Assignment 2: Implementation of Phases in Robotics

Course: AIML ZG528 – AI and ML for Robotics
2024aa05612- Najmuddin A k Warsi - 100%

2024aa05661- Sumit Kumar - 100%

2024aa05662- Binit Kumar - 100%

2024aa05693- Radhakrushna Mohanty - 100%
Domain: Warehouse Logistics and Inventory Movement

1. Objective

The objective of this assignment is to design and implement key phases of a mobile robotic system in a warehouse logistics environment. The robot must be capable of localizing itself, building a map of the environment, planning an optimal path, and navigating autonomously. Machine learning techniques are integrated to enhance environmental understanding and obstacle classification.

2. Domain

The selected domain is warehouse logistics and inventory movement. In this domain, autonomous robots are deployed to transport goods efficiently between storage locations and delivery points. These robots must operate safely in environments with obstacles such as shelves, walls, and dynamic objects. Automation improves operational efficiency, reduces human effort, and minimizes errors.

3. Scope of Work

The scope of this project includes the simulation and implementation of key robotics phases using Python. The project simulates a warehouse environment represented as a grid map. The robot performs localization using Monte Carlo Localization, builds a map using SLAM techniques, and finds optimal paths using the A* planning algorithm. Additionally, a machine learning algorithm is used to classify obstacles and improve navigation efficiency.

4. Future Enhancement and Optimization

Future enhancements may include integration with real robotic hardware and sensors such as LiDAR and cameras. Advanced SLAM algorithms such as Graph SLAM can improve mapping accuracy. Deep learning methods can be used for object detection and classification. Real-time ROS2 integration and multi-robot coordination can further enhance system scalability and efficiency.

5. Key Design Aspects of Mobile Robotics Phases

A. Deployment Platform

The robot is deployed in a simulated warehouse environment represented as a grid. The platform supports autonomous navigation, localization, and obstacle avoidance. Python and simulation tools are used to model the robot and environment.

B. Sensors and Types

The robot uses simulated sensors to detect obstacles and landmarks. These include virtual proximity sensors and landmark detection mechanisms. These sensors provide positional and environmental information required for localization and mapping.

C. Perception: Sensor Measurement Model and Motion Model

The motion model updates the robot's position based on movement commands. The sensor measurement model calculates distances between the robot and landmarks or obstacles. These measurements help in estimating the robot’s position accurately.

D. Localization and Mapping

Monte Carlo Localization (MCL) is used to estimate the robot's position using particle filters. Feature-based SLAM is used to create a map of the environment and track landmarks. These methods allow the robot to understand its position and surroundings simultaneously.

E. Trajectory Planning

The A* algorithm is used for trajectory planning. It calculates the shortest path from the start position to the goal while avoiding obstacles. The heuristic function improves efficiency by guiding the search process toward the goal.

F. Control Technique

The robot follows the planned path using discrete movement commands. The control system ensures the robot moves safely without colliding with obstacles. This allows efficient navigation within the warehouse.

G. Communication Infrastructure

The system uses software-based communication between modules such as localization, mapping, planning, and machine learning components. These modules share environmental and positional information for coordinated operation.

6. Machine Learning Integration

A Decision Tree machine learning algorithm is used to classify grid positions as obstacles or free space. The model is trained using environment data and predicts obstacle locations.

This improves navigation by allowing the robot to make intelligent decisions based on learned information.

Justification:
Machine learning enhances environmental understanding and improves obstacle detection accuracy. It helps the robot adapt to different environments and improves navigation performance.

7. Implementation and Results

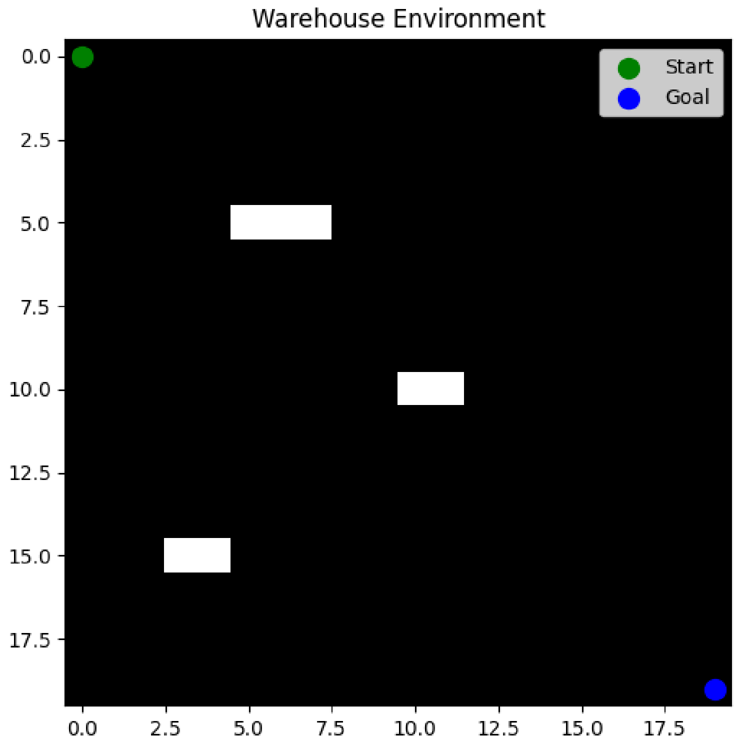
The system was implemented using Python and simulated in a warehouse grid environment. The following algorithms were successfully implemented:

