Sensors modelling for Servosila Engineer crawler robot in Webots simulator

Alexandra Dobrokvashina
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
dobrokvashina@it.kfu.ru

Roman Lavrenov
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
lavrenov@it.kfu.ru

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

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

Evgeni Magid
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
HSE University
Moscow, Russia
magid@it.kfu.ru

2022 Moscow Workshop on Electronic and Networking Technologies (MWENT) | 978-1-6654-9666-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/MWENT55238.2022.9802400

***Abstract*—Simulations became a convenient, easy and safe way to interact with a robot in a virtual world in order to reproduce complex tasks and environments, preliminary evaluate new concepts and algorithms instead of directly building complicated testing field constructions in real world. Yet, a simulation requires an adequate model of a robot that corresponds well to its hardware original. This paper presents an improved Webots simulation model of the Russian crawler-type robot Servosila Engineer. We extended a previously constructed model with sensors of the real robot and added connection to the Robot Operating System (ROS), while ROS/Webots connection was also improved for a greater convenience.**

***Index Terms*—ROS, Webots, Servosila Engineer, simulation, sensors**

I. Introduction

Nowadays robotics industry became one of the most rapidly developing areas. Robots are commonly used various tasks of our daily routine, e.g., they are employed as house cleaners and increase human security [1], used in infrastructure inspection [2], autonomous manufacturing [3], entertainment and advertisement [4], daily services [5] etc. Yet, one of the most significant purpose of robotics is a human search and rescue. Urban search and rescue (USAR) is one of the most useful and perspective way of using robots [6]. It associates with such elements as human-robot interaction [7], human assistance [8] and scouting in extraordinary events or dangerous environment [9]. Working on USAR technologies improvements requires ability to test newly created concepts and algorithms in real environments [10], [11]. This could be quite tricky in a case of real disasters [12], e.g., such as a radioactive pollution accidents [13]. That is one of the reasons that allowed simulation and virtual experiments to gradually turn into an important part of modern robotics. Simulated environment is useful for checking risky algorithms that might require expensive hardware. It also provides scientists with an ability to recreate dangerous and dynamic environmental conditions for a robot, which are hard or even impossible to reproduce in a real life [14].

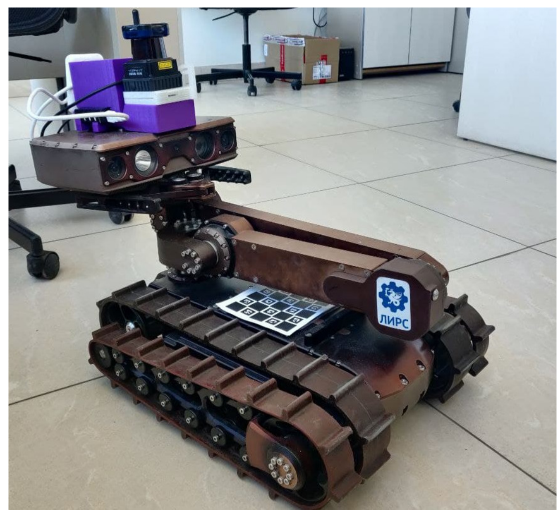

Fig. 1. Servosila Engineer crawler-type mobile robot

In this work we presented an integration of sensors into Webots simulation [15] model of the Russian crawler-type robot Servosila Engineer (Fig. 1). This is an extension of our previous work of creating a simulation model for this robot [16]. Additionally, we added and improved ROS integration for the entire original model. Our solution provides a standard control for the robot via ROS-topics. Same configurations were used previously for our Gazebo simulation model and for the real robot [17]. Such modifications allow to ease a transfer of all previously implemented algorithms to the new model using Robot Operating System (ROS).

II. Previous work

Simulation is a widely used instrument among robotics developers. There are several popular simulation products, e.g., Matlab [18], Gazebo, Webots [19], VRep [20], and Morse [21], and dozens less popular ones. Each of these simulators is effective for particular purposes and due to a broad variety they have been gradually replacing old-fashion in-lab self-made simulators [22].

