UDC 621.8

Review and analysis of software simulators for robotic information systems

Sofia Efremova [1], *Dmitry Mishchuk* [2], *Evgeny Horbatyuk* [3]

[1,2,3]Kyiv National University of Construction and Architecture,
31 Air Force Avenue, Kyiv, Ukraine, 03037,
[1]yefremova_so-2022@knuba.edu.ua,
[2]mischuk.do@knuba.edu.ua, https://orcid.org/0000-0002-8263-9400,
[3]gek_gor@i.ua, https://orcid.org/0000-0002-8148-5323

Received: 18.03.2024; Accepted: 18.04.2024
https://doi.org/10.32347/gbdmm.2024.103.0501

Abstract. The rapid development of educational technologies requires the active involvement of new means and effective teaching methods. One of the innovative tools of learning in the field of robotics is the use of robot simulators in the educational process, which open up many opportunities for better assimilation of knowledge and development of practical skills among students.

Robot simulators are software complexes that allow you to visualize and simulate the behavior of real robots. With these tools, students can:

- program robots in a virtual environment, test code and debug without the risk of damaging real equipment;
- to study robotics and the principles of robot operation, exploring different types of sensors, actuators and algorithms of robot behavior;
- to solve problems in various fields, such as production automation, logistics, construction, etc.;
- practices your skills in a safe and controlled environment.

Among the benefits of using robotic simulators are next:

- availability for a large number of students, as simulators are much cheaper than real jobs;
- safety, as the virtual environment eliminates the risks associated with working with real robots;
- flexibility that allows students to work at their own pace, explore different scenarios and experiment with different parameters;
- interactivity with a realistic 3D-environment and visualization.

This work presents an overview and analysis of software that allows modeling robot control systems and simulating their operation for various algorithms.

A review and analysis of the proposed software tools for simulating robotic systems, which are used to teach bachelor's and master's degree students in the specialties of "Computer Science", "Information Systems", "Industrial Mechanical Engineering" and "Applied Mechanics" and have proven their effectiveness and effectiveness in distance learning at the Kyiv National University of Construction and Architecture.

Keywords: robotic simulators, robotic information system, Arduino simulators, visual simulators, wokwi, tinkercad, Gazebo, Webots, CoppelisSim.

INTRODUCTION

Robotics is a complex science that combines knowledge of mechanics, electronics, programming, artificial intelligence and other fields. Robotics aims to create robots that can perform various tasks, imitate human actions or explore dangerous and unusual environments [1-2].

Industrial robotics are using in automate production processes on factories and warehouses. Service robotics are using to perform tasks in the field of service, such as cleaning, delivering food, assisting people with disabilities. Medical robotics are useful for surgical operations, rehabilitation, diagnostics and other medical procedures. Space robotics are using for space exploration, repair of satellites and other space vehicles [2-3].

The main advantages of using robots are [4-6]:

- productivity improvement (robots can perform tasks faster, more accurately and longer than humans);
- reduction of risks to human health (robots can perform dangerous or harmful tasks);
- improving the quality of human life (robots can help people with disabilities, as well as the elderly);
- providing new opportunities for the implementation of economic and research activities (robots can explore new environments, such as space or deep-sea spaces, and perform tasks that are impossible for humans).

The main trends in the development of robotics are [7-9]:

- high-precision modeling and simulation of robots;
- optimization of dynamic robot systems and movement modes;
- make of synthetic data for artificial intelligence models;
- cloud computing;
- development of realistic and fast models of the physical world;
- development and standardization of frameworks and software libraries;
- development of management programs based on natural language;
- development of technologies for human-robot interaction;
- a combination of robots of different types;
- development of humanoid robots;
- space exploration with the help of robots.

The robotics science combines such disciplines as theoretical mechanics, the theory of mechanisms and machines, electrical engineering, microprocessor engineering, computer science, higher mathematics, theory and systems of automatic control, computer science, and theory of algorithms. Such a significant set of knowledge from various disciplines makes it much more difficult to train quality personnel [9-11].

Also, all this is superimposed with the problems of practical training, when each applicant must independently complete tasks and laboratory work at personal workplaces. Working in groups on collective projects has revealed some problems that limit its effectiveness, in particular, it is not an even distribution of responsibilities among applicants who perform their practical tasks collectively on the same laboratory stand or model. The solution to this problem can be a personal workplace with a set of tools for developing and programming robots and simulating their behavior [10, 11].

A set of software complexes and tools significantly simplifies the possibilities of high quality and fast training of specialists in the field of robotics.

The Kyiv National University of Construction and Architecture has developed a number of training courses in the field of robotic systems, in particular, such training courses (educational components) was offer next:

- robots and manipulators;
- design and construction of robotic systems;
- programming of information robotic systems;
- synthesis of robotic systems in machine building.

All courses are offer as optional educational components for higher education applicants at bachelor and master's level and cover the following areas.

The course "Robots and manipulators" intend for a general introduction to the field of robotics through the prism of robot mechanics. The focus of this course is on the study of kinematics and dynamics of robots and manipulators, the study of hydraulic, pneumatic and electric drive mechanisms and programming tools, both the components of robot mechanisms and robots as a whole on general templates.

The course "Design and construction of robotic systems" is designed to master the abilities and skills of the synthesis of robotic systems, in particular, the course covers the synthesis of kinematic open chains of the manipulator, the construction and calculation of motion transmission mechanisms, in particular, the creation of chain, belt, screw and gear gears, planetary and wave reducers.

The course "Programming of information robotic systems" is design to provide skills and knowledge in the development of software for managing information robotic systems. The main attention is pay to methods of communication with robot devices, collecting and pro-

ISSN(online)2709-6149. Mining, constructional, road and melioration machines, 103, 2024, 71-85

cessing information from robot sensors, and developing robot control programs.

The course "Synthesis of robotic systems in mechanical engineering" is intend for the training of masters-pedagogues in the direction of robotics and includes sections on kinematics, dynamics, structural synthesis of the constituent elements of robots and programming of robotic systems.

All considered courses in robotics use software complexes for robot simulation, which significantly automates the acquisition of the necessary skills and knowledge.

PURPOSE OF THE ARTICLE

Analyze software complexes for modeling, programming and simulating robotic systems that can be used for training in distance learning conditions.

PRESENTING MAIN MATERIAL

It known that any robotic system can be present architecturally in the form of three main systems: mechanical, information and control system. In this context, the definition of a mechanical system means electromechanical, hydro mechanical, pneumatic, and their combination. It is clear that these systems have significant structural differences, but in principle, they perform similar work. In this way, the functional scheme of any robot can be present in the form of basic structural blocks that are interconnect (Fig. 1). At the same time, the information and measurement system and the control system of the robot can be influence by the external environment, which forms random signals, and therefore quite often the external environment is also introduced into the robot structure as a separate functional element.

The mechanical system of the robot ensures the actual processing of the control program for all degrees of mobility and consists of an executive system, a drive and a working body. The executive system with the drive (effectors) performs all the driving functions of the robot, with the help of which movements are realize according to degrees of mobility. The relative movements of the mechanical robot system divide into: orientation (local), transport (regional) and coordinate (global). Orientation movements provide the working body of the robot with the necessary orientation at a given point in the working area. Transportation movements are intend for moving the working body within the working space. Coordinate movements ensure the movement of the robot between individual production positions. Basic movements include all movements of the mechanical system without taking into account capture, orienting and additional movements. CAD systems such as SolidWorks, FreeCAD or similar can be used to model the robot's mechanical system [10, 13].

