ME 597 Lab 3 Report
Group 2 (Team Farley)

Iain Peet Andrei Danaila Kevin Kyeong Abdel Hamid Ahmed Salam

December 2, 2011

Introduction

In this lab exercise the Potential Fields and Wavefront planning methods are to be implemented for planning the trajectory of the robot around a pre-defined obstacle grid map. The methods produce a path that the robot can follow to travel from a given start position towards a pre-defined goal while avoiding known obstacles.

1 Potential Fields Path Planning

1.1 Theory

The potential fields method is based on the concept of potential function of which value is viewed as energy and gradient is force. The gradient is a vector $\nabla U(q) = DU(q)^T = [\frac{\partial U}{\partial q_1}, ..., \frac{\partial U}{\partial q_m}]^T$ which points in the direction that locally maximally increases U[1]. This gradient then defines a vector field on an occupancy grid, and directs a robot as if it were a particle moving in a gradient vector field[1]. Intuitively speaking, this approach can be viewed as if the gradients are forces acting on a positively charged particle which is being attracted to the negatively charged target goal[1]. Additionally, the obstacles are considered as positively charged and defined as repulsive forces thus directing the robot away from the obstacles. Then, combining the potential functions of attractive and repulsive forces and negating the gradient of the potential function, a robot can simply follow the path of the gradient descent to reach the target goal where the gradient vanishes.

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{1}$$

$$\nabla U(q) = V(q) = V_{att}(q) + V_{rep}(q) \tag{2}$$

1.1.1 The Attractive Potential $K_{attractive}$

There are different ways to define the attractive potential. The method used in this lab is one that grows quadratically with the distance to q_{goal}. Hence, the attractive potential function is defined as

$$U_{att}(q) = \frac{1}{2} K_{att} d^2(q, q_{goal}) \tag{3}$$

and the gradient is

$$\nabla U_{att}(q) = \nabla(\frac{1}{2} K_{att} d^2(q, q_{goal})) = K_{att}(q - q_{goal}) \tag{4}$$

where K_{att} is the attractive potential coefficient, essentially a scale factor determining the effect of the attractive potential in the potential function. Such attractive potential can be illustrated as Figure 1.

[1]S. Thrun. *Principles of Robot Motion.* The MIT Press, 2005. (pg.77)

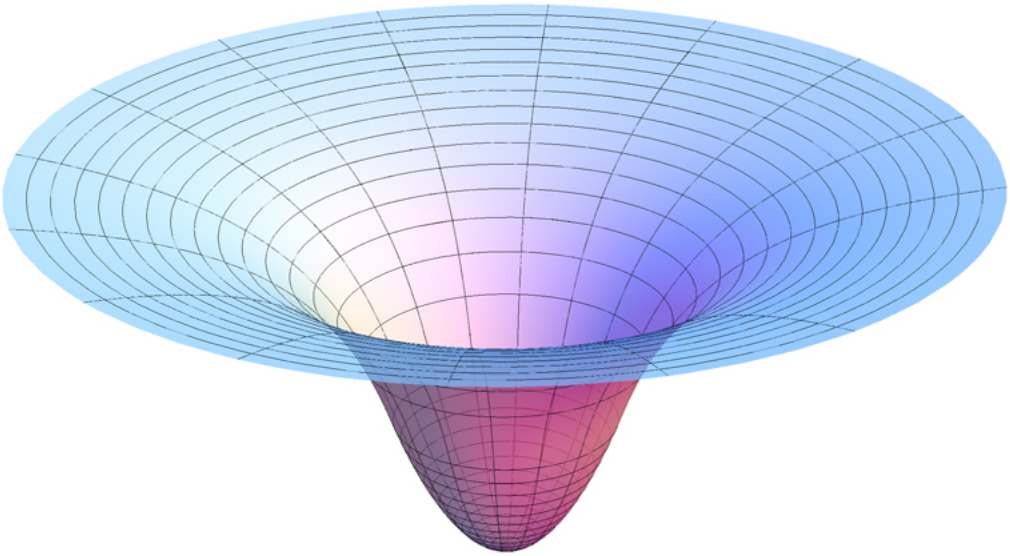

Figure 1: Graph of the Attractive Potential

1.1.2 The Repulsive Potential $K_{repulsive}$

While the attractive potential is attracting the robot particle to the target goal, the repulsive potential is defined throughout the occupancy grid map to push the robot away from the obstacles. The repulsive potential function is inversely quadratically proportional to the distance of obstacles and defined as

