one hundred blocks for an RSE size of 10x10.

It proposed to have series of simulation and real-life experiments using both simulated and real RSE in the nearest future.

V. Conclusion

In this paper, we presented the next step of working on the Gazebo simulation model of the Russian crawler robot Servosila Engineer. Number of improvements were made, such as actualizing the model with the real robot and fixing inertial block. New navigation stack was added to the robot, as for real one so for its simulation model. There also were several experiments done in laboratory environment and in simulation, which helped to validate and tune navigation algorithms.

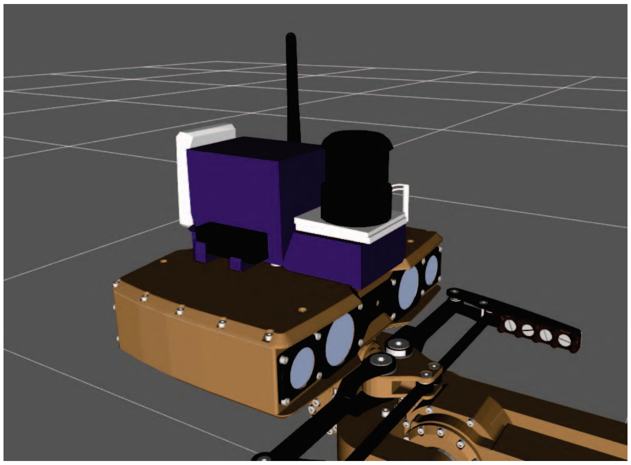

Fig. 9. New model of the stand for laser scanner.

Fig. 10. Example of random step environment (RSE).

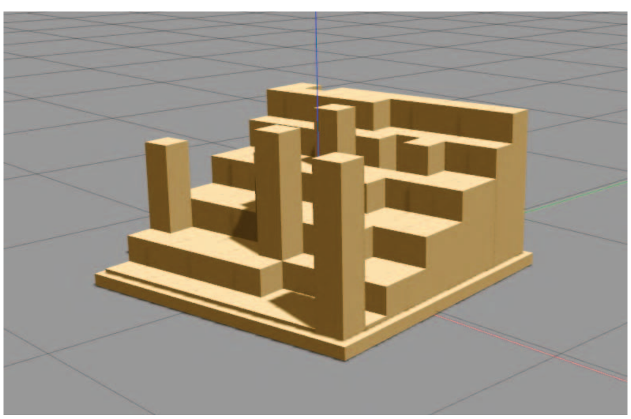

Fig. 11. Simulation model of random step environment (RSE).

ACKNOWLEDGMENT

This work was supported by the Russian Foundation for Basic Research (RFBR), project ID 19-58-70002. The fifth author acknowledges the support of the Japan Science and Technology Agency, the JST Strategic International Collaborative Research Program, Project No. 18065977.

REFERENCES

[1] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2149–2154.

[2] C. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. Rivero, and G. Pratt, "Inside the virtual robotics challenge: Simulating real-time robotic disaster response," *Automation Science and Engineering*, vol. 12, pp. 494–506, 2015.

[3] O. Michel, "Webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, p. 5, 2004.

[4] B. Abbyasov, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Svinin, and E. Magid, "Automatic tool for gazebo world construction: from a grayscale image to a 3d solid model," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7226–7232.

[5] L. Wang, K. Tan, and V. Prahlad, "Developing khepera robot applications in a webots environment," in *Proceedings of 2000 International Symposium on Micromechatronics and Human Science*, 2000, pp. 71–76.

[6] E. Magid, A. Pashkin, N. Simakov, B. Abbyasov, J. Suthakorn, M. Svinin, and F. Matsuno, "Artificial intelligence based framework for robotic search and rescue operations conducted jointly by international teams," in *14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*. Springer, 2020, pp. 15–26.

[7] J. Wang, M. Lewis, and J. Gennari, "Usar: A game based simulation for teleoperation," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 47, no. 3. SAGE Publications Sage CA: Los Angeles, CA, 2003, pp. 493–497.

[8] R. R. Murphy, S. Tadokoro, and A. Kleiner, "Disaster robotics," in *Springer Handbook of Robotics*. Springer, 2016, pp. 1577–1604.

[9] J. D. Bric, D. C. Lumbard, M. J. Frelich, and J. C. Gould, "Current state of virtual reality simulation in robotic surgery training: a review," *Surgical endoscopy*, vol. 30, no. 6, pp. 2169–2178, 2016.

[10] "Servosila official site," https://www.servosila.com/en/index.shtml, [Online; accessed 30-March-2020].

[11] S. Ramil, R. Lavrenov, T. Tsoy, M. Svinin, and E. Magid, "Real-time video server implementation for a mobile robot," in *2018 11th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2018, pp. 180–185.

[12] T. Tsoy, A. Zakiev, K. Shabalina, R. Safin, E. Magid, and S. K. Saha, "Validation of fiducial marker systems performance with rescue robot servosila engineer onboard camera in laboratory environment," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2019, pp. 495–499.

[13] A. Dobrokvashina, R. Safin, R. Lavrenov, and Y. Bai, "Improved graphical user interface for crawler robot servosila engineer," in *2021 International Siberian Conference on Control and Communications (SIBCON)*. IEEE, 2021, in press.

[14] A. Dobrokvashina, R. Lavrenov, E. Martinez-Garcia, and Y. Bai, "Improving model of crawler robot servosila "engineer" for simulation in ros/gazebo," in *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*. IEEE, 2020, in press.

[15] R. Lavrenov and E. Magid, "Towards heterogeneous robot team path planning: acquisition of multiple routes with a modified spline-based algorithm," in *MATEC Web of Conferences*, vol. 113, 2017, p. 02015.

[16] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd, 2018.

[17] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, "Ros based autonomous mobile robot navigation using 2d lidar and rgb-d camera," in *First international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP)*. IEEE, 2019, pp. 151–154.

[18] S. Zaman, W. Slany, and G. Steinbauer, "Ros-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues," in *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*. IEEE, 2011, pp. 1–5.

[19] Y. Cheng and G. Y. Wang, "Mobile robot navigation based on lidar," in *Chinese Control And Decision Conference*, 2018, pp. 1243–1246.
[20] R. Reid, A. Cann, C. Meiklejohn, L. Poli, and A. Boeing, "Cooperative multi-robot navigation, exploration, mapping and object detection with ros," in *Intelligent Vehicles Symposium*, 2013, pp. 1083–1088.
[21] S. Allali and M. Benchaiba, "Safe route guidance of rescue robots and agents based on hazard areas dissemination," in *International Conference on Mathematics and Information Technology (ICMIT)*, 2017, pp. 29–37.
[22] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *International Symposium on Safety, Security and Rescue Robotics (SSRR)*, 2011.
[23] K. Zheng, "Ros navigation tuning guide," *arXiv preprint arXiv:1706.09068*, 2017.
[24] "Experiments with the navigation stack of the crawler robot servosila engineer using data from lidar," https://youtu.be/PwwKyJuyxGo, [Online; accessed 31-March-2020].
[25] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "Rviz: a toolkit for real domain data visualization," *Telecommunication Systems*, vol. 60, no. 2, pp. 337–345, 2015.
[26] K. Zakharov, A. Saveliev, and O. Sivchenko, "Energy-efficient path planning algorithm on three-dimensional large-scale terrain maps for mobile robots," in *International Conference on Interactive Collaborative Robotics*. Springer, 2020, pp. 319–330.
[27] E. Magid and T. Tsubouchi, "Static balance for rescue robot navigation: Discretizing rotational motion within random step environment," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2010, pp. 423–435.