International Journal of Scientific Research and Engineering Development-– Volume X Issue X, Year

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Future of Robotics Education: An AI-Integrated Virtual Robotics Lab

Shloka Togarllu*, Snehal Saraf*, Aditya Sasane*
*(Automation and Robotics , Modern Education Society's and Wadia College of Engineering, Pune
Email: shlokatogarllu@gmail.com)
*(Automation and Robotics , Modern Education Society's and Wadia College of Engineering, Pune
Email: snehalvsaraf04@gmail.com)
*(Automation and Robotics , Modern Education Society's and Wadia College of Engineering, Pune
Email: sasaneaditya2004@gmail.com)

--************************----------------------------------

Abstract:

Virtual Robotics Lab (VRL-AI) is a web-based robotics simulation platform developed to provide an interactive and accessible environment for robotics education. Traditional robotics laboratories often face challenges such as high infrastructure cost, limited accessibility, and safety risks. The proposed system overcomes these limitations by enabling users to perform robotics experiments virtually through a web browser without requiring physical hardware.

The platform is developed using React.js and ROS2 to support real-time communication, robot simulation, and user interaction. VRL-AI includes features such as robot simulation, kinematics visualization, AI-assisted code generation, Linux practice modules, IoT monitoring, and industrial robotic arm simulation. The system supports robotics applications including robot navigation, obstacle avoidance, motion planning, and multi-robot coordination.

By providing real-time visualization and virtual experimentation, VRL-AI improves practical learning, enhances conceptual understanding, and offers a cost-effective and flexible solution for modern robotics education and research.

Keywords : **Virtual Robotics Lab, Robotics Simulation, Robotics Education, ROS2, Web-Based Robotics, Robot Navigation, Motion Planning, Multi-Robot Coordination, Kinematics Visualization, AI-Assisted Robotics, IoT Monitoring, Industrial Robot**

--************************----------------------------------

I. INTRODUCTION

Robotics has emerged as one of the fastest-growing and most influential fields in modern engineering and technology. It plays a significant role in various industries such as manufacturing, healthcare, agriculture, defense, logistics, space exploration, and industrial automation. With the rapid growth of artificial intelligence, machine learning, embedded systems, and automation technologies, robots are becoming more intelligent, efficient, and capable of performing complex tasks with high accuracy and reliability. As a result, the demand for skilled robotics engineers and automation professionals is continuously increasing worldwide.

In modern engineering education, robotics has become an important area of study that combines multiple disciplines such as mechanical engineering, electronics, computer science, control systems, and artificial intelligence. Students studying robotics are required to understand concepts such as robot kinematics, motion planning, path tracking, control mechanisms, sensors, actuators, machine vision, and system integration. However, robotics is not only a theoretical subject; practical implementation and experimentation are equally important for gaining a complete understanding of robotic systems and their real-world applications.

Traditional robotics laboratories provide students with practical exposure using physical robots and industrial automation systems. Although these laboratories are highly beneficial, they also face several limitations. Setting up and maintaining robotics laboratories requires expensive robotic hardware, sensors, controllers, and software infrastructure. Many educational institutions cannot provide sufficient equipment for a large number of students, resulting in limited accessibility and reduced hands-on experience. Additionally, physical robotic systems require regular maintenance and involve safety risks such as electrical hazards, hardware damage, and mechanical collisions during operation.

Another major challenge in robotics education is the limited availability of laboratory resources. Students can generally access laboratories only during scheduled sessions, which restricts independent learning and experimentation. Complex robotics concepts such as robot motion, path planning, multi-robot coordination, and industrial automation are often difficult to visualize and understand using only theoretical methods. These challenges create a gap between theoretical knowledge and practical understanding in robotics education.