The Gazebo simulator is one of the most popular open source solutions associated with ROS [23]. One of the reason of its popularity is a powerful technical support from the developers and a large global ROS community. ROS/Gazebo is a standard set that is recommended for ROS users and supplied out of the box. Gazebo simulator uses OGRE 3D engine [24] and works with several other physics engines, including ODE [25], Bullet [26] and Dart [27]. Yet, despite of its popularity Gazebo still has a number of disadvantages. For example, it suffers from instability in minor cases or absence of important constructive modules and parts, e.g., a detailed crawlers' simulation.

Another popular solution for the robot simulation is Webots [19]. It is mostly autonomous simulator, which allows ROS-integration as an additional feature. Similarly to the Gazebo, it uses OGRE graphical engine, but it works only with ODE physics engine, which is yet an effective solution. As another important benefit, Webots has a large variety of tools and built-in models that simulate different types of environment. Unfortunately, because of having the ROS-integration only as an add-on, it does not have a well-documented connection to ROS.

VRep, recently known as CoppeliaSim, is another popular option among simulators [28], [29]. It is a cross-platform simulator, which could use a 3D engine as an internal one as well as an external one. For the physics engine it allows such options as ODE, Bullet, Vortex and Newton. Similarly to Webots, VRep is positioned as an autonomous simulator. It also declares a ROS-integration with an effective connection [29].

MORSE simulator stands for Modular Open Robots Simulation Engine [30]. It is based on an open-source code of Blender. MORSE is a cross-platform simulator, which also supports ROS-integration.

Previously a number of research works presenting simulation models for the crawler robot Servosila Engineer were presented by our laboratory. After a number of preliminary models, the first successful version of the simulation model was constructed in the Gazebo simulator (Fig. 2) [17]. Yet, it faced a number of problems, including an absence of a proper simulation of robot crawlers (tracks) in the Gazebo simulator. To solve this problem we proposed to approximate a crawler physics with an array of small pseudo-wheels, which is one of popular solutions for a crawler simulation [31]. A resulting two hundred pseudo-wheels robot model demonstrated a significantly reduced real-time factor (RTF), which does not allow a comfortable online usage that assumes at least $RTF \geq 0.3$ [32]. Furthermore wheels were shown as quiet questionable solution for good traversability and navigation [33].

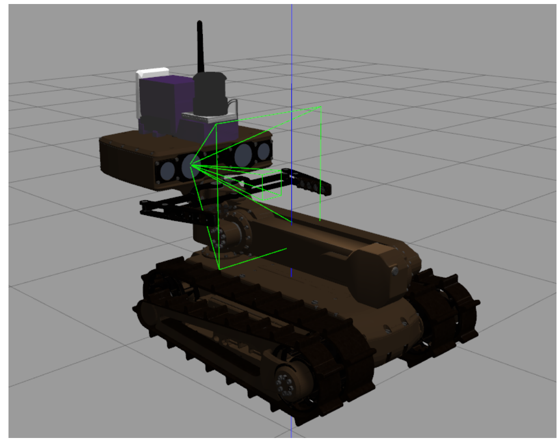

Fig. 2. Gazebo simulation model for Servosila Engineer

To extend the current Gazebo solution and to minimize the issues with the previous crawler module implementation, along with ongoing improvement of the Gazebo crawlers with pseudo gear wheels approach, it was decided to create a model of the Servosila Engineer robot within another simulator. Webots was used as a new platform for the simulation model [15]. This simulator became quite popular among users mostly because of being open source and having worth integration with widely used Robot Operating System (ROS) [34]–[36]. For our purposes it already contains an effective module of a crawler simulation. In one of the previous works presented a new simulation model of the Servosila Engineer robot in Webots(Fig. 4) [16], and in this work we integrate sensors and create a decent ROS integration for them and for the robot controllers.

III. Sensors modelling

A. Laser range finder

A laser range finder (LRF) is a widely used sensor in robotics. It provides information about obstacles in front and around of a robot. Data from an LRF is frequently used for laser-data based simultaneous localization and mapping (SLAM), navigation and human-robot interaction algorithms.