- Monte Carlo Localization for position estimation
- Feature-based SLAM for map creation
- A* algorithm for path planning
- Machine learning for obstacle classification

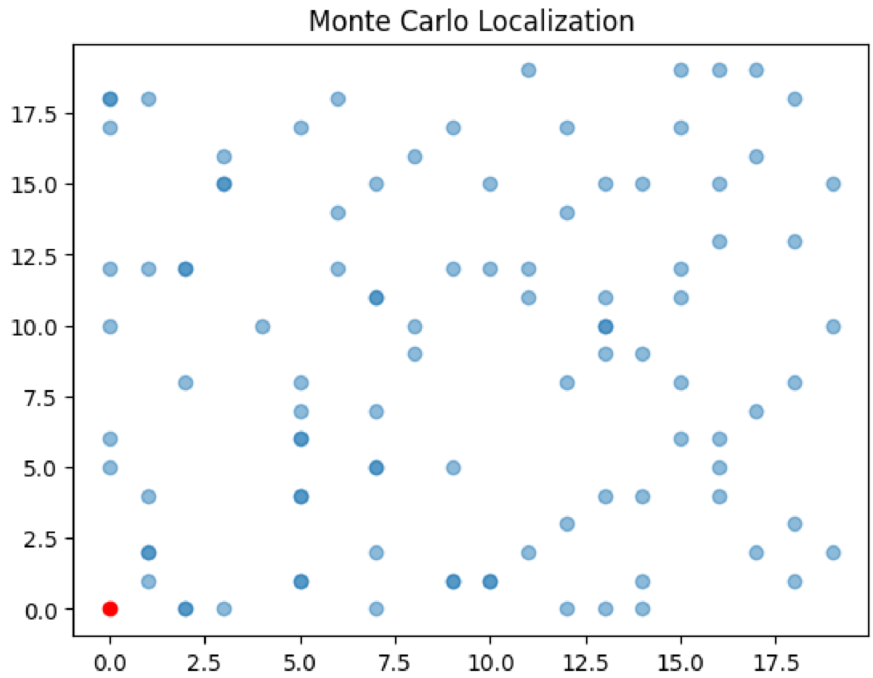
The robot successfully localized itself, created a map, planned an optimal path, and navigated to the goal without collision.