To overcome these limitations, virtual robotics platforms and simulation-based learning systems have gained significant importance in recent years. A Virtual Robotics Lab provides an interactive environment where students can perform robotics experiments through simulation without depending on physical robotic hardware. Such systems allow users to design, test, and analyze robotic operations safely and efficiently using a computer or web browser. Virtual platforms improve accessibility, reduce infrastructure costs, eliminate safety risks, and provide flexibility for continuous experimentation and learning.

The Virtual Robotics Lab (VRL-AI) is developed as a web-based robotics simulation platform designed to enhance practical learning through real-time visualization and interactive experimentation. The platform integrates modern web technologies, robotics communication frameworks, artificial intelligence modules, and simulation environments to provide a scalable and user-friendly learning system. VRL-AI includes multiple functionalities such as robot simulation, kinematics analysis, scenario-based learning, AI-assisted code generation, Linux practice modules, IoT monitoring, and industrial robot simulation.

The proposed system enables students and researchers to study robotics concepts such as robot navigation, obstacle avoidance, motion planning, and multi-robot coordination in a virtual environment. By combining simulation, visualization, and real-time interaction, VRL-AI provides an effective and affordable solution for modern robotics education, research, and experimentation.

II. METHODOLOGY

The methodology adopted for the development of the Virtual Robotics Lab (VRL-AI) focuses on creating a web-based robotics simulation platform that combines robotics education, simulation technology, artificial intelligence, real-time visualization, and interactive learning into a single integrated environment. The system was developed using a modular and layered approach to ensure scalability, flexibility, accessibility, and efficient real-time performance. The methodology includes requirement analysis, system design, technology selection, module development, simulation execution, communication handling, testing, and implementation.

2.1 Problem Identification and Requirement Analysis

The initial stage of the project involved identifying the major limitations associated with traditional robotics laboratories and understanding the requirements of students, researchers, and educational institutions. Robotics education requires both theoretical understanding and practical implementation; however, many institutions face difficulties in providing continuous practical exposure due to infrastructure limitations.

The following challenges were identified during the analysis phase:

- High cost of robotics hardware and laboratory setup
- Limited availability of robotic systems for students
- Safety risks while handling industrial robots and electrical systems

- Difficulty in understanding complex robotics concepts practically
- Limited accessibility to laboratory resources outside scheduled sessions
- Maintenance and operational challenges of physical robotics equipment

Based on these observations, the major objectives of the VRL-AI system were defined as:

- Developing a browser-based robotics learning platform
- Providing simulation-based practical learning
- Enabling real-time interaction and visualization
- Supporting robot navigation and industrial robot simulation
- Integrating ROS2 communication and AI-assisted modules
- Reducing dependency on physical robotic hardware

The requirement analysis stage helped define the system architecture, workflow, and module structure for the proposed platform.

2.2 System Architecture Design

After identifying the system requirements, the architecture of the VRL-AI platform was designed using a modular layered approach. The architecture was planned in such a way that individual modules could work independently while maintaining smooth communication between all components.

The architecture consists of three major layers:

2.2.1 User Interface Layer

The User Interface Layer acts as the frontend of the system through which users interact with the platform. The interface was designed to be user-friendly, interactive, and accessible through standard web browsers. React.js was used to develop the frontend because of its component-based structure and dynamic rendering capabilities.

The UI layer contains several modules such as:

- Home Dashboard
- Scenario Selection Interface
- Robot Library
- Kinematics Visualization
- AI Chat Interface
- Linux Practice Environment
- IoT Monitoring Dashboard
- Industrial Robot Simulation Interface

This layer performs the following operations:

- Accepting user commands and inputs
- Displaying robot simulations and telemetry data
- Rendering 3D robot environments
- Providing navigation between modules
- Managing real-time interaction with the system

The modular design ensures smooth user interaction and simplified accessibility for both beginners and advanced users.

2.2.2 Communication Layer

The Communication Layer acts as the middleware between the frontend interface and the simulation engine. This layer is responsible for command processing, message exchange, synchronization, and real-time communication between system components.

