Mobile Robot Control – Navigation (1/2)

MAY 3, 2023

Mobile Robot Control 2023

Mechanical Engineering, Control Systems Technology, Robotics Lab

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Motion Behaviour of Last Years

- What are the motion requirements?
- When do they change?

EMC 2019 – Group 5

Motion Behaviour of Last Years

- What went well?
- What can be improved?

EMC 2019 – Group 9

Outline

- Motion planning problem
  - Motion constraints
- Motion planning algorithms
  - Specifications & properties
  - Taxonomy
  - World representation
  - Graph-based solutions
  - Local motion planning algorithms
- Control objectives

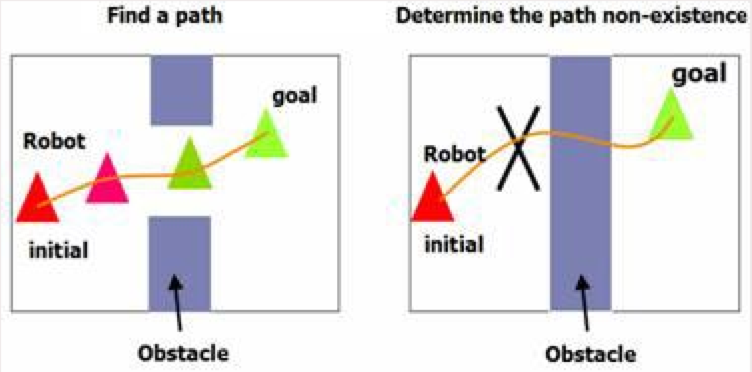
Outline

- Motion planning problem
  - Motion constraints
- Motion planning algorithms
  - Specifications & properties
  - Taxonomy
  - World representation
  - Graph-based solutions
  - Local motion planning algorithms
- Control objectives

Motion Planning Problem

Given an initial pose, determine the control outputs such that, via a sequence of valid configurations, its desired final pose is reached

- When is the final pose achieved?
  - Constant or variable requirements
- Motion constraints
- Environment static or dynamic?
- Measurement/localization uncertainty



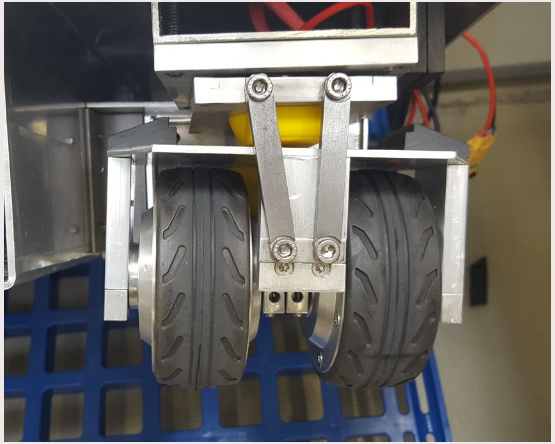
Source: http://gamma.cs.unc.edu/NOPATH/

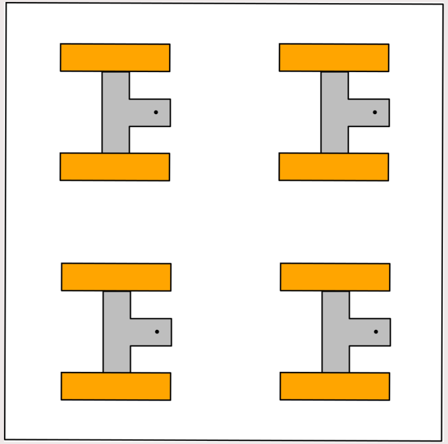
Motion Constraints

- Which motion constraints does the robot have?
  - Holonomic/non-holonomic?
  - Actuator & physical limits? Maximum acceleration, velocity?
  - Compliant with environment-constraints?
  - Multiple robots?
  - How strict is the “trajectory”?

