Simulation and Implementation of RSCUAD Walking Robot Based on ROS and Gazebo Simulator

Danu Andrean [1], Nuryono Satya Widodo [2,*]
[1,2] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
Email: [1] Danu1700022001@webmail.uad.ac.id, [2] nuryono.sw@ee.uad.ac.id
*Corresponding Author

***Abstract*— This research describes the virtual humanoid robot R-SCUAD using the Gazebo simulator. In its development, humanoid robots often perform movements that have a negative impact on the robot's hardware, therefore the development of a virtual robot model is a solution to overcome this problem. So that the robot can be simulated before running. Gazebo is a robot simulator that allows to accurately simulate, design and test robots in various environments. Gazebo itself is a simulation used by ROS (robotic operating system). The simulation is built by doing a 3D design process in solidwork software and exported to a URDF file that matches the format on the ROS. Tests carried out on robots are by comparing virtual robots with real robots. From the tests carried out on the robot, it was found that the virtual robot can walk according to the real robot, such as falling if the robot's condition is not balanced. The simulation robot also moves according to the real robot when the controls are carried out.**

***Keywords*—ROS, Gazebo, Robot Simulation, Kinematics and Dynamics, Control**

I. Introduction

The Indonesian Robot Contest (KRI) is an event organized by the Ministry of Education and Culture (Kemendikbud). This event aims to be a competition for design and engineering in the field of robotics. Robot Soccer Universitas Ahmad Dahlan (R-SCUAD) is one of the divisions of the Ahmad Dahlan University Robotics Team that participates in the Indonesian robot contest in the humanoid division (KRSBI-humanoid). R-SCUAD is a humanoid robot with basic abilities that resemble human movements in playing soccer.

Humanoid robots generally face challenges in their development, as they often fall, leading to serious hardware damage and significant costs. Therefore, developing a virtual robot model is a solution to tackle these issues. A simulator is a program that functions to simulate or mimic the real movements of a device or 3D object. In the case of robots, Gazebo is a commonly used simulator. Gazebo is a robot simulator that allows accurate simulation, design, and testing of robots in various environments. RSCUAD has been developing Gazebo, but the movement is still limited to joint movements. In this research, it is expected that walking movements can provide a solution to these issues.

II. Method

In the conducted research, the R-SCUAD robot is used, which has the Darwin-OP platform configuration. The Darwin-OP platform consists of 20 DOF (degrees of freedom) [1], and it exhibits a clear correlation between oscillator parameters and dynamic systems for gait pattern generation [2], which is also applied to the R-SCUAD robot. The design and placement of robot components can be seen in Fig. 1.

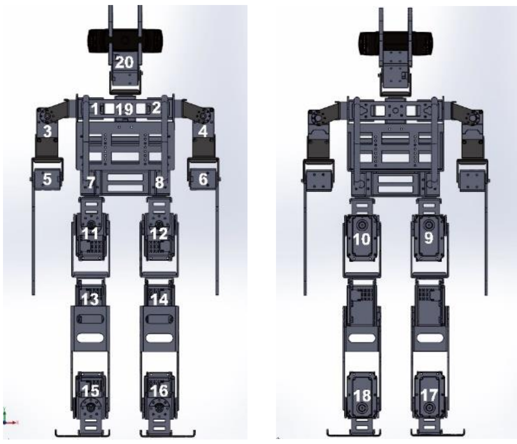


Fig. 1. R-SCUAD Robot [3]

Broadly speaking, the working system is divided into 2 parts: the embedded system on the robot and the separate system, where a computer acts as a client to display the robot's simulation.

In the robot system, it consists of various components. The Intel NUC serves as the main controller of the robot, responsible for providing general commands to the robot and processing important information, such as environment identification from sensor data. Additionally, the main controller also handles crucial tasks like decision-making. On the other hand, the Open CR acts as the sub controller. Unlike the main controller, the sub controller's role is to receive and send data from the main controller to the servos [3].

