REVIEW ARTICLE

Advances in ROS-Based Robot Simulation Technology: A Mini Review

Hui Zhou[1,*]

[1] School of Electromechanical Engineering, Nanjing Xiaozhuang University, Nanjing 211171, China

Abstract

Robot simulation provides a repeatable and risk-controlled environment for developing increasingly integrated robotic systems. Within this field, the Robot Operating System (ROS) links robot models, sensors, controllers, and algorithms through modular communication interfaces and has formed a broad ecosystem with simulators such as Gazebo, Webots, and NVIDIA Isaac Sim. Recent progress has shifted from basic rigid-body and sensor simulation toward ROS 2-based interface unification, high-fidelity perception, GPU-accelerated synthetic-data generation, and Sim-to-Real transfer. This paper reviews these advances, discusses key developments in modeling, navigation, manipulation, and sensor simulation, and summarizes remaining challenges in fidelity, real-time performance, and virtual–real consistency.

Keywords: ROS, ROS 2, robot simulation, Gazebo, Sim-to-Real, virtual-real integration.

Submitted: 21 July 2026
Revised: 21 August 2026
Accepted: 03 September 2026
Published: 16 September 2026

Vol. 1, **No.** 3, 2026.
10.62762/TICPS.2026.304297

***Corresponding author:**
✉ Hui Zhou
15261498478@163.com

1 Introduction

Modern robots integrate mechanics, actuators, perception, localization, planning, and task decision-making, and such tightly coupled systems are increasingly developed and validated in simulation before deployment [1]. Directly testing unstable algorithms on physical platforms is costly, difficult to reproduce, and may cause collisions or hardware damage. Simulation therefore serves as a system-level validation environment rather than merely a substitute for physical experiments, offering repeatable, safe, and parallelizable evaluation. ROS further enables modular integration through nodes, topics, and services, allowing simulators to function as "virtual hardware" while preserving the software interfaces used by higher-level algorithms [2].

Early ROS simulation centered on ROS 1 and Gazebo Classic, mainly supporting robot modeling, rigid-body dynamics, and sensor-data generation [3]. ROS 2 introduced DDS/RTPS communication, richer QoS policies, and lifecycle management, while modern Gazebo, `ros2_control`, and GPU-based simulators expanded simulation toward distributed execution, perception, manipulation, and learning [4]. Together, these developments increasingly connect virtual validation with engineering deployment on real robots, narrowing the gap between simulated prototypes and field-ready systems.

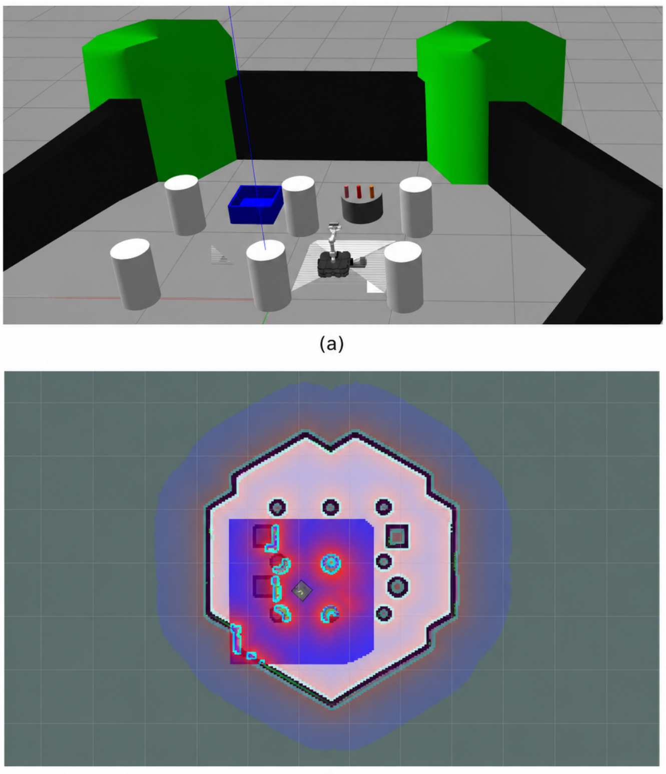
Citation
Zhou, H. (2026). Advances in ROS-Based Robot Simulation Technology: A Mini Review. *ICCK Transactions on Intelligent Cyber-Physical Systems*, 1(3), 93–96.

2 Evolution of the ROS Robot Simulation Framework

2.1 From 3D Physical Simulation to Closed-Loop ROS System Validation

A ROS simulation system typically combines a robot model, physical environment, sensor models, and control interfaces. URDF/Xacro describes links, joints, inertial properties, and collision geometry, while virtual cameras, LiDARs, and IMUs publish data as ROS messages. Mobile bases commonly receive commands through `/cmd_vel`, and manipulators execute trajectories through joint controllers. Simulation has therefore evolved from stand-alone 3D animation into a closed loop of perception, planning, control, and feedback.

Gazebo and RViz play complementary roles. Gazebo provides 3D environments, dynamics, contact, and sensor simulation, whereas RViz visualizes robot states, maps, TF frames, and navigation information. As shown in Figure 1, a mobile-manipulation environment can use Gazebo to model the robot and obstacles and RViz to display pose and costmap information, enabling navigation and motion verification before physical hardware is connected.



(a)

(b)

Figure 1. ROS robot simulation environment: (a) Gazebo 3D scene; (b) RViz navigation costmap visualization.

