eternal

Autonomous Rover for Warehouse Rack Inventory Management

Inter IIT Tech Meet 14.0

Resonating Cascades

End-Term Report

Team ID: 42

6th December 2025

***Abstract*—This report describes the design procedure of an Autonomous Mobile Robot capable of navigating warehouse environments and performing vertical scanning operations on storage racks up to 2 meters high.. The AMR is equipped with a high-resolution visual data capture system that scans the rack contents, processes the information, and stores it in a real-time database for inventory analysis and management. Detailed design concepts, accompanied by CAD files and videos that demonstrate live operation, are included as part of the performance evaluation.**

I. Executive Summary

Inventory management is a mission-critical task for modern warehouses requiring precise tracking of Stock Keeping Units (SKUs), rapid cycle counts, and continuous updates to inventory databases. Traditionally, this has been done manually, where workers spend a large portion of their time walking long distances, searching for SKUs, climbing racks, and performing repetitive scanning tasks, which leads to fatigue, slow operations, and a high likelihood of human error. This is where Autonomous Mobile Robots (AMRs) greatly improve warehouse inventory management compared to traditional human-based systems by addressing several long-standing operational challenges. AMRs can take over repetitive and physically demanding tasks, reducing workplace accidents and increasing accuracy in less time—more suited to today's fast paced world of quick orders and deliveries. They provide continuous, reliable, and data-driven inventory tracking that enhances productivity, ensures real-time visibility, and supports faster, error-free order fulfillment.

A. Overview of Solution Approach

Here, we are going to discuss our approach to designing an Autonomous Mobile Robot capable of navigating the warehouse and performing vertical scanning. The robot is built with commercially available off-the-shelf components that can be easily assembled anywhere, making it a suitable choice for this purpose.

The proposed solution integrates SLAM-based autonomous navigation, QR scanning, a vertical lift mechanism, and data logging. The robot uses a modular hardware architecture combining differential drive motion, a camera-based scanning module, and ROS 2 powered perception and planning algorithms.

The core challenge is to develop a vertical scanning mechanism capable of positioning and reading QR codes at pre-defined rack heights across a 2-meter vertical range, while ensuring accurate alignment, stable motion control, and reliable data acquisition.

B. Key Metrics

The performance of the autonomous robotic system has been evaluated and benchmarked across motion, sensing, perception, and operational endurance. The key performance metrics are summarized below:

- **Positioning Accuracy (X–Y Axis)**
  - The robot maintains a positional accuracy of ± 25 cm in the X–Y plane over a full traversal length of up to 10 meters.
- **Vertical Lift Accuracy (Z Axis)**
  - The stepper-motor-driven lifting system achieves a precise vertical positioning accuracy of up to 2 cm.
- **Z-Axis Lift Speed**
  - The maximum achievable velocity of the vertical lift stepper motor is 100 mm/s, enabling rapid elevation adjustments for scanning and object interaction.
- **Motor Speed and Braking Performance**
  - The mobile base motors attain a maximum translational speed of 0.6 m/s.
  - Motor stopping time from 0.6 m/s to a complete stop: 2 seconds.
- **QR Code Scanning Performance**
  - Vision-based scanning detects and decodes QR markers in less than 1 second.
- **Full Rack Scanning Efficiency**
  - The entire storage rack can be scanned within 2 minutes, including robot motion, positioning, and image acquisition cycles.
- **LiDAR Range and Mapping Accuracy**
  - The LiDAR sensor provides accurate distance measurements up to 1.2 meters.
- **System Control Frequency**
  - The primary motion control loop operates at 20 Hz, ensuring stable feedback for velocity regulation and localization updates in real time.
- **Battery Life**

- The on-board power system ensures up to 1 hour of continuous operation under typical motion and perception workloads.

II. Problem Understanding and Motivation

A. Productivity

Warehouse staff spend a lot of time walking through the warehouse, searching for SKUs, climbing racks and performing repetitive scanning tasks. This results in low operational throughput and high-cycle count times, especially in case of large-scaled facilities.

AMRs can autonomously navigate through the pre-defined layouts of the warehouse, perform routine checks at intervals the operators are free to decide. This would reduce cycle-count times by 3-5 times and significantly improve operations.

B. Susceptibility to Human Errors

Manual scanning introduces inconsistencies due to operator fatigue, variable scanning angles, and improper documentation. This leads to inaccurate inventory records, misaligned stock levels, and higher chances of order fulfillment errors.

With controlled motion, fixed scanning geometry, and computer vision algorithms, AMRs ensure consistent barcode/QR readability and eliminate variability caused by human operators.

C. Limited Accessibility to Vertical Storage

Racks extending up to 2 meters require operators to climb ladders or use lifting equipment for scanning. This increases inspection time and introduces ergonomic and safety risks—particularly in narrow warehouse aisles.

The AMR integrates a vertical lift mechanism that moves the camera from bottom to top automatically. This ensures safe, fast, and complete scanning of all shelf levels without requiring human intervention.

D. Scalability

As SKU volume grows, manual inventory workflows do not scale well. Long traversal paths and repetitive scanning behaviors drastically increase workload and reduce operational agility, especially during peak seasons.

AMRs scale effortlessly—we would just need to deploy more robots or increase the scanning frequency. Their autonomous operation allows covering large areas efficiently and consistently. Robots can also operate continuously with predictable behavior and accuracy, enabling 24/7 inventory monitoring with minimal downtime.