A. ROS (Robotic Operating System)

ROS (Robot Operating System) was initially developed in 2007 by the Stanford Artificial Intelligence Laboratory to support the Stanford AI Robot project [4]. ROS is an open-source operating system that provides libraries for developing robot software [5]. It serves as a middleware that connects robot hardware with a computer's operating system in a flexible manner. The processes that can be executed in ROS are called Nodes, and the interprocess communication follows a Publish/Subscribe model. The communication data is referred to as Topics. Publisher processes can publish one or multiple Topics, and processes that subscribe to specific Topics can receive their contents [6]. The main goal of ROS is to facilitate robot developers in creating software without starting from scratch [4]-[5].

B. ROS Modeling (Gazebo)

Gazebo is a simulation environment that enables testing of complex systems. It has various applications, including testing the dynamics of control systems before their actual implementation [7]. Moreover, Gazebo allows easy creation of robots, actuators, sensors, and objects [8]. The Gazebo environment consists of a division between the server and the client, provided by two executable programs: “gzserver” for simulating physics, rendering, and sensors, and “gzclient” for the graphical interface to visualize and interact with the simulation [9].

The first step in modeling within ROS requires creating a 3D environment model. The 3D model is generated using software like Solidworks. Once the model is complete, it is exported in STL format meshes to be integrated into the ROS environment [4]. This integration is made possible with the help of robot model packages that include URDF (Unified Robot Description Format) parser files.

The URDF file is used to record all information about the virtual robot. It is common to create a robot model using Computer Aided Design (CAD) tools such as Solidworks, Pro-engineer, or Blender [10]. The URDF file contains descriptions of the model and the robot to be imported into ROS. These descriptions include the object's name, network structure, and additional information about the visual components of an object, such as color and texture [4]. The <robot> tag contains the robot's name, which will be displayed across all ROS subsystems. The <joint> tag is a child tag of the robot, which describes the type of movement (Fixed, Revolute, Continuous, Prismatic, Planar, or Floating). The concept of links and joints' structure can be seen in Fig. 2.

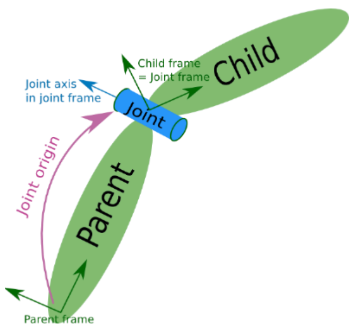


Fig. 2. The structure of links and joints

C. Virtual robot

To facilitate the research and creation of a 3D robot design, developers use Solidworks software. The developers then simplify the process of exporting the 3D robot design into the URDF file format, which is compatible with ROS, by using the URDF Exporter plugin. In the robot's joint structure, each joint connects a parent link to a child link, defining the relationship between different parts of the robot. For this research, the base of the robot utilizes the robot's body structure. The structure of links and joints for the R-SCUAD robot, which was recently created, can be seen in Fig. 3.

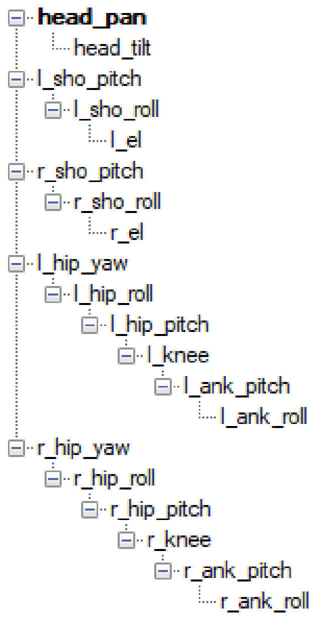


Fig. 3. The joint structure of the R-SCUAD robot

In Gazebo simulator, it has a specific hierarchy that allows seamless integration between programs to work effectively. Simulations are executed by calling a launch file. The launch file itself contains an XML program that configures the controller and robot description [11].

In the virtual robot, a configuration is made according to the ROS protocol to enable the virtual robot to run in the Gazebo simulator [12]-[14]. In this research, the program flow is designed as shown in Fig. 4.

