2023 International Conference on Smart Cities, Automation & Intelligent Computing Systems
Bali, Indonesia, 6-8 December 2023

Barelang FC: An Adult-Size Humanoid Robot Simulation Model on The ROS and Gazebo

2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS) | 979-8-3503-1829-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICON-SONICS59898.2023.10434954

1st Indra Kurniawan
Department of Electrical Engineering
Politeknik Negeri Batam
Batam, Indonesia
indra.kurniawan09158@gmail.com

2nd Nico Breygin
Department of Electrical Engineering
Politeknik Negeri Batam
Batam, Indonesia
nicobreygin23@gmail.com

3rd Irga Julian Diputra
Department of Electrical Engineering
Politeknik Negeri Batam
Batam, Indonesia
diputra184@gmail.com

4th Julian Reza Erwandi
Department of Electrical Engineering
Politeknik Negeri Batam
Batam, Indonesia
jrerwandi@gmail.com

5th Eko Rudiawan Jamzuri
Department of Electrical Engineering
Politeknik Negeri Batam
Batam, Indonesia
ekorudiawan@polibatam.ac.id

***Abstract*— Investigations in the domain of humanoid robots entail significant expenses. This condition is determined by the auxiliary components and the intricacy of the manufacturing of the parts, which incurs substantial expenses. As the robot's size increases, so does the corresponding construction expense. Aside from the exorbitant expense, the maintenance and repair procedures for the robot pose a separate issue. This problem poses a significant challenge for aspiring researchers seeking to delve into the realm of humanoid robots. Various options for creating affordable humanoid robots have been suggested using 3D-printed materials. Nevertheless, this continues to challenge researchers who lack proficiency in the mechanical intricacies. Researchers may sometimes lack the specialized expertise necessary to produce and assemble robots. This study suggests the development of a simulator for a humanoid robot intended for research. The simulation is based on the Barelang FC adult-size humanoid robot, which has a height of 131 cm. The robot is constructed using 30 actuators. Furthermore, the robot is outfitted with auxiliary sensors, including stereo cameras, IMU, and position sensors at each joint. The findings from the development of the Barelang FC simulators indicate that this simulation model outperforms the Robotis OP3 and Thormang3 simulators regarding real-time factors. The Barelang FC robot has a real-time factor of 0.9 in an upright position and 0.87 when lying down. The supine and standing robot pose exhibit real-time factor performances below 0.75 for the other two robots. These findings demonstrate that the Barelang FC simulator effectively replicates the real-time performance of robots, a crucial aspect of robot modeling.**

Keywords—humanoid robot, Robot Operating System, Gazebo, ROS, robot simulation

I. INTRODUCTION

Humanoid robots are robotic devices that imitate humans' physical movements and behavioral patterns. The creature possesses anatomical features identical to humans, including a head, arms, and legs. These components are specifically engineered to engage with humans or their surroundings seamlessly. Humanoid robots are specifically engineered to execute various activities, including the ability to identify and interpret movements, navigate around obstacles, and engage with tools and the surrounding environment. Conversely, humanoid robots can also be operated remotely to minimize human exertion in arduous activities [1].

Humanoid robots have applications in diverse domains, including industrial automation, military operations, healthcare, and transportation. Furthermore, robots can serve as personal aides. For instance, to assist individuals who are ill or advanced in age [2] and perform tasks that are unclean or pose risks [3]. Humanoid robots have the potential to carry out any task that humans are capable of as long as the robot is equipped with the necessary algorithm [4].

Humanoid robots are specifically engineered to imitate human expressions, achieving high similarity to humans. These expressions may encompass ocular movements, cranial motions, and verbal communication. These expressions can be utilized to investigate the emotional communication between people and machines. The expressions and behavior of humanoid robots are intriguing when used in entertainment media. Various performances feature humanoid robots as the primary actors, such as robot magicians [5], [6], [7], robot marathon races [8], [9], and robot soccer matches [10].

The research and development of humanoid robots present numerous problems. The primary obstacle arises from the factor of cost. Humanoid robots, particularly adult-size ones, incur significant costs in production or acquisition from external sources. Researchers from small laboratories are constrained in their ability to delve into the realm of humanoid robots. The cost and difficulty of developing this task are affected by the robot's qualities, capabilities, and supporting components [11]. Humanoid robots are intricate mechatronic systems that necessitate meticulous precision. Augmenting the dimensions of humanoid robots results in a proportional escalation in the mass and cost of the accompanying elements. For instance, when the robot's mass increases, the torque needed for the actuator also increases, making the price costly. Furthermore, the actuator's operation becomes more challenging as it necessitates the usage of a specialized controller. This challenge also impacts the maintenance and repair procedure. These considerations demonstrate the necessity for creating humanoid robot platforms using cost-effective alternative methods. This solution will facilitate the accessibility for researchers in the first stages of studying humanoid robots.

