Mathematical Modeling and Simulation of an Autonomous Warehouse Robot System

Devi Krishna R, Gowri G, Arundhathy Devi L, Abhinand S, Adhi Lekshmi P Binu
Department of Robotic And Automation

Abstract

In this study, an Autonomous Warehouse Robot System is mathematically modeled and simulated to achieve efficient and collision-free navigation within a structured warehouse environment. The system integrates path planning, motion control, and kinematic modeling to enable autonomous movement between predefined pick-up and delivery points. The A* algorithm is employed for optimal path generation by minimizing travel distance, while a PID controller ensures smooth and stable trajectory tracking along the planned route. Transformation matrices are utilized to relate wheel angular velocities to the robot's linear and angular motion. Python-based simulations demonstrate accurate path generation, stable control response, and effective 3D visualization of robot navigation. The results validate the effectiveness of mathematical modeling and control strategies for developing reliable and intelligent robotic systems in warehouse automation.

—

1 Introduction

Automation has become an essential element in modern industrial systems, particularly in warehouse management, where efficiency, accuracy, and operational safety are critical for handling large-scale logistics operations. Traditional manual methods of material handling are often time-consuming, labor-intensive, and prone to human error. To overcome these challenges, the use of autonomous mobile robots (A.M.R.S) has emerged as a key solution for achieving intelligent and efficient warehouse automation. This study addresses the problem of autonomous navigation and path optimization for a warehouse robot operating in a structured environment with fixed obstacles. The primary objective is to design and simulate a robot system that can autonomously move between pick-up and delivery points while minimizing travel distance and avoiding collisions. The system integrates mathematical modeling, path planning, and control algorithms to ensure efficient and stable navigation. To accomplish this, the A* algorithm is utilized for optimal path planning, providing a computationally efficient method for determining the shortest collision-free route between predefined points in the warehouse layout. A Proportional-Integral-Derivative (PID) controller is implemented to regulate the robot's motion along the planned trajectory, ensuring smooth turns, precise velocity control, and minimal position error. Additionally, kinematic transformations are applied to relate wheel angular velocities to the robot's linear and angular motion, enabling accurate control of robot dynamics. The overall system is modeled and simulated in Python using computational tools such as NumPy, Matplotlib, and VPython for visualization. The simulation environment replicates a warehouse grid where the robot autonomously navigates between target points, validating the performance of both the path planning and control systems. Through this work, the study demonstrates how mathematical modeling, control theory, and simulation techniques can be integrated to design an efficient autonomous robot suitable for real-world warehouse automation applications.

Figure 1: Conceptual image of the A.M.R.S (created with Gemeni)

2 Methodology

The methodology is divided into five major stages: Environmental Setup And Mapping ,Path Planning, Motion Control Using PID, Kinematic Transformation And Simulation and Result Visualization And Analysis

2.1 Module 1: Environmental Setup And Mapping

The warehouse environment is modeled as a 2D grid with defined start and goal positions. Each grid cell represents a possible robot location. The robot's motion is governed by non-holonomic differential equations:

$$\dot{x} = v\cos(\theta), \quad \dot{y} = v\sin(\theta), \quad \dot{\theta} = \omega$$

where

x, y are the position coordinates of the robot

θ is the robot's orientation

v is linear velocity

ω is angular velocity. These equations describe the robot's differential drive motion and form the basis for control and trajectory simulation.

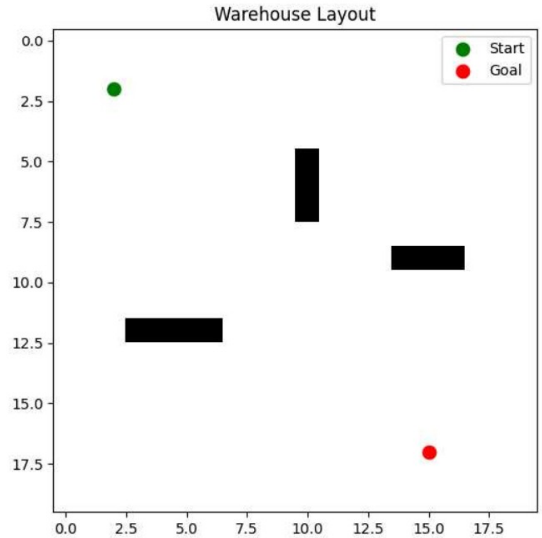

Figure 2: 2D visualization of the warehouse grid with start (green), goal (red), and obstacles (black).

2.2 Module 2: Path Planning using A^* Algorithm

The A* algorithm is implemented to compute the optimal path between the start and goal nodes while avoiding obstacles. The cost function is defined as:

$$f(n) = g(n) + h(n)$$

where

$g(n)$ is the cost from the start node to the current node

$h(n)$ is the heuristic estimate to the goal (using Euclidean or Manhattan distance).