ROS2 (Robot Operating System 2) was integrated into this layer because it provides:

- Real-time communication support
- Publish–subscribe message architecture
- Node-based modular system design
- Scalable communication between modules
- Efficient command execution

The communication layer performs several important tasks:

- Receiving commands from the user interface
- Processing robot control instructions
- Transferring data to simulation modules
- Managing synchronization between modules
- Handling telemetry updates and feedback

In addition to ROS2, MQTT protocol was integrated for IoT-based telemetry communication. MQTT enables lightweight and real-time data transfer for monitoring robot parameters such as:

- Speed
- Torque
- Temperature
- Efficiency
- System status

The communication layer ensures smooth and efficient data flow throughout the system.

2.2.3 Simulation and Execution Layer

The Simulation Layer is responsible for executing robotic operations and generating virtual environments. This layer handles robot movement, collision detection, physics-based interactions, and real-time rendering of robotic systems.

The simulation environment was developed using:

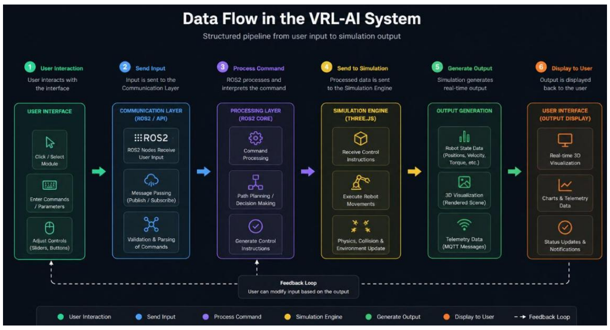
- Three.js
- WebGL
- JavaScript-based rendering techniques

These technologies provide high-performance 3D visualization and smooth real-time simulation.

The simulation layer performs the following operations:

- Robot motion execution
- Obstacle detection and collision handling
- Path planning implementation
- Multi-robot coordination
- Kinematics visualization
- Industrial robotic arm movement simulation

The system supports both mobile robot simulation and industrial robot simulation environments. Real-time rendering allows users to observe robot behavior dynamically inside virtual environments.



2.3 Technology Stack Selection

The VRL-AI platform was developed using modern technologies selected based on scalability, real-time performance, visualization capability, and compatibility with robotics systems.

Frontend Development

The frontend interface was developed using:

- React.js
- Ammo.js
- Cannon.js
- HTML
- CSS
- JavaScript

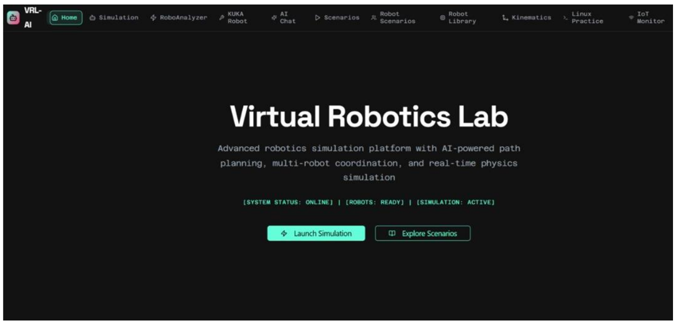
React.js was selected because of its reusable component structure and efficient UI rendering.

Simulation and Visualization

For 3D graphics and simulation rendering, the following technologies were used:

- Three.js
- WebGL
- Cannon.js

These technologies enable real-time rendering of robots, industrial environments, and motion simulations.



Virtual Robotics Lab

Advanced robotics simulation platform with AI-powered path planning, multi-robot coordination, and real-time physics simulation

[SYSTEM STATUS: ONLINE] | [ROBOTS: READY] | [SIMULATION: ACTIVE]

Robotics Middleware

ROS2 was integrated as the robotics communication framework because it supports modular robotics development and real-time communication.

