Motion Planning

IN5060 - Simulation

Introduction

- **Motion planning** plans the state sequence of the robot without conflict between the start and goal.
- It includes Path planning and Trajectory planning.
  - **Path planning:** based on path constraints (such as obstacles)
  - **Trajectory planning:** based on kinematics, dynamics constraints and path sequence.
- More information in the following repository:
  - https://github.com/ai-winter/matlab_motion_planning/blob/master/README.md

Search-based Planning

- **A*:** A Formal Basis for the heuristic Determination of Minimum Cost Paths
- **JPS:** Online Graph Pruning for Pathfinding On Grid Maps
- **Lifelong Planning A*:** Lifelong Planning A*
- **D*:** Optimal and Efficient Path Planning for Partially-Known Environments
- **D* Lite:** D* Lite
- **Theta*:** Theta*: Any-Angle Path Planning on Grids
- **Lazy Theta*:** Lazy Theta*: Any-Angle Path Planning and Path Length Analysis in 3D

Sample-based Planning

- **RRT:** Rapidly-Exploring Random Trees: A New Tool for Path Planning
- **RRT-Connect:** RRT-Connect: An Efficient Approach to Single-Query Path Planning
- **RRT* Planning A*:** Sampling-based algorithms for optimal motion planning
- **Informed RRT*:** Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal heuristic

