Motion Planning

Jonas Tietz

May 3, 2018

Introduction

Problem input:

- World description
- Robot description
- Starting configuration
- Target configuration

Output: Collision free trajectory of the manipulator

Restrictions

- Not dynamic (no moving objects)
- Single robot manipulator
- Only rotational joints

Configuration Space

- Space of all possible configuration of the robot
- for each degree of freedom k one dimension $\rightarrow \mathbb{R}^k$
- every configuration of the manipulator is one point in the configuration space

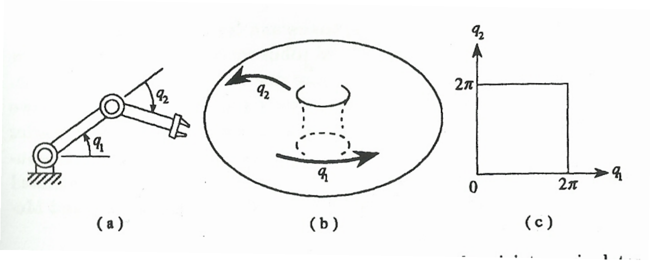

Figure: Robot Motion Planning - Jean-Claude Latombe page 387

Cell Decomposition

The goal is to find a representation of all the collision free space in the configuration space. Then a simple path finding algorithm can be used to find trajectory of the manipulator.

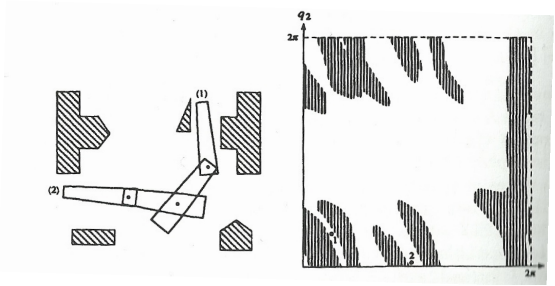

Figure: Robot Motion Planning - Jean-Claude Latombe page 394

Cell Decomposition

For this algorithm a tree structure is used to represent this free space.

1. Start with the first joint
2. Check at a given resolution all angles for collision
3. Add all collision-free cells to the tree
4. Repeat for all new nodes with the next joint

In an static environment this data structure can be precomputed.

Path finding in the tree

From this tree a connectivity graph can be constructed, which in turn can be used in a simple path finding algorithm like A*.

Questions?

Potential Field Method

Idea:

- Configuration is a "particle" in the configuration space
- Obstacles emit a repulsive force
- The target emits a attractive force

Potential Field Method

- The potential field can be easily imagined when dealing with only 2 dimensions
- The repulsive forces are hills in the field
- The attractive forces are valleys
- The algorithm tries to find a path which follows the downward gradient

Potential Field Method

- may get stuck in local minima
- does not necessarily find a trajectory if there is one
- faster than more exact methods

Best First Planning

- to deal with local minima, we need to enhance the algorithm
- one option is do discretize the C-Space into a grid
- use an Path-Finding algorithm, which walks the graph in direction of the gradient until it reaches a minimum
- if it is not the target configuration, backtrack
- if the whole graph is explored an the target is not reached, return false

Randomization Techniques

- Instead of exhaustively searching the graph, randomization strategies to escape local minima can lead to better performance
- Path searching along the gradient
- if in local minimum, move along a random direction and start searching again.
- repeat this k times or until another minimum is found.
- if after k times no other minimum is found, backtrack
- the movement along a random direction needs to be checked for collision

Questions?

Sources

- Robot Motion Planning - Jean-Claude Latombe
- A Simple Motion Planning Algorithm For General Robot Manipulators - Tomás Lozano-Pérez
- Strategies For Solving Collision-free Trajectory Problems For Mobile And Manipulator Robots - Laurent Gouzènes
- Path Planning Using Potential Fields For Highly Redundant Manipulators - Erdinc Sahin Conkur