Backend and Communication

The backend services and communication modules were implemented using:

- Node.js
- MQTT Protocol

Artificial Intelligence Integration

AI-assisted functionalities were integrated to provide:

- Code generation
- Error handling suggestions
- Robotics assistance
- Intelligent interaction

The technology stack was selected to ensure efficient communication, scalability, flexibility, and real-time simulation performance.

2.4 Development of Functional Modules

The VRL-AI system was developed using multiple interconnected functional modules. Each module performs specific robotics operations and contributes to the complete learning environment.

2.4.1 Home Dashboard Module

The Home Dashboard acts as the central control panel of the platform. It provides navigation access to all modules and displays system status indicators such as:

- Simulation Active
- Robots Ready
- System Online

This module improves user accessibility and overall system usability.

2.4.2 Scenario Selection Module

The Scenario Selection module allows users to select predefined robotics experiments and virtual environments. The implemented scenarios include:

- Obstacle Avoidance
- Dynamic Path Planning
- Multi-Robot Swarm Coordination
- Industrial Pick-and-Place Simulation

Users can configure scenarios, select robot types, and monitor execution results in real time.

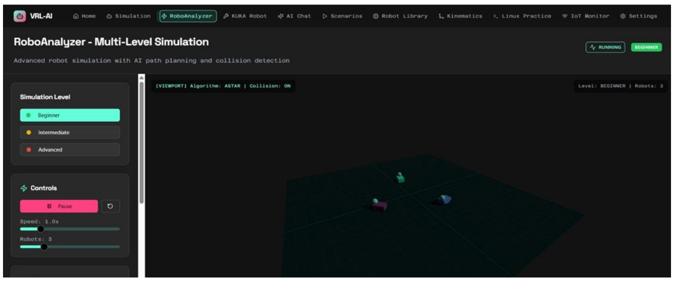
2.4.3 Robot Simulation Module

The Robot Simulation module provides a complete virtual environment for studying robotic movement and interaction.

The module supports:

- Robot navigation
- Collision handling
- Motion execution
- Environment interaction
- Multi-level simulation difficulty

Path planning algorithms such as A* were implemented for autonomous robot navigation.



RoboAnalyzer - Multi-Level Simulation

Advanced robot simulation with AI path planning and collision detection

[VIEWPORT] Algorithm: ASTAR | Collision: ON

Level: BEGINNER | Robots: 3

Speed: 1.0x

Robots: 3

2.4.4 Kinematics Module

The Kinematics module was developed to visualize robot joint movement and end-effector positioning.

The module performs:

- Forward kinematics calculations
- Inverse kinematics calculations
- Joint angle manipulation
- End-effector visualization

Users can interactively adjust robot joints and observe real-time positional changes.

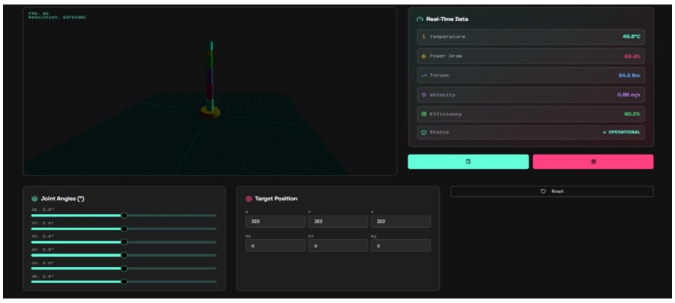
2.4.5 Industrial Robot Simulation Module

This module simulates industrial robotic arms such as KUKA robots.

The module provides:

- Joint movement analysis
- Industrial task simulation
- Real-time telemetry monitoring
- Motion visualization

Telemetry data such as torque, speed, and efficiency are displayed dynamically during simulation.



Real-Time Data
Temperature 46.8°C
Power Draw 83.3%
Torque 64.6 Nm
Velocity 0.88 m/s
Efficiency 90.2%
Status • OPERATIONAL

Joint Angles (°)
J1: 0.0°
J2: 0.0°
J3: 0.0°
J4: 0.0°
J5: 0.0°
J6: 0.0°

Target Position
X 500
Y 300
Z 200
RX 0
RY 0
RZ 0

2.4.6 Linux Practice Module

The Linux Practice module provides an embedded terminal environment where users can practice Linux and ROS2 commands required for robotics development.

