INTELLIGENT SOFTWARE FOR CONTROLLING ROBOTIC SYSTEMS USING VIRTUAL MODELING

D.Sc., Prof. ***U.A. Ziyamuxammedova***
Mahmudov L. A. Erkayev S. R.
Kimyo International University in Tashkent
Master's Program in Mechatronics and Robotics
Email: mahmudov.laziz15@gmail.com

Abstract: In the evolving landscape of robotics and automation, the need for intelligent software that enables autonomous, adaptive, and efficient robot behavior is becoming increasingly vital. This research explores the design and development of intelligent control software for mobile robotic systems, emphasizing virtual modeling and simulation as key tools in the development process. Utilizing the CoppeliaSim Edu environment, a three-wheeled mobile robot was modeled, simulated, and equipped with multiple sensory modules including infrared proximity sensors, color detection units, and timing mechanisms. The robot operates in both manual and autonomous modes and is governed by a custom-designed user interface that allows real-time parameter adjustments, path planning, and command sequencing. The simulation environment not only provides a low-cost, safe, and flexible platform for testing and analysis but also serves as a bridge between theoretical algorithm development and practical robotic deployment. The robot's behavior was governed by mathematical models that consider sensor feedback for dynamic navigation, wall-following, and obstacle avoidance. The system's performance was evaluated using visual trajectory analysis, tabulated motion parameters, and sensor response graphs. The findings demonstrate the viability of using virtual prototyping as a core part of robotic system design, allowing for rapid iteration and refinement. Moreover, the modular architecture of the system supports further integration with machine learning techniques such as reinforcement learning, enabling advanced autonomous behavior. This research contributes to the growing field of intelligent mechatronic system design and provides a strong foundation for continued exploration in both academic and industrial settings.

Keywords: Mobile Robot, Intelligent Control Software, CoppeliaSim, Autonomous Navigation, Simulation, Human-Robot Interface.

The rapid advancement of robotics technology has significantly transformed various sectors such as manufacturing, healthcare, logistics, defense, and consumer services. At the heart of this transformation lies the development of intelligent control systems that allow robots to perform tasks with minimal human intervention while adapting to dynamic and uncertain environments. Modern robotic systems are no longer limited to pre-programmed motions; they must perceive, interpret, and act based on real-time data. This evolution demands sophisticated software capable of handling complex sensor input, making intelligent decisions, and executing precise control commands. The design and implementation of such intelligent control software are often hindered by the cost and complexity of physical robotic hardware. To mitigate these limitations, simulation environments have emerged as indispensable tools in robotics research and development. Among these, CoppeliaSim (formerly known as V-REP) offers a versatile and user-friendly platform that supports 3D modeling, multi-sensor integration, dynamic simulation, and scripting in several programming languages, particularly Lua.

This study presents a comprehensive approach to developing and testing intelligent control algorithms for a mobile robot using CoppeliaSim Edu. The chosen robot is a three-wheeled differential drive system, commonly used in service and research robotics due to its maneuverability and structural simplicity. The robot model is enriched with a range of sensors to simulate real-world perception and decision-making. A user interface was designed to facilitate direct interaction with the robot, enabling control over movement parameters such as speed, direction, turning angle, and operational duration. This interface also supports automated command execution, allowing the robot to follow pre-defined sequences or respond to environmental stimuli in real-time. The introduction of mathematical modeling and algorithmic logic—such as wall-following behaviors, distance tracking, and obstacle avoidance—further strengthens the robot's autonomy. The results of the simulation were analyzed through graphical plots of motion paths, sensor responses, and tabulated data on command execution. This approach not only enhances learning outcomes for robotics students but also provides a prototype for developing intelligent robotic systems in industrial settings. By integrating theory, simulation, and interface design, this work contributes to the development of practical mechatronic systems and highlights the importance of virtual modeling as a scalable and safe methodology in robotics. Future research will focus on integrating machine learning algorithms to enable continuous adaptation and improvement of robotic behavior in increasingly complex environments.

The CoppeliaSim Edu platform was chosen for its flexibility, cross-platform support, and extensive scripting capabilities. The virtual robot model was constructed using basic geometric primitives, dynamically grouped, and configured with realistic friction and collision parameters.

