AUE-893 Autonomy: Science and Systems

Repository: https://github.com/rmerco-clemson/AuE893Autonomy.git

Commands for launching the applications

The assignment is totally contained in the package “assignment_03” with its own launch scripts.
The package “assignment_03” has the following folder tree:

- Launch:
  - empty_world.launch. It launches the empty Gazebo world by typing the command:
    ```
    roslaunch assignment_03 empty_world.launch
    ```
  - playground_world.launch. It launches the playground Gazebo world by typing the command:
    ```
    roslaunch assignment_03 playground_world.launch
    ```
  - keyboard_teleop.launch. It launches teleoperation using keyboard.
    ```
    roslaunch assignment_03 keyboard_teleop.launch
    ```
  - turtlebot_app.launch. It launches the python script able to run the turtlebot autonomously in order to avoid the obstacle.
    ```
    roslaunch assignment_03 turtlebot_app.launch
    ```
- worlds:
  - empty.world: Gazebo world totally empty.
  - playground.world: default Gazebo world.
- Scripts:
  - turtlebot_app.py. It is the python script which runs the nodes able to create the logic that the turtlebot uses to avoid the obstacles.
  - fake_odom_node.py. Create a fake robot which follow a circular trajectory. It is useful to create a target for the turtlebot.

Logic of obstacles avoidance

The robot is provided of a front laser sensor able to identify the presence of obstacles in front of itself in a range of (-0.52,0.52) radians. This range is divided by interval of 0.001636 radians in order to provide a matrix where each element represents a portion of the cone of visibility. Each elements provide a value between 0.45 and 10 which represents how far (in meters) is the detected object in that particular sector.
The logic is able to detect the closest object detected in all the sectors and identify the overall sector in left and right.
The logic can be summarized as in the following steps:

1) define the target to reach and try to reach it:

- calculate the orientation
- rotate to the orientation with a PD controller
- once the orientation is reached, go to target straight using a PD controller for the speed

2) If the closest detected object is at less than 0.6 meters:
   - the robot stops
   - the robot turns in the opposite direction of the detected sector using an increment of 25 degrees: if a right side object is detected the robot will turn on left, while if a left side object is detected the robot will turn on right;
   - the robot tries to go forward using the same orientation for 1 meter, if not possible, go again to the point 2;
   - the robot orientates again as in the point 1 trying to point the target.
3) the algorithm will conclude once the target is reached.