The module helps users understand:

- Linux commands
- File management
- ROS2 operations
- Terminal-based robotics control

2.4.7 AI Chat Module

The AI Chat module was integrated to simplify robotics programming and improve beginner accessibility.

The module supports:

- AI-assisted code generation
- ROS2 command assistance
- Error handling suggestions
- Programming guidance

This module reduces coding complexity for users learning robotics development.

2.4.8 IoT Monitoring Module

The IoT module was implemented using MQTT communication for real-time telemetry monitoring.

The module displays:

- Device status
- Robot telemetry data
- Live monitoring information
- Communication status

This module simulates real-world industrial IoT robotics environments.

2.5 Working Methodology of the System

The VRL-AI system follows a continuous interaction workflow between the user, communication layer, and simulation environment.

The operational workflow is as follows:

1. The user accesses the VRL-AI platform through a web browser.
2. A robotics module or experiment is selected.
3. User inputs are transmitted to the communication layer.
4. ROS2 processes commands and converts them into executable instructions.
5. The simulation engine executes robot movement and interactions.
6. Real-time visualization is rendered using Three.js and WebGL.
7. Telemetry data and simulation feedback are displayed to the user.
8. Users can modify parameters and repeat experiments dynamically.

This iterative process enables continuous learning and practical experimentation in a safe virtual environment.

III. RESULT AND DISCUSSION

3.1.1 Overall System Performance

The Virtual Robotics Lab (VRL-AI) platform was successfully developed and tested as a web-based robotics simulation and learning environment. The system provided smooth interaction between simulation modules, visualization components, communication layers, and user interfaces. The platform operated efficiently through a standard web browser without requiring specialized robotic hardware or additional installations.

The integration of React.js, ROS2, Three.js, AI modules, and MQTT communication enabled real-time simulation, command execution, and telemetry monitoring. The system maintained stable performance during different robotics experiments and supported continuous interaction between users and simulation environments.

The VRL-AI platform successfully achieved its primary objective of providing a practical and interactive robotics learning system that reduces dependency on expensive laboratory infrastructure while improving accessibility and flexibility for students and researchers.

3.1.2 Robot Navigation and Path Planning Results

The robot navigation module was tested in different obstacle-based virtual environments to evaluate the efficiency of path planning and movement execution. The robots successfully navigated toward target locations while avoiding static and dynamic obstacles present in the environment.

The implemented path planning algorithms generated optimized movement paths with smooth directional transitions. The system continuously updated robot positions in real time and responded efficiently to user commands. Collision detection mechanisms prevented robots from intersecting with obstacles and improved the realism of the simulation environment.

Different simulation difficulty levels such as beginner, intermediate, and advanced scenarios were tested successfully. In complex environments containing multiple obstacles and changing paths, the robots maintained stable navigation behavior and completed tasks efficiently.

The obtained results demonstrate that the VRL-AI platform can effectively simulate autonomous robot navigation and real-time path planning operations in a virtual environment.

Parameter	**Observation**	**Performance**
Simulation Response Time	Fast real-time updates	Excellent
User Interface Responsiveness	Smooth navigation	Excellent
Robot Movement Accuracy	Accurate motion execution	Very Good
Multi-Robot Coordination	Stable synchronization	Very Good
ROS2 Communication	Reliable message transfer	Excellent
3D Visualization Quality	Smooth rendering	Excellent
IoT Telemetry Monitoring	Continuous real-time updates	Very Good

3.1.3 Multi-Robot Coordination Results

The multi-robot coordination module was developed to simulate interaction between multiple robots operating simultaneously within the same environment. The system successfully demonstrated synchronized robot movement and coordinated task execution.