E. Need for Data-Driven, High-Resolution Inventory Records

Modern supply chains rely heavily on fast, accurate, and data-rich inventory snapshots. Manual scanning rarely provides high-resolution images or structured digital logs, limiting traceability.

AMRs capture high-resolution shelf images, QR/Barcode metadata, and timestamped logs in a structured digital format suitable for analytics and automated reconciliation.

F. Alignment with Industry 4.0 and Smart Warehousing

Adopting AMR-based scanning aligns with Industry 4.0 goals—smart warehousing, data-driven analytics, and integration of cyber-physical systems. The shift from manual audits to automated continuous scanning significantly enhances operational visibility and scalability.

AMRs serve as cyber-physical systems that integrate mobility, perception, and real-time data communication—laying the foundation for smart warehousing, predictive analytics, and full automation.

G. Performance Objectives

Through the design and construction of a functional AMR suitable for Warehouse management system, we would like to achieve the following added improvements over the presently used solutions:

- Demonstrate a scalable warehouse automation solution
- Reduce manual efforts by more than 80%.
- Improve safety while scanning vertical heights.
- Achieve better accuracy with data logging and inventory management.
- Provide Real-Time Inventory Visibility
- Showcase Integration with Existing Warehouse Systems
- Optimize costs and durability.
- Prepare a safe, collision avoidant system for humans and robots to work together.

III. SYSTEM ARCHITECTURE

A. Power and Motor Control

The drive motors are powered directly from the robot's primary battery system, chosen to match the current and voltage requirements of the motor drivers. High-current motor driver modules interface between the motors and the embedded control electronics. Each motor driver provides:

- Direction control
- PWM-based speed control
- Overcurrent prevention
- Heat-sinks for thermal protection

This ensures reliable motor performance even during extended operation in 0℃–40℃ warehouse conditions.

B. Control Flow Architecture

The robot uses a two-tier control system to balance high-level intelligence and processing power with low-level real-time responsiveness for achieving maximum output in minimum time.

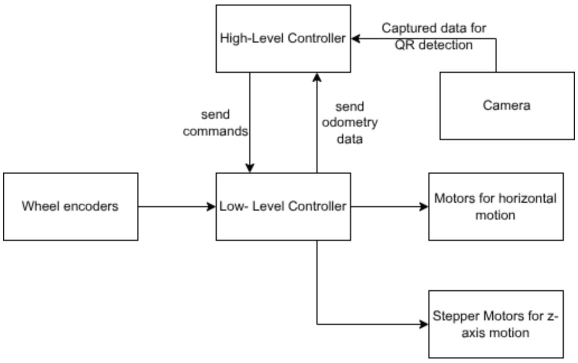

Fig. 1: 2-level Controller

1) *Low-Level Controller:* The Arduino Mega 2560 Rev3 board has been used as the onboard controller because it provides an excellent real-time performance with sufficient GPIO pins, multiple hardware serial ports, strong interrupt handling capabilities, seamless compatibility with motor drivers. It is used for:

- Receiving high-level commands from the high level controller Jetson Nano through UART/I2C/serial
- Generating PWM signals for motor drivers
- Executing differential-drive kinematics in real time
- Handling wheel encoder feedback
- Ensuring immediate response for braking or velocity correction

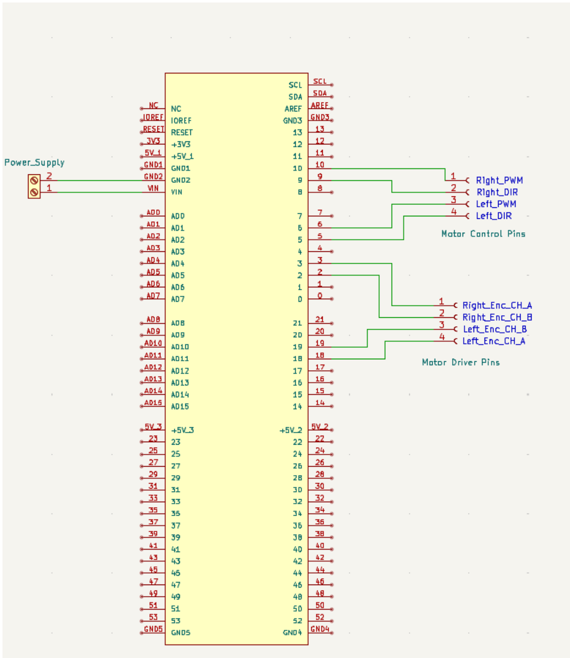

Fig. 2: Schematics

2) *High-Level Controller:* The NVIDIA Jetson Nano functions as the high-level computational unit responsible for autonomous navigation, perception, data processing, and system coordination. It runs the full ROS 2 middleware, acting as the central node that integrates the robot's sensing, decision-making, and communication subsystems. It handles the following critical tasks:

- **ROS2 based Autonomous Navigation**
  The Nano runs ROS 2 nodes for:
  - SLAM (Simultaneous Localization and Mapping) Using LiDAR data to construct and update a real-time 2D map of the warehouse.
  - Localization Tracking the robot's position on the generated map.
  - Path Planning and Path Following Generating obstacle-free trajectories and converting them into high-level movement commands.
  - The LiDAR is directly connected to the Jetson Nano via USB or serial, and its data is processed through ROS 2 packages such as slam-toolbox, nav2, or equivalent.

