

M. S. Kuprin¹, kms0207@ya.ru, **I. A. Osipov**¹, igoros7755@gmail.com,
A. V. Klyuchikov², krok9407@mail.ru, **N. E. Samokhin**¹, nikitasmokhin123@yandex.ru,
¹Yuri Gagarin State Technical University of Saratov, Saratov, 410054, Russian Federation,
²Saratov State University of Genetics, Biotechnology and Engineering named after N. I. Vavilov,
Saratov, 410012, Russian Federation

Corresponding author: Klyuchikov A. V., Ph.D. in Technology, Junior Researcher, Laboratory of Fundamental and Applied Research, Acting Head of the department "Digital process control in the agro-industrial complex" Saratov State University of Genetics, Biotechnology and Engineering named after N. I. Vavilov, Saratov, 410012, Russian Federation, e-mail: krok9407@mail.ru

Accepted on November 22, 2022

Simulation Modeling of Mobile Robotic Complexes Tool Analysis According to Physical Laws (a Review)

Abstract

This article is about the technologies of simulation modeling of mobile robotic complexes. The basic demands for a simulation environment are defined such as realistic results, open-source code, extensibility, performance of system, possibility of usage of low-level code for simulation. Moreover, the most significant characteristics for physics engines are described in the article, specifically solid-state physics, torque setup, stability of "axle" connection, stability of the simulated many-body system. The frequently used physics engines such as PhysX, ODE, MuJoCo, Bullet, Havok are analyzed and the search of modeling services is made in order to make the comparative table. The platform Unity is in focus of this work to show its abilities in modeling of mechanical and electronic parts of mobile robots. The aim of this work is minimizing mobile robots development outgoings. Programming product is given as result of simulation modeling of robotic complexes. The Unity platform is used as the engine for development of simulation tools for mechanical and electronic parts of robotic complex. The set of tools based on Unity engine is developed in order to create virtual models of mobile robots. The architecture of the project was developed in order to determine aspects of creating mobile robots in a simulation modeling environment. The logic of scripts for modeling the solid-state physics, kinematic chains and joints with different degrees of freedom is presented as a result of developing architecture of the project. The package was tested, a model of a mobile four-wheeled robotic platform was built as a result of the test.

Keywords: simulation modeling, computer modeling, mobile robots, physics engine, Unity

For citation:

Kuprin M. S., Osipov I. A., Klyuchikov A. V., Samokhin N. E. Simulation Modeling of Mobile Robotic Complexes Tool Analysis According to Physical Laws (a Review), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2023, vol. 24, no. 3, pp. 152–157.

DOI: 10.17587/mau.24.152-157

УДК 007.52, 004.75

DOI: 10.17587/mau.24.152-157

М. С. Куприн¹, студент, kms0207@ya.ru, **И. А. Осипов**¹, студент, igoros7755@gmail.com,
А. В. Ключиков², канд. техн. наук, мл. науч. сотр., krok9407@mail.ru,
Н. Е. Самохин¹, студент, nikitasmokhin123@yandex.ru,
¹СГТУ имени Гагарина Ю. А., г. Саратов
²ФГБОУ ВО Вавиловский университет, г. Саратов

Анализ инструментов имитационного моделирования мобильных робототехнических платформ с учетом физических законов (обзор)

Рассмотрены технологии имитационного моделирования мобильных робототехнических комплексов. Определены основные требования к среде моделирования, такие как: реалистичность результатов, открытый исходный код, расширяемость, производительность системы, возможность использовать код низкого уровня для симуляции. Выявлены наи-

более значимые для моделирования характеристики физические движки, а именно: учет физических свойств твердых тел, возможность настройки крутящего момента, стабильность соединений типа "ось", стабильность моделируемой системы многих тел. Проанализированы современные физические движки, такие как: PhysX, ODE, MuJoCo, Bullet, Havok. Выполнен поиск средств моделирования и составлена сравнительная таблица. Исследована возможность использования платформы Unity для разработки средств моделирования механических и электронных компонентов мобильных роботов. Поставлена цель минимизировать издержки на разработку мобильного робота. Поставлена задача создать программный продукт для имитационного моделирования мобильных робототехнических комплексов. Предложено использовать платформу Unity для разработки средств имитации механических и электронных компонентов робототехнической системы. Разработан набор инструментов для создания виртуальных моделей мобильных роботов с использованием Unity. Составлена схема архитектуры полученного программного пакета. Проведены испытания пакета, построена модель мобильной четырехколесной робототехнической платформы. Представлены алгоритмы работы скриптов для моделирования физики твердых тел, кинематических цепей и механических соединений с различным числом степеней свободы.

