Autonomous Mobile Robot (AMR) System

Mechanical Engineering

[Task 2]

Mohamad Abdulmoula,R. Rama

ramamohameed9@gmail.com

Department of Software Engineer

Under the Supervision Of:

Eng Asmaa Duramae

July 10, 2025

Table of Contents

Glossary of Terms

Term	Definition
AMR (Autonomous Mobile Robot)	A mobile robot that navigates and performs warehouse tasks without human guidance.
WMS (Warehouse Management System)	Software for managing inventory, tasks, and robot coordination.
SLAM (Simultaneous Localization and Mapping)	A technique for constructing or updating a map while keeping track of the robot's location.
ROS 2	A robotics middleware for building robot software and enabling communication between components.
LIDAR	A sensor that uses laser pulses to measure distance and create spatial maps.
IMU (Inertial Measurement Unit)	A sensor that measures acceleration and angular velocity.
QR Code	A 2D barcode used for localization and identification within the warehouse.
PID Controller	A control loop mechanism for regulating motor and robot motion.
IEEE 1872–1	Standard for formalizing common terms and capabilities in robotics.
Digital Twin	A virtual model representing the physical warehouse environment.
DDS (Data Distribution Service)	A middleware protocol for real-time data sharing in ROS 2.

1. Project Planning

1.1 Goal

To develop a fully autonomous warehouse automation system using mobile robots that can lift and transport inventory pods to workstation zones efficiently and safely.

1.2 Scope

1. Autonomous navigation within a structured warehouse grid
2. Goods-to-person order fulfillment
3. Realistic 3D modeling in Blender
4. Path planning, docking, and lifting simulation
5. Integration of IEEE 1872-1 ontology concepts

2. Requirement Analysis

2.1 Functional Requirements

1. Autonomous movement between storage and workstation zones
2. Lifting platform for pod transport
3. Collision avoidance
4. Task execution based on WMS commands

2.2 Non-Functional Requirements

1. Modular software architecture
2. Scalable for multi-robot environments
3. High reliability and safety compliance
4. Ontological compliance with IEEE 1872–1

2.3 Tools & Technologies

1. Blender (3D modeling & animation)
2. Python (simulation scripting)
3. ROS 2 (for future integration)
4. IEEE 1872–1 standard (ontology)

3. Execution Algorithm –Style Autonomous Mobile Robot (AMR) System

Objective

To design and implement a fully autonomous robotic system capable of executing **goods t operson operations** in a warehouse. The mobile robot autonomously navigates beneath inventory pods, lifts them, and delivers them to designated workstations for order fulfillment. The design adheres to the ontology and terminology defined in **IEEE 1872–1: Standard for Ontologies for Robotics and Automation**.

High-Level Execution Algorithm

This algorithm outlines the sequential control flow of **AMR**, integrating motion planning, task execution, and inter-system communication:

1. System Initialization

1) Load the digital twin of the warehouse map, including:
    a) Receiving Zones
    b) Storage Grid
    c) Packing Stations
    d) Charging Docks
2) Activate all robot subsystems:
    a) SLAM (Simultaneous Localization and Mapping)
    b) LIDAR and proximity sensors
    c) Motor controllers and power systems
3) Establish communication with the **Warehouse Management System (WMS)** via ROS 2 middleware or MQTT broker.
4) Enter standby mode until a task is dispatched.

2. Task Assignment

1) Receive a task packet from WMS (e.g., "Transport Pod_203 to Station_4").
2) Extract pod ID, source location, and destination zone from the digital map.
3) Trigger the motion planning module to generate a shortest-path trajectory using **A*** or **Dijkstra's Algorithm**, considering current traffic and dynamic obstacles.

3. Navigation to Pod Location

1) Enable motion controller and follow the planned trajectory using **PID control**.
2) Continuously update the robot's belief state using:
   a) LIDAR for 2D spatial awareness
   b) Wheel odometry and IMU for dead-reckoning
   c) Visual markers or QR grid tags for fine localization
3) Detect and resolve obstacles via real-time sensor fusion and dynamic path re-planning.

