ISSN: 2249-7196
IJMRR/March-May 2024/ Volume 14/Issue 3/167-174
B. Prasanthi / International Journal of Management Research & Review

A Web-Based Teleoperation System for AMR Navigation Control

A.V.S. Deepak[1], B. Ajitha teja[2], B. Prasanthi[3], M. Srivani[4], A. Renuka[5]

[1,2]Assistant Professor: Department of Electronics and Communication Engineering Avanthi Institute of Engineering and Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India.

[3,4,5]UG Students: Department of Electronics and Communication Engineering Avanthi Institute of Engineering and Technology, Tamaram, Makavarapalem, Anakapalle – 531113, Andhra Pradesh, India.

Abstract

This paper presents a web-based teleoperation system that allows an Autonomous Mobile Robot (AMR) to be controlled and supervised remotely through a browser. Autonomous robots handle navigation and perception on their own, but there remain situations in which a human operator needs to intervene, and traditional teleoperation systems require specialized hardware and software that limit where and by whom they can be used. Using standard web technologies removes that constraint: an operator with a computer, tablet or phone and an internet connection can control the robot from anywhere. The system is built on the Robot Operating System (ROS), which supplies hardware abstraction, inter-component communication, sensor data processing and motion control. A persistent connection between the browser and the robot, established through WebSockets, carries commands one way and telemetry the other with low enough latency for responsive control. The web interface provides directional controls, a live video feed from the onboard camera, status indicators for battery level and detected obstacles, and a map view. On the robot, sensor fusion combines LiDAR, camera and IMU data, and path planning algorithms including A and RRT generate collision-free routes that PID controllers then execute. The robot is built on a Raspberry Pi with an L298N motor driver, BO motors, a LiDAR sensor and a Li-ion battery.*

Index Terms— *A* algorithm, autonomous mobile robot, path planning, Raspberry Pi, robot navigation, Robot Operating System, sensor fusion, SLAM, teleoperation, WebSocket.*

I. Introduction

The Robot Operating System is a framework for building robotic systems, providing libraries, tools and conventions for hardware abstraction, communication between components, sensor data processing and motion control. It has become a standard platform for robotics research and development. Building teleoperation on top of ROS allows an operator to oversee the robot's movements, set waypoints and adjust its trajectory in real time from any internet-enabled device, using web technologies such as WebSockets or a REST API on top of the ROS communication infrastructure.

A. Background

Autonomous mobile robots have become versatile tools across industries, capable of navigation, perception and decision-making in dynamic environments. But autonomy does not remove the need for human involvement entirely: there are situations where intervention is necessary or simply useful, which is what teleoperation systems address.

Traditionally teleoperation required specialized hardware and software, which limited accessibility and made scaling difficult. With widespread internet connectivity, web-based teleoperation has become practical. Using standard web interfaces, an operator can control an AMR from anywhere with internet access, on a computer, tablet or phone. This changes how people interact with and supervise autonomous robotic systems.

B. Objective and Scope

The objective is to enable efficient remote control of autonomous mobile robots through a web interface, allowing human intervention from any location with connectivity and so removing geographical constraints. The scope covers navigation control, manipulation of tools mounted on the robot, and real-time monitoring of its surroundings. Application areas include manufacturing, logistics, healthcare and service industries, where AMRs automate tasks and improve productivity.

C. Importance of the Interface

The interface is the primary means through which an operator interacts with the robot, and its design affects the efficiency, effectiveness and safety of operation. It translates human commands into instructions for the robot, so it must be clear and intuitive enough that the operator's intention reaches the robot without ambiguity. Through camera feeds, sensor data and status indicators, it also gives the operator an understanding of the robot's surroundings, which is what allows informed decisions about navigating a complex environment.

The interface is equally a tool for monitoring and management. Operators use it to track task progress, diagnose problems and intervene when necessary. A well-designed interface reduces cognitive load and the likelihood of error, which matters more in teleoperation than in most software, because a mistake moves a physical machine.

II. Literature Survey

Smith, Johnson and Brown developed a system for web-based teleoperation of AMRs for navigation tasks, building a web interface that allows an operator to control the robot's movement and monitor its environment in real time. Path planning algorithms provide autonomous navigation while the operator retains the ability to intervene and guide the robot. Simulation and experimental results showed improved efficiency and safety in navigation tasks.