$$U_{rep}(q) = \begin{cases} \frac{1}{2}K_{rep}(\frac{1}{D(q)} - \frac{1}{Q^*})^2, & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases} \tag{5}$$

and the gradient is

$$\nabla U_{rep}(q) = \begin{cases} K_{rep}(\frac{1}{Q^*} - \frac{1}{D(q)})\frac{1}{D^2(q)}\nabla D(q), & D(q) \leq Q^* \\ 0, & D(q) > Q^* \end{cases} \tag{6}$$

where K_{rep} is a gain factor on the repulsive gradient and Q^* is the threshold value that allows the robot to either ignore the effect of the repulsive potential from the particular obstacle when it's sufficiently far away from it.

Additionally, it is to be noted that the repulsive potential function should be defined in terms of distances to individual obstacles rather than distance to the closet obstacle. Otherwise, when a path is formed through points where the distances to obstacles are equidistant, some oscillations may be formed.

Therefore, $D(q)$ is redefined as

$$d_i(q) = \min d(q, c) \tag{7}$$

where c includes the closest points to all the obstacles in the grid map.

Then, the gradient of that is

$$\nabla d_i(q) = \frac{q - c}{d(q, c)} \tag{8}$$

Finally, the repulsive potential function is redefined as

$$U_{rep_i}(q) = \begin{cases} \frac{1}{2} K_{rep} (\frac{1}{d_i(q)} - \frac{1}{Q_i^*})^2, & \text{if } d_i(q) \leq Q_i^* \\ 0, & \text{if } d_i(q) > Q_i^* \end{cases} \tag{9}$$

1.2 Results and Observations

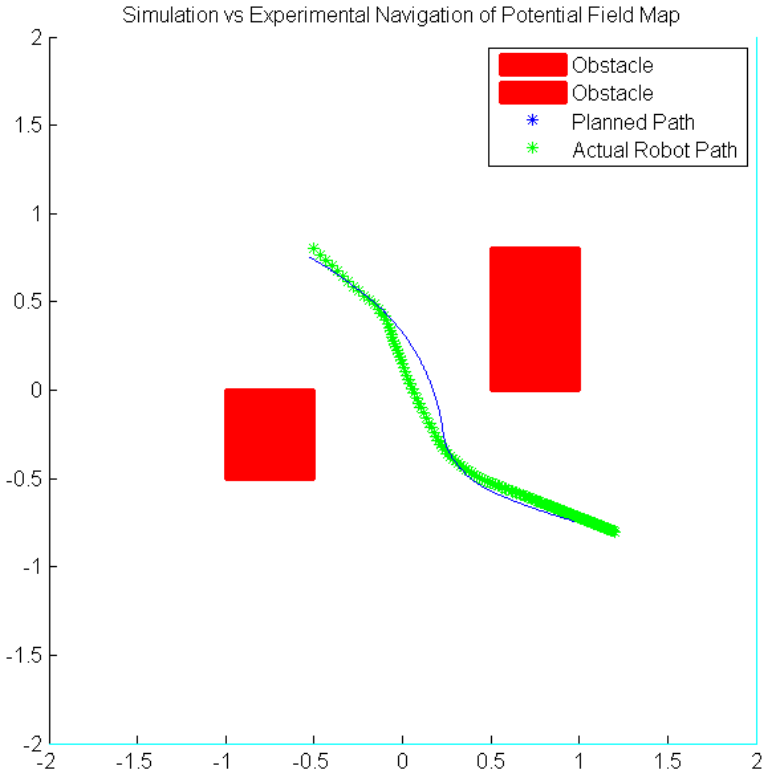

Figure 2: Traversed path through the obstacle map using the potential field algorithm.