A practical approach to reducing the expenses of researching humanoid robots is utilizing a platform that can be autonomously built and assembled. This approach has been accomplished by researchers [12], [13], and [14] who developed humanoid robots with materials amenable to 3D printing technology. Utilizing this substance allows the robot to be assembled within the laboratory without external assistance. Nevertheless, this approach continues to challenge computer science researchers with limited knowledge of mechanics, a crucial domain for constructing and assembling robots.

The utilization of simulators can reduce research expenses in humanoid robots. Presently, there is a significant surge in the use of simulators [15]. The Nao humanoid robot has been simulated using many simulator devices to compare computing performance in several simulators, including Gazebo, Webots, and CoppeliaSim [16]. Based on the findings of this investigation, Gazebo is identified as the simulator that incurs the most significant CPU demand. However, Gazebo offers benefits in terms of precision and authentic simulation outcomes. Furthermore, Gazebo is presently the preeminent simulator employed by researchers. Gazebo boasts a substantial and engaged community, which ensures ample assistance and resources for its users.

This study aims to create a model of a humanoid robot the same size as an adult human. This model will be developed using the Gazebo simulator and the Robot Operating System (ROS). The simulation model is designed to be an affordable research platform in the simulation environment. The research focuses on developing a simulation model of the adult-size humanoid robot called Barelang FC, which features a parallel kinematics structure in its legs. The simulation model is constructed using Unified Robotics Description Format (URDF) scripts, providing a realistic robot model and achieving a high real-time factor.

The specific explanation of this paper will be deliberated over in the subsequent segment. Section II exposes the materials and methods employed in developing the simulation model. Additionally, Section III shall elucidate the achieved development outcomes, encompassing comparisons with alternative simulation models. The paper is concluded in Section IV.

II. Materials and Methods

This section will explain the materials and methodologies for developing the Barelang FC adult-size humanoid robot simulator. Initially, we shall elucidate the structure and configuration of the humanoid robot that will undergo simulation. Next, we will provide a detailed description of the robot simulation environment that will be utilized. Lastly, we will elucidate the process of modeling utilizing the URDF.

A. Robot Platform

The research will focus on modeling and simulating the Barelang FC adult-size humanoid robot as the chosen robot platform. Barelang FC is an advanced robotic platform designed to conduct research in robotics and Artificial Intelligence (AI). The Barelang FC robot was designed in two different dimensions: one in a smaller size called kid-size and another in a larger size called adult-size. The Barelang FC kid-size robot often participates in the Kontes Robot Indonesia (KRI) and RoboCup, consistently achieving notable results yearly.

The adult-size version of Barelang FC is an advancement of the kid-size robot platform [17], [18]. Fig. 1 displays the 3D CAD of this robot. To provide a summary, the height of this robot platform measures 131 cm. The robot uses 30 units of Dynamixel XH540-W270 digital servo motors. The leg section comprises eight actuators, while the hand section comprises four. The head of the robot is equipped with two additional actuators that enable it to rotate horizontally (to the right and left) and vertically (up and down).

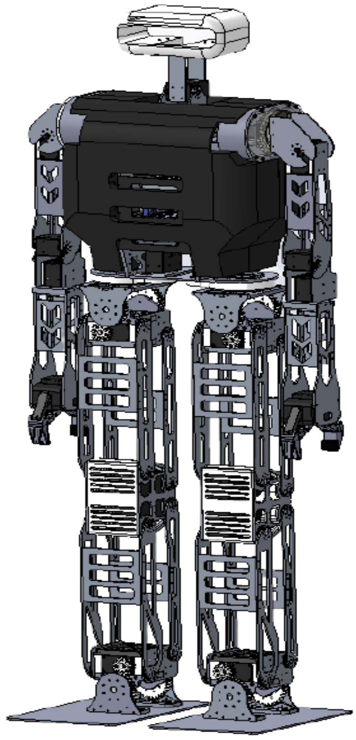


Fig. 1. CAD design of Barelang FC adult-size.