SolidWorks and FreeCAD allow you to create mechanical components of robots with specified physical and mechanical properties and contain a significant set of data formats for saving and transferring electronic models to other software products. SolidWorks and FreeCAD contain parametric modeling modes and allow you to simulate the behavior of developed information models in the mode of movement along given trajectories. This usually requires the use of additional modules, for

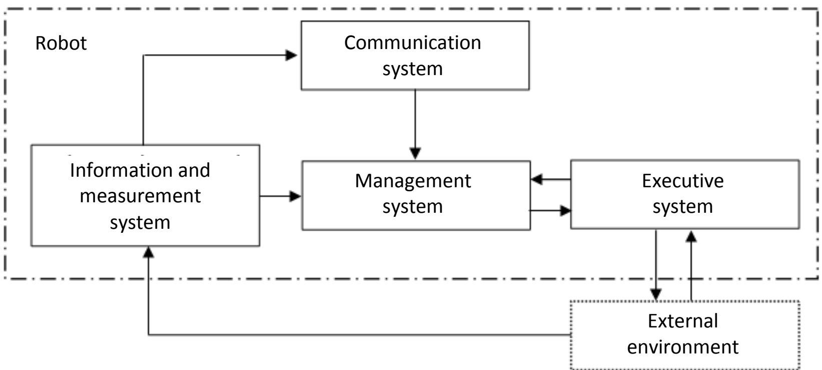


Fig. 1. The robotic functional scheme [12]

example, for robots it is the RoboDK Plug-in or RoboWorks. You can also create your own programs in SolidWorks through the built-in api [14, 15].

Fig. 2 shows the interface and the main window of the RoboDK program, which shows the appearance of the robot control interface and an example of a manipulator simulation model. The disadvantage of such software for its use in education is distribution on a paid basis, as well as the use of an abstract controller to control the robot, which may not correspond to a real controller.

SolidWorks is also a paid software environment, which is not always suitable for university studies. At the same time, electronic models created in SolidWorks can be using in other robot simulators by importing these models into them. The SolidWorks program using by the Kawasaki Company to simulate the behavior of industrial manipulator robots in its own products with the RoboWorks library. This application allows engineers to modeling, debugging and virtually testing robotic systems, including manipulator movements, workflow optimization and analysis of robot interactions with the environment.

FreeCAD is a free and open source software for modeling robot components. With the Robot FreeCAD library, it is can be model the behavior of some manipulator robots. FreeCAD has support for python applications through its own APIs. FreeCAD includes a fairly abstract controller for modeling the simulation of robot behavior. This is its drawback.

To programing and simulate robots, it is necessary to be able to work with real devices of the information and measurement system of the robot - sensors.

"Sensorics" is a system of interconnected artificial sense organs (sensors or sensors) that are designed to perceive and transform information about the state of the external environment and the state of the robot itself. Based on this statement, it is necessary to take into account that simulation and modeling systems allow to take into account similar relationships and factors that are present in the works.

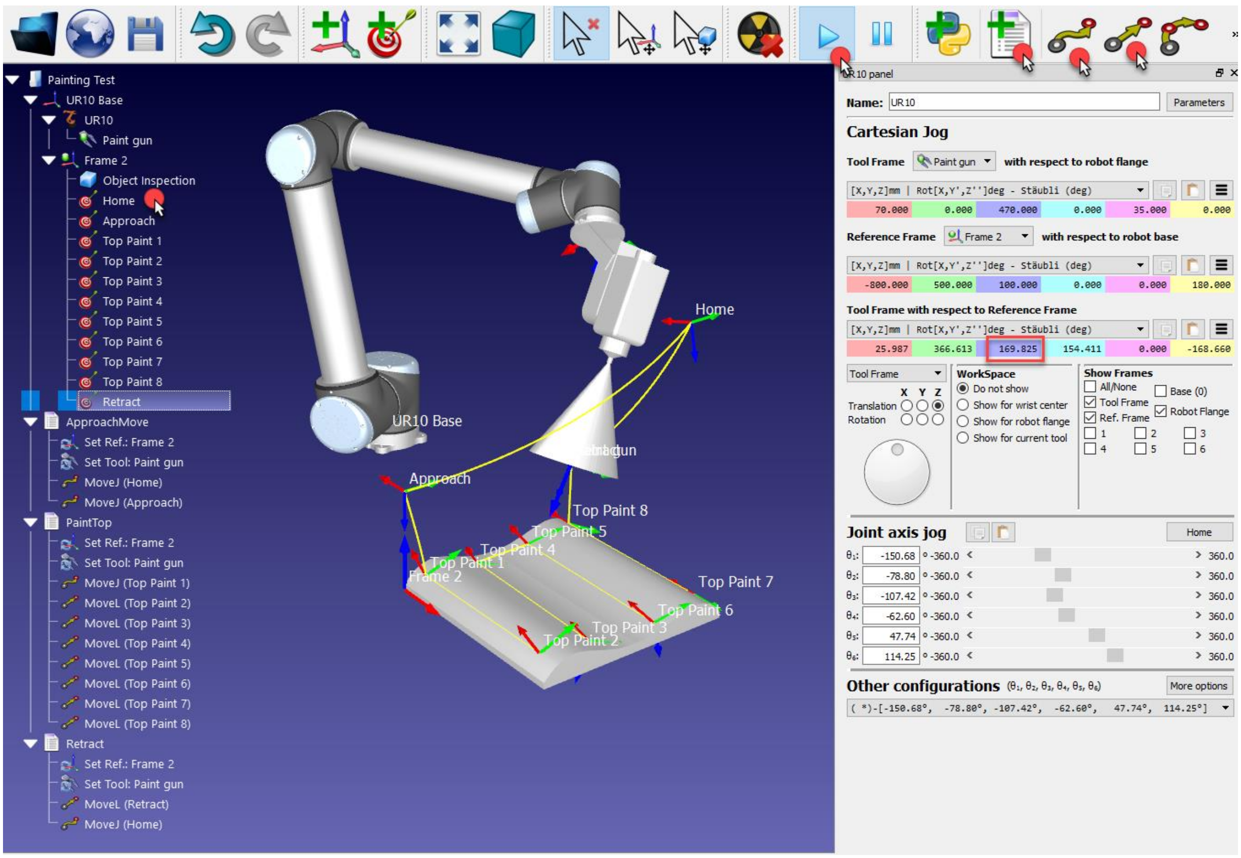


Fig. 2. The main view of the RoboDK program [15]

Arduino is a project development platform that allows you to quickly and easily experiment with electronics and, in particular, learn how to work with robot sensors. One of the key features of Arduino is the ability to work with both analog and digital signals, allowing you to interact with real analog sensors and control numerically controlled devices. Online and offline platforms can be used as a simulator of Arduino boards.

Among the online platforms for programming Arduino and simulation of sensors of robots, the following services have proven themselves well https://wokwi.com and https://www.tinker-cad.com (see Fig. 3). These services are free of charge. The advantage of such services is that the teacher can develop basic projects in advance, which will later be improve individually by each applicant. For example, on Fig. 3, and an example of work with a stepper motor, an A4988 driver and a potentiometer sensor with a mechanical discrete switch in the Wokwi system is show. On Fig. 3 shows how Tinkercad can work with the HC-SR04 ultrasonic distance sensor. Each of the specified simulators has a built-in library of electronic components, such as a potentiometer, photoresist, resistor, LED, servomotor, breadboard, etc. Both simulators support work with the Arduino port monitor and contain internal APIs for connecting third-party libraries such as Servo, Stepper, etc.

