ME 597 Lab 3 Report

Iain Peet   Andrei Danaila   Kevin Kyeong   Abdel Hamid   Douche Salam

December 1, 2011

1   Introduction

In this lab exercise the Potential Fields and Wavefront methods are implemented for planning the robot's path around a know obstacle grid map. The methods produce a path that the robot can follow to travel from a given start position towards a predefined goal while avoiding known obstacles.

2   Potential Fields Path Planning

2.1   Standard Method

The potential field method defines

2.2   Steering

The robot follows the planned path by steering towards the angle of steepest decent in the potential field map. This allows the robot to calculate the steering angle from any position on the map where the gradient is defined, which is essentially in every free cell in the grid.

Alternatively, we can define waypoints along the path of steepest decent and use the Stanley non-linear steering controller to follow it, but this would be more error prone due to the robot's poor steering which makes it more likely to deviate significantly from the intended path.

2.3   Extended Potential Fields

Placeholder

3   Wavefront Path Planning

3.1   Description of Algorithm

The wavefront planning algorithm is fairly simple in principle. Cost is propagated outward from the goal, with each cell being assigned a cost one greater than the minimum cost of cells reachable from that cell. It is equivalent to interpret the occupancy grid as a graph where cells are nodes and cells which are reachable from each other are connected by edges. All edges are assigned a constant cost, and the costs of reaching each cell from the goal are found using Dijkstra's Algorithm.

For this particular application, all cells which are horizonatally, vertically, or diagonally adjacent to a cell are considered to be reachable from that cell.

The wavefront cost computation algorithm is as follows:

- Initialize the cost map to 0.

- Place the goal cell in the open set, with cost 1.

- While there are cells in the open set,

  - Remove a cell from the open set. This is the current cell.

  - For each cell reachable from the current cell,

    * If the cell is occupied, move it directly to the closed set, with cost 0.

    * If the cell is in the unreached set, move it to the open set and assign it a cost one greater than the current open cell.

  - Place the current cell into the closed set.

In the completed cost map, cells which have a cost of zero are obstacles. For all other cells, the shortest path to the goal requires $n - 1$ moves, where n is the cost of the cell. In order to traverse a shortest path to the goal, the actor must always move to a cell which has lower cost than the current cell.

For the actual robot, a wavefront descent steering controller is defined as follows. First, the optimal descent direction is computed as follows:

$$\theta_{ref} = tan^{-1}\left(\frac{-\frac{\partial C}{\partial y}}{-\frac{\partial C}{\partial x}}\right) \tag{1}$$

The steering controller is a simple unity feedback controller:

$$\delta = \theta_{ref} - \theta \tag{2}$$

3.2   Simulation Results