Algorithm 1 SLAM Algorithm for Autonomous Mobile Robot (AMR)

Initialization:
Initialize robot pose estimate $\hat{x}_0$
Initialize map representation M_0 (empty or prior map)
Initialize motion model and sensor model parameters
Set time step $t = 0$

while robot is operational **do**
    $t \leftarrow t + 1$
    Input:
    Receive control commands u_t (odometry, wheel encoders)
    Receive sensor measurements z_t (LiDAR)

    Map Update:
    Add new landmarks or update existing ones in M_t
    Update occupancy grid / feature map using z_t and $\hat{x}_t$

    Output: Updated pose estimate $\hat{x}_t$ and map M_t
end while

- **Sensor Fusion and Perception** The Nano is connected to LiDAR for scanning the environment for running Simultaneous Localization and Mapping (SLAM) algorithms and generating point clouds. The visual data captured in the camera is also processed by the Nano for QR code detection and data retrieval. The data obtained is stored in the Warehouse servers.
- **Server Communication and Monitoring** The Jetson Nano maintains wireless connectivity with warehouse infrastructure to:
  - Upload camera scans
  - Stream robot status
  - Send mapping/navigation logs
  - Fetch mission instructions or scheduled scanning tasks

  This allows real-time synchronization between the robot and the warehouse management system (WMS).

IV. HARDWARE DESIGN

The hardware is built with commercially available off-the-shelf components. The vertical lifting subsystem is designed to enable precise, stable, and repeatable elevation of the camera module for high-rack inventory scanning within warehouse environments.

A. Horizontal motion

The horizontal motion of the AMR is achieved by a differential drive mechanism, designed for high maneuverability, robustness, and precise motion control within warehouse environments.

1) Differential Drive Configuration: The platform utilizes four DC motors, arranged with two motors on each side to maximize traction, payload capacity, and stability during operation. They are controlled by commands from the onboard Microcontroller. Independent control of left and right motor groups allows the robot to: Move forward/backward by driving both sides in the same direction, perform turns by differential wheel speeds, execute zero-radius (pivot) turns by driving the sides in opposite directions. This configuration enables smooth navigation through narrow warehouse aisles, tight corners, and dynamic environments with obstacles and human movement.

B. Vertical Motion and Lift Mechanism

The robot utilizes a 2-meter aluminum double-extrusion profile as the primary structural guide, offering high rigidity, low weight, and excellent resistance to flexing under vertical loads.

A belt-driven linear actuation system is implemented to provide smooth and controlled vertical motion. At the base of the extrusion, a NEMA-series stepper motor is mounted securely using a motor-mounting plate. The motor shaft drives a timing pulley, which engages with a reinforced GT2/HTD timing belt running along the length of the extrusion.

At the upper end of the extrusion profile, an idler pulley assembly is positioned using an idler plate. The belt passes over this idler pulley, allowing for proper tensioning and minimizing backlash. The belt loop is routed such that one segment is fixed to a V-slot gantry plate equipped with precision rollers. As the stepper motor rotates, the belt's motion directly translates into linear displacement of the gantry plate along the extrusion.

The V-slot gantry plate provides stable 4- or 6-wheel contact with the extrusion profile to ensure smooth, low-friction vertical motion while maintaining alignment under lateral forces. The camera module is rigidly mounted to this gantry plate, enabling it to traverse 1.60-meter height for rack scanning operations.