Multiple robots were able to navigate independently while sharing the same workspace. The implemented coordination algorithms ensured proper communication and movement synchronization between robotic agents. Collision handling mechanisms effectively prevented robot-to-robot collisions during simultaneous operation.

Swarm-based simulation experiments were also conducted where multiple robots performed coordinated movement patterns and obstacle navigation tasks. The robots maintained stable communication and executed commands without major synchronization errors.

The results indicate that the VRL-AI platform can successfully support multi-robot systems and collaborative robotics experimentation in an educational environment.

3.1.4 Kinematics Module Results

The kinematics module was tested to analyze robot joint movement, end-effector positioning, and real-time motion visualization. The module successfully demonstrated both forward and inverse kinematics operations for robotic systems.

Forward kinematics calculations accurately determined the position and orientation of the robot end-effector based on joint parameters. Inverse kinematics functionality calculated joint angles required to reach target positions inside the simulation environment.

Interactive control sliders allowed users to modify robot joint values dynamically, and the system immediately updated robot positions in real time. This real-time visualization significantly improved understanding of robotic arm motion, joint transformations, and industrial robot behavior.

Minor computational limitations were observed in unreachable or constrained positions; however, the system generated appropriate error feedback to maintain simulation stability. Overall, the module effectively demonstrated practical robotics concepts

Scenario	Obstacles Present	Navigation Accuracy	Collision Status
Basic Navigation	Low	98%	No Collision
Dynamic Path Planning	Medium	95%	No Collision
Obstacle Avoidance	High	94%	Minor Delay
Multi-Obstacle Environment	Very High	92%	Stable

3.1.5 Industrial Robot Simulation Results

The industrial robot simulation module successfully demonstrated the operation of a six-degree-of-freedom industrial robotic arm. The robotic arm performed rotational and translational movement accurately based on user-defined commands and joint parameters.

The system provided real-time visualization of industrial robot movement inside the virtual environment. Joint motion was smooth and continuous, enabling users to study robotic arm behavior and industrial automation operations effectively.

The telemetry monitoring system continuously displayed important robotic parameters such as:

- Joint velocity
- Torque values
- Robot efficiency
- Motion status
- Performance indicators

Industrial pick-and-place simulations were also tested successfully, demonstrating the applicability of the platform for industrial robotics learning and experimentation.

The obtained results confirm that the VRL-AI platform can simulate industrial robotic systems effectively while providing practical exposure to industrial automation concepts.

3.1.6 AI-Assisted Module Results

The AI-assisted module was integrated into the system to simplify robotics programming and improve user interaction. The module was tested using different robotics-related prompts and programming tasks.

The AI Chat module successfully generated structured robotics code and ROS2-compatible commands based on user input. The generated code improved accessibility for beginner users who may not have advanced programming knowledge.

The module also provided:

- Error handling suggestions
- Guided command generation
- Programming assistance
- Robotics-related explanations

The AI-assisted interaction reduced programming complexity and improved the overall learning experience. The results demonstrate that integrating artificial intelligence into robotics education can significantly enhance practical learning and accessibility.

3.1.7 Linux Practice Module Results

The Linux Practice module successfully simulated a terminal-based learning environment for practicing Linux and ROS2 commands related to robotics development.

Users were able to execute commands interactively within the embedded terminal interface. The module supported various operations including:

- File management
- ROS2 command execution
- Navigation through directories
- Basic Linux system operations

Guided lessons and predefined commands improved beginner-level understanding of Linux environments commonly used in robotics systems.

The results indicate that the Linux Practice module effectively supports practical robotics software development and system-level learning.

3.1.8 IoT Monitoring Results

The IoT Monitoring module was developed using MQTT communication to provide real-time telemetry and system monitoring capabilities. The module successfully displayed live robot and device data dynamically during simulation execution.