The potential field method for path planning successfully plotted a map through the obstacles. Figure 2 showd. the potential field descent path, both as-planned and as-executed. It may be noted that the theoretical path required an infeasibly sharpt turn, resulting in noticeably imperfect tracking. Since the robot was not able to perform such sharp turns, the volume of the each obstacle was increased in simulation so as to provide a margin of safety to compensate for the lack of steering bandwidth.

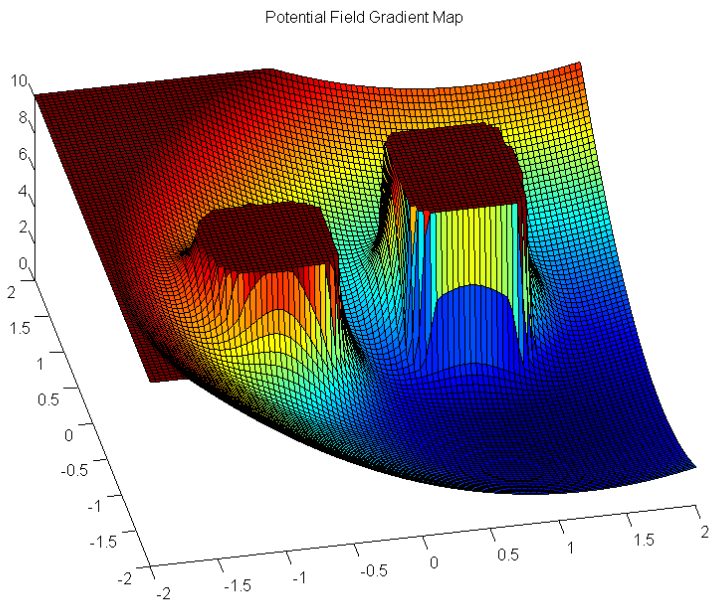

Figure 3: Potential field map surface plot.

1.3 Extended Potential Fields

The extended potential fields method follows from the theory of the potential fields described above, but extendes it by allowing the magnitude of the repulsive field for each obstacle to be weighted based on their bearing from the the robot's heading as it navigates through the map. This allows the robot to prioritize the avoidance of obstacles which are more likely to interfere with its motion; by setting higher weights for obstacles in front of the robot and reducing them for ones that are behind or to the side. The weighting function can be arbitrarily decided for the problem, for example by applying gradual decay or using a sharp cutoff.

In our implementation of the method we defined a field of view in degrees that is centered around the front of the robot. Any obstacles that fall within the field of view are weighted equally, and anything outside of it is ignored (weighted by zero). Furthermore, this method was only implemented in simulation since it requires the local gradient to be calculated online based on the current position as the robot navigates the map; this differs significantly from the two other planning methods which allow the robot to read a pre-calculated gradient map.

The following figure shows the result of planning using the extended potential fields method versus the standard method for a map with two obstacles. In this test the potential field is defined as 90 degrees. The figure shows the advantage of using the extended method as it plans a shorter path once the robot is clear of most of the obstacles.