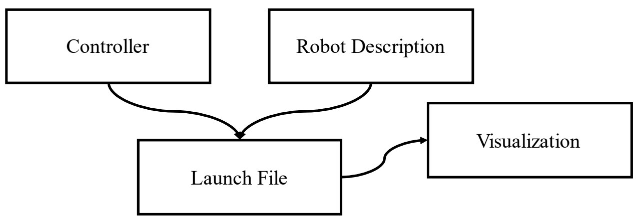


Fig. 4. Virtual Robot Flow

D. Physical Robot

The real robot uses a converter module that is used to transmit data to the robot actuator, namely Sevo Dynamixel. For the robot flow can be seen in Fig. 5.

Fig. 5. Physical Robot Flow

With the use of the U2D2 module, kinematic programs and simple servo movement programs can be implemented on the robot's actuators. Based on the output values from the kinematics, which will later be input into the robot's joints using radian values [15], the following formula is used to perform the conversion:

$$\theta = rad\frac{180}{\pi} \quad (1)$$

After calculating the angles, the next step is to convert them into Dynamixel servo values.

$$n = \theta\frac{4096}{360°} \quad (2)$$

By using the values 4096 as the maximum angle for the Dynamixel servo and 360 as the maximum angle in degrees, you can perform the necessary calculations to map the joint movements with the corresponding angles to control the Dynamixel servos effectively [16]. This conversion allows you to achieve the desired joint movements by controlling the servo angles accordingly.

E. Block Diagram

In the research using the R-SCUAD robot, there is a block diagram consisting of several parts, including the main controller, sub-controller, actuator, camera, Wi-Fi module, and several other components, with the system block diagram design as shown in Fig. 6.

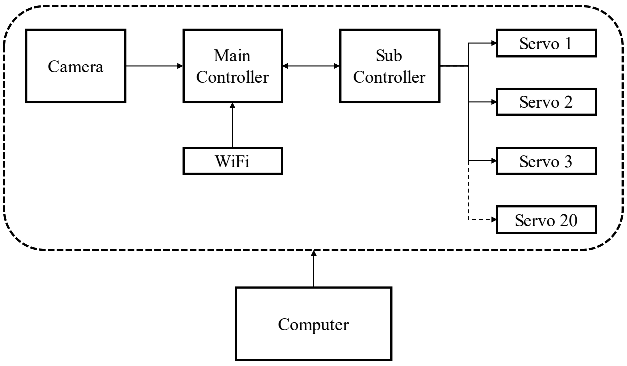


Fig. 6. System Block Diagram of the R-SCUAD Robot

In the robot system design, there is a mini PC that functions as the main controller, which accesses the sensors. The main controller determines the robot's movements by sending commands to the sub-controller, which then forwards them to the actuators. The computer itself works as a robot simulation used for visualization. In the block diagram above, the role of the computer is not directly connected to the robot. This is because the computer can simulate with or without hardware, making it a solution when the robot being simulated has hardware issues [17].

F. Flowchart

In simulating the robot, there are several processes that need to be performed before conducting the actual simulation. These processes are known as initialization and configuration in the system. In the system flow diagram, there are various conditions that can be executed, whether using software or hardware [18], as shown in Fig. 7.

In the flow diagram shown in Fig. 7, there are conditions for both using the robot and not using the robot or hardware. This differentiation is because the processes involved run differently, allowing them to be utilized effectively based on the specific needs and objectives of the simulation [19].

