Assignment-2

-Dhrumil Lotiya

21d070026

- **Scan Callback:**

The scan_callback function receives laser scan data, which provides distance measurements from the robot to obstacles in its surroundings.

It stores these measurements in the Range variable.

- **Forces Calculation:**

The Forces function calculates forces based on the stored laser scan data.

The forces acting on the robot are calculated based on the inverse distance from obstacles. Given the coordinates of each obstacle detected by laser scan data, the force components fx and fy are computed as follows:

Fx =1/x1

Fy =1/y1

Where x1 and y1 are the distances to the obstacle along the x-axis and y-axis, respectively.

If the distance x1 or y1 is zero (indicating that the obstacle is directly in front of or beside the robot), the force is set to zero to avoid undefined behavior.

The magnitude of the force at each point (size) is also calculated using vector magnitude formula: size=$\sqrt{\{(Fx^{\{2\}}\} + Fy^2)}$

- **Calculating Net Force:**

Once the individual forces are computed for each obstacle, the net force acting on the robot is determined by summing up all the individual forces along each axis:

- **Motion Calculation:**

The linear and angular velocities of the robot are then adjusted based on the net force. The linear velocity (linear.x) is adjusted along the x-axis, while the angular velocity (angular.z) is adjusted along the z-axis:

linear.x=100−F.nety/100

angular.z= Fnet.x /300

These equations determine how the robot should move to navigate away from obstacles based on the calculated net force.

Reasoning behind use of potential function:

I have implemented a potential field-based approach for obstacle avoidance, where obstacles exert repulsive forces on the robot, and the goal exerts an attractive force. The chosen potential function is inversely proportional to the distance from obstacles.

The potential function is chosen to be inversely proportional to the distance from obstacles. This means that as the robot gets closer to an obstacle, the potential (and consequently the force) increases rapidly, pushing the robot away from the obstacle to avoid a collision.

This choice ensures that the repulsive force grows stronger as the robot approaches an obstacle, facilitating effective obstacle avoidance behavior.

The inverse proportional relationship is a simple and computationally efficient choice for calculating forces based on distance.

It avoids complex mathematical operations and allows for quick computation of forces, which is beneficial for real-time applications like robot navigation.

The choice of an inverse proportional potential function allows for easy tuning of control parameters. For example, the strength of the repulsive force can be adjusted by scaling the inverse distance.

This flexibility enables fine-tuning of the robot's behavior to navigate through different environments effectively.

What are the limitations of your approach if any.

The potential field method can lead to local minima where the robot becomes trapped due to the attractive forces from nearby obstacles. In such cases, the robot may struggle to navigate out of the local minima and may require additional strategies for escape, such as random perturbations or sophisticated path planning algorithms.

Here I have used the potential field approach which assumes static obstacles and a known environment. In dynamic environments where obstacles or the robot's goal location change over time, the precomputed potential fields may become outdated, leading to suboptimal or unsafe behavior.

How can you improve the planner if possible.

I have not considered the velocity of obstacle. So, implementing dynamic potential fields that adapt to changes in the environment or robot's goals can enhance the planner's responsiveness to dynamic obstacles and evolving situations.

Instead of simply measuring the distance to the nearest obstacle, I could use more sophisticated techniques such as clustering or segmentation to identify different obstacles in the environment. This would allow the robot to navigate around multiple obstacles more effectively.