The Barelang FC adult-size robot features a parallel link configuration in its thigh and knee, specifically in its leg design. Fig. 1 illustrates the parallel linkages regulated by eight actuators positioned at each knee joint. Bearings are put in the thigh and ankle joint to allow unfettered movement. In order to achieve a specific angular configuration, the movement of the actuator on the knee will cause the links on the hip and knee to move accordingly. By employing this paradigm, the reliance on actuators for the hip and knee can be minimized. Nevertheless, the knee actuator needs an actuator with sufficient torque.

B. Robot Operating System (ROS) and Gazaebo Simulator

The simulation model created in this research is intended explicitly for utilization within the Robot Operating System and Gazebo Simulator frameworks. We utilized the ROS Noetic Ninjemys and Gazebo version 11 installed on the Ubuntu 20.04 LTS (Focal Fossa) operating system.

A Robot Operating System (ROS) is a middleware operating system that offers tools and frameworks for developing robot software [19]. Due to its nature as middleware, ROS is incapable of functioning independently on a computer. Installation of ROS requires a Linux operating system, specifically the Ubuntu distribution. However, ROS2 can be utilized on several operating systems, including Windows and macOS. The reason for selecting this simulation environment is due to robotics researchers' widespread adoption of ROS. Moreover, ROS offers a comprehensive framework for developing algorithms related to autonomous navigation, speech processing, object detection, and recognition [20]. Furthermore, ROS exhibits versatility, modularity, and seamless integration with diverse hardware and software components [21].

The Gazebo simulator is a software application replicating and evaluating diverse robotic systems within a computer-generated environment. Researchers and developers can use this to conduct experiments and evaluate control algorithms' effectiveness without relying on costly and sometimes hazardous field studies [22], [23], [24]. Research has been conducted by [25] on utilizing Gazebo as a simulator for humanoid robots. The study focuses on employing the Darwin OP2 robot simulator model as a means of teaching English through the usage of robots. Integrating Gazebo with ROS is comparatively simpler than integrating it with alternative robot simulators. This reason is because ROS already possesses numerous software packages that can be utilized to connect with the Gazebo simulator.

C. Unified Robot Description Format (URDF)

The Unified Robot Description Format (URDF) is a widely used script for robot modeling in robotics [26]. URDF scripts specify a robot's kinematics or dynamics model, essential for precise robot control, estimation, and simulation. The robot's kinematic structure, which refers to the geometric relationship between its linkages and joints, is represented in Extensible Markup Language (XML) format. Currently, URDF is a modeling script that receives complete support from ROS and Gazebo.

The process of constructing a robot model using the URDF script is illustrated in the flowchart depicted in Fig. 2. The process commences with examining the robot's design and structure in the Computer-Aided Design (CAD) file. Currently, the complete physical structure of the robot is divided into multiple sub-assemblies, each representing a specific link of the robot. The separated 3D design of the link is afterward transformed into Standard Triangle Language (STL). Subsequently, this STL file is employed as a visual representation within the simulation model. On the other hand, the 3D design of the link is simplified into a rectangular shape encompassing the link's entire area to generate a collision model. Subsequently, the rectangular shape is transformed into an STL file format and employed as a collision model within the URDF script. Additionally, the link sub-assembly file evaluates the link's mass estimation, its center of mass (CoM) position, and the parameters of its inertia matrix. This examination establishes the composition of the material utilized in the link. The CAD software can readily extract the mass, CoM, and inertia matrix values.

Once all the link parameters have been acquired from the CAD file, the URDF scripting can be executed. The URDF script is created by designating the base link as the primary root of the robot model's hierarchy. Next, the joints attached to this base link are identified and explained in connection to the other links. The URDF script that specifies the Barelang FC adult-size robot can be viewed at the provided URL https://github.com/BarelangFC/BarelangFC-AdultSize-Simulation.

D. Collision Model Simplification

The collision model of the robot link is one factor that impacts the computational load on the simulator program. The simulator utilizes this collision model to replicate collisions between solid objects accurately. The CPU's computing process lengthens as the collision model becomes intricate. The intricacy of the collision model is evident from the magnitude of the STL file. The greater the complexity of the generated STL file, the higher its size.

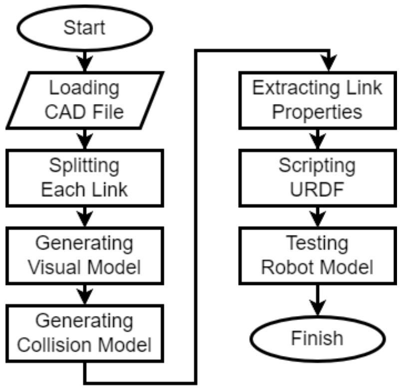
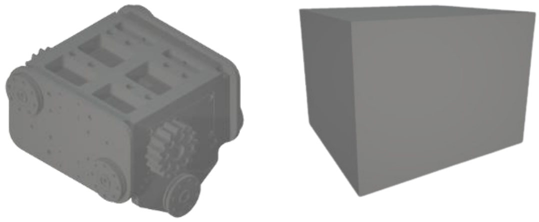


Fig. 2. Process flowchart for modelling humanoid robot.