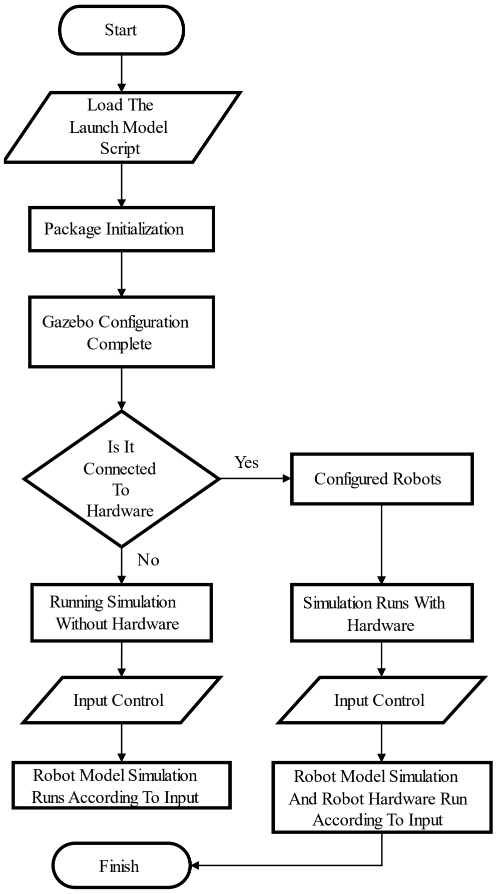


Fig. 7. Flowchart

III. Result and Discussion

The robot testing was conducted by making the robot walk a distance of 100 cm and then making the robot turn right for a distance of 50 cm. The purpose of this testing was to compare the performance of the virtual robot with the physical robot. Several results were obtained from the testing.

However, the specific details of the results and their implications are not mentioned in the provided information. If you would like to know more about the results or have any specific questions related to the testing or its outcomes, please feel free to ask, and I'll be glad to provide further assistance.

A. Virtual Robot Model

In the virtual robot testing, data acquisition is performed by observing the robot's position using odometry in Gazebo, which is then displayed on a graph [20]. To display the odometry, Python programming language is used, and it is called by typing scripts in the terminal as follows:

1. The robot straight-line walking test
   The straight-line walking test of the robot used the following main parameters: DSP ratio 0.04, x amplitude 10, and hip pitch 7 during the straight movement. The result shown in Table 1.
2. Testing the robot walking straight and turning
   In testing the straight robot and turning 90 degrees using the main parameters, namely DSP ratio of 0.04, x amplitude of 10 and hip pitch of 7 when moving straight and x amplitude of 10 and hip-pitch of 5 when turning with a total of 45 periods. The result shown in Table 2.

Table 1. Virtual robot straight-line testing

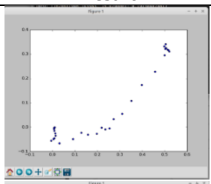
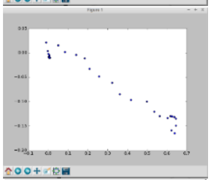
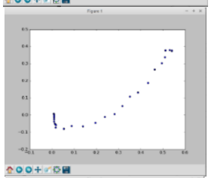
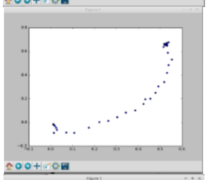
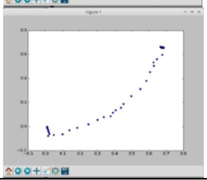
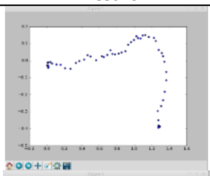
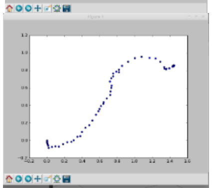
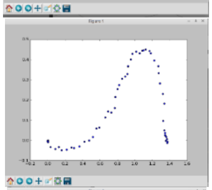
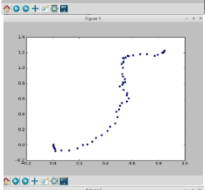
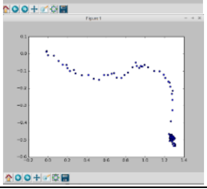
No	Distance	Information	Result
1	50 cm	The robot turns left	
2	65 cm	The robot turns right	
3	55 cm	The robot turns left	
4	55 cm	The robot turns left	
5	70cm	The robot turns left	

Table 2. The virtual robot tests straight and turns 90°

No	Distance	Information	Result
1	96.81cm	The robot slightly shifted 10° to the left	
2	123.24cm	Straight Robot	
3	127.86cm	The robot slightly shifted 5° to the left.	
4	135.58cm	The robot slightly shifted 10° to the left.	
5	97.20 cm	Straight Robot	

B. Physical Robot Model

In real robot testing, data collection is done by tracking the robot using a camera. This method is usually called object tracking. The results on the test can be seen as.