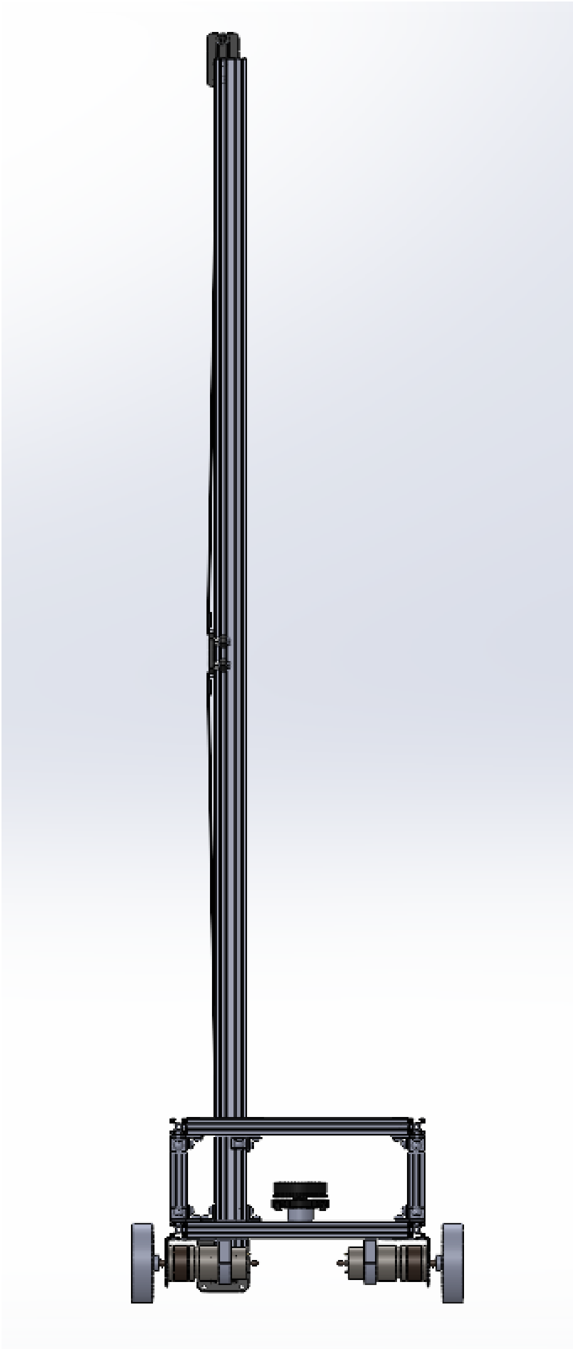

Fig. 3: AMR front view

Physical Dimensions (in mm)	
Length	550
Width	320
Track Width	400
Height (base)	152
Maximum Height	1600
Ground Clearance	83
Wheel Diameter	100

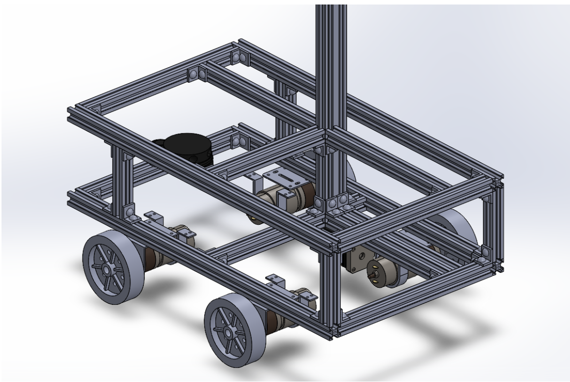

Fig. 4: AMR Isometric view

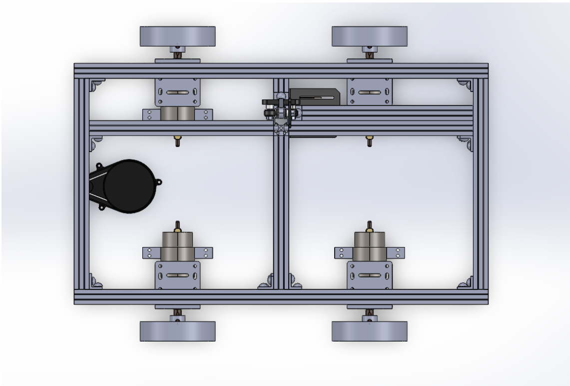

Fig. 5: Top view

Mechanical Characteristics	
Total Robot Weight	15kg
Structural Design	Double-layer extrusion frame with cross-bracing
Mounting Interfaces	V-slot rails, universal mounting plate

Power System	
Battery Type	LiPo (lithium-polymer)
Voltage & Capacity	22.2V, 20000 mAh
Battery Placement	Central compartment for balanced weight

Optimum Environmental Conditions	
Operating Temperature	0℃ to 40℃
Humidity Tolerance	20 %
Dust Protection	Basic housing

V. SOFTWARE ARCHITECTURE

The software system of the AMR is built on top of a modular, ROS 2–based architecture that integrates navigation, perception, motion control, and high-level coordination. The software stack

Current Analysis	Value
Jetson Power Draw	2A
Arduino Mega (each)	0.5A
DC Motor Power (ideal)	2A $\times$ 4 = 8A
DC Motor Power (maximum)	4A $\times$ 4 = 16A
Stepper Motor (each)	2.8A
Camera & LiDAR	Powered by Jetson
Motor Driver Rating	10A (normal), 30A (peak)
Stepper Motor Driver Rating	4A

runs primarily on the NVIDIA Jetson Nano, while low-level actuation is handled by the Arduino Mega. The system supports autonomous warehouse navigation, LiDAR-based mapping, Z-axis scanning, QR-code detection, and structured inventory data storage.

A. Operating Framework: ROS 2

The robot uses ROS 2 as its core middleware. ROS 2 provides:Real-time communication between software nodes, Sensor integration, Navigation algorithms, Visualization tools, Modular software scalability

ROS 2 nodes running on the Jetson Nano interact with the LiDAR, Z-axis lift systems, camera, and the Arduino Mega.

B. SLAM and Localization

For mapping and continuous localization, the system uses slam_toolbox, a ROS 2 package optimized for real-time warehouse-scale SLAM.

- 2D LiDAR-based SLAM
- Pose-graph optimization
- Loop closure
- Real-time map correction
- Persistent map saving

C. Odometry

Odometry is obtained by fusing multiple sensor sources:

- Wheel Encoders measure how much the wheels have rotated, allowing the robot to estimate how much it has moved. When we combine these measures for left and right wheels, we can get the position and orientation.

D. Navigation Stack

The autonomous navigation layer is built using Nav2, the standard ROS 2 navigation framework. Nav2 handles:

- Global path planning
- Local obstacle avoidance
- Costmap generation
- Path smoothening
- Behavior trees for mission execution

Using LiDAR data and the SLAM-generated map, Nav2 computes collision-free trajectories between rack aisles and scanning locations.

E. High-Level Command Execution

The Jetson Nano acts as the decision-making brain. All path planning, SLAM, and perception algorithms run on the Nano. Once the desired linear and angular velocities are computed, they are transmitted to the Arduino Mega using a serial protocol.

