# SC-627 Motion Planning & Coordination of Autonomous Vehicles

Assignment-2

Danish Behnal (190010018)
Instructor: Prof. Arpita Sinha
5 March 2022

## Artificial Potential Field Path Planning

In the potential field planning problem we model obstacles as a repulsive objects and a goal as an attractive object. This creates a potential map which is used by our bot to navigate. We made two main helper functions in the helper.py file:

- ***attract_pot():*** This returns the gradient of the attractive potential using the goal according to this formula:

$$\nabla U_{\text{att}}(q) = \begin{cases} \zeta(q - q_{\text{goal}}), & d(q, q_{\text{goal}}) \leq d^*_{\text{goal}}, \\ \dfrac{d^*_{\text{goal}}\zeta(q - q_{\text{goal}})}{d(q, q_{\text{goal}})}, & d(q, q_{\text{goal}}) > d^*_{\text{goal}}, \end{cases}$$

- ***repulsive_pot()***: This function returns the gradient of the repulsive potential when a bot comes close enough to any obstacle. The formula used is:

$$\nabla U_{\text{rep}}(q) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{D(q)}\right)\frac{1}{D^2(q)}\nabla D(q), & D(q) \leq Q^*, \\ 0, & D(q) > Q^*, \end{cases}$$

The implementation for the attractive potential is pretty straightforward, the tricky part to calculate was the repulsive potential. The implementation was like this:

First I calculate the distance between the bot and the polynomial if it is inside the region of influence, we identify the edge closest to the bot and its type (0,1,2). If the edge is of type 0, the direction of the repulsive gradient is orthogonal to the edge. If it is type 1 or 2 edge the direction of the repulsive gradient is radially outward from the vertex of that edge. We do this for all the obstacles in the environment to calculate the net repulsive gradient.

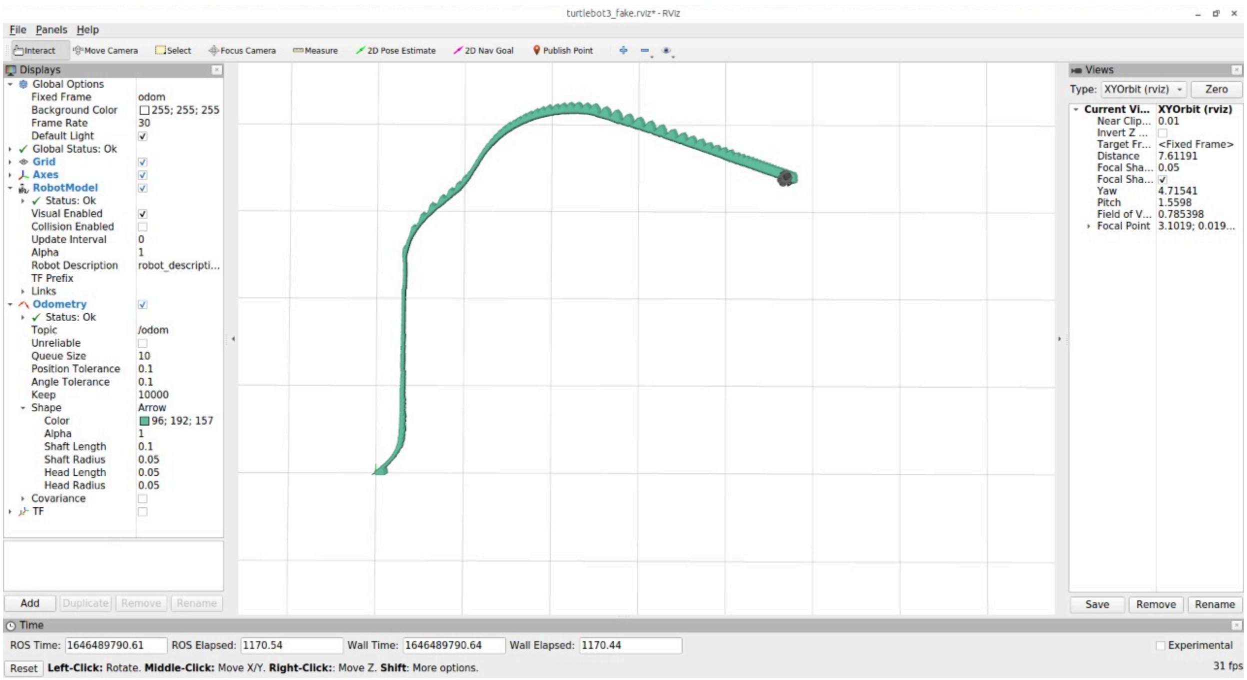
## Potential Field Planner Implementation

Using the *attract_pot()* function we obtain the gradients in x and y. Also using the *repulsive_pot()* function we obtain the gradients in x and y due to opposing force (if any). Then we sum up the gradients and we obtain a direction in which the bot shall move in order to get to the goal point. We follow a basic gradient descent algorithm and we finally reach the goal point.

## Problems encountered

The algorithm is very slow due to the small step length. Also since the bot slowly reaches towards the goal the value of the gradients decrease to levels where it would take a lot more time to reach the goal. To save time the process is terminated early. To show sufficiency,  the bot reaches the goal with an error of about (+10% or -10%).

## Results



Artificial Potential Field Planning