TU/e

Motion Planning Problem: Example





Motion Planning Problem: Example

Outline

- Motion planning problem
  - Motion constraints
- Motion planning algorithms
  - Specifications & properties
  - Taxonomy
  - World representation
  - Graph-based solutions
  - Local motion planning algorithms
- Control objectives

Motion Planning Algorithms: Specifications & Properties

- Completeness: finding a path if one exists
- Optimality: finding the optimal path
- Computational complexity
- Robustness against a dynamic environment
- Robustness against uncertainty
- Kinematic and dynamic constraints

Motion Planning: Taxonomy

Global vs. Local

Bio Inspired

- Genetic Algorithms
- Particle Swarm Optimization

Graph Based

- Topological
  - Semantics Based
  - Probabilistic Roadmap
    - Voronoi Graph
    - Visibility Graph
- Cell/Grid Based

Learning

- Reinforcement Learning
- Deep Learning

Overview far from complete!

Hierarchical Planning: Global vs. Local

Reduction of complexity: divide the planning problem into global and local planner:

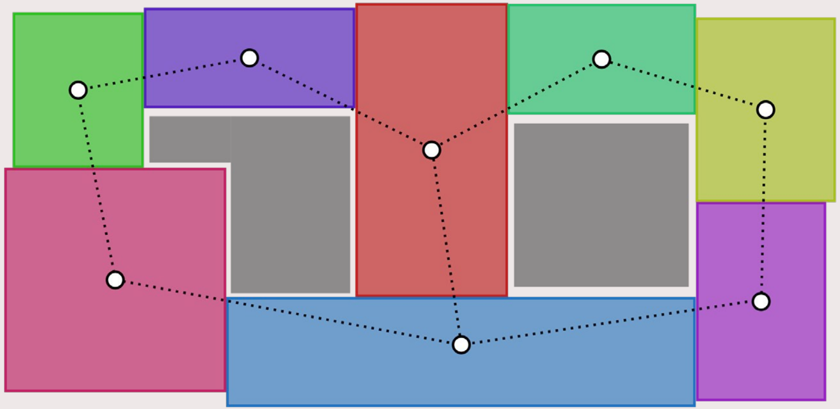
- Global planner: computes a path from start to goal
- Local planner: satisfy kinodynamic constraints

"What is the route from Eindhoven to Amsterdam" vs. "I need to pass the car in front of me"

Explicitly describe what you mean by global and local, it might create confusion

Topological Map

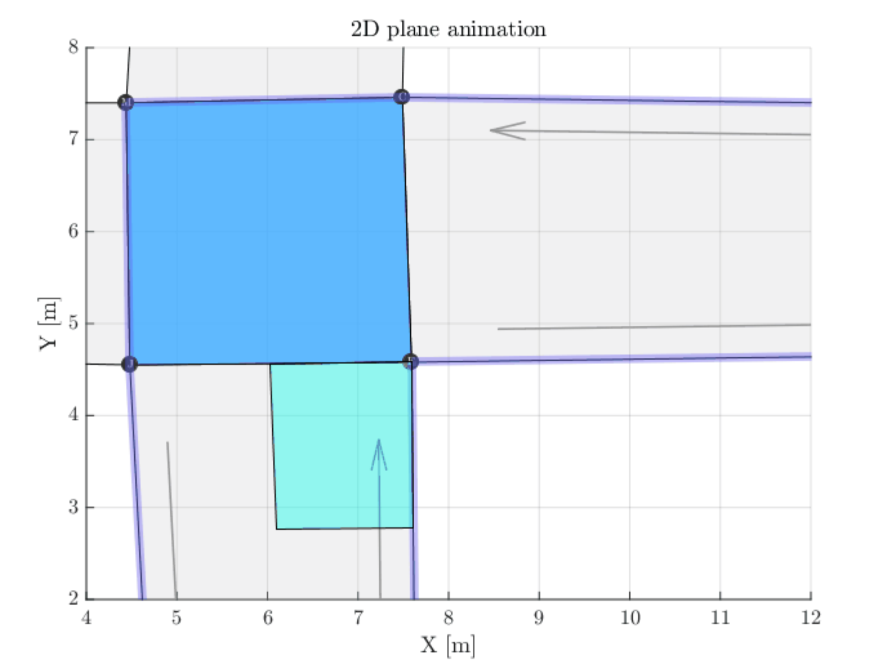
- Where to place the nodes?
- Which semantics might be relevant?
- Which semantics are missing?