The algorithm continuously updates open and closed node sets until the optimal path from start to goal is found. This path is stored as a sequence of coordinates for robot movement.

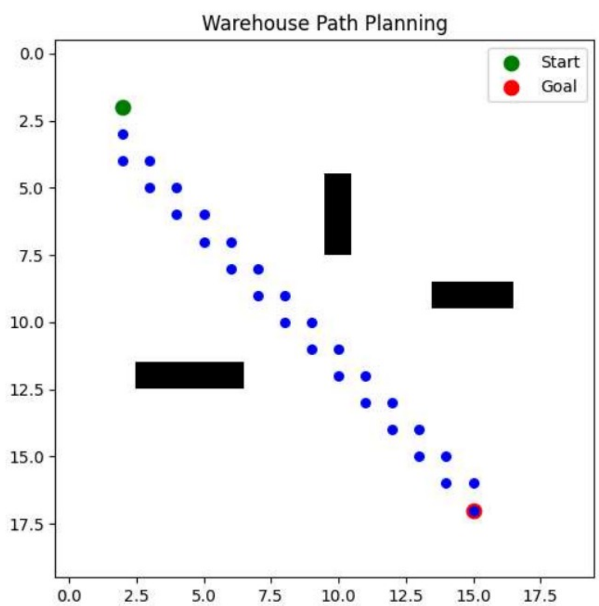

Figure 3: The optimal path (blue dots) computed by the A^* algorithm, navigating around all obstacles.

2.3 Module 3: Motion Control Using PID

follow the planned path accurately, a PID controller adjusts the robot's wheel velocities based on positional error. The control law is expressed as:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}$$

where $e(t)$ is the position error between target and actual coordinates.

The proportional term reduces present error, the integral term minimizes accumulated offset, and the derivative term prevents overshoot by damping oscillations. This control law ensures smooth turning, accurate alignment, and speed regulation throughout navigation.

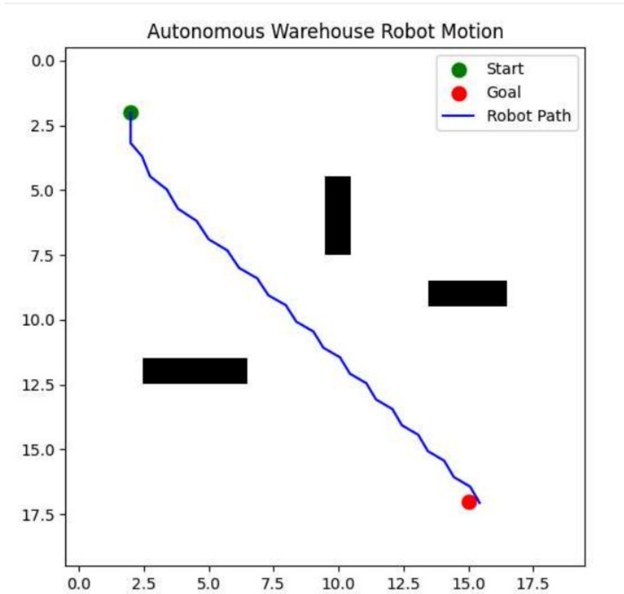

Figure 4: The smoothed trajectory of the robot (blue line) as it follows the path generated by the A^* algorithm, controlled by the PID system.

2.4 Module 4: Kinematic Transformation And Simulation

To represent and control the robot's position and orientation (pose) in space using mathematical transformations. Linear algebra, specifically 3x3 homogeneous transformation matrices, is used to calculate the robot's translation and rotation [**?**]. The orientation angle θ is determined by the direction of movement along the path.

$$T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix}$$

The robot's pose was successfully tracked throughout its trajectory. At various points along the path, orientation vectors were calculated and visualized, correctly showing the robot's heading as it navigated towards the goal (Figure 5).

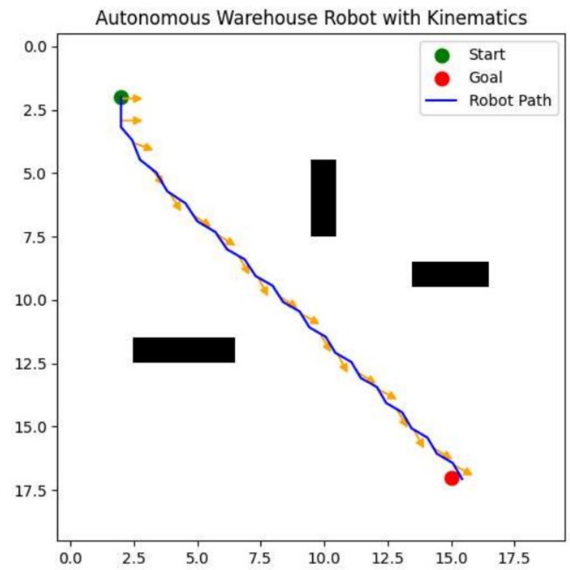

Figure 5: The robot's path with orientation vectors (orange arrows) shown at intervals, indicating its heading as it moves.