Jetson to Arduino Communication:

- Jetson Nano publishes /cmd_vel velocity commands
- A bridge script converts ROS 2 velocity messages into serial commands
- Arduino Mega receives:
- Target linear velocity
- Target angular velocity
- Motion mode (forward, rotate, stop)
- Arduino handles real-time PWM generation for all four motors

This ensures smooth, deterministic execution of motion commands without OS-induced delays on the Jetson.

F. Imaging & QR Code Scanning Pipeline

The AMR performs vertical rack scanning using a stepper-driven Z-axis lift equipped with a camera. The scanning pipeline operates in discrete height levels that match warehouse shelf elevations.

1) Controlled Z-Axis Positioning: During a scan:

- Nav2 positions the robot in front of the rack.
- Motion is halted to eliminate vibrations.
- The Z-axis controller raises/lowers the camera to predefined shelf heights.
- Stabilization delay ensures sharp images.

Algorithm 2 Camera Position Control using Stepper Motors

Input: Target rack heights $H = \{45, 87, 130\}$ cm
Constants:
  $S_{rev} = 200$ steps per revolution
  $D_{rev} = 5.027$ cm belt travel per revolution
  $S_{cm} = S_{rev}/D_{rev}$ steps per cm

ROS2 Node (Publisher):
Initialize $prev\_height = 0$
for each h in H **do**
  Compute incremental distance:

$$d = h - prev\_height$$

  Compute required steps:

$$steps = d \times S_{cm}$$

  Publish $steps$ on topic `/camera_lift/steps`
  Update previous height:

$$prev\_height = h$$

end for
Arduino Node (Subscriber):
while message received from topic `/camera_lift/steps` **do**
  Read incoming step command $steps$
  Drive the stepper motor for $steps$ steps
  Stop motor movement
  Wait for 5 seconds to capture QR
end while

Camera Operation:
for each halt position **do**
  Run detection script
  Extract QR data and append to JSON file
end for
End Algorithm

2) QR Code Detection (OpenCV + Python): A Python script running on the Jetson Nano uses OpenCV QR detection to process each captured image:

- Detects and decodes QR codes
- Extracted data is of the form `RACKID_SHELFID_ITEMCODE` which is parsed and stored in a CSV file
- Validates code integrity
- Handles low-contrast cases with preprocessing (thresholding, sharpening, etc.)

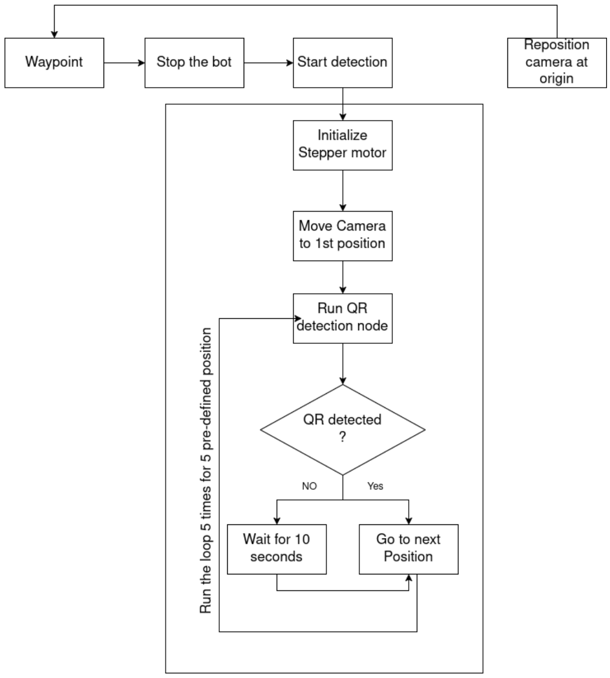

Fig. 6: QR detection mechanism

VI. Integration and Control

The autonomous mobile robot (AMR) integrates hardware and software components to achieve precise navigation and task execution within warehouse environments. Control and motion execution are handled using the ROS2 framework, which provides modular and real-time communication between nodes.

A. Motion Control

- The AMR uses **ROS2 Control** to interface with motor drivers and sensors, providing a hardware-agnostic control layer. Wheel velocities and actuator commands are sent via ROS2 controllers, allowing seamless integration with higher-level planning.
- Sensor feedback, including wheel encoders
- A PID controller processes velocity errors and generates motor commands, ensuring accurate tracking of the desired trajectory.

B. Navigation

- The AMR employs **Nav2** stack for autonomous navigation. It performs mapping, localization, path planning, and obstacle avoidance within the warehouse.
- Real-time sensor data is fused to create an up-to-date occupancy grid, which is used by the planner to compute collision-free paths.

- Motion commands from the planner are translated into wheel velocity setpoints, which are tracked by the PID controller via ROS2 Control.

C. Integration Workflow

- High-level navigation goals are sent to the Nav2 stack, which computes safe paths for the robot to follow.
- ROS2 topics and services handle communication between the navigation stack, perception nodes, and low-level control nodes.
- Safety mechanisms, including emergency stops and collision detection, are integrated to override commands if required, ensuring reliable and safe operation.

