Obstacle Avoidance On Turtlebot Using Artificial Potential Field 2.0

Soumick Pyne

***Abstract*— Navigating a bot through an obstacle course to reach a desired goal by using the model of artificial potential field**

The obstacle avoidance model is analogous to a proton moving through a electrostatic potential field to settle in a potential minima (global or otherwise).This intuitive approach can be used for navigation in any state space not just for mobile vehicles.

I. INTRODUCTION

The approach for an obstacle avoidance problem should realize the fact that a bot rarely has a global picture of the configuration space because its state estimation occurs using sensors that have a finite range. Artificial Potential Field was chosen because it works fine with local information and proves as a practical candidate solution.

II. PROBLEM STATEMENT

Any robot in a new/unfamiliar environment doesn't have global information(eg.the position of every obstacle) so it's not possible to have the entire configuration space ready right away for optimal path planning. The problem is to avoid obstacles in near proximity while simultaneously moving toward the goal in in the long run.

The project was implemented on turtlesim where the bot could only be given velocity command along one of it's body axis and angular velocity command. The objective was to build 2 cases, one with global information available and hence preplanning a path(given the obstacles stay stationary), the second case was of using a local planner, which should work for dynamic obstacle too.

III. RELATED WORK

There are several other algorithms like probabilistic roadmaps that are useful for a bot that is meant for repeated navigation in the same environment. Randomly Exploring Random trees is a very good algorithm for practical application as it's probilistic complete and is very good for concave obstacle courses and mazes.

IV. INITIAL ATTEMPTS

To get started I had to define the potential field functions for the obstacles and the goal. Ideally the bot's motion should be unaffected by an obstacle unless it comes really close, and this nature should be manifested in the potential field of the obstacle. Hence we choose:

$$p_i = k \times e^{-d_i^2} \tag{1}$$

*

d_i being the distance between centre of obstacle i and bot p_i being the potential at centre of the bot because of the said obstacle The function used above flatlines rapidly after an acute radius.

The potential field of the goal is defined as:

$$P = -K \times (1/D^2) \tag{2}$$

D being the distance between centre of bot and the goal point P being the potential at centre of the bot because of the goal point Write all the equations and results with pictures if applicable.

After a contour has been defined the bot calculates the potential at n equidistant points on a circle of very small radius γ with itself as the centre,after finding the direction of minimum potential the bot moves to that point using PID controllers on linear and angular velocity. The same process is repeated once it reaches the setpoint and so on.

Initial definition of potential at a point was:

$$P_f = \sum P_i + P \tag{3}$$

V. FINAL APPROACH

The initial attempt didn't manifest the dimensions of obstacles or bot and it was far from what the solution was intended to be. The final approach tackles some of those drawbacks.

To incorporate dimension of obstacle and bot in the the model the definition of p_i was changed as follows:

$$p_i = k \times e^{-(d_i^2 - R^2 - r_i^2)} \tag{4}$$

Where R i the radius of the bot and r_i is the radius of i^th obstacle.

Any oblique shaped obstacle can be approximately decomposed into circular parts with their individual potential potential fields defined. The potential p_i spikes for centre to centre distance less than $R + r_i$. Since the bot should make a maneuver only to evade the closest obstacle, not bothered by obstacles at a distance, the P_f function is modified as:

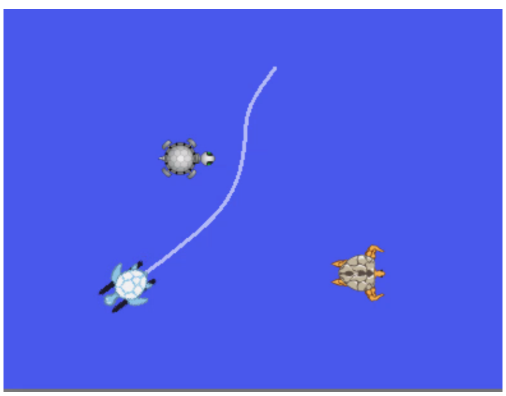
$$P_f = Max(P_i) + P \tag{5}$$

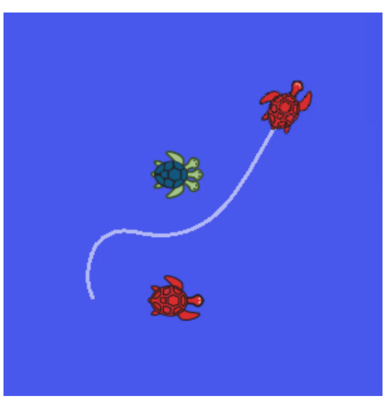
The repeated evaluation of potential 360 deg takes up enough time to make the journey of bot jerky(move, stop, scan and repeat); this was not desirable. Analysing the functions of the model and the final contour of the potential field we understand that it's a differentiable surface, so there will not be any abrupt changes in direction of heading if the obstacles are stationary.

Hence I decided not to check 360 degree around the bot each time but instead sweep an angle of about 90 degree in the present heading direction. This dropped computation time by 75%. I also used the same algorithm for global path planning before starting the bot and used a computationally far less expensive local planner that made the journey smooth and considerably fast.

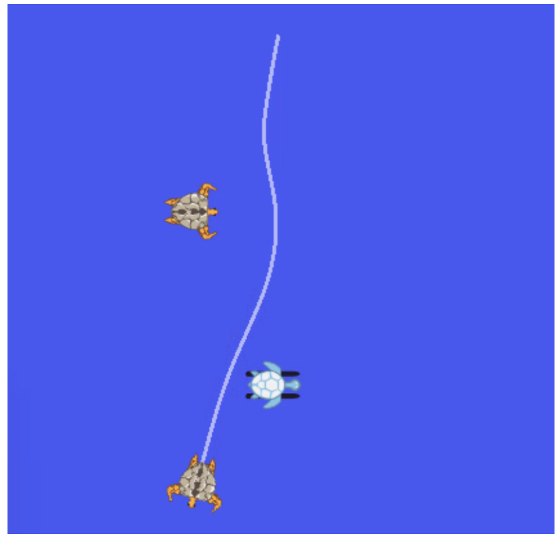
VI. Results and Observation

Some results of implementation in turtlesim were as below:





We see that keeping the step size small enough the path is quite smooth but on the downside the bot moves slowly. When I increased the step size, one case I found important is shown below:



The bot almost rams into the second obstacle (white turtle), because of increasing the step size curve paths get approximated to straight lines which increases the risk of bumping into thin/small extensions of obstacles.
When tested on dynamic obstacles the performance was appreciable only for very slowly moving obstacles and it's only logical because after the bot calculates the potential value at point the value changes because the obstacle changes it's position. The error will reduce if the potential calculation occurs very fast compared to the rate at which nearby obstacles move.

VII. Future Work

Write about the problems in your algorithm / approach and limitations in testing (if any due to hardware or otherwise) and how to tackle them and any future work which can be done to improve the results further.

Conclusion

Write overall about what the problem was, how you solved it, difficulties faced and the net output with it's usefulness to ARK or in general.

References

[1] Khatib, O., "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," International Journal of Robotics Research, Vol. 5, No. 1, pp 90-98, 1986.

[2] Write all the papers and links you have referenced to complete your project.One example for format is given above, Format followed should be ::

[3] Author Names, "Paper Name", Conference / Journal where the paper was published , Year of Publication