To work with electronic components in the Wokwi and Tinkercad simulators, you need to select the desired element from the library of digital duplicates of electronic devices and insert it into the working window of the program. Next, the select components are connecting according to the rules of electrical engineering and physics. The program code record in the corresponding work area and use only for boards with a microcontroller, and such elements as resistor, potentiometer, diode, electric motor, etc. have their own physics programmed by the developers [7, 9, 10].

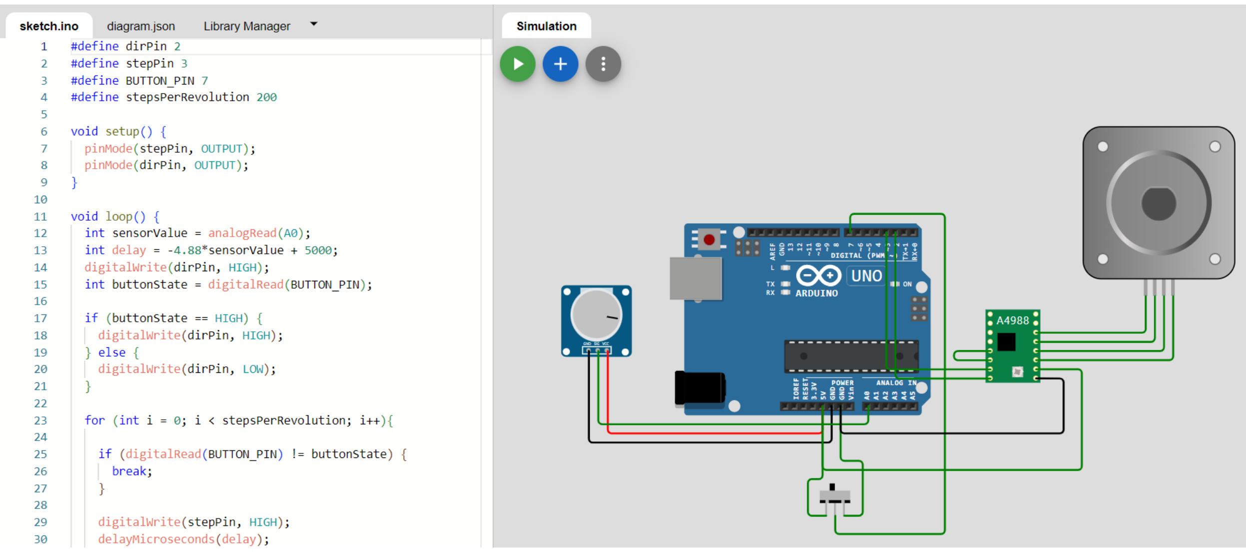
The code built in these emulators is identical to the real code that will be upload to the physical Arduino board. When studying control programs in these simulators, you can use the C programming approach or use the special Wiring library for the Arduino platform.

The C code fragment have been show below:

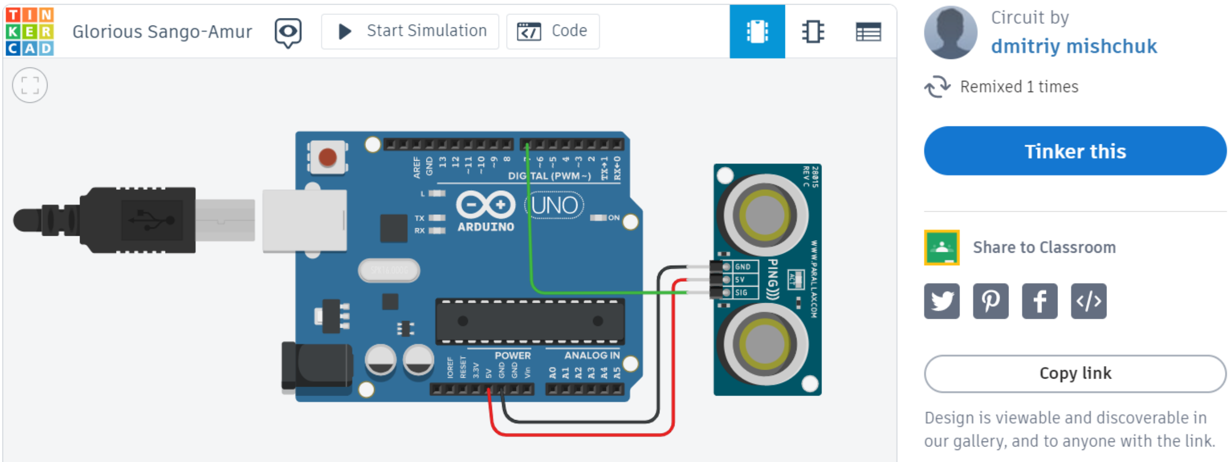
```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
int main (void) {
    DDRB=0b00100000;
    while (1) {
      PORTB = 0b00100000;
      _delay_ms(1000);
      PORTB = 0b00000000;
     _delay_ms(1000);
    }
}
```

For example to check the operation of the HC-SR04 sensor, download the following program "Ultrasonic_Ranging", which is built on the foundations of the wiring library:

```
#define ECHO   2
#define TRIG   3
int Distance = 0;
int Distance_test() {
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  float Fdistance = pulseIn(ECHO, HIGH);
  Fdistance= Fdistance/58;
  return (int)Fdistance;
}
void setup() {
  Serial.begin(115200);
  pinMode(ECHO, INPUT);
  pinMode(TRIG, OUTPUT);
}
void loop() {
  Distance = Distance_test();
  if((2 < Distance) && (Distance < 400)) {
    Serial.print("Distance = ");
    Serial.print(Distance);
    Serial.println("cm");
  }
  else  {
    Serial.println("!!! Out of range");
  }
  delay(250);
}
```



a



b

Fig. 3. Online service Wokwi (*a*) and Tinkercad (*b*)

Among the offline tools for learning Arduino, a free software product is recommend for use open source UnoArduSim. This software environment must be download from the official website of the developer and run the corresponding executable file. The interface of the UnoArduSim program is show on Fig. 4.

In appearance, the program interface is much simpler than Wokwi or ThinkerCAD, but it has similar functionality. To configure the hardware that will be used to work with the Arduino and configure the I/O ports, it is necessary to go to the section configure → I/O Devices. There it will need to select the desired devices, their number and the Arduino contacts to which these devices will be connect (see Fig. 5). UnoArduSim allows to work with 20 devices such as button, piezo, LED, segment LED, switch, servo motor, DC motor, stepper motor, pulse generator, LCD display. Devices such as photoresistor, encoder and potentiometer unlike Wokwi or ThinkerCAD are missing in UnoArduSim. However, there are special components such as a switched resistor and an analog slider that will allow you to simulate the behavior of digital and analog sensors [17].

