УДК 004

АНАЛІЗ СУЧАСНИХ УНІВЕРСАЛЬНИХ 3D-СИМУЛЯТОРІВ РОБОТІВ

Поддубняк І.А.

Харківський національний університет радіоелектроніки, Харків

З розвитком галузі робототехніки поширилися й різновиди роботів, яких треба моделювати у симуляціях. Через це важливо мати знання про сучасні перевірені програмні середовища тривимірних симуляцій робототехніки – так можна уникнути таких проблем, що виникають при виборі симуляторного середовища, де обране програмне забезпечення (ПЗ) несподівано виявляється не сумісним: з метою проекту розробки, з наявними навичками розробника, з робочим апаратним забезпеченням розробника тощо [1].

Отже, метою стає аналіз сучасного ПЗ для симуляції функціонування роботів – як надає уяву огляд літератури з галузі моделювань для робототехніки, на рівні нескладного впровадження базових апаратної [2] [3] [4], програмної [5] та додаткової частин симуляції [6], симуляторні середовища не мають істотної різниці в виведенні результатів спільного вигляду [7], але на рівні розробки комплексних проектів виникають розгалуження в можливостях та обмеженнях серед симуляторних ПЗ, про які розробникам та дослідникам варто знати [8].

В таблицях 1 та 2 приведено назви та властивості програмних середовищ робототехнічних симуляцій, які використовуються для розробок [9] [10].

На основі додаткових досліджень, що використовувалися в аналізі [11] [12] [13] [14] [15], в результаті можна звести порівняльні висновки для кожного симуляційного ПЗ.

Gazebo, завдяки інтеграції структури пакету програмної розробки ROS2 (Robot Operating System 2), є найоптимальнішим для праці одночасно з модельованою та апаратною частинами однієї робототехнічної системи.

Gazebo Ignition – окрема версія Gazebo на такому модульному принципі виконання, який дозволяє легко змінювати програмні компоненти самого симуляторного ПЗ. Єдиний істотний недолік – проблеми моделювання пружності.

Таблиця 1 – Базові дані властивостей сучасних симуляційних ПЗ

Назва	Мова програмування	Операційна система	Підтримка безмоніторного (headless) функціонування
Gazebo	C, C++, Python	Linux, macOS, Windows	Вбудована
Gazebo Ignition	C++, Python	Linux, macOS, Windows	Вбудована
WeBots	C, C++, Python, Java, MATLAB	Linux, macOS, Windows	Зовнішня
Isaac Sim	C++	Linux	Вбудована
Project Chrono	C++, Python	Linux, macOS, Windows	Вбудована
PyBullet	Python	Linux, macOS, Windows	Вбудована
CoppeliaSim (V-REP)	C, C++, Python, Java, Lua, MATLAB, Octave	Linux, macOS, Windows	Вбудована
MuJoCo	C	Linux, macOS, Windows	Вбудована
ARGoS	Lua, C++	Linux, macOS, Windows	Вбудована

Таблиця 2 – Додаткові дані властивостей сучасних симуляційних ПЗ

Назва	Фізичний рушій	Відкритий доступ	Підтримка ROS2	Підтримка машинного навчання
Gazebo	Bullet, DART, ODE, Simbody	Так	Так	Зовнішня
Ignition	DART	Так	Так	Зовнішня
WeBots	ODE	Так	Так	Зовнішня
Isaac Sim	PhysX	Ні	Так	Вбудована
Project Chrono	Chrono::Engine	Так	Так	Зовнішня
PyBullet	Bullet	Так	Так	Зовнішня
CoppeliaSim (V-REP)	Bullet, Newton, ODE, Vortex Dynamics	Ні	Так	Зовнішня
MuJoCo	MuJoCo	Так	Ні	Зовнішня
ARGoS	ODE, 3D particle engine, 2D-dynamics open-source physics engine library Chipmunk, 2D-kinematics engine	Так	Ні	Вбудована

WeBots підтримує кілька мов кодування та кілька типів симулювань сенсорів, має нескладний але повнофункціональний користувацький інтерфейс, гарну оптимізацію та точність довгого симулювання. Деякі відсутні функціональності WeBots є

адресованими користувацькими доповненнями (такими як DeepBots для машинного навчання та WeBots.HPC для паралельних симуляцій у «headless» режимі).

Isaac Sim має сумісність з найсучаснішими методами та інструментами машинного навчання для роботів (через програмне розширення Isaac Gym), але працює тільки на відеокартах NVIDIA та тільки з певними стабільними версіями ОС типу Ubuntu.

