8. Introduction to Motion Planning

Given a robot able to build a map of its workspace as well as to localize itself in such a map, the next step for being operational is to provide the algorithms to reach a given destination, i.e. to plan how to reach a goal point and safely complete such navigation. In the real world this is complicated by various factors, e.g. the geometry of the robot, the variability of the obstacles, the non-holonomic constraints of the movement, etc.

Two major concepts are related with this *planning and navigation* idea:

- **Path planning**
  - Pursues how to get from point A to point B.
  - Factor like time and robot kinematics or cinematics are not considered.
- **Motion planning**
  - In charge of computing speed and turning commands to be sent to the robot (non-holonomic constraints apply) to follow a desired trajectory.
  - Explicit consideration of time (considers a trajectory instead of a path). This is interesting, for example, where two robots are operating in the same space.

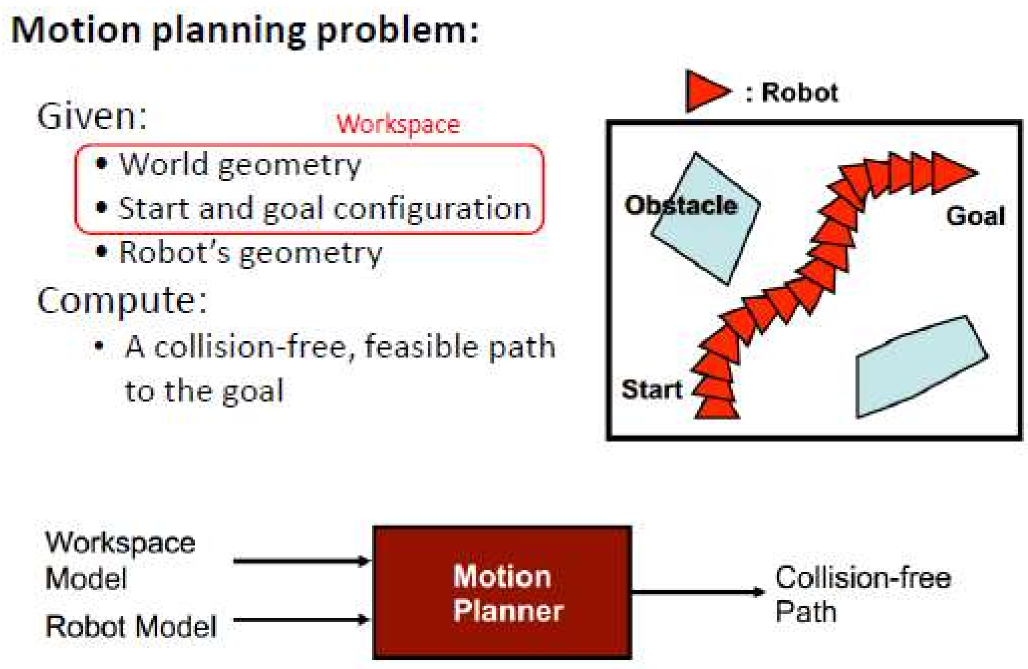

Fig. 1: Main actors in the Motion planning concept.

They are different concepts, but sometimes both terms are used as synonymous.

Usually, motion planning is further decomposed into two problems:

- **Global navigation**: Which uses the available map to find a sequence of movements to reach the goal (it finds a sequence of waypoints between the start and the goal). It is also called **roadmap navigation** since it could be viewed as a kind of topological map that represents a set of paths (roads) between two points in the environment that the robot can travel on without collision.

  - *Input*: the whole (known) map.

    - *Techniques*: search method for global finding of a (optimal) path to the goal (e.g. A*).
        - Bug algorithm.
        - Visibility graph.
        - Generalized Voronoi Diagram.
        - Cell decomposition.
        - Probabilistic Road Map.
- **Local reactive navigation**: Which uses sensor information to avoid the obstacles in the robot's path, even those which don't appear in the map, that is, it navigates between consecutive waypints while avoiding obstacles.