(a) (b)

Fig. 3 Illustration of (a) visual model for knee link and (b) simplified collision model.

In order to alleviate this computational load, it is advisable to simplify the geometry of the connection. Mainly when it is intended for use as a collision model. The collision model may be defined using a basic rectangular form to do this. Fig. 3 illustrates this simplification. Fig. 3 (a) depicts a visual representation of a knee link directly converted from a CAD design. Fig. 3 (b) depicts the collision model of the same link. Following the simplification procedure, the collision model file size was reduced to 169 KBytes from its original 2.37 MBytes, as shown in Fig. 3 (b).

III. Result and Discussion

This section outlines the tests performed to validate the created robot model. In addition, we evaluate the test outcomes by comparing them to several open-source robot models. The specifics of this explanation will be elucidated in the subsequent sub-sections.

A. Evaluation Metrics

In order to assess the effectiveness of the created simulation model, we employed three evaluation indicators. The three assessment criteria are CPU utilization, memory usage, and real-time performance. CPU load refers to the CPU resources needed to execute a simulation model. Memory consumption refers to the overall amount of memory utilized during the execution of the simulation model. Moreover, the real-time component pertains to the software's capacity to deliver immediate and up-to-date outcomes and updates without substantial delays [27].

The CPU load and memory consumption values were acquired by directly viewing the values in the htop software on Linux. These values were noticed upon loading the simulation model into the Gazebo simulator. A simulation model is better when it has lower CPU load and memory consumption values. Conversely, the real-time factor can be immediately derived from the data provided in Gazebo. The real-time factor value falls within the range of 0 to 1. A higher

TABLE I. COMPUTER SPECIFICATION FOR BENCHMARKING.

Device	Specification
Central Processing Unit (CPU)	Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz
CPU Core(s) per socket	8
CPU Thread(s) per core	2
Memory	16 GBytes
Graphics Processing Unit (GPU)	GeForce GTX 1080

TABLE II. SIMULATION TESTING WHILE ROBOT IS STANDING.

Simulation Model	Evaluation Metrics		
	CPU Load (%)	***Memory Consumption (%)***	***Real-time Factor***
Barelang FC adult-size	223.3	3.4	**0.90**
Robotis OP3	**210.6**	3.9	0.72
Thormang3	211.0	**2.4**	0.53

TABLE III. SIMULATION TESTING WHILE ROBOT IS FALLING.

Simulation Model	Evaluation Metrics		
	CPU Load (%)	***Memory Consumption (%)***	***Real-time Factor***
Barelang FC adult-size	223.2	3.4	**0.87**
Robotis OP3	**157.8**	3.9	0.11
Thormang3	209.6	**2.4**	0.49

value indicates that the simulation closely matches the actual time, while a lower real-time factor value results in a significant deviation from the actual time. This test is conducted on a machine with identical characteristics. The computer specifications for the test are outlined in Table I.

B. *Evaluation Results*

We conducted our testing using two different methods. One method involves placing the robot model upright, while the other involves placing it on its back. These two methods are employed due to the substantial disparities in their effects under both circumstances. Only the sole comes into contact with the floor surface when the robot is upright. While the robot is lying flat, multiple parts will come into contact with the floor. This testing will influence the simulator's ability to calculate the effects on the surface accurately. As the number of collisions calculated increases, the CPU burden generated also increases.

In order to assess the performance of the resultant robot model, we conducted a benchmarking analysis, comparing it with other robots. We conducted a benchmark test on the Thormang3 and Robotis OP3 humanoid robot models. The manufacturers have provided the simulation models for both robots. The depiction of the Barelang FC robot simulator is shown in Fig. 4. The image displays the spatial coordinates assigned to each joint of the robot's frame.

The outcomes of this examination are displayed in Table II and Table III. Table II demonstrates that the CPU load necessary to simulate the Barelang FC robot is significantly higher, at 223.3%, when the robot stands compared to other robots. The Robotis OP3 simulation model necessitates only 210.6%, while Thormang3 necessitates 211.0%. However, regarding memory usage, the Barelang FC simulation model ranks second, with a consumption rate of only 3.4%, surpassed only by Thormang3. The Barelang FC simulator is a simulation model with the most significant real-time factor value of 0.9.