Blöchliger et al. (2017). Topomap: Topological Mapping and Navigation Based on Visual SLAM Maps. CoRR, http://arxiv.org/abs/1709.05533

World Representation

- Robot

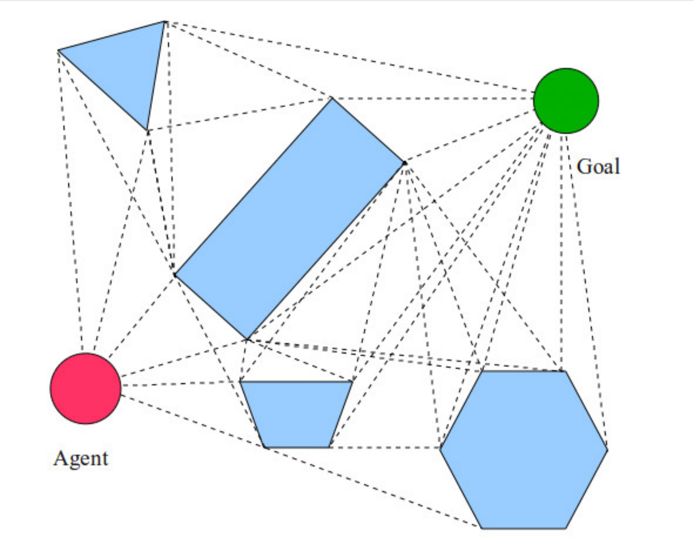


M.S. de Wildt, C.A. Lopez Martinez, M.J.G. van de Molengraft and H.P.J. Bruyninckx. (2018). Tube Driving Mobile Robot Navigation Using Semantic Features. Master's Thesis

World Representation

- Robot

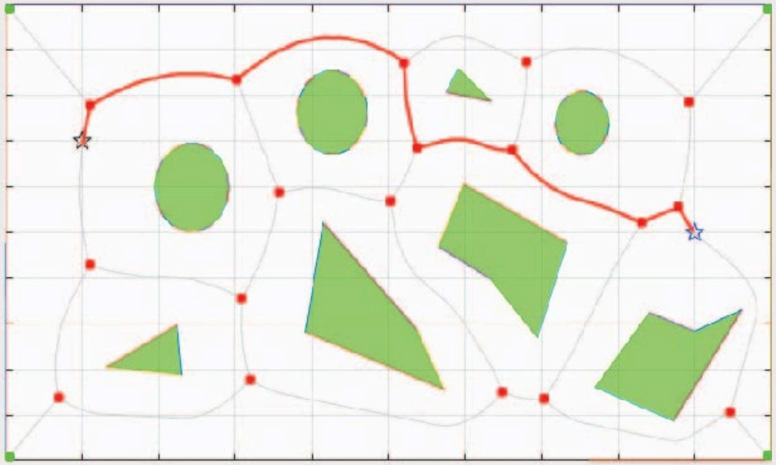
- Exact: Roadmaps
  - Visibility Graph



Niu, Hanlin & Lu, Yu & Savvaris, Al & Tsourdos, Antonios. (2018). An energy-efficient path planning algorithm for unmanned surface vehicles. Ocean Engineering. 161. 308-321. 10.1016/j.oceaneng.2018.01.025.