8. Simulation Output

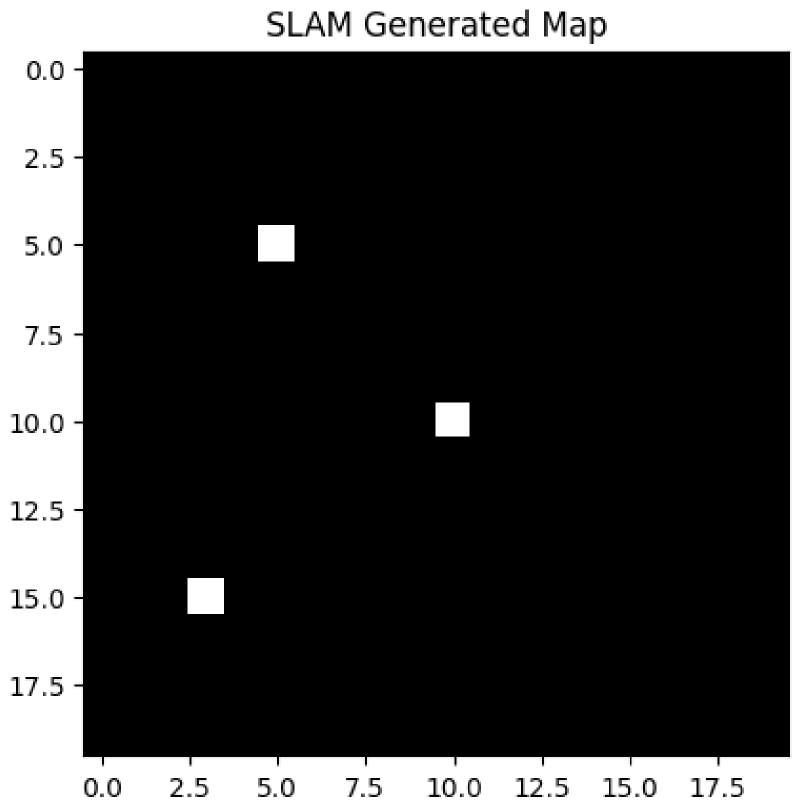
1. Warehouse Environment Grid



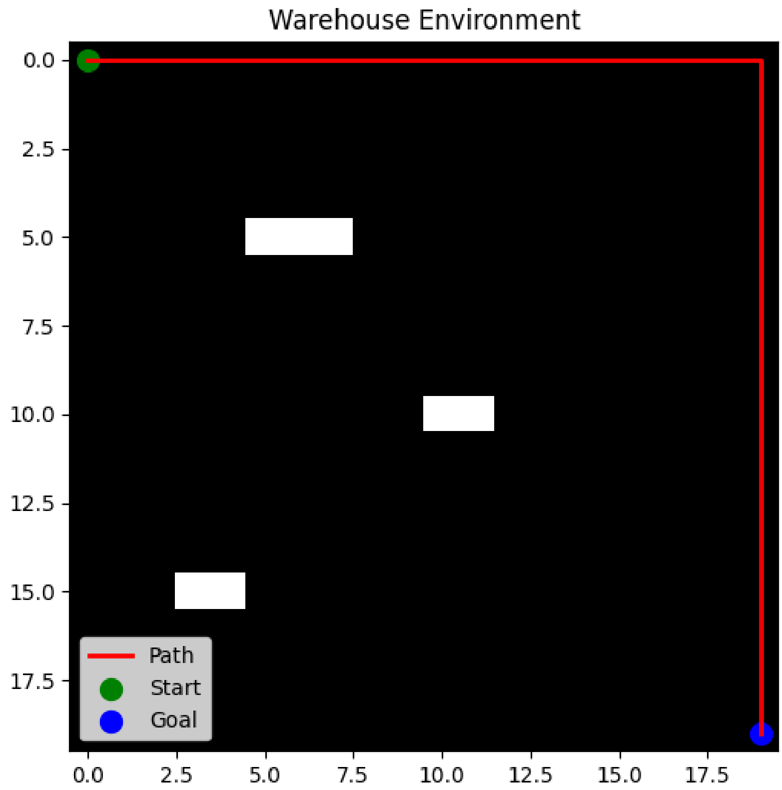
• Monte Carlo Localization (Particles Visualization)



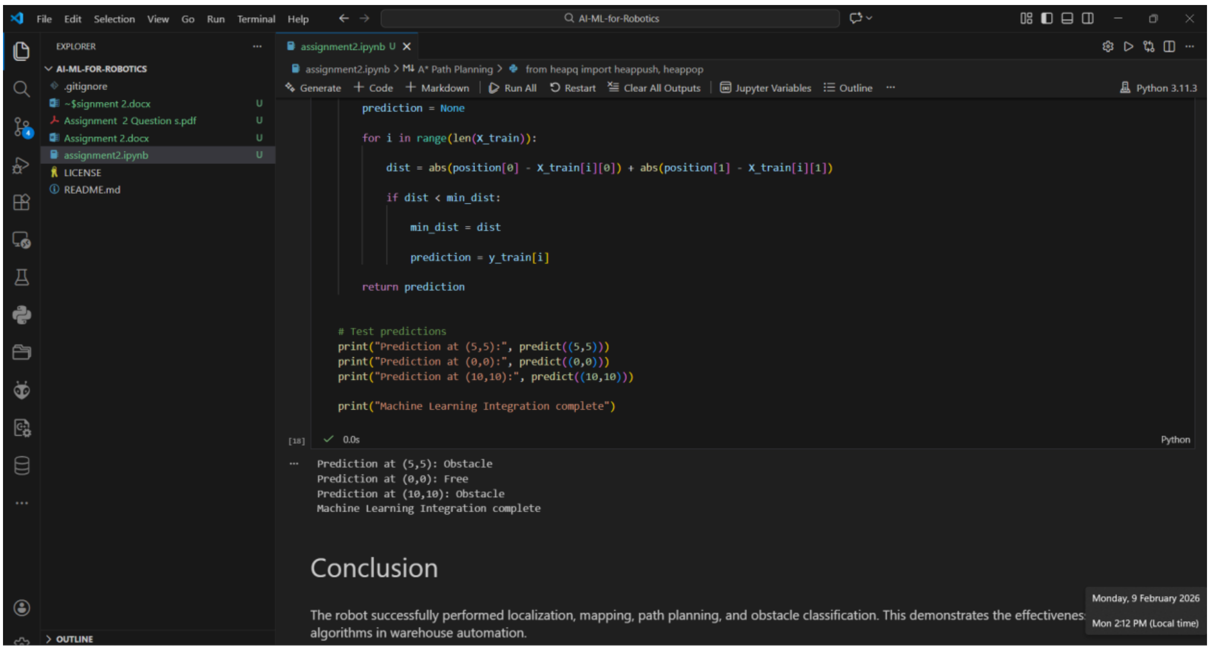
• SLAM Map



• A* Path Planning Output



• Machine Learning Output



9. Conclusion

This project successfully demonstrates the implementation of core robotics phases including localization, mapping, planning, and machine learning integration. The robot was able to navigate autonomously in a simulated warehouse environment. The results validate the effectiveness of Monte Carlo Localization, SLAM, and A* planning algorithms. Machine learning improved obstacle classification and decision-making. This project highlights the importance of robotics and AI in modern warehouse automation systems.