Assignment 2: Dynamic obstacle avoidance using APF

course code:-SC-627

FEBRUARY 2024

1 DESCRIPTION

In this assignment, you have to implement a collision avoidance algorithm for the turtlebot to move from its starting position to its goals while avoiding dynamic and static obstacles. The code repository contains a ROS Noetic package which includes launch files, worlds, and a `planner.py` script. You have to edit the `planner.py` file inside the `src` folder, which should navigate to its goal while avoiding all the obstacles.

You are free to use any type of potential field functions. The launch file launches the `dynamic_obstacle` world with a Turtlebot. The code can be adapted to use the launch the static world as well. Turtlebot has a LiDAR mounted which has to be used to detect the obstacle and plan the trajectory accordingly. The simulations are to be replicated with the Turtlebot3 Burger in the arms lab.

2 TASK

The planner.py file will host the code for collision avoidance algorithm. The planner should be able to take a goal position and navigate to the goal without any collisions. Execute following to launch gazebo with turtlebot

```
roslaunch assignment2 assignment.launch
```

Your task is to write a collision avoidance algorithm based on artificial potential field to avoid static and dyanmic obstacles. The algorithm has to be tested in simulation as well as in the hardware.

3 Steps before using launch plugin:

```
roscd assignment2/worlds
mkdir build
cd build
cmake ../
make
export GAZEBO_PLUGIN_PATH=`pwd`:$GAZEBO_PLUGIN_PATH

gazebo dynamic_obstacle.world
```

Check if it's working; you should see a box moving around near origin

4 Report

your report should consists of

- A brief explaination of your approach including all the equations used.
- Reasoning behind the used of choosen potential function.
- What are the limitations of your approach if any.
- How can you improve the planner if possible.

5 Grading

Grading will be based on following parameters

- The correctness of your implementations.
- Simulation Results.
- Experimental Results.
- Report.

6 submission

- A folder named code that contains all relevant python files.
- submit your report as **NAME-IITBID.PDF** format.
- you can combine both the codes and report in a single zip file named same as mentioned previously.