World Representation

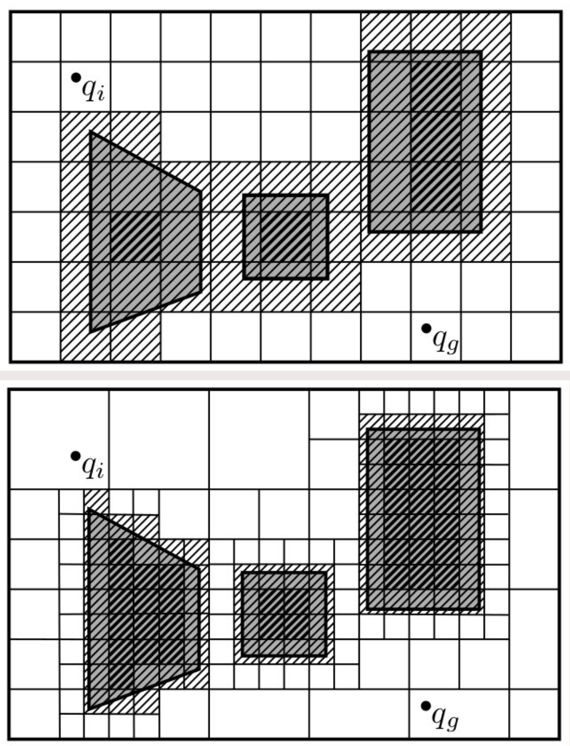
- Robot
- Exact: Roadmaps
  - Visibility Graph
  - Voronoi Diagram



Magid, Evgeni et al. "Voronoi-based trajectory optimization for UGV path planning." 2017 International Conference on Mechanical, System and Control Engineering (ICMSC) (2017): 383-387.

World Representation

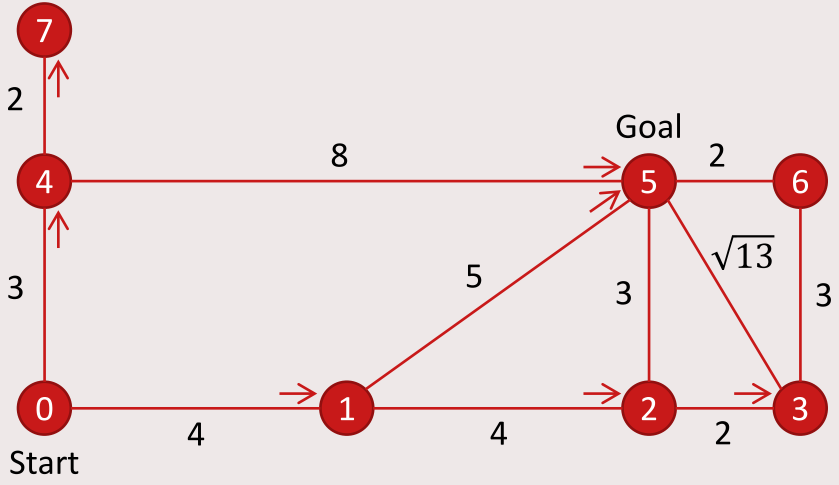
- Robot
- Exact: Roadmaps
  - Visibility Graph
  - Voronoi Diagram
- Approximate: Cell decompositions
  - Occupied vs Free
  - (Adaptive) Cell size
  - Semantics



Coenen, S.A.M. (2012). Motion Planning for Mobile Robots - A Guide. Master’s thesis

Graph Based Solutions

- Dijkstra's algorithm[1]
  - Start at initial node as current node
  - While goal not closed:
    - For all connected non-closed nodes:
      - Calculate cost via current node
      - If unvisited: open and store
      - Else: update if lower
    - Close current node
    - Open node with minimum costs becomes current node
  - Trace back the path from goal to start

