Obstacle Manager Block Design

Block Owner: Qusai Alawlaqi
Date: Feb-09-2021

Design Details

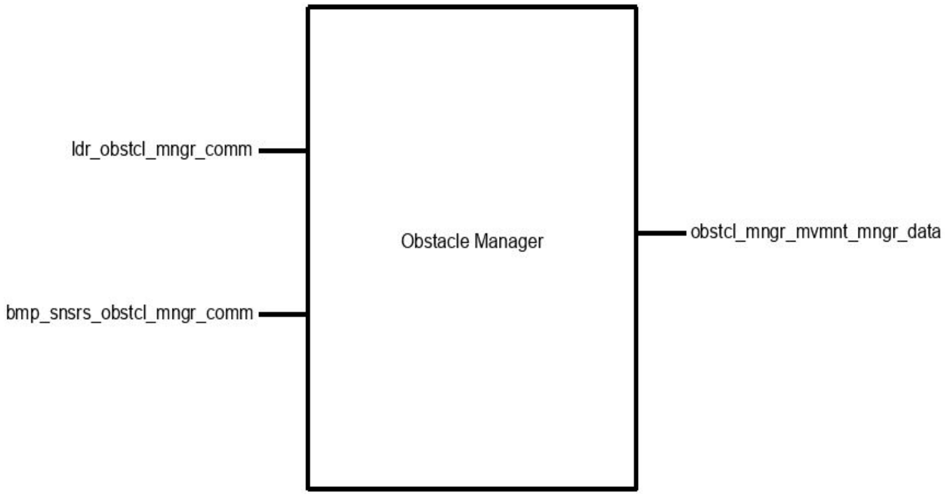

Figure 1: Black box for Obstacle Manager

The interfaces:

ldr_obstcl_mngr_comm: The LiDAR sensor should send/publish the data to a topic called /scan. And the obstacle manager should subscribe to it and convert the data to a named regions.

bmp_snsr_obstcl_mngr_comm: The Bump sensor should send/publish the data to a topic called /bump. And the obstacle manager should subscribe to it and assign it to a boolean variable.

obstcl_mngr_mvmnt_mngr_data: At the end of deciding which case we are on, we should send linear and angular speeds to the movement manager by publishing it to a /cmd_vel topic.

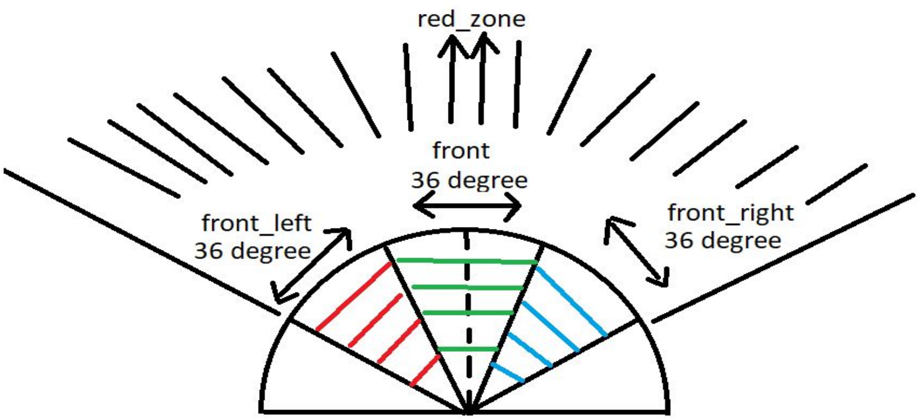

Figure 2: LiDAR Data

We can see in Figure 2, how we are going to divide the front 3 sections of the LiDAR data:

Red Region: front_left with 36 degree

Green Region: front with 36 degree

Blue Region: front_right with 36 degree

The Red Zone is the area where we start taking actions.

So, the total angle is 108 degrees for now, and it can be easily adjustable after testing it in real-world.

Cases	front_left	front	front_right	Decision
0				Moving Forward
1		X		Turn Right/Left
2			X	Turn Left
3		X	X	Turn Left
4	X			Turn Right
5	X	X		Turn Right
6	X		X	Moving Forward (Slowly)
7	X	X	X	Turn Right/Left

Table 1: All possible cases for the LiDAR

Pseudocode:

```
Getting data from LiDAR and Bump Sensors. (Subscribe to a ROS topics)
    Call a moving function (Whenever we receive a data from the subscribed topic(callback))
Moving function
    Initialize linear and angular speeds.
    Set the maximum distance that an object can be away from us (red_zone)
    Divide the LiDAR data to 3 different regions (front_Left, front, front_right)
    Get the minimum value(distance)of each region.(only if it’s less than the red_zone value)
    If bump sensor collides with an object
        Set linear speed to zero
        Set angular speed to zero
    If-else there is no object detected in the red_zone
        Set linear speed to a positive speed
        Set angular speed to zero
    If-else  there is an object entered the red_zone from the front or from all regions
        Set linear speed to zero
        Set the angular to either positive or negative value (turning right or left)
    If-else there is object entered the red_zone from the front_right or front and front_right
        Set linear speed to zero
        Set the angular to a negative value (turning left)
    If-else there is an object entered the red_zone from the front_left front and front_left
        Set linear speed to zero
        Set the angular to a positive value (turning right)
    If-else there is an object entered the red_zone from the front_right and front_left
        Set linear speed to a positive speed (Slowly)
        Set angular speed to zero
    Send the linear and angular speeds to the movement manager (Publish to a ROS topic)
```

Design Validation Overview

The idea of this block is to avoid any objects or obstacles we may face during our package delivery. The movement manager is responsible for switching between avoid obstacles mode and path following mode. And the priority is always for avoiding obstacles.

So, for the above details we can make sure that if any object comes to our red zone we can avoid it safely from all directions (left, right, and moving forward) and if by any reason we collide with an object we can immediately stop the robot and send an emergency message to our webpage.

We can improve the robot’s intelligence by adding more cases and sub-cases and think of both obstacle avoidance and path following simutancly.