1. Robot testing goes straight
   Straight robot testing uses the main parameters, namely DSP ratio of 0.04, x amplitude of 10 and hip pitch of 7 when moving straight. The result shown in Table 3.
2. Testing the robot walking straight and turning
   In testing the walking and turning robot using the main parameters, namely the DSP ratio of 0.04, x amplitude of 10 and hip-pitch of 7 when moving straight and x amplitude of 10 and hip-pitch of 5 when turning with a total of 45 periods. The result shown in Table 4.

In testing the walking and turning robot using the main parameters, namely the DSP ratio of 0.04, x amplitude of 10 and hip-pitch of 7 when moving straight and x amplitude of 10 and hip-pitch of 5 when turning with a total of 45 periods.

In this test, the virtual robot can follow the movements of the real robot. both realtime and not. Virtual robots tend to have slower processing speeds and poor balance compared to real robots, this makes the robot's movements not the same as real robots.

IV. CONCLUSION

Based on the results of research and testing conducted on virtual robots and real robots. Then it can be concluded as follows.

1. Virtual Robot
   In the research conducted, it was found that virtual robots can follow the movements of real robots. both realtime and not. Virtual robots have problems in real-time simulations with real robots, namely the incompatibility of robot movements caused by virtual robot inertia conditions, so that virtual robots fall easily if they use the same parameters as real robots.
2. Physical Robots
   In the research conducted, it was found that robots have responses that are in accordance with simple commands, such as turning robots and straight robot movements. Real robots have a lack of accelero as a balance of the robot so that it makes the robot fall easily due to lack of balance. This thesis can be used for the development of the R-SCUAD robot in making algorithms and motion movements which were previously carried out directly on real robots so that it burdens the servo conditions.

ACKNOWLEDGEMENT

Thanks to the editors and reviewers for all the suggestions, input and assistance in the process of publishing the manuscript. Acknowledgments are also addressed to those who have supported the research and provided moral and material assistance.

Table 3. Real robot straight testing

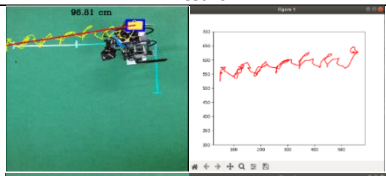
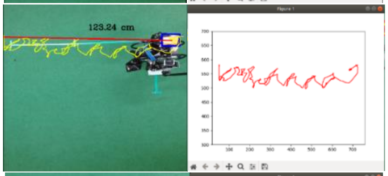
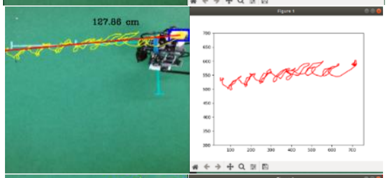
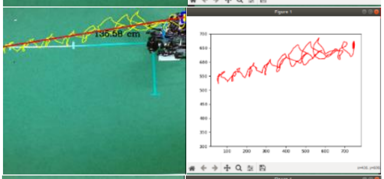
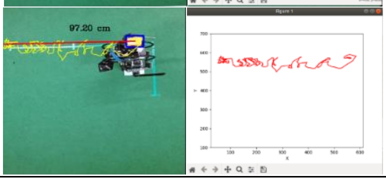
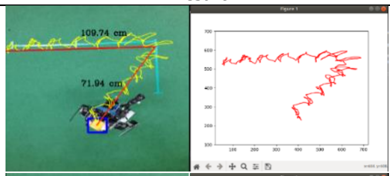
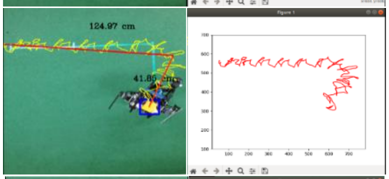
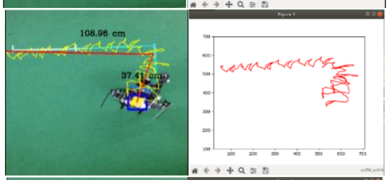
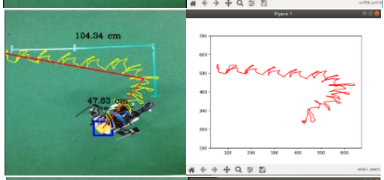
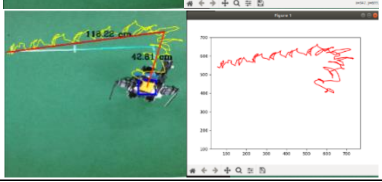
No	Distance	Information	Result
1	96.81cm	Robot slightly shifted 10° to the left	
2	123.24cm	Straight robots	
3	127.86cm	Robot slightly shifted 5° to the left	
4	135.58cm	Robot slightly shifted 10° to the left	
5	97.20 cm	Straight robots	