2.2 ROS 2 and Unified Software–Hardware Control Interfaces

ROS 2 has increased the emphasis on consistent interfaces between simulation and physical hardware. `ros2_control`, building on the control abstraction concepts of its ROS 1 predecessor [5], abstracts joints, sensors, and system hardware through standardized command and state interfaces, while the Controller Manager coordinates control execution. High-level controllers can therefore be reused with different simulation or hardware backends. Modern Gazebo, following the 2025 end-of-life of Gazebo Classic, integrates with ROS 2 through `ros_gz` packages, strengthening the separation between simulator, controller, and application layers.

3 Key Technical Advances in ROS Simulation

3.1 Sensor and High-Fidelity Environment Simulation

As robot intelligence increases, sensor simulation must reproduce more than ideal laser scans or RGB images. Visual manipulation and autonomous navigation require realistic lighting, occlusion, depth, noise, and material appearance. Webots remains effective for education and rapid prototyping because of its integrated robot and sensor models [7]. Quantitative comparisons of simulators including Gazebo, Webots, CoppeliaSim, and MORSE have shown meaningful differences in motion accuracy across platforms [6], underscoring the importance of simulator selection for specific application requirements. Isaac Sim adds GPU-accelerated rendering, synthetic RGB/depth/segmentation data, and bidirectional ROS 2 communication [8], extending simulation from control validation to perception-data generation.

3.2 Closed-Loop Validation of Navigation, Manipulation, and Mobile Manipulation

Modern ROS simulation increasingly evaluates complete robotic workflows. Mobile-robot environments can jointly test SLAM, localization, planning, costmaps, and base control, while manipulator simulation combines kinematics, collision checking, and trajectory execution. For mobile manipulators, the base and arm must additionally coordinate tasks such as approach, local alignment, and grasping. Integrated simulation can reveal coordinate-frame, topic-interface, timing, and sequencing problems that isolated subsystem tests may miss.

3.3 Sim-to-Real Transfer and Data-Driven Simulation

The remaining gap between simulation and reality has made Sim-to-Real transfer a major research topic [10]. Domain randomization varies textures, lighting, object poses, mass, friction, and related parameters to reduce dependence on a single simulated condition [9]. More recent Real-to-Sim-to-Real methods reconstruct virtual scenes from real observations and generate training data within them. Re3Sim, for example, combines 3D reconstruction with photorealistic rendering to reduce geometric and visual domain gaps for robotic manipulation [11].

4 Major Challenges and Development Trends

Three challenges remain prominent. First, parameters such as backlash, mechanical clearance, tire–ground friction, compliance, and contact dynamics are difficult to model accurately. Second, real sensors introduce exposure changes, motion blur, occlusion, noise, and latency that are often simplified in simulation. Third, simulation time and wall-clock time may differ, particularly under computationally intensive vision or physics. Reliable evaluation should therefore consider not only trajectory accuracy but also control period, communication latency, and computational load.

Future development will likely focus on three directions: standardized interfaces that allow higher-level algorithms to be retained while hardware backends are replaced; GPU-accelerated, high-fidelity, parallel simulation for learning and synthetic-data generation; and tighter integration of real-scene reconstruction, online feedback, and digital-twin techniques. For mobile manipulators, unified simulation of navigation, visual perception, and grasping will be especially important for validating whole-system closed-loop behavior before deployment.

5 Conclusion

ROS robot simulation has evolved from basic 3D motion verification into integrated validation of perception, planning, control, and learning. Modern Gazebo and `ros2_control` improve interface consistency between virtual and physical systems, while Isaac Sim extends high-fidelity perception and synthetic-data generation. Sim-to-Real research further addresses the virtual–real gap. Future progress will depend on consistent physical models, sensor characteristics, control timing, and software interfaces so that simulation can more reliably support deployment on real robotic systems.

Data Availability Statement

Not applicable.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

AI Use Statement

The author declares that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Collins, J., Chand, S., Vanderkop, A., & Howard, D. (2021). A review of physics simulators for robotic applications. *IEEE Access, 9*, 51416–51431. [CrossRef]

[2] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5). http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS.pdf

[3] Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)* (Vol. 3, pp. 2149-2154). IEEE. [CrossRef]

[4] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., & Woodall, W. (2022). Robot operating system 2: Design, architecture, and uses in the wild. *Science robotics, 7*(66), eabm6074. [CrossRef]

[5] Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Tsouroukdissian, A. R., Bohren, J., ... & Perdomo, E. F. (2017). ros_control: A generic and simple control framework for ROS. *Journal of Open Source Software, 2*(20), 456-456. [CrossRef]

[6] Farley, A., Wang, J., & Marshall, J. A. (2022). How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory, 120*, 102629. [CrossRef]

[7] Michel, O. (2004). Cyberbotics ltd. webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, *1*(1), 5. [CrossRef]

[8] NVIDIA. (2026). *Isaac Sim Documentation: ROS 2 Tutorials*. NVIDIA Corporation. Retrieved from https://docs.isaacsim.omniverse.nvidia.com/latest/ros2_tutorials/index.html

[9] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017, September). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 23-30). IEEE. [CrossRef]

[10] Zhao, W., Queralta, J. P., & Westerlund, T. (2020, December). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)* (pp. 737-744). IEEE. [CrossRef]

[11] Han, X., Yu, J., Liu, M., Chen, Y., Lyu, X., Tian, Y., ... & Pang, J. (2025). Re $^ 3$ Sim: Generating High-Fidelity Simulation Data via 3D-Photorealistic Real-to-Sim for Robotic Manipulation. *arXiv preprint arXiv:2502.08645*. [CrossRef]

Hui Zhou is with the School of Electromechanical Engineering, Nanjing Xiaozhuang University, Nanjing, China. His research interests include robot simulation, the Robot Operating System (ROS), and mobile manipulation. (Email: 15261498478@163.com)