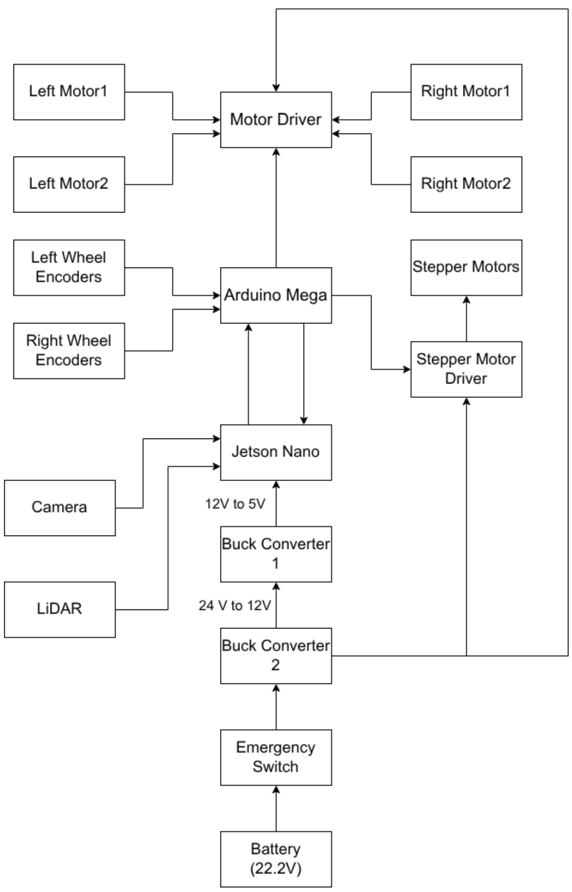

Fig. 7: Circuit Connection Flow

Algorithm 3 Updating positions from Wheel Encoder

Initialization:
Set PID gains: K_p, K_i, K_d
Initialize error, integral, and derivative terms to zero
Initialize robot pose variables (x, y, θ)

Loop executed every fixed time interval:
Read the current encoder tick counts for the left and right wheels.
Compute tick difference:
$\Delta ticks = ticks_{\text{current}} - ticks_{\text{previous}}$
Calculate RPM from wheel encoder data:
$\omega = \frac{\Delta ticks}{N_{\text{rev}}} \cdot \frac{60}{\Delta t}$, where N_{rev} = ticks per revolution
Convert wheel speeds into linear velocities for each wheel:
$v_{\text{measured}} = \omega \cdot R \cdot \frac{2\pi}{60}$
Compute measured body velocities:
$\omega_{\text{body}} = \frac{v_{\text{right}} - v_{\text{left}}}{W}$
$v_{\text{body}} = \frac{v_{\text{left}} + v_{\text{right}}}{2}$
Update robot pose (x, y, θ):
$x \leftarrow x + v_{\text{body}} \cos(\theta)\, \Delta t$
$y \leftarrow y + v_{\text{body}} \sin(\theta)\, \Delta t$

Algorithm 4 PID Error and Target RPM Calculation

Compute target wheel RPM from commanded velocity:
$\omega_{\text{target}} = \frac{v_{\text{target}} \cdot 60}{2\pi R}$
Calculate wheel velocity error:
$\omega_{\text{error}} = \omega_{\text{target}} - \omega_{\text{measured}}$
Update integral term:
$I \leftarrow I + \omega_{\text{error}} \cdot \Delta t$
Compute derivative term:
$D = \frac{\omega_{\text{error}} - \omega_{\text{error}}^{\text{prev}}}{\Delta t}$
Store current error for next iteration:
$\omega_{\text{error}}^{\text{prev}} \leftarrow \omega_{\text{error}}$
Compute PID output as: $u = K_p \cdot \omega_{\text{error}} + K_i \cdot I + K_d \cdot D$

VII. Testing and Validation

To evaluate the performance, reliability, and operational robustness of the autonomous warehouse-inspection robot, a structured testing and validation procedure was conducted. All experiments were performed inside a controlled arena designed to resemble a real warehouse layout. The arena consisted of consecutive racks arranged in parallel aisles, allowing the robot to navigate, scan, and record data from each rack in a manner consistent with industrial storage environments. Each rack contained multiple QR-coded markers, and the robot was tasked with scanning them sequentially using its onboard perception and vertical lift system.

A. Motion and Navigation Validation

The robot's movement through the warehouse aisles was assessed by measuring its X–Y planar positioning accuracy over repeated traversals of up to 10 meters. The system consistently maintained a positional deviation of ±25 cm, confirming that the fusion of wheel odometry and LiDAR-based sensing provided stable, repeatable navigation in cluttered indoor spaces.

The motor speed and braking performance were also evaluated. The mobile base achieved a peak translational velocity of 0.6 m/s, and braking tests showed that the robot could decelerate from maximum speed to a complete stop within 2 seconds. These tests validated that the motion controllers ensured both responsiveness and safety when operating around racks and confined spaces.

B. Vertical Lift System Testing

The vertical scanning mechanism, powered by a stepper motor, was tested across multiple rack levels to verify precision and repeatability. The lift system demonstrated a vertical positioning accuracy of 2 cm, allowing the camera to align correctly with QR codes placed at different heights. The lift could reach its maximum speed of 100 mm/s, enabling quick transitions between rack levels without compromising stability.

C. Perception and QR Scanning Validation

The vision pipeline was evaluated using QR markers affixed on every rack surface. Testing showed that the system could detect and decode each QR code in under 1 second, even under minor lighting variations. For full-rack testing, the robot completed the entire scanning sequence—including navigation, lift adjustments, and image acquisition—within 2 minutes per rack, demonstrating high operational efficiency.