When working with a switched resistor, it is necessary:

1. to move the "resistor switch" component from the component panel to the work area;

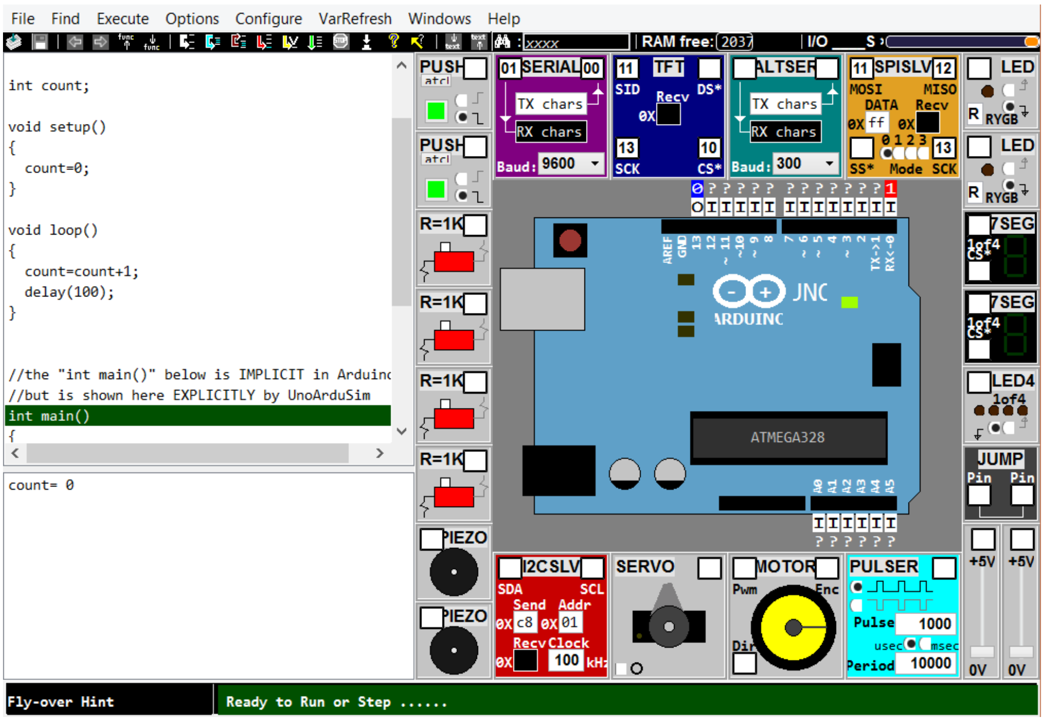


Fig. 4. UnoArduSim V2.9.2 simulator interface

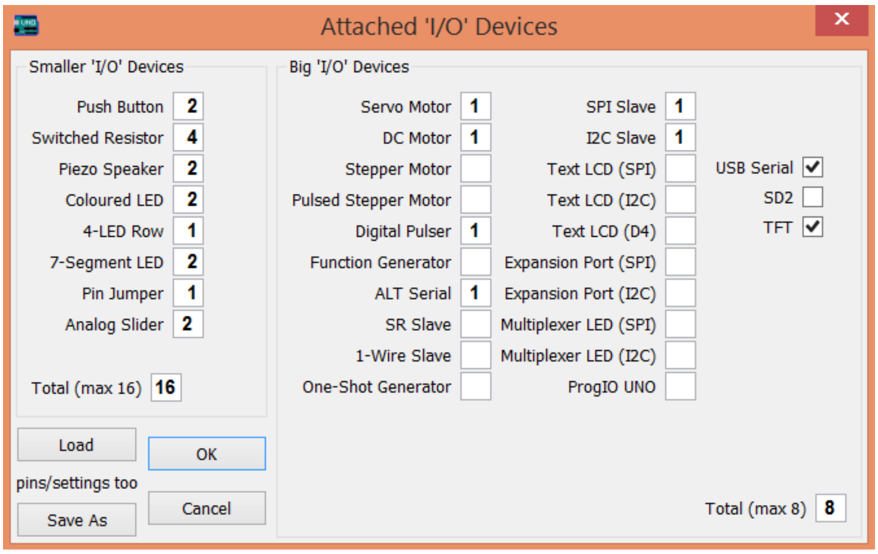


Fig. 5. Interface for select I/O Devices

2. change the resistor rating depending on the power of the selected circuit;
3. select the type of switch: SPDT - a single-pole double-position switch that has two positions or DPDT - a double-pole two-position switch that has four positions.

The switched resistor works with discrete high and low level signals.

The analog slider in UnoArduSim is a virtual component used to simulate the operation of a real potentiometer. With the help of the slider, you can change the value of the analog signal supplied to other components of the circuit.

A switched resistor can be using to simulate the operation of a light switch, where one resistor corresponds to the "on" state, and the other to the "off" state.

An analog slider can be used to simulate a volume control, where the value of the signal supplied to the amplifier corresponds to the volume of the sound.

Along with the Arduino simulator, the developers of UnoArduSim also provide an Arduino based wheeled robot simulator called Q2WDBotSim. It is a free and open source robot simulator designed for educational and research purposes that allows users to visualize, simulate and test the behavior of robots in a virtual environment. The general view of the interface of the Q2WDBotSim program is show on Fig. 6.

In Q2WDBotSim, for the settings of the controller contacts that will be connected to the drive motors and the PWM contacts that will be used to adjust the speed of the robot, you need to go to the menu configure → Wire up Pins (Fig. 7). In the Uno Pin Assignment settings menu, it will be need to set the pins that will using for the connection.

For example, the following code can be using to move the robot forward:

```
const int leftMotorDirectionPin = 4;
const int rightMotorDirectionPin = 7;
const int leftMotorPWMPin = 5;
const int rightMotorPWMPin = 6;
int count;
int motorSpeed;
void setup(){
   count = 0;
   motorSpeed = 150;
}
void loop(){
   move();
   count = count + 1;
   delay(100);
   stop();
   delay(500);
}
void move(){
 digitalWrite(leftMotorDirectionPin, HIGH);
 digitalWrite(rightMotorDirectionPin, HIGH);
 analogWrite(leftMotorPWMPin, motor-
Speed);
  analogWrite(rightMotorPWMPin, motor-
Speed);
}
void stop(){
  digitalWrite(leftMotorDirectionPin, LOW);
  digitalWrite(rightMotorDirectionPin, LOW);
}
```