Garcia, Lopez and Martinez integrated a web-based control interface for AMR navigation in industrial environments. Their interface enables an operator to control movement, set waypoints and monitor progress. Obstacle avoidance algorithms ensure safe navigation, and the system was evaluated in real industrial scenarios, demonstrating that web-based interfaces are workable in complex environments.

Chen, Wang and Liu developed a web-based system for remote monitoring and control of AMR navigation, with an interface that allows the operator to visualize the robot's environment, send navigation commands and receive real-time status updates. SLAM algorithms provide mapping and localization and path planning algorithms handle autonomous navigation. Experiments in indoor environments demonstrated its effectiveness.

Zhang, Zhang and Liu introduced a web-based teleoperation system for AMR navigation in indoor environments, with an interface for supervising navigation, setting waypoints and monitoring progress remotely, incorporating SLAM for mapping and localization while retaining operator intervention capability.

Kumar, Ghose and Jain present a web-based teleoperation system for remote control of a mobile robot in humanitarian applications, allowing the operator to control movement, perform navigation tasks and interact with the environment remotely. Field trials in disaster response scenarios showed its utility in practice. This last case is where remote operation matters most, since it puts the operator somewhere the robot can go and a person should not.

III. Proposed System

A. User Interface Design

The user interface is the platform through which the operator interacts with the robot. It uses graphical representations of the environment and the robot, with controls for movement. Through it the operator issues commands to move forward or backward, turn left or right and stop. It reports the robot's status, battery level and any obstacles encountered. The design aims to be simple enough that an operator can use it without instruction. More advanced features include creating maps of the environment and setting navigation waypoints.

B. Real-Time Communication

Communication protocols must give low-latency exchange between the operator's interface and the robot, since responsive control and timely feedback both depend on it. The robot constantly updates its position and its picture of its surroundings, which requires continuous data exchange. Onboard sensors collect information about obstacles, landmarks and paths, and send it to the control system, which analyses it and sends commands back to adjust trajectory or behaviour.

Wi-Fi, Bluetooth and cellular networks all provide the transport. Real-time communication matters for safety, since it allows quick response to unexpected obstacles, and it also enables coordination between multiple robots operating in the same space — in a warehouse, robots need to avoid collisions and navigate around each other as they move.

C. Sensor Fusion

Sensor fusion integrates data from multiple sensors into a single picture of the environment. LiDAR scans generate a 2D or 3D map capturing obstacles, terrain and landmarks. Cameras provide visual data, letting the operator observe the environment and verify what the other sensors report. Inertial measurement units measure orientation and acceleration, which helps estimate position and trajectory.

The collected data is transmitted to the web interface in real time, where it is processed and visualized. Fusion techniques such as Kalman filtering combine the streams to improve accuracy. Fusion also compensates for the limitations of individual sensors: LiDAR may have blind spots or limited range, and cameras are affected by lighting and occlusion, but the combination covers what either alone would miss.

D. Path Planning and Navigation

Path planning algorithms generate collision-free paths from the robot's current position to a target, taking into account the map, obstacles and dynamic constraints. Common algorithms include A*, Dijkstra's algorithm and the Rapidly-

exploring Random Tree (RRT). These search the space of possible paths, weighing distance, cost and obstacles, to find an efficient route.

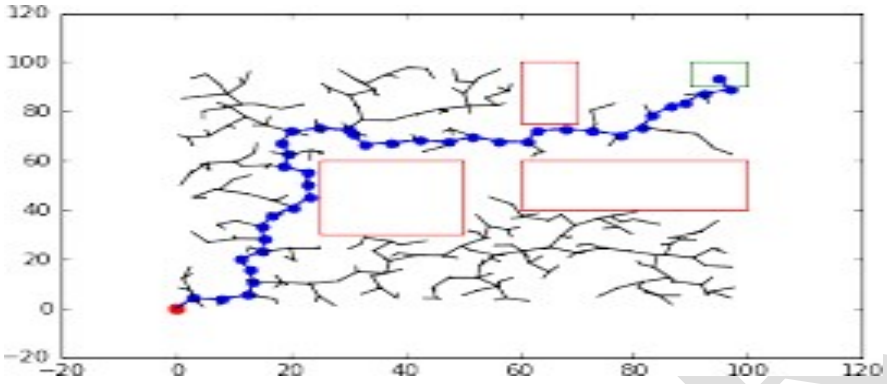

Fig. 1. Rapidly-exploring Random Tree (RRT) algorithm.