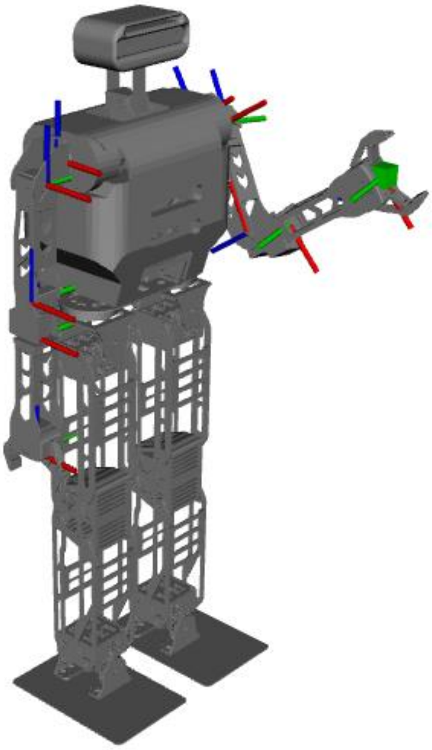
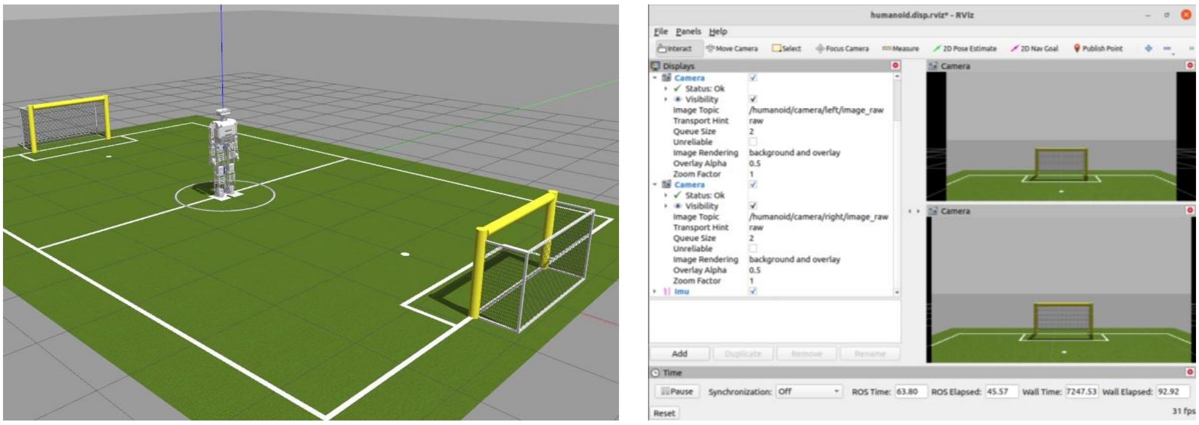