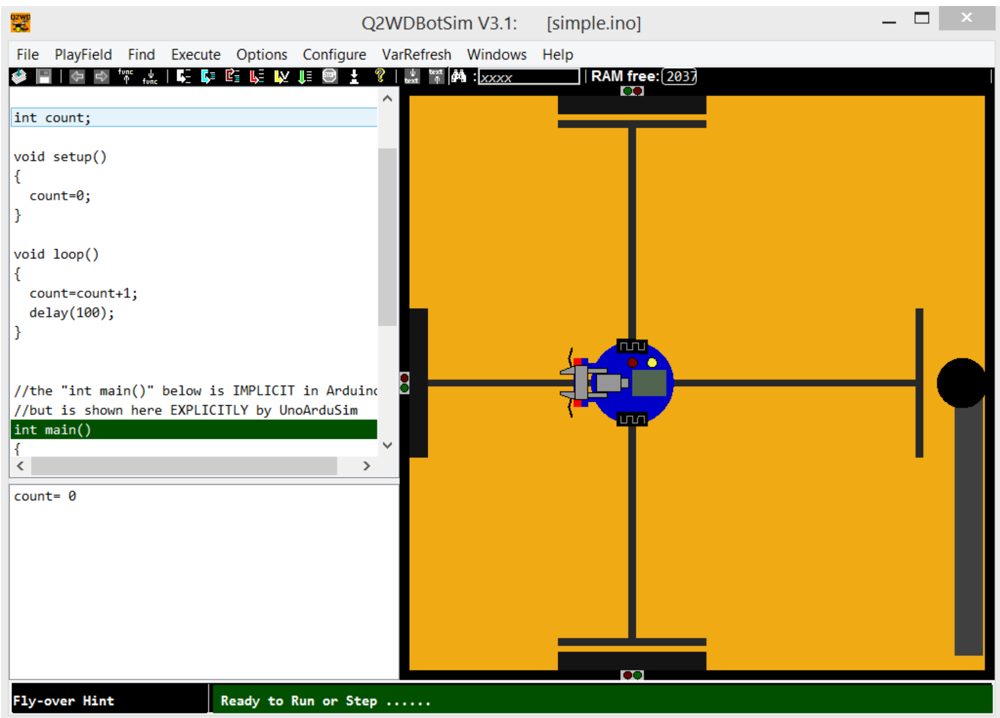


Fig. 6. Q2WDBotSim interface

ISSN(online)2709-6149. Mining, constructional, road and melioration machines, 103, 2024, 71-85

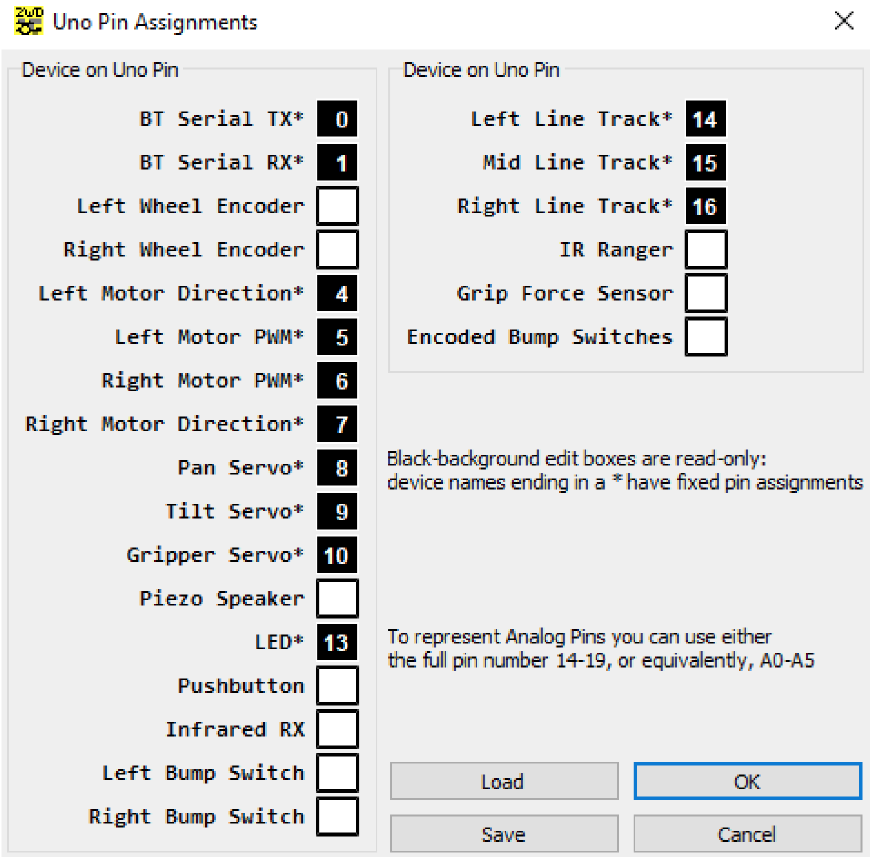


Fig. 7. Interface for select I/O pins

In the Q2WDBotSim simulator, you can simulate the imperfection of motors due to the deviation of their speeds in the Wheel Speed Mistmatch menu. In this way, it is possible to simulate algorithms with adjustment of the speeds of the left and right wheels of the robot and adjustment of its longitudinal stability. In addition, encoders, distance sensors and line detection sensors can be work with in this simulator.

CoppeliaSim is an offline physics simulation software for robotic systems developed by Coppelia Robotics. CoppeliaSim includes a free student version of EDU. CoppeliaSim allows you to simulate dynamic systems, sensors and environments in which robots work and can be using for debugging, testing and visualization of robot control algorithms, as well as for research in the field of autonomous systems and artificial intelligence [18, 19].

CoppeliaSim has a wide range of built-in robot models of various types and configurations, including manipulators, mobile robots, drones, robots with soft bodies, etc. (Fig. 8). You can also create your own models using the built-in modeling tools [20].

CoppeliaSim can simulate a variety of environments, including laboratories, factories, manufacturing facilities, city streets, and more. This allows testing robots in realistic conditions and studying their interaction with the environment [20].

CoppeliaSim supports a wide range of sensors such as cameras, distance sensors, laser sensors, gyroscopes and accelerometers, as well as different types of actuators including motors, servos and pneumatic actuators.

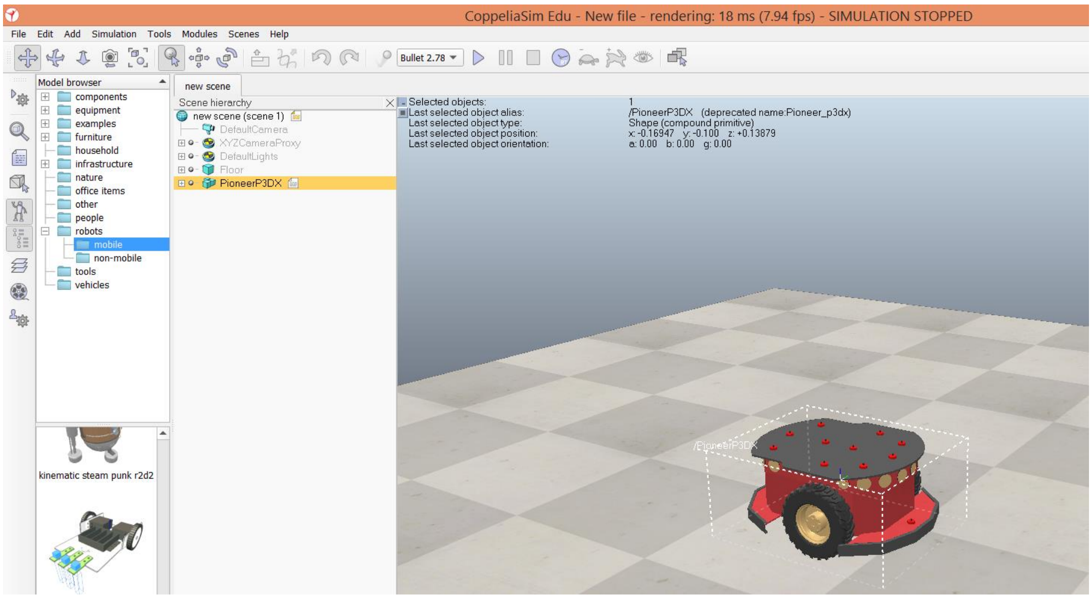


Fig. 8. CoppeliaSim interface [19]

CoppeliaSim has integration with MATLAB and Python, which allows you to develop and test control algorithms and artificial intelligence directly in the simulation environment. In CoppeliaSim, you can create displays of robot movement, graphs of sensors and actuators, as well as analysis of system behavior in real time.