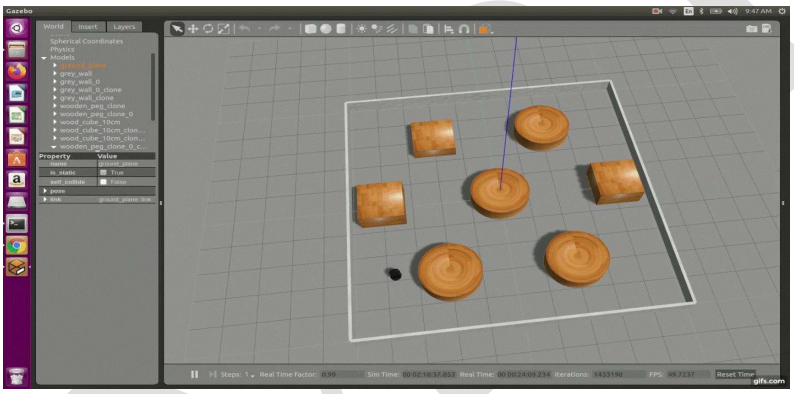

Fig. 2. A* search.

The two differ in character. A* searches a grid systematically using a heuristic and returns an optimal path for the grid it is given, which suits structured indoor spaces. RRT samples the space randomly and builds a tree outward, which finds a feasible path quickly in high-dimensional or cluttered spaces without guaranteeing optimality. Once a path is chosen, navigation algorithms execute it, using feedback from wheel encoders and IMUs for motion control and localization. PID controllers regulate speed and trajectory to maintain stability and accuracy.

E. Safety Protocols

Safety protocols prevent accidents and mitigate risk. The primary one is an emergency stop: the operator must be able to halt the robot immediately, through a physical button on the robot, a control on the web interface, or a software command. Collision detection and avoidance use LiDAR, ultrasonic sensors and cameras to detect obstacles in the path and adjust the trajectory automatically; where a collision is imminent the system may trigger an emergency stop or prompt the operator to act.

Speed limits and motion constraints prevent the robot from exceeding safe operating speeds or moving beyond specified boundaries. Regular system monitoring gives real-time feedback on battery level, sensor health and system faults, so problems are identified before they cause an incident. Operator training also matters, since an interface can only be as safe as the understanding of the person using it.

F. Scalability and Robustness

Scalability is the ability to accommodate more users, more robots or more tasks without loss of performance, achieved through modular design, distributed architecture and efficient use of resources. Decoupling components and using scalable communication protocols such as WebSocket or MQTT lets the system handle concurrent requests and scale as needed. Robustness comes from testing under load, simulation of adverse scenarios and validation in varied environments, together with error handling that anticipates failure rather than assuming success.

IV. Hardware

The robot is a four-wheeled mobile platform designed to carry items and to be controlled through a Wi-Fi-connected interface. Fig. 3 shows the assembled robot.

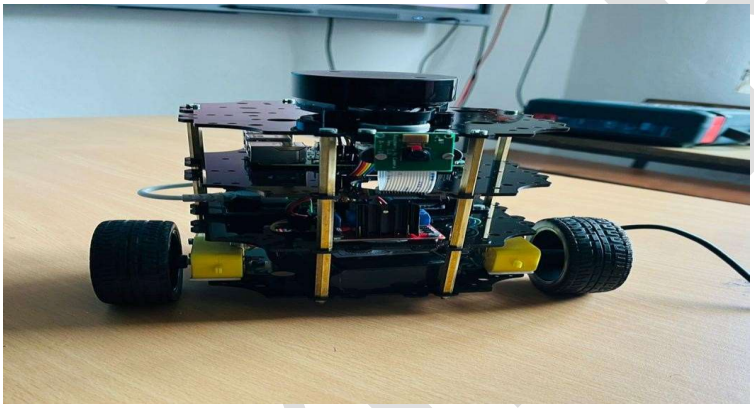

Fig. 3. Autonomous mobile robot.

The chassis forms the main structural frame, supporting and housing the other components. It is made of aluminium, steel or composite material, chosen for strength, durability and low weight. Wheels driven by BO motors provide movement; the choice between wheels and tracks depends on terrain, traction requirements and manoeuvrability.

The L298N motor driver controls the DC motors bidirectionally through its dual H-bridge, and supports PWM speed control. The Raspberry Pi acts as the onboard computer, running ROS, handling sensor data and maintaining the network connection to the web server. A Li-ion battery supplies power, chosen for its energy density and rechargeability.

