Assignment - Avoiding Obstacles with VFF

Introduction

In this assignment you will implement a simple behavior that makes the turtle bot move forward avoiding obstacles using the Virtual Force Field algorithm.

As discussed in class, this algorithm is based on using three 2D vectors to calculate the speed command that will deviate the robot away from the obstacle.

1 - **Attractive vector**: This vector always points forward since the robot wants to move in a straight line in the absence of obstacles.

2 - **Repulsive vector**: This vector is calculated from the laser sensor readings. Feel free to experiment and adopt different approaches to calculate this vector. The simplest way is to consider the obstacle closest to the robot that produces a repulsive vector inversely proportional to its distance.

3 - **Result vector**: This vector is the sum of the two previous vectors and will calculate the control speed. Linear speed depends on the resulting vector module, and the angle to turn depends on the resulting vector angle.

Package Structure

```
vff_avoidance/
├── CMakeLists.txt
├── package.xml
├── config/
│   └── AvoidanceNodeConfig.yaml
├── include/
│   └── vff_avoidance/
│       └── AvoidanceNode.hpp
├── launch/
│   └── avoidance_vff.launch.py
└── src/
    ├── avoidance_vff_main.cpp
    └── vff_avoidance/
        └── AvoidanceNode.cpp
```

Control Logic

The **AvoidanceNode** implements the VFF algorithm to generate the control commands based on the laser scan readings.

- This node has to execute the algorithm iteratively at 20Hz using a timer.

- Ensure that the laser scan message is valid (The pointer to the message is not null and the message currently being processed is not older than 2 seconds).
- Get the three vectors
- Use the result vector to get the output speed
- Create the output cmd_vel message and make sure to clamp the speeds to a safe range.

Debugging with Visual Markers

Your **AvoidanceNode** should publish visual markers for debugging purposes that will be displayed in Rviz2.

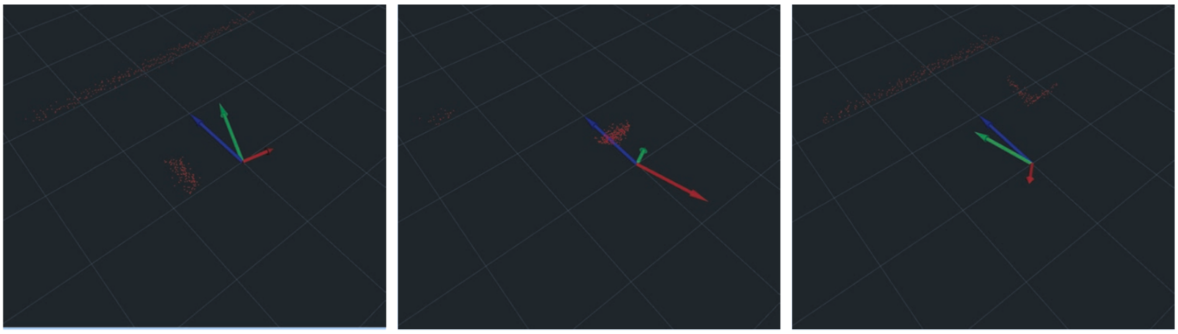
- The type of message published is **visualization_msgs/msg/MarkerArray.**
- The type of each marker is **visualization_msgs/msg/Marker** and should be an arrow.
- Make sure to publish the debugging messages if and only if the number of subscribers to the debugging topic is greater than zero.

Running the AvoidanceNode

- Use a single threaded executor to run your avoidance node.
- Make sure to use config files to configure the different parameters of the node and algorithm (topic name, threshold distance, gains, ...)

Sample Run

Blue: Attractive vector
Red: Repulsive vector
Green: Result vector



Submission Guidelines

GitHub Repository Creation

- Create a new GitHub repository named GroupName-VFF-Avoidance.
- Initialize the repository with a README.md file that briefly describes the project and its objectives.

- Make sure to mention the group members as authors in your README.md
- Ensure the repository is public so that it can be accessed for evaluation.

Code and Documentation

- Ensure all code is well-commented, and each file has appropriate documentation describing its functionality.
- Include comments in your code explaining any unique approaches or significant modifications to the algorithm.

Submission Deadline

- Deadline: Submit your GitHub repository link by Monday