**Design an autonomous robot for indoor navigation using A\* search for path planning while avoiding obstacles. Equipped with sensors, it dynamically updates its environment knowledge, ensuring real-time path adjustments.**

**Introduction**

This report presents the development and implementation of two essential classes—Node and Robot—used in autonomous robot path planning and navigation. The Node class is fundamental to the A\* search algorithm, while the Robot class employs unicycle kinematics for motion control and path execution. The integration of these classes enables efficient pathfinding and real-time navigation.

**Node Class**

**Overview**

The Node class is a fundamental data structure in the A\* path planning algorithm. It stores positional data and associated path costs, facilitating efficient shortest-path computations.

**Key Components**

**-Position:** Stores node coordinates (x, y).

**-Parent:** Tracks the predecessor node in the path.

- **Cost Variables:**

- `g`: Cost from start node to current node.

- `h`: Estimated heuristic cost to the goal.

- `f`: Total cost (f = g + h).

**Robot Class**

**Overview**

The Robot class models a mobile robot employing unicycle kinematics. It integrates A\* path planning, obstacle avoidance, and motion control.

**Key Features**

**Unicycle Model Parameters:**

- Position (x, y)

- Orientation (theta)

- Wheel velocity parameters

- Time step integration

**Path Planning:**

- A\* algorithm for optimal path selection

- Dynamic obstacle avoidance

- Real-time path replanning

**Visualization**:

- Graphical rendering of paths and obstacles

- Real-time pose updates

**Performance Characteristics**

Path planning efficiency: O(n log n) complexity.

Real-time adaptability: Adjusts paths dynamically.

Obstacle handling: Efficiently avoids and navigates around obstacles.

**Integration and Usage**

The Node and Robot classes collaborate through the following workflow:

**Robot Initialization:** The robot's initial state and environment are set up.

**Path Planning:** The A\* algorithm computes the optimal path using Node instances.

**Motion Execution:** The robot follows the computed waypoints using the unicycle model.

**Dynamic Replanning**: If obstacles are detected, the robot dynamically adjusts its path.

**Conclusion**

The Node and Robot classes provide a foundational framework for autonomous navigation. By integrating A\* search with unicycle-based motion control, the system ensures efficient and adaptive path planning. These implementations facilitate real-world applications in autonomous robotics and intelligent navigation systems.