4. Pod Lifting Sequence

1) Precisely align beneath the target pod using fine-tuned docking sensors.
2) Activate the **lifting mechanism** (telescopic or scissor-based platform).
3) Monitor load cell or hall-effect sensors to confirm pod engagement and weight.
4) Lock platform and secure pod before movement.

5. Transport to Delivery Station

1) Generate a secondary trajectory from the current location to the destination (e.g., Station_4).
2) Maintain safe operating speed and reactive obstacle avoidance.
3) Prioritize routes based on global traffic rules (e.g., one-way aisles, human-safe zones).

6. Pod Delivery

1) Align with designated drop-off zone.
2) Lower the pod smoothly to the floor using descent control logic.
3) Notify WMS via message: "Delivery of Pod_203 completed at Station_4".

7. Energy Monitoring & Recharge

1) Monitor battery SOC (State of Charge) continuously.
2) If SOC < 20%, schedule low-priority return to **Charging Dock**.
3) Perform automated docking and initiate charging cycle.
4) Resume tasking post-recharge or upon reaching SOC threshold.

4.Components of the Working Envelope

Defined per IEEE 1872–1 as the **"Operational Workspace"**—the total physical and functional space in which a robot performs valid actions.

1. Mechanical Subsystems

Component	**Description**
Chassis Base	Rigid, circular platform housing all actuators and control electronics.
Omni/Mecanum Wheels	Enable holonomic movement across 2D plane. Controlled via differential drive.
Lifting Platform	Vertical linear actuator or scissor-lift capable of lifting payloads up to 500 kg.
Battery Pack	Rechargeable lithium-ion with integrated BMS and thermal protection.

2. Perception & Sensor Subsystems

Sensor Type	Functional Role
2D/3D LIDAR	Global mapping, obstacle detection, SLAM input.
Ultrasonic / IR Sensors	Docking alignment and proximity detection.
Line / QR Code Readers	Ground-based localization and guidance.
IMU / Gyroscope	Orientation estimation, pitch/yaw/roll correction.
Wheel Encoders	Precise motion feedback for PID control loops.

3. Software Architecture

Module	Functional Role
Navigation Stack	Pathfinding via A* or Dijkstra with obstacle inflation layer.
Task Scheduler	FIFO/LIFO queue system pulling from WMS task list.
ROS 2 Communication Layer	Inter-node messaging using DDS (Data Distribution Service).
SLAM/Localization	Based on gmapping, Cartographer, or fiducial marker system.
State Machine Controller	Manages finite state transitions (Idle → Navigate → Lift → Deliver).

4. Warehouse Environment Interfaces

Zone	Function
Receiving Area	Initial intake zone for goods and inventory.
Storage Grid	Grid-based pod layout with positional tags for easy localization.
Packing Station	Human-in-the-loop interface where items are picked from delivered pods.
Charging Dock	Autonomous robot recharging interface with wireless or contact points.

5.Summary Overview

Category	Description
Robot Type	Autonomous Mobile Robot (AMR)
Primary Functions	Navigate, Dock, Lift, Deliver, Recharge
Control System	Hybrid: Centralized WMS + Onboard ROS 2 Autonomy
Operating Envelope	2D warehouse grid + 0.5m vertical lift (Z-axis support)
Safety Measures	LIDAR fail-safes, E-stop, watchdog timers, and thermal protection

6.Professional & Ontological Standards

IEEE 1872–1:2017 – Standard Ontologies for Robotics and Automation

Concept	Application
Autonomous Agent	Robot operates without continuous external control.
Task Frame	A structured set of goals from the WMS or Scheduler.
Workspace	The robot's reachable zone in 3D space (with Z-axis lift).
Environment Entity	Physical zones like storage grids, stations, and walls.
Capability	Robot's ability to perceive, move, and manipulate pods autonomously.

7.References

1. IEEE Robotics and Automation Society. (2017). *IEEE Standard Ontologies for Robotics and Automation (IEEE 1872–1)*. IEEE Standards Association. https://standards.ieee.org/ieee/1872.1/6286/