Integrating the LRF combined a standard process of adding a sensor and a visual model improvement. The LRF is placed on the robot head on a special stand with an adjustable plate (Fig. 1), which allows to set a selected angle of the laser ray with regard to a supporting surface (i.e., the ground surface) that allows the sensor to identify negative obstacles such as pits in front of the robot.

For the simulation we used the same custom CAD model of the LRF stand, which had been used to print this stand on a 3D printer. Visual blocks were added in order to emulate a Wi-Fi router, a battery element and a USB-hub, which were placed precisely according the real stand (Fig. 3).

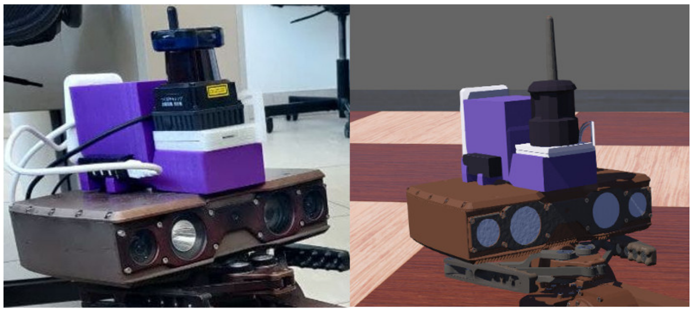

Fig. 3. The real stand (left) and the simulation model of the stand in Webots (right).

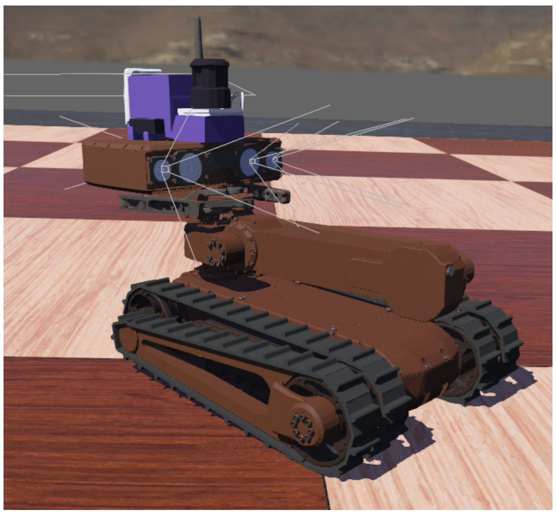

Fig. 4. The final simulation model of the Servosila Engineer robot in Webots.

The LRF settings corresponded to the real Hokuyo UTM-30LX-EW LRF of the robot. We set up its scanning range, angle resolution and scan time via a scanning frequency (Table I). And the final view of the model is demonstrated in Figure 4.

TABLE I
LASER RANGE FINDER PARAMETERS

Parameter	Value
Scanning range	0.1 - 30 m
	270 degrees
Angle resolution	0.25 degree
Scanning frequency	40 Hz

B. IMU sensor

An inertial Measurement Unit (IMU) provides information about an orientation of a robot in space, and sends data about rotation of the sensor with regard to the coordinate system's axes. It helps an operator (while in a teleoperation mode) or the robot (in an automatic operation) to capture underlying terrain unevenness that allows preventing turning upside down of the robot while traversing the terrain or negotiating with obstacles.

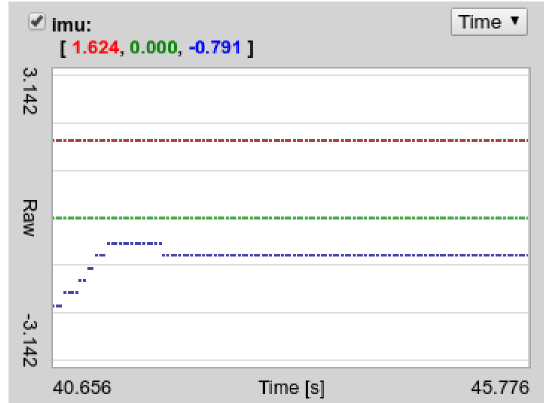

Fig. 5. Example of the data from the IMU sensor in Webots simulator

By contrast with the LRF integration, an IMU sensor does not require special changes in a visual model of the robot. The IMU sensor of the Servosila Engineer is a part of Zubax GNSS 2. It provides 3-axis compass data, and an example of such data is demonstrated in Figure 5.