Ключевые слова: имитационное моделирование, компьютерное моделирование, мобильные роботы, физический движок, Unity

Introduction

The number of mobile robots in the world is rapidly increasing. The number of mobile robots sold will increase by 31 % annually from 2020 to 2023 according to IFR (International Federation of Robotics) predictions [1]. With the increase in the number of mobile robots, there is a need for means of testing new solutions, modeling the behavior of the prototype, checking performance and evaluating the most important parameters such as maneuverability and the ability to overcome obstacles [2]. Approbation is a resource-intensive process because of the wear of the mechanical parts of the robot, possibility of failure of electronic components and the need to purchase consumables.

Computer calculations are carried out during the development of 3D models of the housing, prototyping of electronic circuits, mathematical modeling of individual links of the robotic system (RTK) [3, 4], as well as the creation and testing of algorithms for the software [5, 6].

Simulation modeling combines all these aspects. Simulation modeling is the process of creating a virtual model, when using which the system under study is replaced by a simplified version of the object. A simulation model is a computer program that accurately describes the structure of the system. The simulation model is capable of reproducing the behavior of the system in real time [7].

Defining requirements for the simulation environment

The process of modeling mobile robots is labor-intensive as a result of the complexity of the orientation of the robot in area, navigation and overcoming obstacles. The main problems in developing simulation environments are:

1. Calculation stability [8]. Many physics engines are used in computer games for animations where accurate results are not the primary concern so accuracy is neglected in favor of performance. The PhysX physics engine shows good results in calculating collisions but optimization "send to sleep" the object;

2. Ability to use as a predictive engine in real control systems, which requires a high speed of calculations from the engine;

3. Rigid body physics and soft body physics. Inaccurate calculations lead to incorrect results;

4. Ability to simulate various types of sensors as well as adding and customizing your own [9].

The most important factors when choosing a modeling tool are:

- realism of results, closeness of simulation to reality;
- open-source code;
- the ability to use the same code for both simulation and robot control;
- extensibility of projects
- possibility of modification and development of own simulation modules;
- performance of the system [10].

Analysis and choice of physics engine

Factors affecting the realism of the simulation of the movement of a mobile robot are:

1. Rigid body physics.
2. Stability of "axis" type joint.
3. Possibility to adjust the torque.
4. Stability of the multibody simulated system [11].

The most advanced physics engines are: ODE, MuJoCo, PhysX, Havok, Bullet [12–15]. Most robotic simulators use these engines (Table 1).

General characteristics of modeling environments

Development platform	Developer	License	Graphics engine	Physics engine	3D modeller	Supported OS
Gazebo	Open-Source Robotics Foundation (OSRF)	Apache 2.0	OGRE	ODE, Bullet, Simbody, DART	Internal	Linux, MacOS X, Windows
MORSE	Academic community	BSD	Blender game engine	Bullet	Blender	Linux, BSD, MacOS X
V-Rep	Coppelia Robotics	Proprietary/ GNU GPL	Internal	ODE/Bullet/ Vortex	Internal	Linux, MacOS X, Windows
Webots	Cyberbotics	Proprietary	OGRE	Custom version of ODE	Internal	Linux, MacOS X, Windows
OpenRave	JSK Robotics Lab	GNU	Internal	Any, connected as plugin	None	Linux, Microsoft Windows, Mac OS X
USARSim	National Science Foundation	GNU	Unreal Engine2	Karma	Internal	Linux, Windows
Unity	Unity Technologies	Proprietary	Direct 3D, OpenGL	PhysX	Internal	Windows, OS X
MuJoCo	Emo Todorov	Free license	OpenGL	MuJoCo	Internal	Windows, Linux and OSX

Table 2

Technical parameters of modeling environments

Development platform	Main programming language	Supported file formats	Extensibility	API	Supported Robotic platforms	Use Interface
Gazebo	C++	SDF/URDF	Plugins (C++)	C++	ROS, Player, Sockets	GUI
MORSE	Python	Unknown	Python	Python	Sockets, YARP, ROS, Pocolibs, MOOS	Command-line
V-Rep	LUA	OBJ, STL, DXF, 3DS, Collada, URDF	API, Addons, Plugins	C/C++, Python, Java, Urbi, Matlab/Octave	Sockets, ROS	GUI
Webots	C++	WBT, VRML'97	Plugins (C++), API	C/C++, Python, Java, Matlab	ROS, URBI, NaoQI	GUI
OpenRave	C++, Python	DAE, ZAE, more with plugins	Plugins (C++, Python), API	C++, Python, Matlab/Octave	CARMEN, ROS, Player	GUI
USARSiM	C++, UnrealScript	UMAP, ASC, ASE, DFX, T3D	Add-ons, Plugins	Pyro	MOAST, ROS, Player	GUI
Unity	C#, JavaScript	3DS, MAX, OBJ, FBX, DAE	Plugins, API	—	Socket	GUI
MuJoCo	MJCF	MJCF, URDF	API, Plugins	C/C++	Socket	GUI