2.5 Module 5: Result Visualization And Analysis

To extend the simulation into a more realistic 3D environment, where the robot must navigate between different floors or levels of the warehouse. The concepts from the previous modules were extended to three dimensions. The environment became a 3D NumPy array, the A* algorithm searched a 3D grid, and robot kinematics were handled using 4x4 homogeneous transformation matrices [**?**].

Result: The system was successfully extended to 3D. The robot planned and executed a smooth trajectory from a start point on one level to a goal point on another, avoiding 3D obstacles. The final simulation provides a comprehensive visualization of the robot's path and orientation in 3D space, as shown in Figure 6.

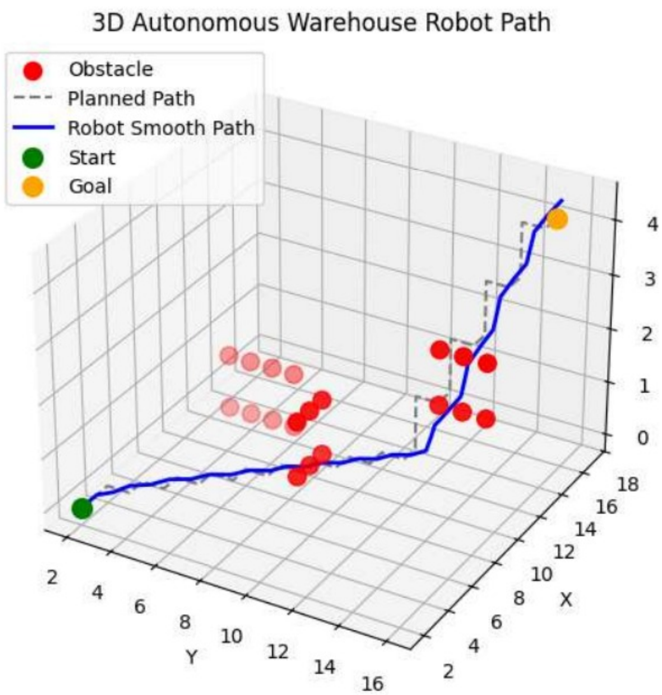

Figure 6: Final 3D simulation showing the robot's smooth path with orientation vectors (green arrows) indicating its heading in 3D space.

—

3 Conclusion

This project successfully implemented the mathematical modeling, control, and simulation of an Autonomous Warehouse Robot System. By combining A* path planning, PID motion control, and kinematic transformations, the system achieved reliable autonomous navigation with minimal error. The study illustrates how mathematical concepts such as vector calculus, optimization, and control theory form the foundation for real-world robotics applications.

4 Limitations

- The simulation assumes an ideal environment without sensor noise.
- Real-time dynamic obstacles were not considered.
- Wheel slip and actuator nonlinearity were neglected.
- Further work could include SLAM integration, real-time sensor feedback, and multi-robot coordination.

5 Acknowledgment

I would like to express sincere gratitude to my **Professor Siju Swamy** ,Department of Mathematics for his invaluable guidance, technical expertise, and continuous support throughout the development of this project. His insightful advice on Differential Equations, Optimization, Control System, Matrix Algebra, Applied Mathematics greatly contributed to the successful completion of the work. I would like to sincerely thank my **Teammates** for their hard work , dedication and collaborative sprint in all stages of the project from conceptualization and modeling to simulation and documentation . Finally, we acknowledge the use of **Generative AI Tools** for providing code, debugging, documentation refinement and improving the overall clarity of the report

6 Data and Code for the Project

All simulation codes and data files are available in the following GitHub repository: https://github.com/Minnusrd/Project

7 References

1. Norman S. Nise, *Control Systems Engineering*, 8th Edition, Wiley, 2020.
2. Richard C. Dorf and Robert H. Bishop, *Modern Control Systems*, 13th Edition, Pearson Education, 2017.
3. Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*, MIT Press, 2005.
4. Jazar, R. N., *Theory of Applied Robotics: Kinematics, Dynamics, and Control*, Springer, 2010.
5. Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D., *Introduction to Autonomous Mobile Robots*, MIT Press, 2011.