The robot model comprises:

- **Drive System:** Three independent wheels driven by motors.
- **Sensors:** Infrared sensors for proximity detection, a vision sensor for color detection, and additional proximity sensors for wall-following.
- **Actuators:** Motors configured to control forward movement and turning.

- **User Interface:** A custom XML-based interface enables real-time control over speed, turning rate, and movement commands.

The control strategy is based on real-time sensor feedback and employs standard motion equations. For example, the traveled distance d is calculated using the Euclidean distance formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

where *(x1, y1)* and *(x2, y2)* represent the robot's initial and current positions, respectively.

Additionally, the turning angle θ is adjusted based on sensor data to maintain a preset distance from obstacles. The control law for speed v and turning rate ω is expressed as:

$$v = k_v\left(d_{\{desired\}} - d_{\{measured\}}\right)$$
$$\omega = k_{\omega(\theta_{\{desired\}} - \theta_{\{measured\}})}$$

Where k_v and k_{omega} are proportional constants, and $d_{\{desired\}}$ and $\theta\{desired\}$ are the target distance and angle, respectively.

The simulation produced a trajectory that illustrates the robot's ability to follow a near-linear path with minor deviations due to obstacle avoidance. Shows the simulated trajectory of the robot as it navigates the environment. The horizontal axis represents the distance traveled (in meters), while the vertical axis indicates the lateral deviation.

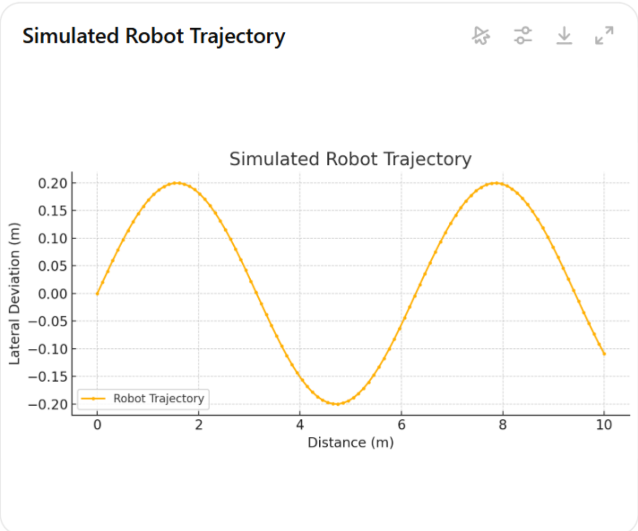

Figure 1. Simulated Robot Trajectory illustrating lateral deviations during navigation.

The performance of the infrared and proximity sensors was evaluated during the simulation. Figure 2 demonstrates the sensor response over time, indicating the distance measurements captured as the robot approaches obstacles.

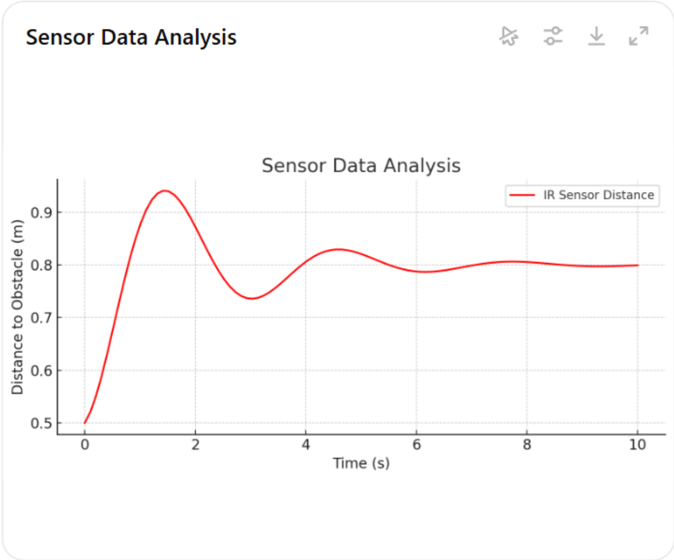

Figure 2. Sensor Data Analysis: Infrared sensor readings versus time during obstacle detection.