D. LiDAR-Based Mapping and Sensing

To validate environmental awareness, the onboard LiDAR was tested for range and measurement consistency. The sensor reliably measured distances up to 6 meters with high accuracy, which was crucial for real-time obstacle detection and localization during aisle traversal.

E. System Control Stability

The robot's real-time performance was further validated through monitoring of the control loops. The primary motion control system operated at 20 Hz, ensuring that velocity control, pose estimation, and feedback corrections occurred with sufficient frequency for stable autonomous behavior in dynamic warehouse environments.

F. Power and Endurance Testing

Finally, battery endurance tests were conducted under typical navigation and perception workloads. The system sustained up to 1 hour of continuous operation, confirming that the onboard power distribution and consumption were optimized for warehouse-scale inspection tasks.

VIII. LIMITATIONS AND FUTURE SCOPE

A. Current Limitations

1) Linear Z-Axis Lift Mechanism (Fixed 2m Rod): The current vertical scanning mechanism uses a rigid 2m linear rod with a sliding camera mount. While functional, this approach poses several limitations:

- **High Structural Load:** The long, fully exposed rod introduces bending stress and requires additional stabilizers to prevent oscillation.
- **Poor Deployability:** The mechanism remains continuously extended, increasing the center of mass and limiting movement under low-clearance obstacles.
- **Vibration Sensitivity:** The long unsupported length amplifies micro-vibrations, causing motion blur and inconsistent QR detection.
- **Energy Inefficiency:** The lift motor must continuously counteract the entire mass of the rod, reducing system efficiency.
- **Mechanical Wear:** Sliding bearings and rails experience increased friction due to long-stroke operation.

2) Wiring, Cable Routing, and Connector Reliability: The current wiring approach limits long-term durability and robustness:

- External wiring along the lift increases the likelihood of cable fatigue and entanglement.
- Lack of strain-relief and proper flex channels leads to connector stress and electrical noise.
- Exposed signal and power lines are vulnerable to dust, collisions, and motion-induced degradation.

3) Structural and System-Level Design Constraints:

- Limited spatial optimization results in tightly packed components that reduce maintainability.
- Sensor mounts rely on basic brackets, limiting calibration accuracy and requiring frequent manual adjustments.
- The non-collapsible mast contributes to suboptimal weight distribution and reduced navigation stability.

4) Partial Automation in Scanning Workflow:

- The scanning process requires predefined height inputs rather than adaptive shelf-height detection.
- Limited scan-field optimization may cause redundant captures or incomplete coverage.

B. Future Scope and Improvements

1) Telescopic Mast-Based Z-Axis Mechanism: Replacing the current rod with a multi-stage telescopic mast can significantly improve performance:

- **Compactness:** Collapses into a low profile during navigation, improving safety and maneuverability.
- **Higher Stability:** Multi-stage structures offer improved rigidity and reduced vibration.
- **Energy Efficiency:** Less mass is lifted per stage, reducing motor load.
- **Mechanical Protection:** Enclosed mast housing protects the camera and cables when retracted.

2) Improved Cable Management Using Drag Chains:

- Integration of cable drag chains ensures stress-free motion along the Z-axis.
- Industrial-grade connectors (M8/M12) improve vibration resistance.
- Shielded cables and ferrite cores can reduce electromagnetic interference.

3) Modular and Service-Friendly Mechanical Architecture:

- Modular sensor mounts enable fast replacement and easy recalibration.
- Optimized chassis design using lightweight profile extrusions enhances internal routing.
- Repositioned battery and electronics improve center-of-gravity and stability.

4) Advanced Scanning Intelligence:

- Dynamic shelf-height detection using depth cameras or ultrasonic sensors.
- Adaptive camera exposure and focus for varying warehouse lighting.
- Real-time completeness validation to ensure no gaps in scanning.
- Fusion of visual SLAM with scan data for improved localization accuracy.

5) Enhanced Autonomy and Navigation:

- Multi-floor map support and coordinated multi-robot navigation.
- Integration with warehouse management systems (WMS) for automated task scheduling.
- Dead-reckoning correction via sensor fusion (encoder + LiDAR).

6) Industrial-Grade Shock Isolation:

- Use of isolation mounts or passive dampers for improved scan stability.
- Future integration of active stabilization for high-precision camera imaging.

IX. USER MANUAL

While the robot is capable of autonomous navigation , basic operational safety standards and practices must be practiced while using it on a day-to-day basis.

A. Safety Guidelines

1) General:

- Do not operate the AMR in areas with unauthorized personnel.
- Keep hands, clothing, and loose items away from wheels, lift mechanisms, and moving parts.
- Ensure the robot operates only on clean, flat surfaces free of debris.
- Do not attach external equipment without approval.

2) Electrical:

- Only trained personnel should handle the battery connection or charger.
- Avoid exposing the robot to liquids, extreme heat, or corrosive environments.
- Disconnect power before performing any repairs or modifications.

3) Mechanical:

- Keep hands away from the vertical lift assembly during operation.
- Do not force or manually pull the lift—this could damage motors or gears.

4) Operational Safety:

- Maintain at least 1 meter of clearance around the robot during operation.
- Ensure emergency stop (E-Stop) buttons are functional before every run.
- Do not obstruct or block the LiDAR, camera, or ultrasonic sensors.
- If the AMR behaves unpredictably, press the E-Stop immediately.

