Robot Path Planning and Navigation using A* Algorithm and Dynamic Window Approach

Muskan Dewangan || Arya Mahesh Bhiwapurkar || Chandan Singh Chauhan

International Institute of Information Technology , Naya Raipur

INTRODUCTION

Path planning is the task of finding a path from a starting point to a goal point in a given environment. In the context of mobile robots, the path must also take into account the presence of obstacles.

A* Algorithm (Global Path Planning)

- A* is a best-first search algorithm that uses a heuristic function to guide the search process
- Combines actual cost from start (g) and estimated cost to goal (h) to form f(n) = g(n) + h(n)
- Explores nodes with the lowest f-cost first, ensuring optimality
- Uses a priority queue (open list) and visited set (closed list) to manage the search
- Efficiently finds the shortest path in static environments

Dynamic Window Approach (Local Navigation)

- DWA is a velocity-based local navigation method that accounts for robot dynamics
- Samples possible velocity commands within the robot's dynamic constraints
- Predicts trajectories for each velocity pair and evaluates them
- Selects the optimal velocity command based on multiple objectives (goal direction, clearance, velocity)
- Enables real-time obstacle avoidance while following the global path

OBJECTIVE

1. To implement the **A* algorithm** for finding an optimal global path from start to goal position in an environment with static obstacles.
2. To develop a Dynamic Window Approach controller that enables a robot to follow the generated path while avoiding obstacles.
3. To demonstrate successful path following behavior through simulation in a 2D environment.
4. To analyze the effectiveness of the combined approach by comparing the planned path with the actual robot trajectory.

INTEGRATION APPROACH

A* algorithm computes the global path once at the beginning
DWA controller:

- Takes current robot state and global path as input
- Samples velocities within robot's dynamic constraints
- Selects optimal velocity commands to follow the path while avoiding obstacles
- Updates at each time step (dt = 0.1s)

The robot follows waypoints in sequence until reaching the goal position

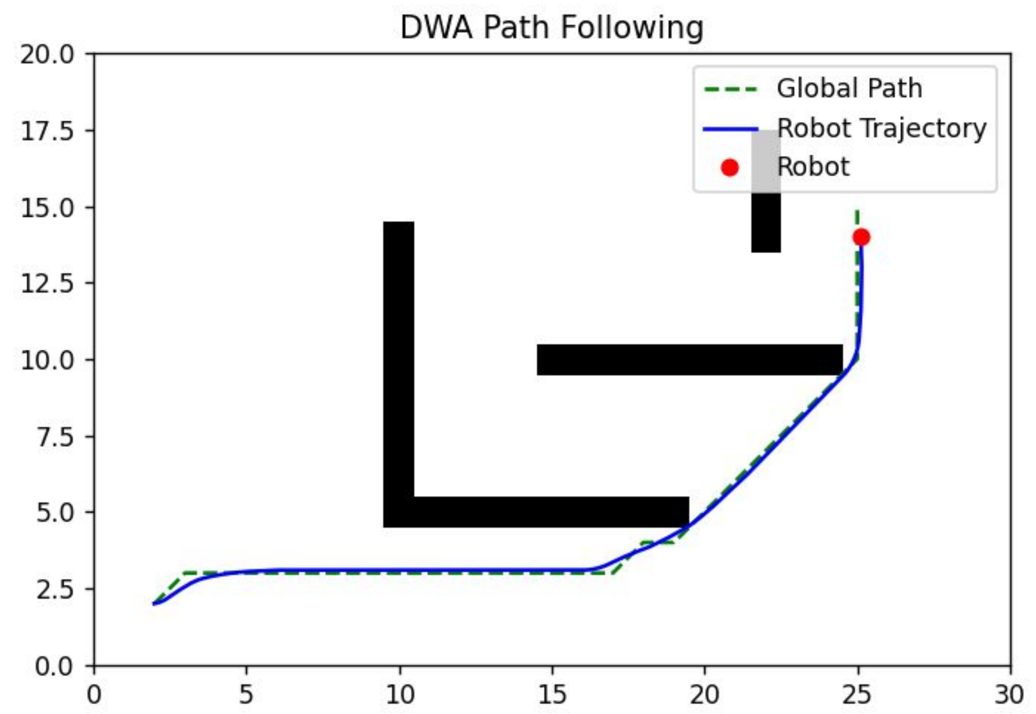
METHODOLOGY

A* Algorithm (Global Path Planning)