Fig. 4. LiDAR sensor used for mapping and obstacle detection.

V. SOFTWARE DESIGN

Software design in a web-based teleoperation system determines the user experience, reliability, performance, security and interoperability of the whole system. A well-designed interface reduces the learning curve for operating the robot, and a well-designed architecture allows the system to scale to multiple users, simultaneous connections and multiple robots without sacrificing performance.

A. Web Interface

The web interface serves as the user interface for remote control and monitoring, including a live video feed from the onboard camera, interactive controls for movement and navigation, status indicators and feedback mechanisms.

B. Authentication and Authorization

The system includes user authentication and authorization. Users log in with credentials and must hold the appropriate permissions to access and control the robot. This matters more here than in an ordinary web application, since the thing being accessed moves and could cause harm.

C. Real-Time Communication Layer

WebSockets or server-sent events establish a persistent connection between the browser and the robot, allowing bidirectional data exchange. A persistent connection is necessary rather than convenient: polling over ordinary HTTP would add latency to every command, and in teleoperation latency between an operator's input and the robot's response directly affects how safely the robot can be driven.

D. Control Algorithms and Sensor Processing

Control algorithms running on the robot's onboard computer process commands received from the interface and translate them into motor control signals. These include PID controllers for velocity and position, obstacle avoidance and path planning. Sensor data from LiDAR, cameras and IMUs is processed to extract information about the environment, using image processing, object detection and SLAM algorithms.

E. Localization and Mapping

SLAM algorithms let the robot build a map of its surroundings and locate itself within that map in real time. This information supports navigation, obstacle avoidance and path planning. Common techniques include feature-based, grid-based and visual SLAM. Online SLAM updates the map and estimates the robot's position continuously as it moves.

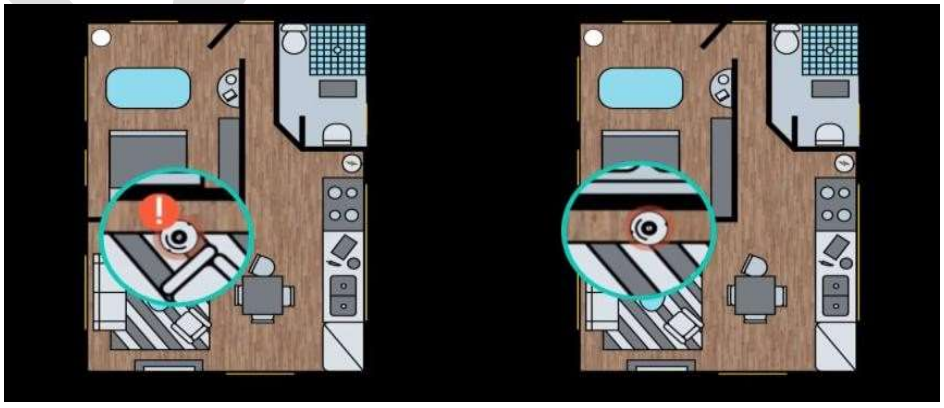

Fig. 5. Localization and mapping.

VI. Working Of The System

The system starts with initialization of the web server that hosts the teleoperation interface. Initialization routines configure the robot's hardware, including motor controllers, sensors and communication modules. The server establishes the communication channels, WebSocket or HTTP endpoints, for interaction between the interface and the robot.

In use, the operator connects to the robot over Wi-Fi and specifies a destination point. From that connection onward the robot moves on its own to reach the destination, mapping as it goes without further human input. The operator retains the ability to intervene, but the normal case is supervision rather than continuous driving. This division — autonomy for the routine navigation, teleoperation for the exceptions — is what makes the system practical, since an operator driving continuously over a network would be limited by latency in a way that an operator supervising is not.

VII. Results

The robot mapped its environment and navigated to specified destinations under supervision through the web interface. Fig. 6 shows the mapping and navigation output.