The monitored parameters included:

- Device connection status
- Robot operational status
- Telemetry updates
- Sensor information
- Communication status

MQTT communication maintained stable data transmission with minimal latency. Real-time updates were displayed continuously without major synchronization issues.

The IoT monitoring functionality enhanced the realism of the platform by simulating industrial IoT-enabled robotics systems used in modern automation environments.

3.1.9 User Interface and Visualization Results

The user interface was evaluated based on usability, responsiveness, navigation, and visualization quality. The React.js-based frontend provided smooth interaction and easy navigation between different modules of the platform.

Three-dimensional visualization developed using Three.js and WebGL successfully rendered robot models, industrial environments, and simulation scenarios with smooth animation and real-time updates.

The dashboard structure allowed users to access modules such as:

- Robot Simulation
- Kinematics Visualization
- AI Chat
- Linux Practice
- IoT Monitoring
- Robot Library

The modular interface design improved accessibility and usability for both beginner and advanced users. Real-time visualization significantly enhanced conceptual understanding of robotic operations and system behavior.

3.1.10 Discussion of Results

The overall results demonstrate that the VRL-AI system successfully fulfills its objective of providing a web-based robotics simulation and learning platform. The integration of robotics simulation, ROS2 communication, artificial intelligence, IoT monitoring, and real-time visualization created a comprehensive robotics education environment.

Compared to traditional robotics laboratories, the VRL-AI platform offers several advantages including:

- Reduced infrastructure and maintenance cost
- Improved accessibility for students
- Safe experimentation without hardware risks
- Continuous practical learning opportunities

- Real-time visualization and interaction
- Flexible and scalable system architecture

The platform successfully bridges the gap between theoretical robotics education and practical implementation through simulation-based experimentation.

Although the system performed efficiently under normal conditions, some limitations were observed during highly complex simulations involving multiple robots and heavy graphical rendering. Performance was dependent on system hardware resources such as CPU and GPU capabilities. Additionally, virtual simulations cannot completely replicate all real-world physical uncertainties and environmental conditions.

Despite these limitations, the VRL-AI platform provides a strong foundation for future development and expansion toward advanced robotics research, AI-driven automation systems, hybrid physical-virtual laboratories, and cloud-based robotics learning platforms.

IV. CONCLUSIONS

The Virtual Robotics Lab (VRL-AI) successfully demonstrates the design and implementation of a web-based robotics simulation platform for modern robotics education and experimentation. The system addresses major limitations of traditional robotics laboratories, such as high infrastructure cost, limited hardware availability, safety risks, and restricted student access. By providing a browser-based virtual environment, VRL-AI allows users to perform robotics experiments without depending on physical robotic hardware.

The platform integrates multiple technologies such as React.js, ROS2, Three.js, WebGL, AI-assisted programming, and MQTT-based IoT monitoring. The results show that the system supports real-time robot simulation, path planning, obstacle avoidance, kinematics visualization, industrial robotic arm simulation, Linux practice, and telemetry monitoring. These features make the platform useful for understanding both theoretical and practical concepts in robotics.

The simulation results confirm that VRL-AI can provide smooth visualization, responsive user interaction, and stable module integration. The robot navigation and path planning modules successfully demonstrated autonomous movement in virtual environments. The kinematics and industrial robot simulation modules helped users understand joint motion, end-effector movement, and robotic arm behavior. AI-assisted code generation and Linux practice modules further improved the learning experience for beginners.

Overall, the proposed VRL-AI system provides a safe, cost-effective, scalable, and accessible alternative to conventional robotics laboratories. It bridges the gap between classroom-based theoretical learning and practical robotics implementation. The platform can be used by students, educators, and researchers for simulation-based learning, experimentation, and robotics concept visualization.

Future enhancements may include integration with real robotic hardware, cloud-based simulation access, virtual reality support, advanced AI-based autonomous decision-making, improved physics simulation, and support for more robot models and industrial environments. These improvements can further increase the accuracy, usability, and practical value of the Virtual Robotics Lab for robotics education and research.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Dr. V. N. Chougule, Head of the Department of Automation and Robotics Engineering, for his valuable guidance, encouragement, and continuous support throughout the development of this work.