PhysX and Havok were developed for games, while they can be used with optimization adjusted and are available for use in the game development system (Unity) [8, 9, 14–16] or as the additional library. The technical characteristics analysis of the simulation environments was made. Data was presented in the comparative Table 2.

Using the Unity game engine for modeling mobile robots

Unity is a development environment for 2D and 3D games and applications. Unity Editor is a multifunctional tool for working with graphics, physics, data, sounds, animations, user interfaces and pro-

gram code [17]. A component-oriented approach is used for development, which means that the properties of an object on the Unity scene are determined by the components that are attached to the object [18]. The standard component library allows you to create a variety of events and objects, but restricts user actions. Unity allows the user to develop unique components—scripts using the C# and JavaScript programming languages [19]. Unity supports additional libraries and plugins that extend the functionality of the development environment [20].

The use of Unity as a tool for modeling the functioning of robots is considered.

The main reasons to use Unity are:

- PhysX physics engine;
- possibility to use Havok physics engine;
- Windows and Linux support;
- C# and JavaScript support;
- clear documentation;
- wide range of file formats support;
- intuitive GUI with flexible settings;
- plugins support and API;
- environmental modeling tools.

Setting the goal and objectives of the study

The goal of study is to analyze simulation modeling tools for robotic systems that correspond to the identified parameters. This will allow developing a software package for designing a simulation model of a mobile robot, taking into account the physical, software and hardware features of the product.

Simulation modeling makes it possible to predict with high accuracy the behavior of a real robotic system using a virtual model.

The process of designing a virtual model takes less time and does not require the cost of physical resources (fasteners, wiring, parts). It is also easy to

edit, requires less production and research into new configurations, algorithms, and increases the information content during designing and testing [21, 22].

Objectives:

- define the most important parameters for modeling mobile robots;
- determine the degree of compliance of Unity with the identified parameters;
- implement close to realistic behavior of mechanics and electronics by using Unity;
- evaluate the possibilities of using the package;
- develop a mobile robot model.

Proposed solution

To reach the goals of study a package tools (Asset) was chosen for data presentation. Development in the Asset format will allow adding 3D models, scripts, materials and images to the database and use it to develop your own application software.

It is necessary to develop the behavior of mechanical joints, develop a basic set of mechanical joints and implement the behavior of electronics and microcontrollers in order to simulate the behavior of a mechatronic platform.

Component-oriented programming was used in order to develop a package of simulation modeling tools. Special script components allow to simulate the behavior of individual elements of a robotic system with the use of the Unity tools [23]. The developed components allow to simulate:

- electronic part of robot:
 - microcontrollers (make microcontrollers programmable, simulate GPIO interface, implement I²C protocol);
 - motor drivers (motor control for generating a control signal);