Project Chrono – орієнтований на робототехніку в симульованих середовищах складних фізичних, динамічних та кінематичних процесів з вбудованою підтримкою функціоналу для елементів середовища типу піску, води та гравію. Підтримка ROS2 є можливою через зовнішнє ПЗ.

PyBullet більш підходить саме для досліджень з робототехніки та машинного навчання. PyBullet працює на клієнтській архітектурі, яка дозволяє легко виконувати паралельні симуляції, та PyBullet має функціональні можливості прямої та зворотної кінематики, середовища машинного навчання з підкріпленням, інтеграції віртуальної реальності, моделювання об'єктів з деформацією та моделювання тканини..

Комерційний CoppeliaSim може бути порівняний з більш точним та оптимізованим WeBots з відкритим кодом, але в першого є свої переваги в підтримці набору інструментів PyRep Python для навчання роботів.

MuJoCo відносно легкий в освоєнні та дозволяє виконувати швидкі розробки керувань з машинним навчанням, але не має таких функціональних можливостей, які мають існуючи середовища симуляцій з відкритим кодом.

ARGoS не є високоточним для задовільного моделювання одиничних роботів, але добре підходить для моделювання великомасштабних зграй роботів з кількома задіяними моделями роботів.

Через різноманітність у функціоналі симуляторних середовищ, що проаналізовані, існують і випадки, коли розробникам доводиться поєднувати використання кількох симуляторів для одної робототехнічної системи, але, з поточними розробками пакетів симуляторного ПЗ, такі методи можуть стати непотрібними, що може бути виявлено у подальших робіт з аналізу цієї галузі.

Література:

1. Afzal, D. S. Katz, C. L. Goues, and C. S. Timperley, “A Study on the Challenges of Using Robotics Simulators for Testing,” *arXiv:2004.07368 [cs]*, 2020.
2. K. Liu and D. Negrut, “The Role of Physics-Based Simulators in Robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 35–58, 2021.
3. J. Yoon, B. Son, and D. Lee, “Comparative Study of Physics Engines for Robot Simulation with Mechanical Interaction,” *Applied Sciences*, vol. 13, no. 2, p. 680, 2023, doi.
4. Q. L. Lidec, W. Jallet, L. Montaut, I. Laptev, C. Schmid, and J. Carpentier, “Contact Models in Robotics: a Comparative Analysis,” *arXiv.org*, 2023.
5. J. Liu, P. Borja, and C. Della Santina, “Physics-informed Neural Networks to Model and Control Robots: a Theoretical and Experimental Investigation,” *arXiv*, 2023.
6. J.-L. Blanco-Claraco, B. Tymchenko, F. J. Mañas-Alvarez, F. Cañadas-Aránega, Á. López-Gázquez, and J. C. Moreno, “MultiVehicle Simulator (MVSim): Lightweight dynamics simulator for multiagents and mobile robotics research,” *SoftwareX*, vol. 23, pp. 101443–101443, 2023.
7. Farley, J. Wang, and J. A. Marshall, “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*, vol. 120, pp. 102629–102629, 2022.
8. Phadke, F. A. Medrano, C. N. Sekharan, and T. Chu, “Designing UAV Swarm Experiments: A Simulator Selection and Experiment Design Process,” *Sensors*, vol. 23, no. 17, p. 7359, 2023.
9. F. P. Audonnet, A. Hamilton, and G. Aragon-Camarasa, “A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2,” *IEEE Xplore*, pp. 755–762, 2022.
10. Elmquist *et al.*, “A software toolkit and hardware platform for investigating and comparing robot autonomy algorithms in simulation and reality,” *arXiv (Cornell University)*, 2022.

11. Z. Chen, J. Yan, B. Ma, K. Shi, Q. Yu, and W. Yuan, “A Survey on Open-Source Simulation Platforms for Multi-Copter UAV Swarms,” *Robotics*, vol. 12, no. 2, pp. 53–53, 2023.
12. Y. Wang and H. Kasaei, “IPPO: Obstacle Avoidance for Robotic Manipulators in Joint Space via Improved Proximal Policy Optimization,” *arXiv (Cornell University)*, 2022.
13. J. Collins, S. Chand, A. Vanderkop, and D. Howard, “A Review of Physics Simulators for Robotic Applications,” *IEEE Access*, vol. 9, pp. 51416–51431, 2021.
14. M. Körber, J. Lange, S. Rediske, S. Steinmann, and R. Glück, “Comparing Popular Simulation Environments in the Scope of Robotics and Reinforcement Learning,” *arXiv:2103.04616 [cs]*, 2021.
15. M. Franchi, “Webots.HPC: A Parallel Robotics Simulation Pipeline for Autonomous Vehicles on High Performance Computing,” *arXiv (Cornell University)*, 2021.