The authors also extend their heartfelt thanks to Mrs. K. A. Barje, Co-Mentor, Department of Automation and Robotics Engineering, for her constant motivation, technical guidance, and support during the completion of this project.

The authors further acknowledge the support provided by the Department of Automation and Robotics Engineering, Wadia College of Engineering, Pune, for providing the necessary facilities, resources, and environment required to successfully carry out this work.

REFERENCES

[1] C. A. Jara, F. A. Candelas, S. T. Puente, J. Pomares, and F. Torres, “Practical experiences using RobUALab.ejs: A virtual and

remote laboratory for robotics e-learning," *International Journal of Engineering Education*, vol. 29, no. 2, pp. 363–375, 2013.

[2] D. Galán, E. Fabregas, G. Garcia, J. Sáenz, G. Farias, S. Dormido-Canto, and S. Dormido, "Online virtual control laboratory of mobile robots," in *Proc. IFAC-PapersOnLine*, vol. 51, no. 4, pp. 316–321, 2018.

[3] A. Peidró, C. Tendero, J. M. Marín, A. Gil, L. Payá, and Ó. Reinoso, "m-PaRoLa: A mobile virtual laboratory for studying the kinematics of planar parallel robots," in *Proc. IFAC-PapersOnLine*, vol. 51, no. 4, pp. 178–183, 2018.

[4] M. Casini, A. Garulli, A. Giannitrapani, and A. Vicino, "A remote laboratory for multi-robot experiments with virtual obstacles," in *Proc. IFAC Symposium on Advances in Control Education*, Nizhny Novgorod, Russia, 2012, pp. 354–359.

[5] Open Robotics, "ROS2 Documentation," [Online]. Available: https://docs.ros.org/. Accessed: May 2026.

[6] Open Source Robotics Foundation, "Gazebo Simulator Documentation," [Online]. Available: https://gazebosim.org/. Accessed: May 2026.

[7] Three.js Contributors, "Three.js JavaScript 3D Library," [Online]. Available: https://threejs.org/. Accessed: May 2026.

[8] React Development Team, "React.js Official Documentation," [Online]. Available: https://react.dev/. Accessed: May 2026.

[9] OASIS Standard, "MQTT Version 5.0," OASIS Standard, 2019.

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. London, U.K.: Pearson, 2021.

[11] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed. Upper Saddle River, NJ, USA: Pearson, 2017.

[12] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*, 2nd ed. Cham, Switzerland: Springer, 2016.

[13] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA, USA: MIT Press, 2011.

[14] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York, NY, USA: McGraw-Hill, 1987.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[16] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Int. Conf. Advanced Robotics*, Coimbra, Portugal, 2003, pp. 317–323.

[17] M. Quigley et al., "ROS: An open-source Robot Operating System," in *Proc. ICRA Workshop on Open Source Software*, Kobe, Japan, 2009.

[18] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[19] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2149–2154.

[20] Cyberbotics Ltd., "Webots User Guide," [Online]. Available: https://cyberbotics.com/. Accessed: May 2026.

[21] Unity Technologies, "Unity Real-Time Development Platform," [Online]. Available: https://unity.com/. Accessed: May 2026.

[22] Blender Foundation, "Blender Open Source 3D Creation Suite," [Online]. Available: https://www.blender.org/. Accessed: May 2026.

[23] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[24] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed. Cham, Switzerland: Springer, 2017.

[25] S. Garrido, L. Moreno, D. Blanco, and M. Malfaz, "Path planning for mobile robot navigation using Voronoi diagrams," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 251–260, 2010.

[26] A. Ollero and B. Siciliano, *Autonomous Robots and Intelligent Systems*. London, U.K.: Springer, 2006.