B. Operation Guide

1) Pre-Operation Checklist: Before powering the AMR:

- Verify battery level is above 30
- Inspect wheels and lift for debris or obstructions.
- Ensure LiDAR and camera lenses are clean.
- Confirm Wi-Fi or wired communication links are active (if applicable).
- Test the E-Stop.

2) Powering ON:

- Turn ON the main power switch.
- Wait for onboard computer to boot (10–20s).
- Check LED indicators for system health:
  Green: ready
  Yellow: warnings (e.g., low battery)
  Red: fault detected

3) Navigation & Scanning Operation:

- Select the scan route from the control interface.
- Robot begins navigation and raises the camera to assigned rack heights.
- QR/Barcode data is captured and logged.
- When the scan run completes:
- Robot returns to the starting point or docking station.
- Data is uploaded to the inventory system.

4) Powering OFF:

- Park the robot in its safe zone/dock.
- Turn OFF the main power switch.

C. Emergency Guidelines

- Press the red Emergency Stop (E-Stop) button to immediately cut power to motors.
- To resume: twist and release the E-Stop, then restart the robot.
- Battery Emergency:
  - If battery overheats or swells:
  - Power off immediately
  - Move robot away from personnel
  - Contact maintenance team

D. Storage Guidelines

- Store the AMR in a cool, dry, dust-free environment.
- Avoid exposure to direct sunlight, moisture, or extreme temperatures.
- Maintain the battery between 40–60% charge for long-term storage.
- Cover LiDAR, camera, and lift mechanism with protective dust caps if unused for more than 7 days.
- Power off the robot completely before storage.

E. Transportation Guidelines

- Use the provided transport locks to secure the lift mechanism during movement.
- Turn off the main power switch before transportation.
- Ensure the robot is positioned upright; avoid tilting or flipping.
- Use a padded crate or trolley for long-distance transport to avoid vibration damage.
- Avoid stacking heavy objects on the robot.
- If transporting by vehicle, secure the robot with straps to prevent shifting.

X. MAINTENANCE

Proper maintenance ensures reliable operation, longer component lifespan, and consistent scanning accuracy. This section outlines preventive maintenance intervals, replacement parts, and guidelines for safe storage and transportation of the AMR.

A. Daily Maintenance

- Clean camera and LiDAR lenses with microfibre cloth.
- Inspect wheels and castor rollers for dust and dirt.
- Ensure the battery level is sufficient and check for abnormal heating.
- Check vertical lift rails for smooth travel and look for obstructions.

B. Weekly Maintenance

- Validate Encoder readings for detecting drift or slippage in wheels
- Examine the lift mechanism, checking belt tension and alignment.
- Inspect all wires for connections and proper insulation.
- Clean air vents, cooling fans, and onboard computer housing.

C. Monthly Maintenance

- Lubricate lift rails, lead screws, or belts (if applicable) using manufacturer-recommended lubricants.
- Inspect motor mounts, gearboxes, and chassis frame for cracks or deformation.
- Test wheel alignment and adjust caster orientation if drift is observed.
- Perform a full software/firmware update, including ROS packages and navigation maps.
- Conduct a complete functional audit: navigation, scanning, lift actuation, and battery performance.

D. Replacement of Parts

Only authorized replacement components should be used to ensure safety and compatibility.

- Wheel assembly: wheel, encoder, motor shaft coupling
- Battery pack: Li-ion or LiFePO4 module
- Lift system components: belts, lead screws, pulleys, bearings
- LiDAR lens cover (if scratched or cracked)
- Camera module: lens, mount, or protective casing
- Fuses, wiring harnesses, connectors
- Cooling fans and filters

E. Replacement Guidelines

- Always power off the AMR before replacing any part.
- Use only manufacturer-recommended components to avoid calibration issues.
- After replacing lift or drive components, perform a calibration check.
- Keep a log of all part replacements for maintenance records.

XI. References

[1] ROS 2 Humble Documentation, Open Robotics, 2020. Available online: https://docs.ros.org/en/humble/index.html (accessed: 2025).

[2] G. Bradski, "The OpenCV Library," *OpenCV.org*, 2000–present. Available: https://opencv.org/ (accessed: 2025).

[3] OpenCV QR Code Detector API Documentation, *OpenCV.org*, 2024. Available: https://docs.opencv.org/ (accessed: 2025).

[4] AccelStepper Library: Stepper Motor Control with Acceleration, *AirSpayce*, 2024. Available: https://www.airspayce.com/mikem/arduino/AccelStepper/ (accessed: 2025).

[5] Nav2 (Navigation2) ROS Package Documentation, *Open Robotics*, 2024. Available: https://github.com/ros-navigation/navigation2/ (accessed: 2025).

[6] S. Kohlbrecher et al., "Hector SLAM: A Flexible SLAM System for Robots and UAVs," *University of Technology Darmstadt*, 2011. Available: https://wiki.ros.org/hector_slam (accessed: 2025).

[7] USB_cam ROS Package Documentation, *ROS Community*, 2023. Available: https://github.com/ros-drivers/usb_cam (accessed: 2025).

[8] Jetson Nano Developer Kit User Guide, *NVIDIA Corporation*, 2024. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed: 2025).

[9] RPLIDAR A1M8 Product Specifications, *Slamtec*, 2023. Available: https://www.slamtec.com/en/Lidar/A1 (accessed: 2025).