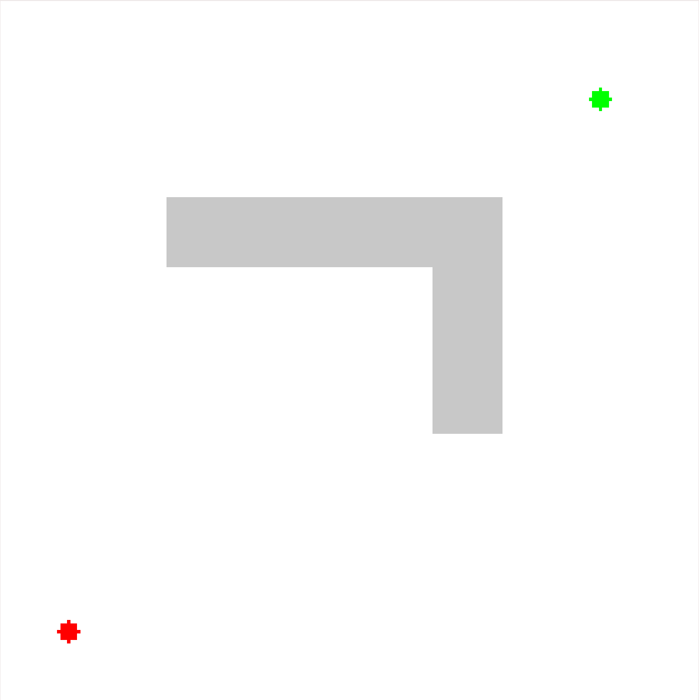


Step	Closed nodes	Open nodes	Unvisited nodes
0		0(0)	1, 2, 3, 4, 5, 6, 7
1	0	1(4), 4(3)	2, 3, 5, 6, 7
2	0, 4	1(4), 5(11), 7(5)	2, 3, 6
3	0, 1, 4	2(8), 5(9), 7(5)	3, 6
4	0, 1, 4, 7	2(8), 5(9)	3, 6
5	0, 2, 1, 4, 7	5(9), 3(10)	6

[1] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

Graph Based Solutions

- Dijkstra's algorithm[1]
  - Start at initial node as current node
  - While goal not closed:
    - For all connected non-closed nodes:
      - Calculate cost via current node
      - If unvisited: open and store
      - Else: update if lower
    - Close current node
    - Open node with minimum costs becomes current node
  - Trace back the path from goal to start

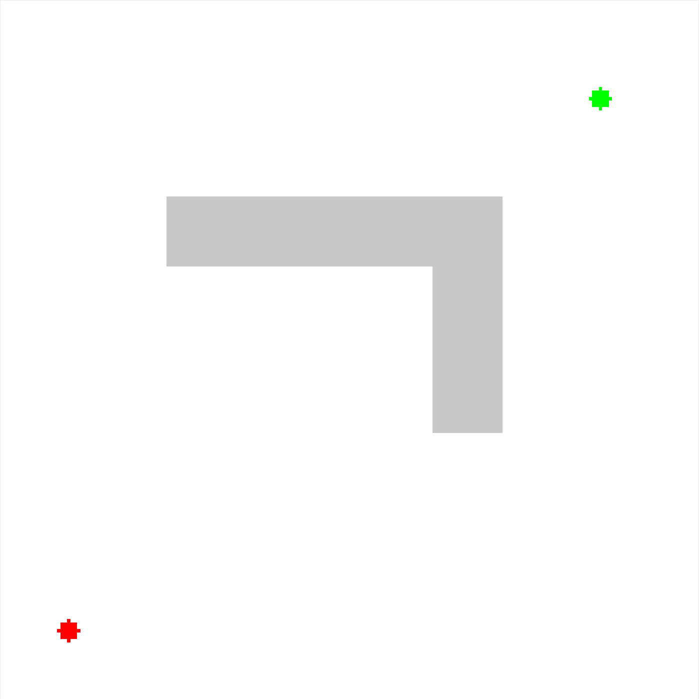


Dijkstra's algorithm. Source: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[1] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

Graph Based Solutions

- Dijkstra's algorithm[1]
- A*
  - 'Dijkstra + heuristic'
  - Heuristic: underestimation



A* algorithm. Source: https://en.wikipedia.org/wiki/A*_search_algorithm

[1] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

Graph Based Solutions

- Dijkstra's algorithm[1]
- A*
- Rapidly-exploring Random Tree (RRT) and RRT*
  - Open space: where to place the nodes?
  - Creates the graph and finds the path

RRT algorithm. Source: https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree

[1] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

Graph Based Solutions

- Dijkstra's algorithm[1]
- A*
- Rapidly-exploring Random Tree (RRT) and RRT*
- And many more!

[1] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

Assignment 1 – A* algorithm

- Write your own implementation of the A* algorithm to find the shortest path from start to finish in a maze
- Provided: list of nodes and connections, index of start and finish nodes
- Required: sequence of node indices that form the shortest path from start to finish