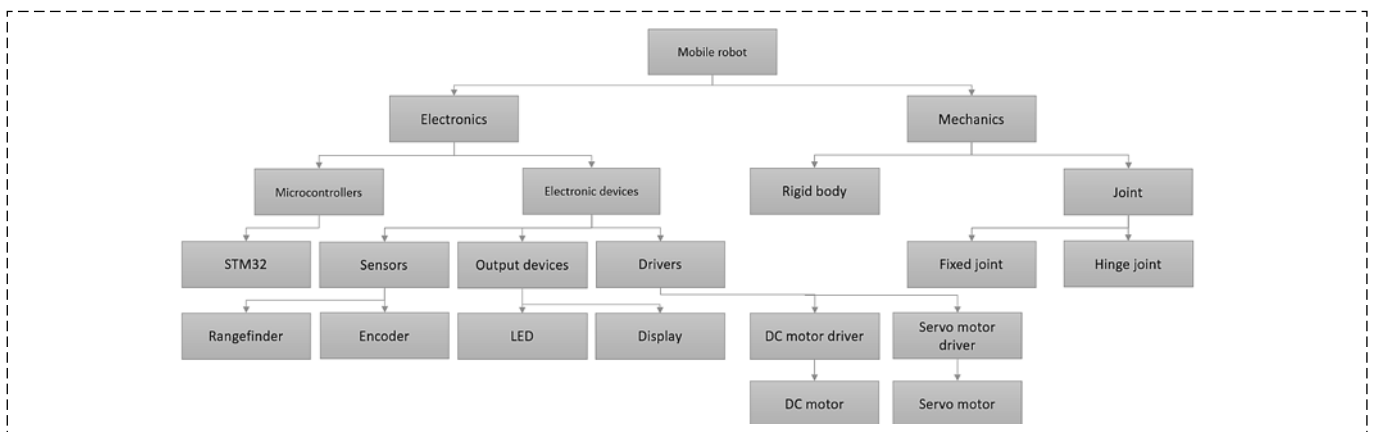


Fig. 1. Architecture diagram of a software package for designing simulation models of mobile robots

- electric motor (creation of torque in the presence of a control signal);
- encoders (detection of changes in the rotation angle of an object);
- rangefinders (distance measurement);
- LEDs (switching on/off in the presence/absence of a control signal);
- text displays (displaying text received from I²C protocol).
- mechanical part of robot [24–27]:
- rigid body physics (robot body elements);
- fixed joint (fixed connection of robot structural elements);
- axis (a movable connection of the robot's structural elements that provides rotation around the axis).

All components have been designed with the possibility of flexible configuration and extension, so the user of the package has the opportunity to develop his own modules for the robot. The architecture of the resulting simulation system is shown in Fig. 1.

The package allows you to simulate mobile robots and test programs for microcontrollers at current state. The mobile robot consisting of the four-wheeled platform, the microcontroller, the display, the rangefinder and the servo drive and four drivers was developed to test the software package in Fig. 2 (see the 3rd side of the cover).

Simulation of mechanical connections and physics of solids

There are special components for working with physics in Unity:

- Rigidbody — Rigid body physics [28];
- Collider — a physical model of an object for collision calculations [29];
- Joins — connections with different number of degrees of freedom [30];
- Articulations — set of articulation bodies [31].

To simplify the configuration of components and automatic attachment to the object, the scripts "ModuleBaseScript" and "HingeBaseScript" were developed, which create and configure a fixed joint and a hinge joint, respectively. When starting the scene, the connected elements are checked for the presence of Rigidbody component, if there is no component, it is automatically attached with the RequireComponent attribute [32].

Conclusion

A search was made for tools for mathematical modeling of the robot behavior in the course of the

study. Tools with extensions for working with physics were identified among them. Software solutions with the ability to visualize the results are also defined.

Taking into account these aspects, Unity development environment was chosen for the simulation of mobile robots. The software allows you to develop your own package with a set of tools for a specific task.

The goal was set to reduce production costs using simulation modeling of mobile robots as a part of the study. The task was set to form the Asset library in Unity development environment with the possibility of developing their own solutions and modeling the designs of mobile robots. Ten modules were developed with the use of Unity to simulate the electronic and mechanical components of a robotic system. The package has been tested. Problems were identified in the process of testing and ways to solve these problems were proposed.

References

1. International Federation of Robotics. "Mobile Robots Revolutionize Industry", available at: <https://ifr.org/ifr-press-releases/news/mobile-robots-revolutionize-industry> (accessed: 23.07.2022).
2. Gonzalez Rodriguez A. G., Gonzalez Rodriguez A. Mobile Robots, *Advanced Mechanics in Robotic Systems*. Springer, 2011, pp. 41–57.
3. Aarnio P., Koskinen K., Salmi S. Simulation of the hybtor robot. International conference climbing walking robots, *Professional Engineering Publishing*, 2000, pp. 267–274.
4. Rose A. Kinematic Design, *Engineering Haptic Devices*, Kern T. (eds), Springer, 2009, pp. 165–190.
5. Germann D., Hiller M., Schramm D. Design and control of the quadruped walking robot ALDURO, *Int Symp Autom Robot Constr ISARC*, 2005.
6. Berkaev A. R., Nenashev A. A., Klyuchikov A. V. Development of localization system and positioning of a mobile robot, *Mathematical methods in engineering and technology. MMTT*, 2020, vol. 12–3, pp. 152–157 (in Russian).
7. Swamidass P. M. SIMULATION MODELS, *Encyclopedia of Production and Manufacturing Management*, Springer, 2000, pp. 699–700.
8. Drumwright E., Shell D. An evaluation of methods for modeling contact in multibody simulation, *IEEE Int. Conf. on Robotics and Automation*, 2011, pp. 1695–1701.
9. Hummel J. et al. An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations, *Lecture Notes in Computer Science. International Symposium on Visual Computing*, 2012, pp. 346–357.
10. Leonova N. L. Simulation modeling: lecture synopsis, Higher School of Technology and Energetics, SPbGTURP — St. Petersburg, 2015, 94 p. (in Russian).
11. Seugling A., Rolin M. Evaluation of physics engines and implementation of a physics module in a 3d-authoring tool, Ume a University, Department of Computing Science, 2006, 89 p.
12. Manyuxina A. N. Game engine usage for robotic devices modeling systems development, *Conference "Integraciya mirovyh nauchnyh protsessov kak osnova obshchennogo progressa"*, 2015, pp. 131–137 (in Russian).