    - Works at a high-frequency (i.e. real time).
    - *Input*: sensor current observation.
    - *Techniques*:
        - Virtual Force Field (VFF).
        - Vector Field Histogram (VFH).
        - Dynamic window.
        - PT-space (developed at **UMA**).

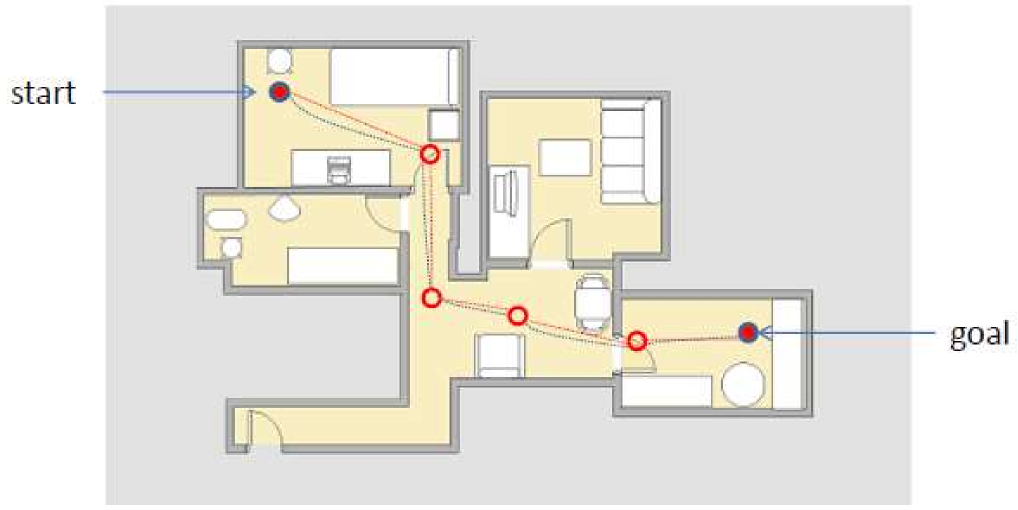

Fig. 2: Example of navigation using Roadmaps. The global navigation set a number of waypoints (red dots) between the start and the goal that are travesable by the robot (dotted red lines), while the local reactive navigation (dotted blue lines) goes from one waypoint to the next one avoiding obstacles.

OPTIONAL

Surf the internet looking for more general information about Motion Planning. You can include additional definitions, examples, images, videos,... anything you find interesting!

Motion Planning with Generalized Voronoi Diagrams

The two-dimensional region in which the robot moves will contain buildings and other types of barriers, each of which can be represented by a convex or concave polygonal obstacle. To find the generalized Voronoi diagram for this collection of polygons, one can either compute the diagram exactly or use an approximation based on the simpler problem of computing the Voronoi diagram for a set of discrete points.

Approximation Strategy: Boundaries of polygonal obstacles are approximated by densely subdividing each side into smaller segments. This results in a substantial number of points outlining the obstacle boundaries. The Voronoi diagram is then computed for this array of approximating points. Voronoi edges with one or both endpoints residing inside obstacles are eliminated. The retained edges form a close approximation of the GVD for the original obstacles.

Robot Path Planning: In the GVD, the robot's starting and stopping points are identified. Voronoi vertices closest to these points are computed. Straight lines are drawn to connect the robot's points with the nearest Voronoi vertices. Special attention is given to cases where these lines intersect with obstacles. A standard search algorithm, like Dijkstra's Algorithm, is employed. It identifies the "best" path, a subset of the Voronoi diagram, connecting the starting and stopping vertices. The determined path is expanded to form a route between the robot's original starting and stopping points.

This method crafts a route that strategically maintains an equidistant trajectory between the robot and its closest obstacles.

Paper: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=88035

Youtube Video:

- https://www.youtube.com/watch?v=YBq_fzAOpVI&ab_channel=CheongKinNg
- https://www.youtube.com/watch?v=Y5_aHsqX22s&ab_channel=AparajitaOjha

END OF OPTIONAL PART