Webots is a free offline software for simulating robotic systems from the Cyberbotics company, which is designed for modeling, testing, and visualizing robots in various environments and conditions (Fig. 9). Webots [21] allows you to simulate different environments, including indoor and outdoor spaces, landscapes, buildings, roads and other objects. In this simulator, you can adjust the parameters of the environment, such as lighting, fog, time of day, and take into account the physical properties of the simulation objects, such as mass, shape, dynamics of movement, forces and pressure.

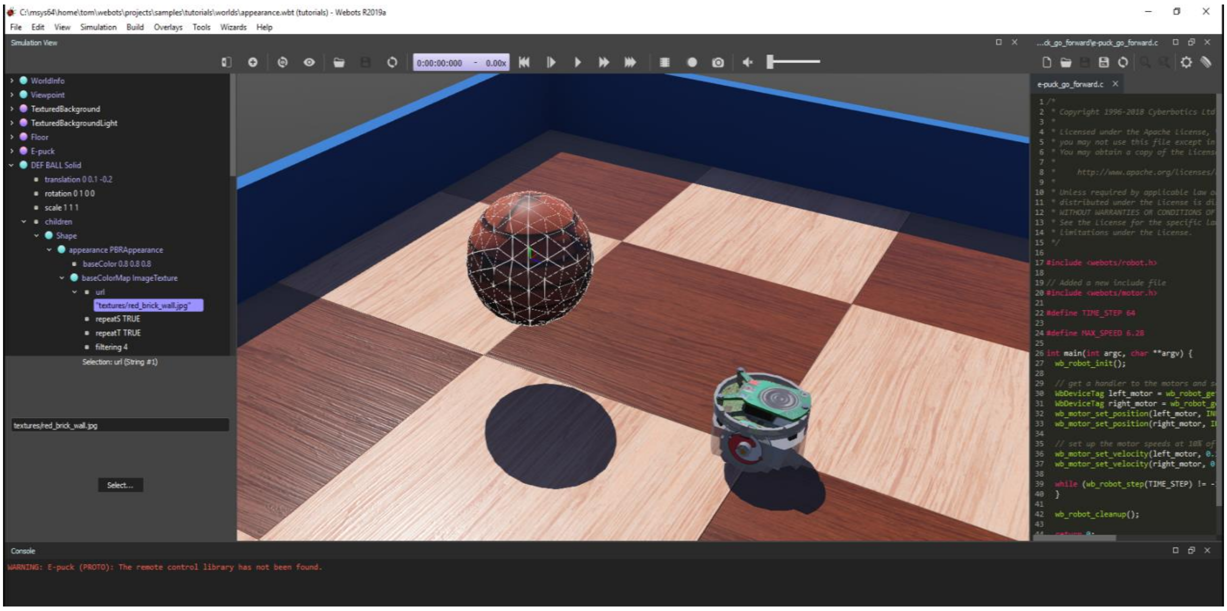


Fig. 9. Webots interface [21]

ISSN(online)2709-6149. Mining, constructional, road and melioration machines, 103, 2024, 71-85

Webots supports simulation of various types of robots, including manipulators, mobile robots, drones, cars. You can simulate the operation of such sensors as cameras, lidars, radio detectors, gyroscopes [22].

Webots integrates with Robot Operating System (ROS) and MATLAB, allowing it to be used as a tool for testing and developing robots in projects that use these platforms. Webots is support on a variety of operating systems, including Windows, macOS, and Linux.

Gazebo is an open source robotic systems simulator developed and maintained by a community of developers led by the Open Source Robotics Foundation (Fig. 10) [23, 24]. Gazebo provides tools for modeling various environments, including indoor and outdoor spaces, landscapes, buildings, roads, and other objects. This allows you to test robots in realistic conditions and study their behavior in different scenarios. This simulator supports simulation of various types of robots, including manipulators, mobile robots, drones, cars and others. It is also possible to model various sensors such as cameras, lidars, radio detectors, gyroscopes and many others. Gazebo integrates with the Robot Operating System (ROS), which allows you to use it as a tool for testing and developing robots in ROS projects [24 - 26].

Gazebo, as well as Webots, require the use of fairly modern equipment on which they will be launched, in particular, 64-bit processors with SSE2 support and a frequency of at least 2 GHz, a minimum of 4 GB of RAM is recommended, a GPU with OpenGL 3.3 support, an operating system of at least Windows 7 64-bit or modern Linux.

Among specialized professional software, we highlight MotoSim EG-VRC (Enhanced Graphics - Virtual Robot Controller). This is software developed by Yaskawa for simulating and programming Motoman series industrial robots. This tool is intended to help manufacturers and engineers in virtual testing, debugging and optimization of programs for robots before their physical use in production processes (Fig. 11). MotoSim EG-VRC allows you to simulate robots of the Motoman series and their working environment, including the equipment with which they interact, you can also create virtual models of factory objects, machines and other equipment of real production [27].

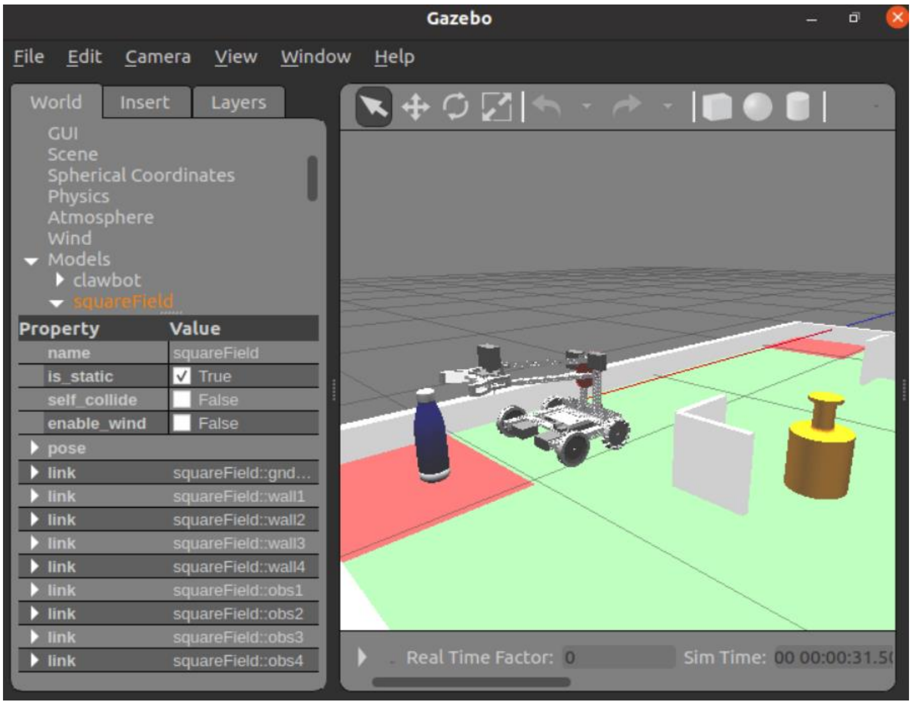


Fig. 10. Gazebo interface [25]