Table 4. Real robot testing straight and turning 90°

No	Distance	Information	Result
1	Straight 109.74 cm, right 71.94 cm	Robot slightly moves right 40° when turning	
2	Straight 124.97 cm, right 41.56 cm	The robot slightly moves right 10° when walking straight and slightly moves right 10° when turning	
3	Straight 108.96cm, right 37.41cm	Robot slightly moves right 10° while turning	
4	Straight 104.34cm, right 47.83cm	The robot slightly moves right 20° when going straight and slightly moves right 30° when turning	
5	Straight 118.22cm, right 42.81cm	The robot moves slightly left 10° when walking straight and slightly right 5° when turning	

REFERENCES

[1] I. Ha, Y. Tamura, H. Asama, J. Han, D. W. Hong, "Development of open humanoid platform DARwIn-OP," *SICE Annual Conference 2011*, 2011, pp. 2178-2181, 2011, https://ieeexplore.ieee.org/document/6060333.

[2] A. Tsukahara, Y. Hasegawa, Y. Sankai, "Gait support for complete spinal cord injury patient by synchronized leg-swing with HAL," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1737-1742, 2011, https://doi.org/10.1109/IROS.2011.6048790.

[3] B. Bingham *et al*., "Toward Maritime Robotic Simulation in Gazebo," *OCEANS 2019 MTS/IEEE SEATTLE*, pp. 1-10, 2019, https://doi.org/10.23919/OCEANS40490.2019.8962724.

[4] T. Linner, A. Shrikathiresan, M. Vetrenko, B. Ellmann, T. Bock, "Modeling and operating robotic environments using Gazebo/ROS," Proc. 28th Int. Symp. Autom. Robot. Constr. ISARC 2011, pp. 957–962, 2011, https://doi.org/10.22260/ISARC2011/0177.

[5] K. Takaya, T. Asai, V. Kroumov, F. Smarandache, "Simulation environment for mobile robots testing using ROS and Gazebo," *2016 20th Int. Conf. Syst. Theory, Control Comput. ICSTCC 2016 - Jt. Conf. SINTES 20, SACCS 16, SIMSIS 20 - Proc.*, pp. 96–101, 2016, https://doi.org/10.1109/ICSTCC.2016.7790647.

[6] A. Dobrokvashina, R. Lavrenov, E. A. Martinez-Garcia, Y. Bai, "Improving model of crawler robot Servosila "Engineer" for simulation in ROS/Gazebo," *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 212-217, 2020, https://doi.org/10.1109/DeSE51703.2020.9450233.

[7] N. Koening, A. Howard, "Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator," *J. Am. Med. Assoc.*, vol. 285, no. 3, pp. 290–292, 2001, https://doi.org/10.1001/jama.285.3.290.