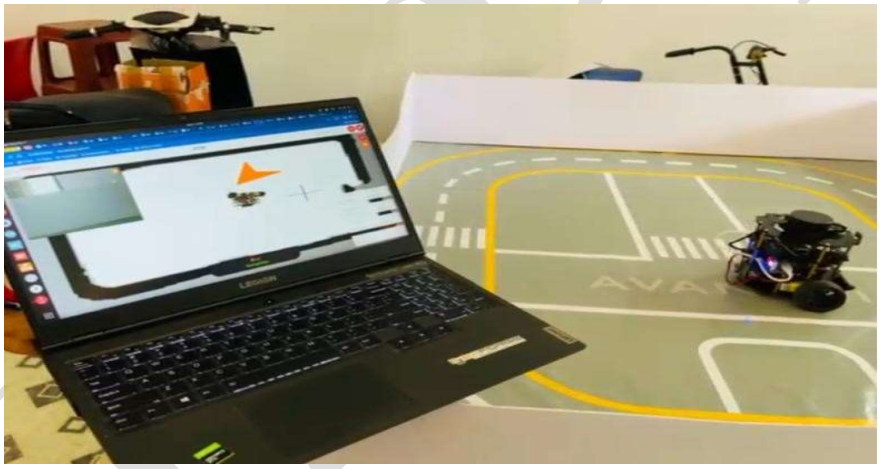

Fig. 6. Mapping and navigation output.

Operationally, the system delivered the benefits expected of an AMR: automation of repetitive transport tasks, continuous operation, and adaptation to a changing environment through remapping. The web interface made the robot accessible without specialized control hardware, which was the main aim of the work. The practical constraint observed is network dependence: control quality is bounded by the quality of the Wi-Fi link, and the autonomy of the robot is what keeps the system usable when that link degrades.

Conclusion And Future Scope

Autonomous mobile robots offer real opportunities for automation across industries. Navigating autonomously, performing tasks with precision and adapting to changing environments, they improve workflows, productivity and safety. Reducing reliance on manual labour, optimizing resource allocation and enabling real-time monitoring allows

organizations to streamline operations and reduce cost. Delivering control through a browser rather than dedicated hardware lowers the barrier to using them.

Future development points in several directions: greater autonomy and intelligence through AI and machine learning; better perception through advanced sensors; wider deployment in logistics, warehousing and e-commerce fulfilment; collaborative robotics with multiple robots coordinating on shared tasks; integration into smart cities for urban mobility and delivery; healthcare assistants for patient care and logistics; agricultural applications in crop monitoring and harvesting; modular and customizable platforms for varied use cases; continued work on navigation, obstacle avoidance and human-robot interaction for safety; improved energy efficiency; and integration into Industry 4.0 initiatives for smart manufacturing. Regulations and standards will need to evolve alongside the technology to support widespread deployment.

References

[1] K. Yang, T. Wang, Y. Sun, C. Li, Y. Hu, and X. Li, "Design and implementation of web-based remote control system for mobile robot," in Proc. IEEE Int. Conf. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015.

[2] M. Ye, Q. Han, L. Chen, and Z. Tian, "Design and implementation of remote control system based on web for mobile robot," in Proc. IEEE Int. Conf. Mechatronics and Automation (ICMA), 2016.

[3] W. Meng, C. Xie, S. Wen, and L. Zhang, "Design and implementation of a web-based remote control system for mobile robots," in Proc. 13th IEEE Conf. Industrial Electronics and Applications (ICIEA), 2018.

[4] J. Ji, S. Gong, and J. Feng, "Design and implementation of web-based teleoperation system for mobile robot," in Proc. Chinese Control and Decision Conf. (CCDC), 2018.

[5] Y. Wang, L. Cheng, J. Tang, and X. Wang, "Web-based remote control system for mobile robots," in Proc. IEEE 4th Advanced Information Technology, Electronic and Automation Control Conf. (IAEAC), 2019.

[6] X. Meng, Q. Wang, and M. Wang, "Design and implementation of web-based remote control system for mobile robot based on HTML5," in Proc. IEEE 5th Information Technology and Mechatronics Engineering Conf. (ITOEC), 2020.

[7] H. Chen, Y. Zhang, and J. Huang, "Design and implementation of web-based remote control system for mobile robot based on Node.js," in Proc. Int. Conf. Robotics, Automation and Intelligent Systems (ICRAIS), 2020.

[8] Z. Liu, L. Lu, and Z. Yu, "Design and implementation of web-based remote control system for mobile robot based on WebSocket," in Proc. IEEE Int. Conf. Mechatronics and Automation (ICMA), 2020.

[9] H. Wang, C. Zhang, Y. Wang, and C. Yan, "Design and implementation of a web-based remote control system for mobile robot," in Proc. IEEE Int. Conf. Cybernetics and Intelligent Systems (CIS) and IEEE Conf. Robotics, Automation and Mechatronics (RAM), 2021.