Fig. 4. Visualization of Barelang FC adult-size in RViz

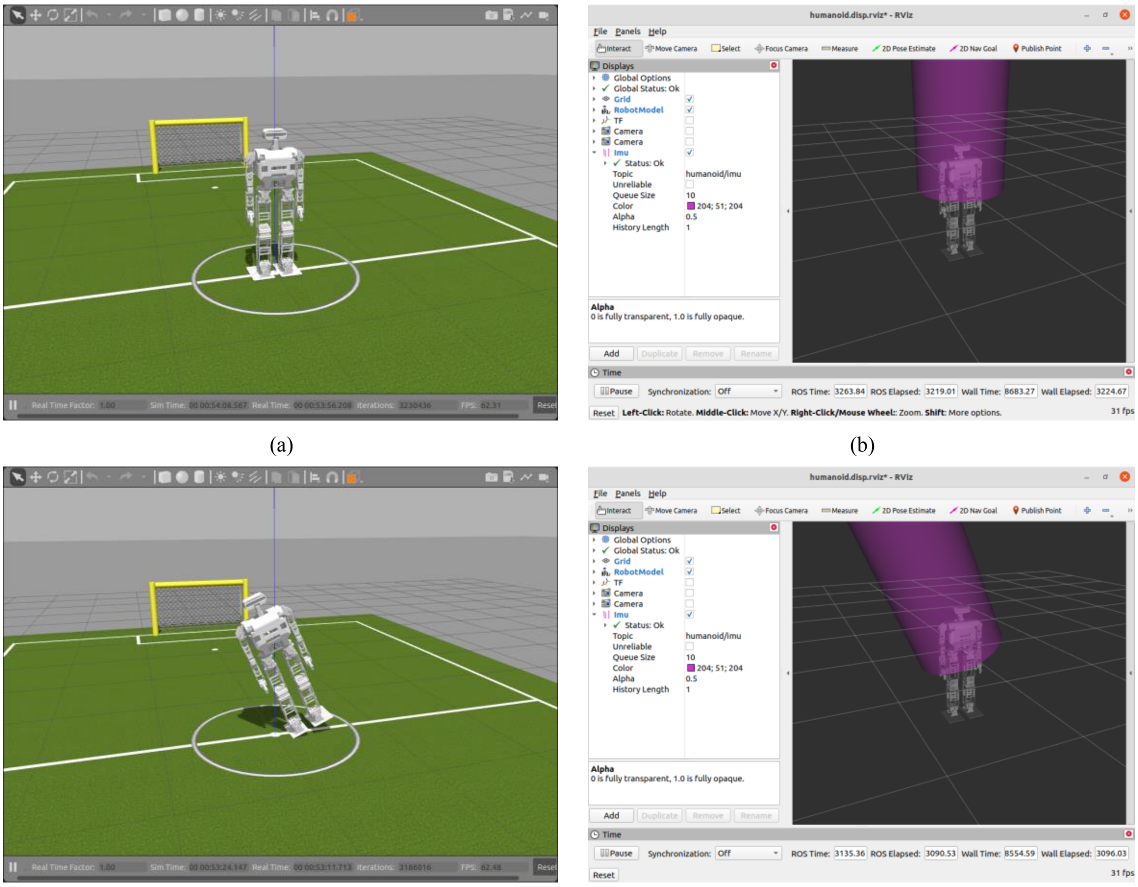
Regarding testing in the robot's inverted position, the outcomes exhibit minimal variation. The Barelang FC simulator CPU load remains nearly unchanged, now at 223.2%, while the memory consumption remains constant at 3.4%. However, the three examined robot models exhibit a notable decrease in the real-time factor value. The real-time factor on the Barelang FC simulator had a minor reduction to 0.87. However, the real-time factor experienced a substantial decline for the remaining two robot models. The real-time factor value in the Robotis OP3 simulator dropped significantly, reaching a value of 0.11. During the use of the Thormang3 simulator, the real-time factor experienced a reduction to 0.49. The Robotis OP3 and Thormang3 robot models are not efficiently designed for their collision model. As a result, while the robot is lying flat, the computational effort needed to simulate collisions becomes greater. As a consequence, the real-time factor value decreases.

The Barelang FC robot simulator has the most significant real-time factor metric compared to the other two. This result is due to our simplification of the collision model. The collision model and visual model in the Robotis OP3 and Thormang3 simulators utilize the identical STL file, which is directly exported from CAD software without undergoing any simplification procedures. In the BarelangFC simulator, the visual model is constructed using STL files obtained from CAD software, while the collision model is built using STL files consisting of basic box shapes.



(a) (b)

Fig. 5. (a) Barelang FC on the Gazebo simulator and (b) stereo camera visualization on RViz



(a) (b)

(c) (d)

Fig. 6. (a), (c) Robot tilt position in gazebo simulator and (b), (d) visualization of marker representing IMU sensor orientation of the base frame.

C. Sensor Model Validation

The Barelang FC adult-size robot model is outfitted with stereo cameras and IMU sensors incorporated into the URDF. We tested the robot's performance on the soccer field area at the Gazebo to verify these sensor models' functionality. Subsequently, we performed surveillance on the RViz software. In RViz, we display the data acquired from the camera sensor presented. Fig. 5 displays the outcomes of this experiment. When the robot directs its gaze toward the target, the image visualization on RViz depicts the display positioned in front of the robot. Due to the utilization of a stereo camera model, two distinct image representations are available, each representing the camera's right and left images.

Simultaneously, we assess the IMU sensor by visually representing the data acquired from the IMU sensor model. In order to evaluate the functionality of this IMU sensor, we deliberately adjusted the robot's position in an upright orientation or at an inclined angle within the Gazebo simulator. Next, carefully monitor the marker's rotation in RViz. If the IMU sensor model output is accurate, the marker's direction will align with the robot's tilt direction. In Fig. 6 (a), the robot is in an upright posture, and the marker visualization in RViz aligns with the robot's orientation, as illustrated in Fig. 6 (b). In addition, when the robot is tilted to the right, the IMU marker visualization also indicates the same direction, as depicted in Fig. 6 (c) and (d).