[8] Z. B. Rivera, M. C. De Simone, D. Guida, "Unmanned ground vehicle modelling in Gazebo/ROS-based environments," *Machines*, vol. 7, no. 2, pp. 1–21, 2019, https://doi.org/10.3390/machines7020042.

[9] W. Qian *et al*., "Manipulation task simulation using ROS and Gazebo," *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 2594-2598, 2014, https://doi.org/10.1109/ROBIO.2014.7090732.

[10] D. Chikurtev, "Mobile Robot Simulation and Navigation in ROS and Gazebo," *2020 International Conference Automatics and Informatics (ICAI)*, pp. 1-6, 2020, https://doi.org/10.1109/ICAI50593.2020.9311330.

[11] T. Wright, A. West, M. Licata, N. Hawes, B. Lennox, "Simulating Ionising Radiation in Gazebo for Robotic Nuclear Inspection Challenges," *Robotics*, vol. 10, no. 3, p. 86, 2021, http://dx.doi.org/10.3390/robotics10030086.

[12] M. Sánchez, J. Morales, J. L. Martínez, J. J. Fernández-Lozano, A. García-Cerezo, "Automatically Annotated Dataset of a Ground Mobile Robot in Natural Environments via Gazebo Simulations," *Sensors*, vol. 22, no. 15, p. 5599, 2022, https://doi.org/10.3390/s22155599.

[13] A. F. Olalekan, J. A. Sagor, M. H. Hasan, A. S. Oluwatobi, "Comparison of Two SLAM Algorithms Provided by ROS (Robot Operating System)," *2021 2nd International Conference for Emerging Technology (INCET)*, pp. 1-5, 2021, https://doi.org/10.1109/INCET51464.2021.9456164.

[14] K. K. P. Gayashani, U. U. S. Rajapaksha and C. Jayawardena, "Moving a Robot In Unknown Areas Without Collision Using Robot Operating System," *2022 2nd International Conference on Advanced Research in Computing (ICARC)*, pp. 84-89, 2022, https://doi.org/10.1109/ICARC54489.2022.9754138.

[15] N. Teslya, A. Smirnov, A. Ionov, A., Kudrov, "Multi-robot coalition formation for precision agriculture scenario based on gazebo simulator," *Proceedings of 15th International Conference on Electromechanics and Robotics" Zavalishin's Readings"*, vol. 187, pp. 329-341, 2021, https://doi.org/10.1007/978-981-15-5580-0_27.

[16] L. He, P. Glogowski, K. Lemmerz, B. Kuhlenkötter, W. Zhang, "Method to integrate human simulation into gazebo for human-robot collaboration," *IOP Conference Series: Materials Science and Engineering*, vol. 825, no. 1, p. 012006, 2020, https://doi.org/10.1088/1757-899X/825/1/012006.

[17] Y. Alborzi, B. S. Jalal, E. Najafi, "ROS-based SLAM and Navigation for a Gazebo-Simulated Autonomous Quadrotor," *2020 21st International Conference on Research and Education in Mechatronics (REM)*, pp. 1-5, 2020, https://doi.org/10.1109/REM49740.2020.9313875.

[18] M. Marian, F. Stîngă, M. -T. Georgescu, H. Roibu, D. Popescu, F. Manta, "A ROS-based Control Application for a Robotic Platform Using the Gazebo 3D Simulator," *2020 21th International Carpathian Control Conference (ICCC)*, pp. 1-5, 2020, https://doi.org/10.1109/ICCC49264.2020.9257256.

[19] J. Platt, K. Ricks, "Comparative Analysis of ROS-Unity3D and ROS-Gazebo for Mobile Ground Robot Simulation," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 4, p. 80, 2022, https://doi.org/10.1007/s10846-022-01766-2.

[20] B. Abbyasov, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Svinin, E. Magid, "Automatic tool for Gazebo world construction: from a grayscale image to a 3D solid model," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226-7232, 2020, https://doi.org/10.1109/ICRA40945.2020.9196621.