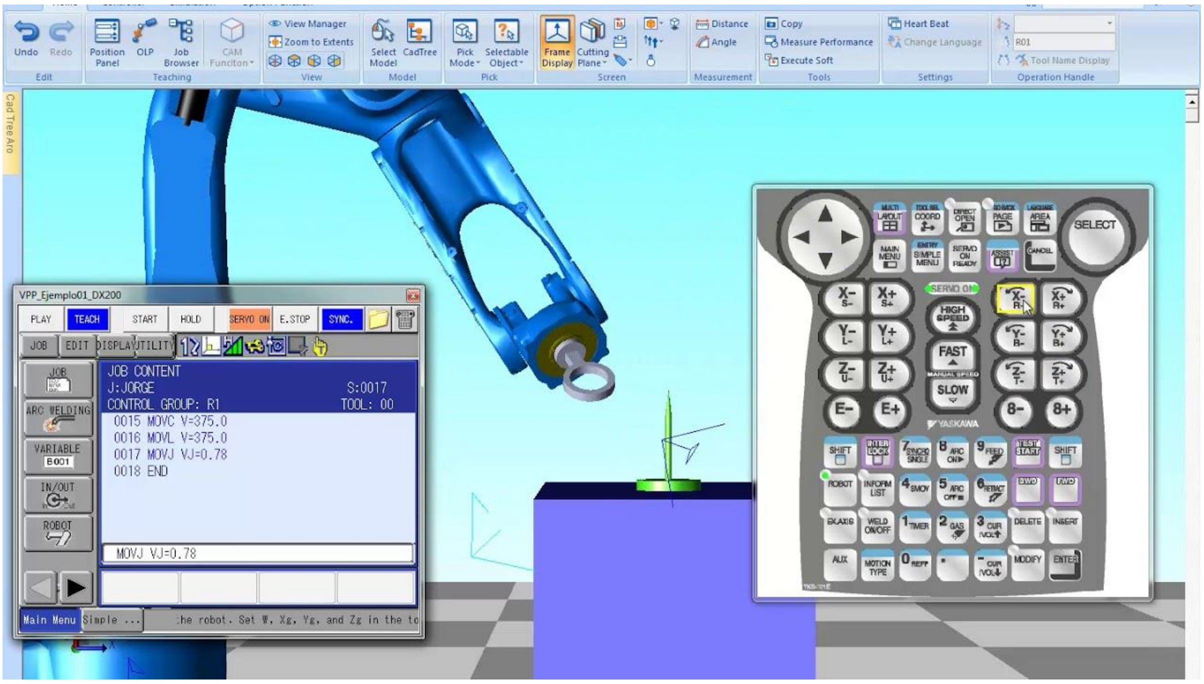


Fig. 11. MotoSim EG-VRC interface [27]

MotoSim EG-VRC supports integration with CAD software, which allows you to import object and work environment models from CAD programs for further use in simulations. MotoSim EG-VRC also has ROS support.

CONCLUSIONS

Evaluation of robot simulators is important to determine their effectiveness and suitability for various robotics tasks [28, 29]. In general, each of the discussed robot simulators has its advantages and disadvantages, and the choice between them depends on the specific needs and requirements. To study industrial robots and create their control programs close to real production conditions, it can be useful to use MotoSim EG-VRC, while for research or training purposes, Gazebo, Webots, CoppeliaSim or Q2WDBotSim can be suitable due to their availability and flexibility.

The advantage of MotoSim EG-VRC is its compatibility with Fanuc industrial robots, making it an ideal choice for companies that use these robots. The disadvantages are limited support for other brands of robots, as well as high cost and limited functionality compared to other simulators.

Gazebo is open source software with a large user community and active development. The main disadvantage is the difficulty of use for the initial level due to the large number of parameters and settings.

Webots is easy to use and contains a starting range of supported robots and sensors. Robots can also be created in CAD programs and uploaded to Webots, there is also a well-documented API and a friendly interface. The disadvantage of Webots is the significant requirements of the computer on which it will be launched, most robots need to be created by yourself in other programs.

CoppeliaSim has high flexibility in programming, supports the use of Python, Lua, contains a large number of available physical components, robots and sensors, has an open API. The disadvantages are the difficulty of use due to the large number of options and parameters. It does not support resource-intensive operating systems.

Q2WDBotSim is easy to use and learn. Does not require significant resources of the operating system, has a sufficient number of sensors to study the basic functions of the robot's movement. Disadvantages are limitations in functionality compared to other simulators.

REFERENCES

1. **Mishchuk D., V. Volyanyuk, E. Gorbatyuk** (2017). Modular principle of the construction of construction work. Girnichi, budivelni, dorozhni ta meliorativni mashini, Nr.89, 90–97.
2. **Mischuk D.** (2013). The review and analysis of designs robots for construction works. Girnichi, budivelni, dorozhni ta meliorativni mashini, Nr.82, 28-37.
3. **Loveikin V., Mishchuk D.** (2022). Analysis of the refined mathematical model of the manipulator boom system with an elastic base. Girnychi, budivelni, dorozhni ta melioratyvni mashyny [Mining, construction, road and melioration machines], Nr.99, 5-14.
4. **Mischuk D. Volianiuk V., Gorbatyuk Eu.** (2019). Determination of geometric parameters of the manipulator according to the characteristics of the working environment. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr.94, 25-34. (*in Ukrainian*). https://doi.org/10.32347/gbdmm2019.94.0201
5. **Volianiuk V., Mischuk D., Gorbatyuk Eu.** (2022). Determination of dynamic loads in the crane lifting mechanism. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr.99, 15-25. (*in Ukrainian*).
6. **Jueyao W., Xiaorui Z., Fude T., Tao Z., Xu X.** (2009). Design of a modular robotic system for archaeological exploration. In: Robotics and Automation. ICRA'09. IEEE International Conference on, Kobe, Japan, May 12-17, 2009, 1435-1440.
7. **Z. C. M. Davidson, S. Dang and X. Vasilakos** (2024). Blended Laboratory Design Using Raspberry Pi Pico for Digital Circuits and Systems. IEEE Transactions on Learning Technologies, Vol.17, 1170-1183. https://doi.org/ 10.1109/TLT.2024.3363230.
8. **Z. Ye and C. Hua**, (2012). An innovative method of teaching electronic system design with PSoC. IEEE Trans. Educ., Vol.55, No.3, 418-424.
9. **S. T. Cahill, P. E. B. Mann, A. F. Worrall and M. I. Stewart.** (2020). Remote teaching of programming in mathematica: Lessons learned, J. Chem. Educ., Vol.97, Nr.9, 3085-3089.
10. **Ruiz J. D., Deierling P. E.** (2021). Introducing mechanical engineering students to online robotics laboratories. Proceedings of asme 2021 international mechanical engineering congress and exposition (IMECE2021), Vol.9. ASME International Mechanical Engineering Congress and Exposition (IMECE), nov 01-05, 2021, Electr Network, Amer Soc Mech Engineers, WOS:000883348000047.
11. **Volianiuk V., Mischuk D., Gorbatyuk Eu**. (2020). Determination of inertial loads of the rotary boom of a self-propelled crane. Girnichi, budivelni, dorozhni ta meliorativni mashini [Mining, construction, road and reclamation machines], Nr.96, 13-21. (*in Ukrainian*). https://doi.org/10.32347/gbdmm2020.96.0201.
12. **Mischuk D., Rashkivskiy V.** (2024). Programuvannja robototehnichnih informacijnih sistem: konspekt lekcij. Kyiv, KNUCA, 214.
13. **Y. Pane, M. H. Arbo, E. Aertbeliën and W. Decré**, (2020). A System Architecture for CAD-Based Robotic Assembly With Sensor-Based Skills. IEEE Transactions on Automation Science and Engineering, Vol. 17, Nr.3, 1237-1249, doi: 10.1109/TASE.2020.2980628
14. **J. P. Carvalho Henriques et al.** (2023). Trajectory Generation Using RoboDK for a Staubli SCARA TS 60 Robot," 2023 11th International Conference on Control, Mechatronics and Automation (ICCMA), Grimstad, Norway, 2023, 121-126, https://doi.org/10.1109/ICCMA59762.2023.10374649.
15. **Online resource**. Simulate Robot Applications. (2024). https://robodk.com.
16. **Mihajlov E. P., Vudvud O. M., Sivokin S. V., Shvec B. R.** (2023). Doslidzhennja algoritmiv ta obladnannja dlja navigacii̇ mobil'nih robotiv za dopomogoju komp'juternogo modeljuvannja. Pidjom.-transp. tehnika. Nr.1(68), 71-86. (*in Ukrainian*).
17. **Montenegro G, Chacón R, Fabregas E, Garcia G, Schröder K, Marroquín A, Dormido-Canto S, Farias G.** (2022). Modeling and Control of a Spherical Robot in the CoppeliaSim Simulator. Sensors. 22(16), 6020. https://doi.org/10.3390/s22166020.
18. **M. A. Pastrana, J. Bautista, J. Mendoza-Peñaloza, L. H. Oliveira, D. A. Mendes and D. M. Muñoz**. (2023). Teaching Control Theory using Mobile Robot Obstacle Following/Avoidance with CoppeliaSim and MFO Algorithm. Latin American Robotics Symposium (LARS), Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE), Salvador, Brazil, 2023, 579-584, https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10333042.
19. **M. A. Pastrana, L. H. Oliveira, D. A. Mendes, Douglas. L. Silva, Jose Mendoza-**