IV. Conclusion and Future Works

According to the acquired test results, the built simulation model can function correctly in the ROS and Gazebo environments. The latter exhibits the highest real-time factor value among the two simulation models: Robotis OP3 and Thormang3. When the Barelang FC robot is placed supine on the floor surface, there is a reduction in the real-time factor value of 0.03. Nevertheless, compared to other robots, the Barelang FC robot exhibits the most significant real-time factor value, which amounts to 0.87. However, when considering CPU consumption, the Barelang FC robot model exhibits the most considerable CPU performance burden compared to the other two models. Regarding CPU usage, the Barelang FC robot's performance is average compared to the other two. The higher the value of the real-time factor, the closer the Barelang FC robot model is to real-time performance. Additionally, further analysis of the Barelang FC simulator's CPU performance is necessary to optimize the simulation and prevent excessive CPU usage.

Acknowledgment

The author would like to thank the Politeknik Negeri Batam and Barelang Robotics and Artificial Intelligence Lab for providing facilities and resources to support the research.

References

[1] K. Darvish *et al.*, "Teleoperation of Humanoid Robots: A Survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, Jun. 2023, doi: 10.1109/TRO.2023.3236952.

[2] T. Tanioka, "Nursing and Rehabilitative Care of the Elderly Using Humanoid Robots," *The Journal of Medical Investigation*, vol. 66, no. 1.2, pp. 19–23, Feb. 2019, doi: 10.2152/jmi.66.19.

[3] Z. Faraj *et al.*, "Facially expressive humanoid robotic face," *HardwareX*, vol. 9, p. e00117, Apr. 2021, doi: 10.1016/j.ohx.2020.e00117.

[4] D. Wang and A. Sharma, "Research on target recognition and mobile job control of humanoid robot based on KM34Z256 for industrial applications," *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, pp. 303–319, Apr. 2021, doi: 10.1108/IJICC-10-2020-0144.

[5] J. Jeong, J. Yang, and J. Baltes, "Robot magic show: human–robot interaction," *Knowl Eng Rev*, vol. 35, p. e15, Apr. 2020, doi: 10.1017/S0269888920000211.

[6] J. Jeong, J. Yang, and J. Baltes, "Robot magic show as testbed for humanoid robot interaction," *Entertain Comput*, vol. 40, p. 100456, Jan. 2022, doi: 10.1016/j.entcom.2021.100456.

[7] J. Yang, J. Jeong, E. R. Jamzuri, and J. Baltes, "Humanoid robot magic show performance," *Multimed Tools Appl*, vol. 82, no. 22, pp. 34609–34630, Sep. 2023, doi: 10.1007/s11042-023-14690-w.

[8] E. R. Jamzuri, H. Mandala, and J. Baltes, "A Fast and Accurate Object Detection Algorithm on Humanoid Marathon Robot," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 8, no. 1, pp. 204–214, Mar. 2020, doi: 10.52549/ijeei.v8i1.1960.

[9] E. R. Jamzuri, H. Mandala, and J. Baltes, "ROS Based Software Framework for Kid-Size Humanoid Marathon Robot," in *2019 2nd International Conference on Applied Engineering (ICAE)*, IEEE, Oct. 2019, pp. 1–6. doi: 10.1109/ICAE47758.2019.9221733.

[10] P. Stone, "Designing Robots to Best World Cup Winners has Inspired Generations of Roboticists," *IEEE Spectr*, vol. 60, no. 7, pp. 40–49, Jul. 2023, doi: 10.1109/MSPEC.2023.10177053.

[11] S. Saeedvand, M. Jafari, H. S. Aghdasi, and J. Baltes, "A comprehensive survey on humanoid robot development," *Knowl Eng Rev*, vol. 34, p. e20, Dec. 2019, doi: 10.1017/S0269888919000158.

[12] S. Saeedvand, H. S. Aghdasi, and J. Baltes, "Mechatronic Design of ARC Humanoid Robot Open Platform: First Fully 3D Printed Kid-Sized Robot," *International Journal of Humanoid Robotics*, vol. 17, no. 03, p. 2050010, Jun. 2020, doi: 10.1142/S0219843620500103.

[13] G. Ficht, P. Allgeuer, H. Farazi, and S. Behnke, "NimbRo-OP2: Grown-up 3D printed open humanoid platform for research," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, IEEE, Nov. 2017, pp. 669–675. doi: 10.1109/HUMANOIDS.2017.8246944.