13. **Todorov E.** Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo, *IEEE Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6054–6061.

14. **Boeing A., Brunl T.** Evaluation of real-time physics simulation systems, ACM Press, 2007, p. 281.

15. **Erez T., Tassa Y., Todorov E.** Simulation Tools for Model-based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX, *IEEE Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 4397–4404,

16. **Tassa Y., Todorov E.** IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015.

17. **Unity** — Unity User Manual 2021.3 (LTS) // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/index.html> (accessed 07.09.2022).

18. **Unity** — Introduction to components // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/Components.html> (accessed 07.09.2022).

19. **Unity** — Scripting // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/ScriptingSection.html> (accessed 07.09.2022).

20. **Unity** — Plug-ins // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/Plugins.html> (accessed 07.09.2022).

21. **Unity** — Scripting API: RequireComponent // Unity Scripting API 2021.3 (LTS), available at: <https://docs.unity3d.com/ScriptReference/RequireComponent.html> (accessed 07.09.2022).

22. **Hummel J., Wolff R., Stein T.** et al. Advances in Visual Computing, Lecture Notes in Computer Science, 2012, vol. 7432, pp. 346–357.

23. **Ivaldi S., Padois V., Nori F.** Tools for dynamics simulation of robots: a survey based on user feedback, *14th IEEE-*

RAS International Conference on Robotics and Automation, 2014, pp. 842–849.

24. **Izyumov A. A., Vinnik A. E., Habibulina N. Y.** Mobile robot movement simulation // TUSUR confederation digest. 2020. P. 83–86 (in Russian).

25. **Mokaeva A. A., Hamukov Y. H., Shautsukova L. Z.** Kinematic scheme development of cable control system of "trunk" manipulator device, *Izvestiya Kabardino-Balkarsky Scientific Center Russian Academy of Sciences*, 2015, pp. 15–22 (in Russian).

26. **Afanasiev B. A.** et al. Design of all-wheel drive wheeled vehicles. Vol. 1, Moscow, Publishing house of MSTU. 2008, pp. 25–56, 144–182, 435–485 (in Russian).

27. **Afanasiev B. A.** et al. Design of all-wheel drive wheeled vehicles. Vol. 2, Moscow, Publishing house of MSTU, 2008, pp. 152–490 (in Russian).

28. **Unity** — Manual: Rigidbody // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/class-Rigidbody.html> (accessed 06.09.2022).

29. **Unity** — Manual: Introduction to collision// Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/CollidersOverview.html> (accessed 06.09.2022).

30. **Unity** — Manual: Introduction to joints // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/Joints.html> (accessed 06.09.2022).

31. **Unity** — Manual: Introduction to physics articulations // Unity User Manual 2021.3 (LTS), available at: <https://docs.unity3d.com/Manual/physics-articulations.html> (accessed 06.09.2022).

32. **Unity** — Scripting API: RequireComponent // Unity Scripting API 2021.3 (LTS), available at: <https://docs.unity3d.com/ScriptReference/RequireComponent.html> (accessed 07.09.2022).



29–31 мая 2023 г.
г. Санкт-Петербург, Россия
АО "Концерн "ЦНИИ "Электроприбор"



XXX Санкт-Петербургская Международная конференция по интегрированным навигационным системам

Тематика конференции

- Инерциальные датчики, системы навигации и ориентации
- Интегрированные системы навигации и управления движением
- Глобальные навигационные спутниковые системы
- Средства гравиметрической поддержки навигации

В рамках каждого направления рассматриваются:

- схемы построения и конструктивные особенности;
- методы и алгоритмы;
- особенности разработки и применения для различных подвижных объектов и условий движения (аэрокосмические, морские, наземные, подземные);
- испытания и метрология.

Контактная информация:

Тел.: +7 (812) 499 82 10 +7 (812) 499 81 57

Факс: +7 (812) 232 33 76 E-mail: icins@eprib.ru