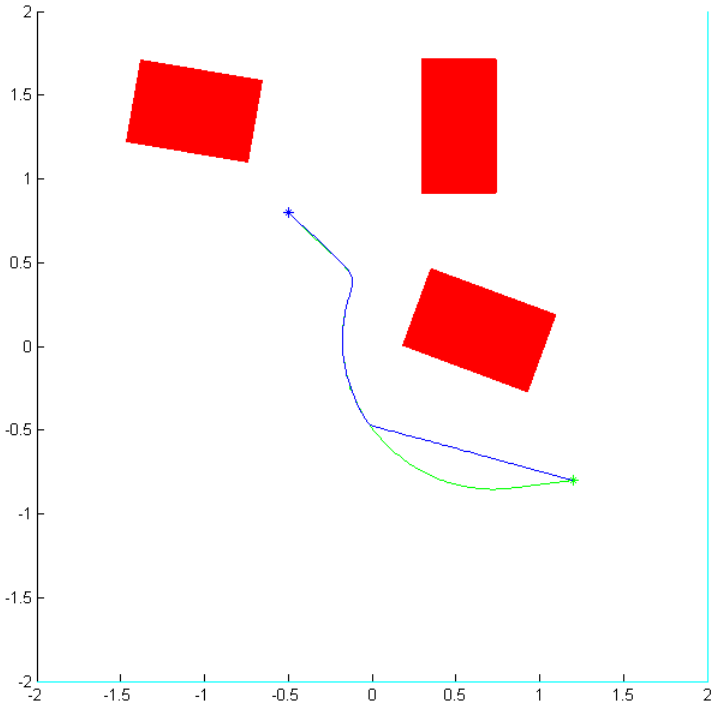

Figure 4: Extended Potential Field planning method

2 Wavefront Path Planning

2.1 Algorithm Description

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. Equivalently, the occupancy grid may be interpreted as a graph where cells are nodes and each cell is connected to its reachable neighbors with 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, an 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:

$$\theta_{ref} = tan^{-1}(\frac{-\frac{\partial C}{\partial y}}{-\frac{\partial C}{\partial x}}) \tag{10}$$

Where C is the wavefront cost map.

The steering controller is a simple unity feedback controller:

$$\delta = \theta_{ref} - \theta \tag{11}$$

It is likely that the shortest descent paths provided by the wavefront algorithm violate the kinematic constraints of the robot. The algorithm is adjusted as follows, to allows for path tracking error.

First, buffer regions are added around obstacles, which are considered to be obstacle region for the purpose of wavefront cost computation. This ensures that the path provides a certain amount of tolerance for tracking error.

Second, obstacle regions are assigned high cost values, such that their gradient always points to the centre of the obstacle. This ensures that, if the robot path overshoots into the buffer region, the cost gradient will be defined such that the robot will steer out of the buffer area and back into the 'good' region.

2.2 Simulation Results

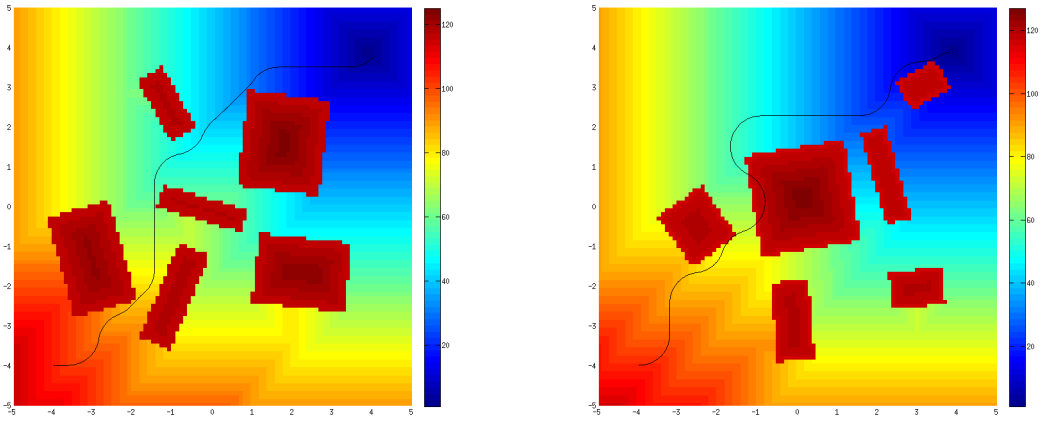

Figure 5: Wavefront costmaps and robot trajectories, for two randomly generated maps.

Figure 5 shows simulated results for two different random maps. Note how, in the second simulation, the robot overshoots the desired trajectories, and recovers as intended.