C. Cameras

The Servosila Engineer robot is equipped with four onboard cameras: three in the front (a stereo pair and zoom camera) of the robot head and one monocular rear camera in the back of the head. This set of cameras provides the robot and/or its operator with information, which could be successfully used for teleoperation procedures [37] or be autonomous visual SLAM algorithms [38]. We used information about cameras resolution, their position and rates of frame per second to set them according to the real ones (Table II).

TABLE II
CAMERA PARAMETERS

Parameter	Stereo pair	Zoom camera	Rear camera
Resolution	640x480	1280x720	640x480
Rate	30 fps	50 fps	30 fps

Figure 6 presents an example of data received from the cameras.

IV. ROS/WEBOTS CONNECTION

Webots simulator is effective in connection with ROS. There is a controller that transfers most of data types from a simulator to the ROS topics and services. These are information from sensors and commands to the robot motors. Although most of the information became available via ROS, a number of additions were still required. One of the most significant parts was an adjustment for the motor controllers. Its became classic for joints in ROS to be controlled using ROS-topics, but in Webots simulator it is recommended to use ROS-services. For unification of control to a standard solution, we created a new ROS-node, which converts classic messages from the ROS-topics to the ROS-service commands. An example of

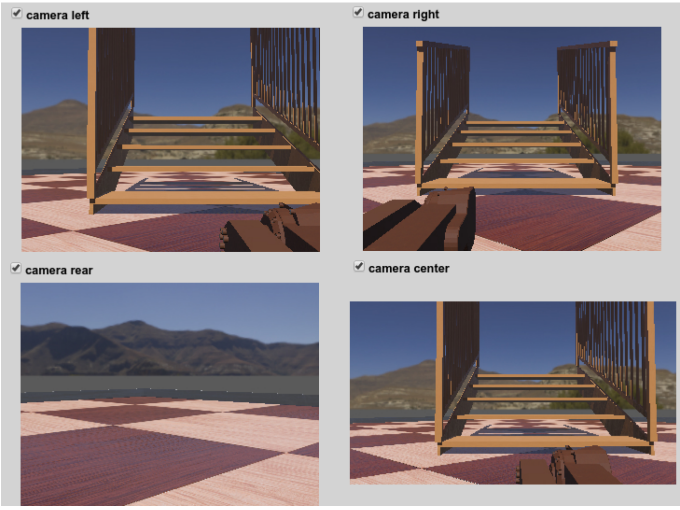

Fig. 6. Example of data from the cameras in Webots simulator

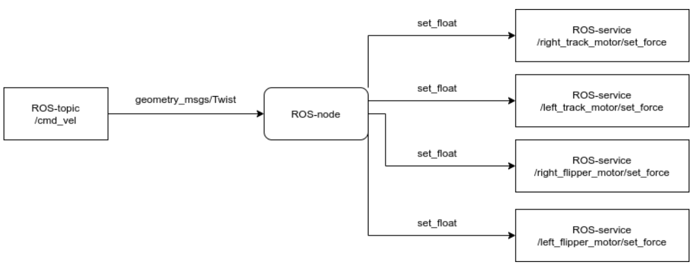

Fig. 7. Converting messages from the ROS-topic to the ROS-services (example with $/cmd\_vel$ topic)

such conversion is presented in Figure 7. There is a standard ROS-message with $geometry\_msgs/Twist$ type. Our node converts data about a speed of the robot from these messages into corresponding commands to each of the two tracks of the robot. In other words, a ROS-topic command is split into four services - two for the main tracks and two for the flippers, which are moving simultaneously with the main ones.

The constructed ROS-node allows to control the robot in the same way we controlled it in Gazebo simulator. Such unification makes easier the process of integration of the Webots simulation model with already created ROS-based algorithms, such as a navigation stack, an autonomous return algorithms and others.