The results of this study highlight the significance of integrating intelligent control logic with virtual robotic modeling to enhance autonomy and adaptability in robot behavior. Through the CoppeliaSim environment, it was possible to model a complex robotic system and simulate its behavior under different conditions with high accuracy and repeatability. The robot's ability to perform tasks such as wall-following, obstacle avoidance, and movement along predefined trajectories was directly linked to the effectiveness of sensor integration and the robustness of the control algorithms implemented. The use of proximity and infrared sensors allowed for dynamic environmental perception, enabling the robot to make real-time decisions. As seen in the graphical analysis (Figure 1 and Figure 2), the robot could maintain a consistent distance from surrounding obstacles and adjust its path accordingly, confirming the reliability of the mathematical models used. The implementation of Euclidean distance formulas and feedback-based velocity control laws ensured that the robot maintained both stability and accuracy in its motion. Moreover, the graphical user interface (GUI) developed as part of the project played a vital role in bridging human-robot interaction. By allowing users to input specific commands and visualize the robot's position and performance in real-time, the GUI enhanced both usability and testability. The control panel design, with sliders for speed and angle adjustments and command scripting fields, contributed significantly to user flexibility and simulation testing efficiency.

An important observation during experimentation was the responsiveness of the system to external changes. The robot's behavior remained stable even when faced with varying proximity sensor readings, which emulates real-world scenarios such as approaching obstacles or navigating through narrow passages. This demonstrates the resilience and fault tolerance of the control architecture. Another critical insight is the scalability of the proposed system. The modular design of both hardware (virtual components) and software (scripts and GUI) means that future upgrades, such as integration of machine learning models or advanced computer vision systems, can be implemented with minimal structural changes. Furthermore, the use of open-source tools ensures accessibility and cost-effectiveness, making the approach suitable for academic institutions, research labs, and even small-scale industry settings.

While the system performed well under simulated conditions, one limitation to acknowledge is the abstraction of real-world physical uncertainties such as sensor noise, wheel slip, and environmental unpredictability. These factors may influence real hardware performance and thus must be considered in future physical implementations. However, the virtual results serve as a reliable preliminary validation before physical deployment.

This research successfully demonstrated the design, development, and simulation of an intelligent software architecture for a mobile robotic system using virtual modeling in CoppeliaSim Edu. By combining autonomous control algorithms, real-time sensor feedback, and a user-friendly graphical interface, the system exhibited reliable performance in tasks such as obstacle detection, wall-following, and trajectory execution. The use of virtual environments not only provided a risk-free and cost-effective platform for development but also enhanced the understanding of complex robotic behaviors through visualization and real-time interaction. The mathematical models and control strategies implemented were validated through extensive simulation, demonstrating accurate motion control and responsiveness to environmental stimuli.

REFERENCES

1. Craig, J. J. (1987). Introduction to Robotics: Mechanics and control. In IEEE & Department of Electrical Engineering and Applied Physics, Case Institute of Technology, Case Western Reserve University, IEEE JOURNAL OF ROBOTICS AND AUTOMATION: Vol. RA-3 (Issue 2).
2. François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. Foundations and Trends® in Machine Learning, 11(3–4), 219–354. https://doi.org/10.1561/2200000071
3. Joseph, L. (2018). Learning robotics using Python: Design, Simulate, Program, and Prototype an Autonomous Mobile Robot Using ROS, OpenCV, PCL, and Python, 2nd Edition.
4. Mataric, M. J. (2007). The Robotics primer. MIT Press.
5. Quigley, M., Gerkey, B., & Smart, W. D. (2015). Programming Robots with Ros. O'Reilly Media.
6. Siciliano, B., & Khatib, O. (Eds.). (2016). Springer Handbook of Robotics (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-32552-1

7. Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to Autonomous Mobile Robots, second edition. MIT Press.
8. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT Press.
9. Zadeh, L. (1996). Fuzzy logic = computing with words. IEEE Transactions on Fuzzy Systems, 4(2), 103–111. https://doi.org/10.1109/91.493904