2.3 Measured Results

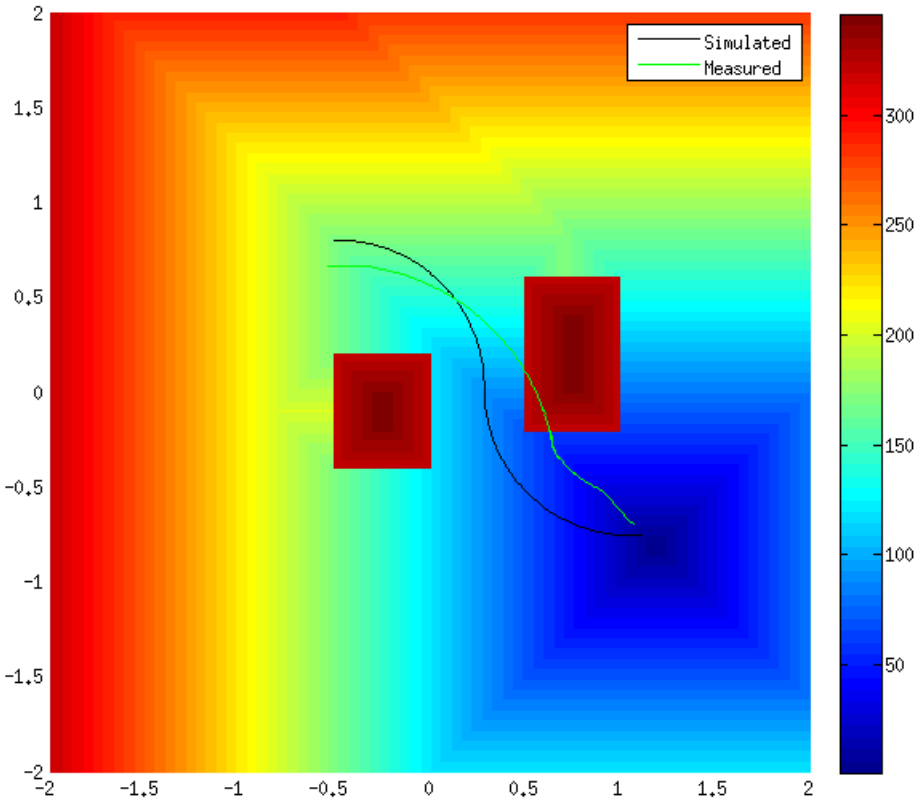

Figure 6: Simulated and measured robot trajectories, for lab obstacle set-up

Figure 6 shows simulated and measured robot trajectories. The true robot does not steer as sharply as the simulation predicts, which suggests that the modelled maximum steering angle is incorrect. Otherwise, the measured behaviour appears consistent with expectation.

2.4 Potential Improvements

A serious limitation of the wavefront algorithm in this application is the inability to account for the robot's kinematic constraints.

One possibility would be to extend the occupancy grid to three dimensions, with heading as the third dimension. The adjacent cells could then be defined to be more plausible state transitions: in the x and y dimensions, the robot must transition to the neighbor nearest to the current heading co-ordinate. Heading may remain the same, or transition to adjacent headings.

It is expected that such a problem definition would result in a smoother path, which would be more easily tracked by the steering controller.

Conclusion

Based on the above discussion both a wavefront and a potential field planning has been implemented successfully. Certain techniques were used in ensuring the success of these algorithms in the real world as opposed to just matlab simulations. Such techniques include include artificially increasing the size and effects of obstacles to overcome turning limitations. Another technique involves assigning high cost gradients, within these artificial buffers, to provide the robot with a way out if they enter these regions. We also demonstrated the superiority of the extended potential fields algorithm (EPF) over the standard potential field. It was observed that the path generated by EPF was shorter, due to the fact that unnecessary turning was eliminated. Finally it is to be noted that proportional steering control was sufficient in obtaining good heading tracking.