V. Conclusions

This article presented an upgraded Webots simulation model for the Russian crawler-type robot Servosila Engineer. We added to a previous model all onboard sensors of the real robot, including a laser range finder, an inertial measurement unit (IMU) and four cameras. All sensors were accurately set up according to the real hardware of the robot. The ROS/Webots connection was improved by constructing a new ROS-node. The new node converts commands from standard topics to custom services created by a built-in ROS/Webots connector, which speeds up the work of Webots via ROS.

Acknowledgment

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program (”PRIORITY-2030”). The third and the forth authors acknowledge the support of the Japan Science and Technology Agency, the JST Strategic International Collaborative Research Program, Project No. 18065977.

References

[1] R. R. Murphy, “Rescue robotics for homeland security,” *Communications of the ACM*, vol. 47, no. 3, pp. 66–68, 2004.

[2] S. Nansai and R. E. Mohan, “A survey of wall climbing robots: recent advances and challenges,” *Robotics*, vol. 5, no. 3, p. 14, 2016.

[3] V. Voronin, M. Zhdanova, E. Semenishchev, A. Zelenskii, Y. Cen, and S. Agaian, “Action recognition for the robotics and manufacturing automation using 3-d binary micro-block difference,” *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 7, pp. 2319–2330, 2021.

[4] J. K. Sheba, R. E. Mohan, and E. A. M. García, “Easiness of acceptance metric for effective human robot interactions in therapeutic pet robots,” in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2012, pp. 150–155.

[5] D. Ryumin, I. Kagirov, A. Axyonov, N. Pavlyuk, A. Saveliev, I. Kipyatkova, M. Zeleny, I. Mporas, and A. Karpov, “A multimodal user interface for an assistive robotic shopping cart,” *Electronics*, vol. 9, no. 12, p. 2093, 2020.

[6] M. Rahnemoonfar, T. Chowdhury, and R. Murphy, “Rescuenet: A high resolution post disaster uav dataset for semantic segmentation,” *UMBC Student Collection*, 2021.

[7] A. Hong, O. Igharoro, Y. Liu, F. Niroui, G. Nejat, and B. Benhabib, “Investigating human-robot teams for learning-based semi-autonomous control in urban search and rescue environments,” *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3, pp. 669–686, 2019.

[8] M. Schnaubelt, S. Kohlbrecher, and O. von Stryk, “Autonomous assistance for versatile grasping with rescue robots,” in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2019, pp. 210–215.

[9] A. Bokovoy, K. Muravyev, and K. Yakovlev, “Real-time vision-based depth reconstruction with nvidia jetson,” in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–6.

[10] R. Wang, Y. Wei, H. Song, Y. Jiang, Y. Guan, X. Song, and X. Li, “From offline towards real-time verification for robot systems,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1712–1721, 2018.

[11] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, “Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[12] J. De Greeff, T. Mioch, W. Van Vught, K. Hindriks, M. A. Neerincx, and I. Kruijff-Korbayová, “Persistent robot-assisted disaster response,” in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 99–100.

[13] R. Smith, E. Cucco, and C. Fairbairn, “Robotic development for the nuclear environment: challenges and strategy,” *Robotics*, vol. 9, no. 4, p. 94, 2020.

[14] M. Lewis, K. Sycara, and I. Nourbakhsh, “Developing a testbed for studying human-robot interaction in urban search and rescue,” in *Proceedings of the 10th International Conference on Human Computer Interaction (HCII’03)*, 2019, pp. 270–274.

[15] O. Michel, “Cyberbotics ltd. webots™: professional mobile robot simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.

[16] A. Dobrokvashina, R. Lavrenov, Y. Bai, M. Svinin, R. Meshcheryakov, and E. Magid, “Servosila engineer crawler robot modelling in webots simulator,” in *2022 International Journal of Mechanical Engineering and Robotics Research(IJMERR)*, 2022.

[17] A. Dobrokvashina, R. Lavrenov, E. A. 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, pp. 212–217.

[18] E. Soria, F. Schiano, and D. Floreano, “Swarmlab: A matlab drone swarm simulator,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8005–8011.

[19] M. Prabhakar, V. Paulraj, J. A. Dhanraj, S. Nagarajan, D. A. K. Kannappan, and A. Hariharan, “Design and simulation of an automated guided vehicle through webots for isolated covid-19 patients in hospitals,” in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*. IEEE, 2020, pp. 1–5.