[14] P. Allgeuer, H. Farazi, G. Ficht, M. Schreiber, and S. Behnke, "The igus Humanoid Open Platform," *KI - Künstliche Intelligenz*, vol. 30, no. 3–4, pp. 315–319, Oct. 2016, doi: 10.1007/s13218-016-0448-6.

[15] S. Woo, M. Kim, and S. Shin, "Trend of Robot Simulators for the Development of Intelligent Robots: A Review," *Journal of Korean Institute of Intelligent Systems*, vol. 33, no. 1, pp. 72–83, Feb. 2023, doi: 10.5391/JKIIS.2023.33.1.72.

[16] A. Ayala, F. Cruz, D. Campos, R. Rubio, B. Fernandes, and R. Dazeley, "A Comparison of Humanoid Robot Simulators: A Quantitative Approach," *ICDL-EpiRob 2020 - 10th IEEE International Conference on Development and Learning and Epigenetic Robotics*, Oct. 2020, doi: 10.1109/ICDL-EPIROB48136.2020.9278116.

[17] S. Susanto, J. A. Ricardo Silitonga, R. Analia, E. R. Jamzuri, and D. S. Pamungkas, "Tiny-YOLO distance measurement and object detection coordination system for the BarelangFC robot," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, p. 6926, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6926-6939.

[18] W. Winarti, S. Susanto, R. Analia, and E. Jamzuri, "Improving The Stereo Distance Measurement Accuracy on The Barelang-FC Humanoid Robot," in *Proceedings of the 5th International Conference on Applied Engineering, ICAE 2022, 5 October 2022, Batam, Indonesia*, EAI, Jun. 2023. doi: 10.4108/eai.5-10-2022.2327753.

[19] A. Jalil, "ROBOT OPERATING SYSTEM (ROS) DAN GAZEBO SEBAGAI MEDIA PEMBELAJARAN ROBOT INTERAKTIF," *ILKOM Jurnal Ilmiah*, vol. 10, no. 3, pp. 284–289, Dec. 2018, doi: 10.33096/ilkom.v10i3.365.284-289.

[20] R. K. Megalingam, V. S. Naick, M. Motheram, J. Yannam, N. C. Gutlapalli, and V. Sivanantham, "Robot operating system based autonomous navigation platform with human robot interaction," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 3, p. 675, Jun. 2023, doi: 10.12928/telkomnika.v21i3.24257.

[21] B. Dieber, B. Breiling, S. Taurer, S. Kacianka, S. Rass, and P. Schartner, "Security for the Robot Operating System," *Rob Auton Syst*, vol. 98, pp. 192–203, Dec. 2017, doi: 10.1016/j.robot.2017.09.017.

[22] M. Nithya and M. R. Rashmi, "Gazebo – Simulink Framework for Trajectory Tracking in a Multi-Quadcopter Environment," *International Review of Automatic Control (IREACO)*, vol. 15, no. 4, p. 164, Jul. 2022, doi: 10.15866/ireaco.v15i4.21432.

[23] C. Riordan and C. O'Donnell, "Fusion of LiDAR and Computer Vision for Autonomous Navigation in Gazebo," in *SoutheastCon 2023*, IEEE, Apr. 2023, pp. 553–558. doi: 10.1109/SoutheastCon51012.2023.10115172.

[24] P. Deng, P. Zhang, Z. Gong, and L. Li, "Simulation System of Underwater Manipulator Based on Gazebo," in *2023 8th International Conference on Control and Robotics Engineering (ICCRE)*, IEEE, Apr. 2023, pp. 115–120. doi: 10.1109/ICCRE57112.2023.10155577.

[25] L. Gavrilova, A. Kotik, T. Tsoy, E. A. Martinez-Garcia, M. Svinin, and E. Magid, "Facilitating a preparatory stage of real-world experiments in a humanoid robot assisted English language teaching using Gazebo simulator," in *2020 13th International Conference on Developments in eSystems Engineering (DeSE)*, IEEE, Dec. 2020, pp. 222–227. doi: 10.1109/DeSE51703.2020.9450749.

[26] L. M. Gjerde Johannessen, M. Hauan Arbo, and J. T. Gravdahl, "Robot Dynamics with URDF & CasADi," in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, Nov. 2019, pp. 1–6. doi: 10.1109/ICCMA46720.2019.8988702.

[27] Xiang Chen, M. Salem, T. Das, and Xiaoqun Chen, "Real Time Software-in-the-Loop Simulation for Control Performance Validation," *Simulation*, vol. 84, no. 8–9, pp. 457–471, Aug. 2008, doi: 10.1177/0037549708097420.