Peñaloza, Daniel M. Muñoz. (2024). Implementation of a PID Controller using Online Tuning Applied to a Mobile Robot Obstacle Following/Avoidance". Argentine Conference on Electronics (CAE), 10-15.
20. **H. Zhang and T. Yang**. (2021). A simulation system for testing robotic navigation based on coppeliasim and ROS. 6th International Conference on Control Robotics and Cybernetics (CRC), 96-100.
21. **X. Gu, A. Zhang, Yuan L. and Xia. Y.** (2021). Design and Dynamics Simulation of a Triphibious Robot in Webots Environment. 2021 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 2021, 1268-1273. https://doi.org/10.1109/ICMA52036.2021.9512574.
22. **A.-C. Stan and M. Oprea.** (2019). A Case Study of Multi-Robot Systems Coordination using PSO simulated in Webots. 2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2019, 1-5. https://doi.org/10.1109/ECAI46879.2019.9042144 .
23. **Gabdrahmanov R. Tsoy, T. Bai Y. Svinin M. Magid E.** (2022). Automatic Generation of Random Step Environment Models for Gazebo Simulator. Robotics for sustainable future. CLAWAR 2021, Vol.324, 408-420. https://doi.org:/10.1007/978-3-030-86294-7_36.
24. **Farley A., Wang J., Joshua A. Marshall**. (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, Vol.120, 102629, https://doi.org/10.1016/j.simpat.2022.102629.
25. **Platt, J., Ricks, K.** (2022). Comparative Analysis of ROS-Unity3D and ROS-Gazebo for Mobile Ground Robot Simulation. J Intell Robot Syst 106, 80. https://doi.org/10.1007/s10846-022-01766-2.
26. **Osorio-Sanchez D and Gonzalez-Hernandez H.** (2022). Sensor fusion for vehicle odometry and path tracking: a ROS simulation 2022 International Symposium on Electromobility (ISEM). https://doi.org/10.1109/ISEM55847.2022.9976548. 978-1-6654-5923-5. (1-6).
27. **MotoSim EG-VRC**. (2024). Offline programming system with 3D simulation. Overview. https://www.yaskawa.eu.com/products/software/seriesdetail/serie/robotics-software_1005.
28. **Mishchuk D. O.** (2017). Study of the dynamics of the boom manipulator mounted on an elastic sup-port. Girnychi, budivelni, dorozhni ta melioratyv-ni mashyny [Mining, construction, road and me-lio-ration machines], Nr90, 11-18.
29. **Loveykin V. S., Mischuk D. O.** (2013). Optimization of the modes of changing the departure of the manipulator with a hydraulic drive. Monograph, Kyiv, CP Komprint, 206. https://doi.org/10.26884/damu.m13opzvmg

Огляд і аналіз програмних симуляторів інформаційних робототехнічних систем

Софія Єфремова[1], *Дмитро Міщук*[2], *Євгеній Горбатюк* [3]

[1,2,3]*Київський національний університет будівництва і архітектури*

Анотація. Стрімкий розвиток освітніх технологій потре-бує активного залучення нових засобів та ефек-тивних методів навчання. Одним із інновацій-них інструментів навчання за напрямком робо-тотехніка є застосування в навчальному процесі симуляторів роботів, які відкривають безліч можливостей для кращого засвоєння знань та розвитку практичних навичок у здобувачів освіти.

Симулятори роботів – це програмні комплекси, які дозволяють візуалізувати та моделюють поведінку реальних роботів. За допомогою цих інструментів студенти можуть:
- програмувати роботів у віртуальному середовищі, тестувати код та відлагоджувати помилки без ризику пошкодити реальне обладнання;
- навчатися робототехніці та принципам роботи роботів, досліджуючи різні типи датчиків, виконавчих механізмів та алгоритмів поведінки роботів;
- вирішувати задачі з різних галузей, таких як автоматизація виробництва, логістика, будівництво тощо;
- практикувати свої навички у безпечному та контрольованому середовищі.

Серед переваг використання симуляторів роботів це:
- доступність для великої кількості студентів, так як симулятори значно дешевші, ніж реальні роботи;
- безпека, так як віртуальне середовище усуває ризики, пов'язані з роботою з реальними роботами;
- гнучкість, що дозволяє студентам працювати у власному темпі, досліджувати різні сцена-

рії та експериментувати з різними параметрами;
- інтерактивність із реалістичним 3D-середовищем та візуалізацією.

В даній роботі показано огляд та аналіз програмного забезпечення, яке дозволяє моделювати системи керування роботів та симулювати їхню роботу для різних алгоритмів.

Проведено огляд та аналіз запропонованих програмних засобів симуляції робототехнічних систем, які застосовуються для навчання здобувачів освіти рівня бакалавр та магістр за спеціальностями «Комп'ютерні науки», «Інформаційні системи», «Галузеве машинобудування» та «Прикладна механіка» і довели свою дієвість та ефективність в умовах дистанційного навчання в Київському національному університеті будівництва і архітектури.

Ключові слова: роботизовані симулятори, роботизована інформаційна система, симулятори Arduino, візуальні симулятори, wokwi, tinkercad, Gazebo, Webots, CoppelisSim.