The A* algorithm creates an optimal path from start to goal by evaluating nodes based on:

- g(n): Cost from start to current node
- h(n): Heuristic estimate (Euclidean distance) to goal
- f(n) = g(n) + h(n): Total estimated cost.

```
Input: grid, start, goal
Output: path from start to goal
Initialize open_list with start node;
Initialize closed_set as empty;
while open_list is not empty do
    current ← node with lowest f(n) from open_list;
    if current = goal then
        return reconstruct_path(current);
    end
    Add current to closed_set;
    foreach neighbor of current do
        if neighbor ∈ closed_set or is obstacle then
            ;
        end
        tentative_g ← g(current) + distance(current, neighbor);
        if neighbor ∉ open_list or tentative_g ¡ g(neighbor) then
            neighbor.parent ← current;
            neighbor.g ← tentative_g;
            neighbor.f ← neighbor.g + h(neighbor, goal);
            if neighbor ∉ open_list then
                add neighbor to open_list;
            end
        end
    end
end
return ”no path found”;
```



Dynamic Window Approach (Local Navigation)

DWA enables real-time obstacle avoidance while following the global path by:

1. Computing dynamic window of possible velocities (v, ω)
2. Generating trajectories for velocity samples
3. Evaluating trajectories based on: Distance to obstacles, Progress toward goal and Alignment with global path

```
Input: robot, goal, config, grid
Output: optimal velocity commands (v, )
Calculate dynamic window [v_min, v_max, ω_min, ω_max];
best_score ← ∞;
best_v, best_ ← 0, 0;
foreach v in range(v_min, v_max) do
    foreach  in range(ω_min, ω_max) do
        traj ← Generate_Trajectory(robot, v, , config.predict_time);
        if Collision(traj, grid) then
            ;
        end
        score ← Evaluate_Trajectory(traj, goal, grid);
        if score ¡ best_score then
            best_score ← score;
            best_v, best_ ← v, ;
        end
    end
end
return best_v, best_;
```

RESULTS

The figure shows:

- **Successful navigation** from start (1.9,2) to goal (25,14.5)
- **Effective obstacle avoidance** around multiple barriers
- **Close adherence** between planned path (green dashed line) and actual trajectory (blue line)
- **Adaptive behavior** in complex regions, especially in the narrow passage near coordinates (20,5)

CONCLUSIONS

Advantages :

1. **Efficient Navigation:** The integration of A* algorithm with DWA provides a computationally efficient solution for autonomous navigation that successfully finds optimal paths while avoiding obstacles.
2. **Real-time Adaptability:** The DWA controller demonstrates excellent capability to adapt to local conditions in real-time while maintaining adherence to the global path objective.
3. **Implementation Simplicity:** The approach uses straightforward algorithms that are easy to implement and tune, making it accessible for various robotic applications without requiring specialized hardware.

Disadvantages:

1. **Parameter Sensitivity:** DWA performance depends heavily on proper tuning of parameters like prediction time and sampling resolution, which may require adjustment for different environments
2. **Static Environment Assumption:** The current implementation works well for static obstacles but would require additional modifications to handle dynamic or moving obstacles..
3. **Computational Scaling:** As environment complexity increases, both A* and DWA face increased computational demands, potentially limiting real-time performance in very large or complex spaces.

REFERENCE

Y. Li et al., "A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach," in IEEE Access, vol. 10, pp. 57736-57747, 2022, doi: 10.1109/ACCESS.2022.3179397.
keywords: {Heuristic algorithms;Mobile robots;Path planning;Interference;Search problems;Path planning;hybrid algorithms;improved A* algorithm;improved DWA}

ACKNOWLEDGEMENTS

Guided by- Dr. Shailesh Khapre