[20] L. Pitonakova, M. Giuliani, A. Pipe, and A. Winfield, “Feature and performance comparison of the v-rep, gazebo and argos robot simulators,” in *Annual Conference Towards Autonomous Robotic Systems*. Springer, 2018, pp. 357–368.

[21] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich, “Simulating complex robotic scenarios with morse,” in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2012, pp. 197–208.

[22] A. Ayala, F. Cruz, D. Campos, R. Rubio, B. Fernandes, and R. Dazeley, “A comparison of humanoid robot simulators: A quantitative approach,” in *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2020, pp. 1–6.

[23] M. Körber, J. Lange, S. Rediske, S. Steinmann, and R. Glück, “Comparing popular simulation environments in the scope of robotics and reinforcement learning,” *arXiv preprint arXiv:2103.04616*, 2021.

[24] F. Ma, Z. Sun, and R. Jia, “Virtual laboratory of robot kinematics based on qt and ogre3d,” in *2nd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2017)*. Atlantis Press, 2016, pp. 112–117.

[25] R. Ohnishi and Y. Hoshino, “Position estimation using stereo camera images and physics engine simulation for robot control,” in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*. IEEE, 2018, pp. 753–757.

[26] T. Erez, Y. Tassa, and E. Todorov, “Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx,” in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4397–4404.

[27] H. Yu, Q. Lai, Y. Liang, Y. Wang, and R. Xiong, “A cascaded deep learning framework for real-time and robust grasp planning,” in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1380–1386.

[28] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, “A v-rep simulator for the da vinci research kit robotic platform,” in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, 2018, pp. 1056–1061.

[29] F. Yumbla, E. Q. Yumbla, and H. Moon, “The bioloid gp robot with different configurations for simulation in v-rep controlled by the robot operating system (ros),” in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2020, pp. 54–58.

[30] M. Askarpour, M. Rossi, and O. Tiryakiler, “Co-simulation of human-robot collaboration: From temporal logic to 3d simulation,” *arXiv preprint arXiv:2007.11737*, 2020.

[31] I. Moskvin, R. Lavrenov, E. Magid, and M. Svinin, “Modelling a crawler robot using wheels as pseudo-tracks: model complexity vs performance,” in *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*. IEEE, 2020, pp. 1–5.

[32] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7226–7232.

[33] A. Dobrokvashina, R. Lavrenov, T. Tsoy, E. A. Martinez-Garcia, and Y. Bai, “Navigation stack for the crawler robot servosila engineer,” in *2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2021, pp. 1907–1912.

[34] O. Michel, “Webots: Symbiosis between virtual and real mobile robots,” in *International Conference on Virtual Worlds*. Springer, 1998, pp. 254–263.

[35] M. Mohan, R. K. Chetty, K. M. Azeem, P. Vishal, B. Poornasai, and V. Sriram, “Modelling and simulation of autonomous indoor robotic wastebin in webots for waste management in smart buildings,” in *IOP Conference Series: Materials Science and Engineering*, vol. 1012, no. 1. IOP Publishing, 2021, p. 012022.

[36] X. Jin, X. Ma, and C. Mu, “Design and implementation of family service robots’ object recognition based on webots,” in *Tenth International Conference on Digital Image Processing (ICDIP 2018)*, vol. 10806. International Society for Optics and Photonics, 2018, p. 108060G.

[37] S. Ramil, R. Lavrenov, E. A. Martínez-García, and E. Magid, “Ros-based multiple cameras video streaming for a teleoperation interface of a crawler robot.” *J. Robotics Netw. Artif. Life*, vol. 5, no. 3, pp. 184–189, 2018.

[38] E. Mingachev, R. Lavrenov, T. Tsoy, F. Matsuno, M. Svinin, J. Suthakorn, and E. Magid, “Comparison of ros-based monocular visual slam methods: Dso, ldso, orb-slam2 and dynaslam,” in *International Conference on Interactive Collaborative Robotics*. Springer, 2020, pp. 222–233.