More algorithms

Evolutionary-based planning

- **ACO:** Ant Colony Optimization: A new Meta-Heuristic

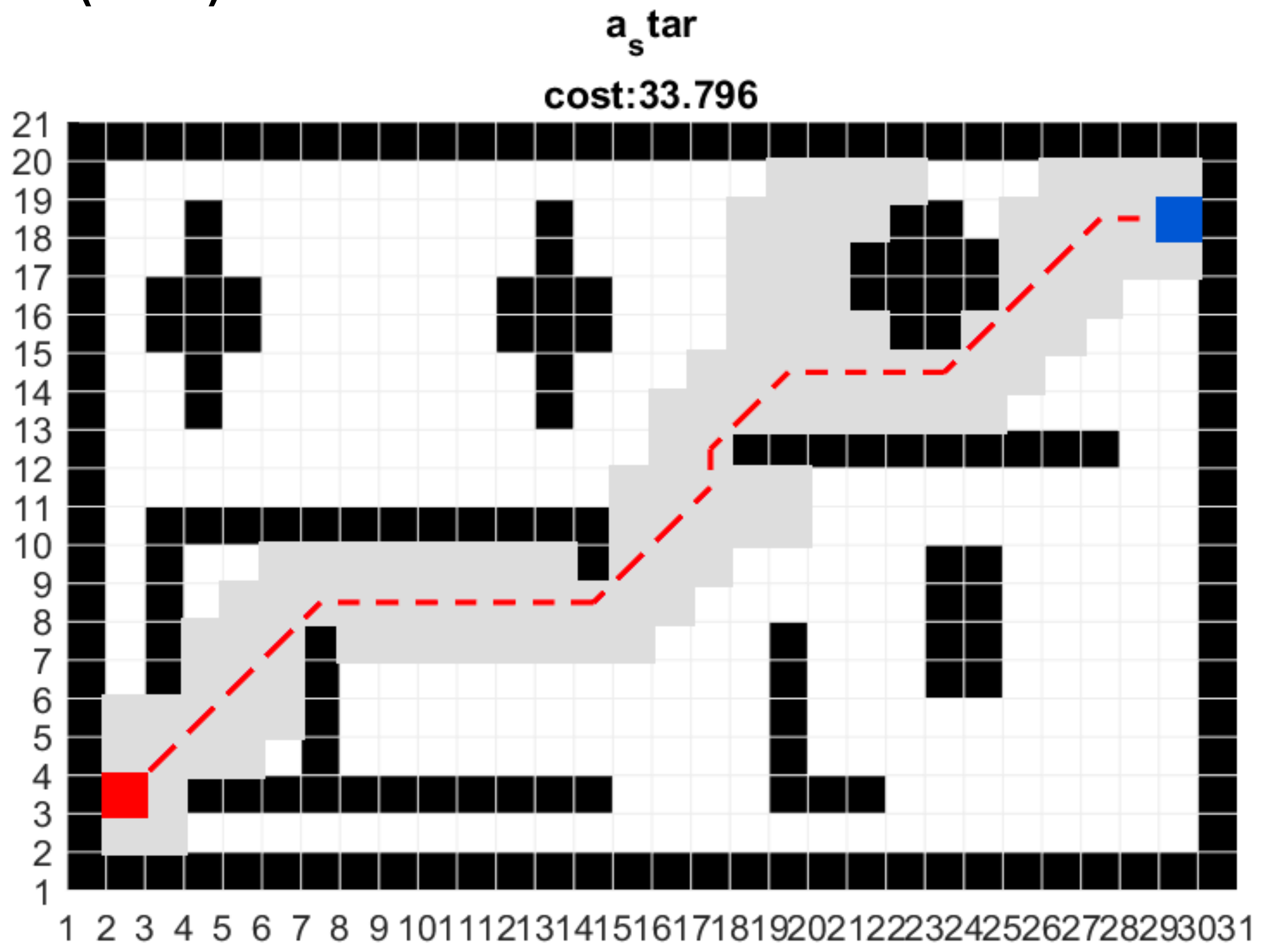
Local Planning

- **DWA:** The Dynamic Window Approach to Collision Avoidance

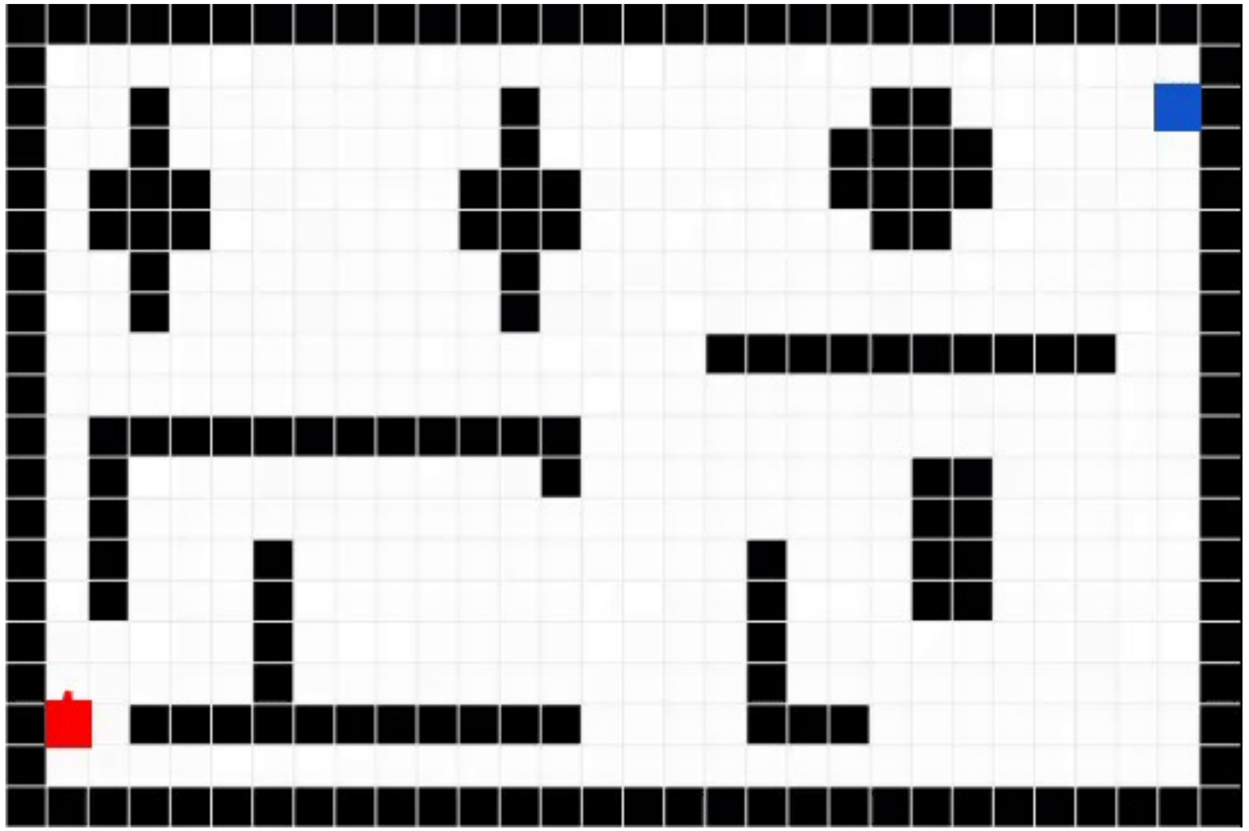
Quick Start Up

- To start simulation, open **./simulation_global.mlx** or **./simulation_local.mlx** and select the algorithm
  - For a complete view on the file structure, check the repository
- Configure the **Start** and **Finish** point on the map.
- Select an algorithm among the pool of available ones
  - See earlier slides

Example (A*)


$a_s$tar
cost:33.796
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Example (DWA) - Interactive



Evaluation

- Available variables
  - ***path:*** path coordinates from the starting to the finish point
  - ***cost:*** associated cost value – can be modified (source code)
- Cost functions
  - Distance
  - Time
  - …

Your task

Just like C or Java, Matlab is not inherently a simulator of any kind.

The Motion Planning examples comprise algorithms of various kind, and only some of these are iterative in nature. Most of them are algorithms that are design to operate on graphs. Since squared fields can be represented as graphs, they can solve the given problem, but it may not be optimally suited.

Your task is to

- define a scenario inspired by real life
- make interesting observation that arise when you compare several of the give algorithms under different conditions.
- use basic statistics to compare different situations and summarize the performance of several algorithms
- Must include Dijkstra and A*

Compare several maps, compare different start and end position, or compare a map that changes while the path is being walked (requiring recomputation).

The main metric to compare is «cost>. You could also look at

  - path length in squares traveled
  - required recomputations (when paths change while moving)