Simulation & Modeling
CSE3102
Lab Exercise-7
Turtle Bot and Irritated Robot

Rishikesh Raj Nair
20BRS1245
24/02/2023

Q1. Exercise 1:

If the robot encounters an obstacle at a threshold distance of 0.5, then the robot engages in a twisted motion or circular motion
If no obstacle, the robot moves forward with a nominal speed.

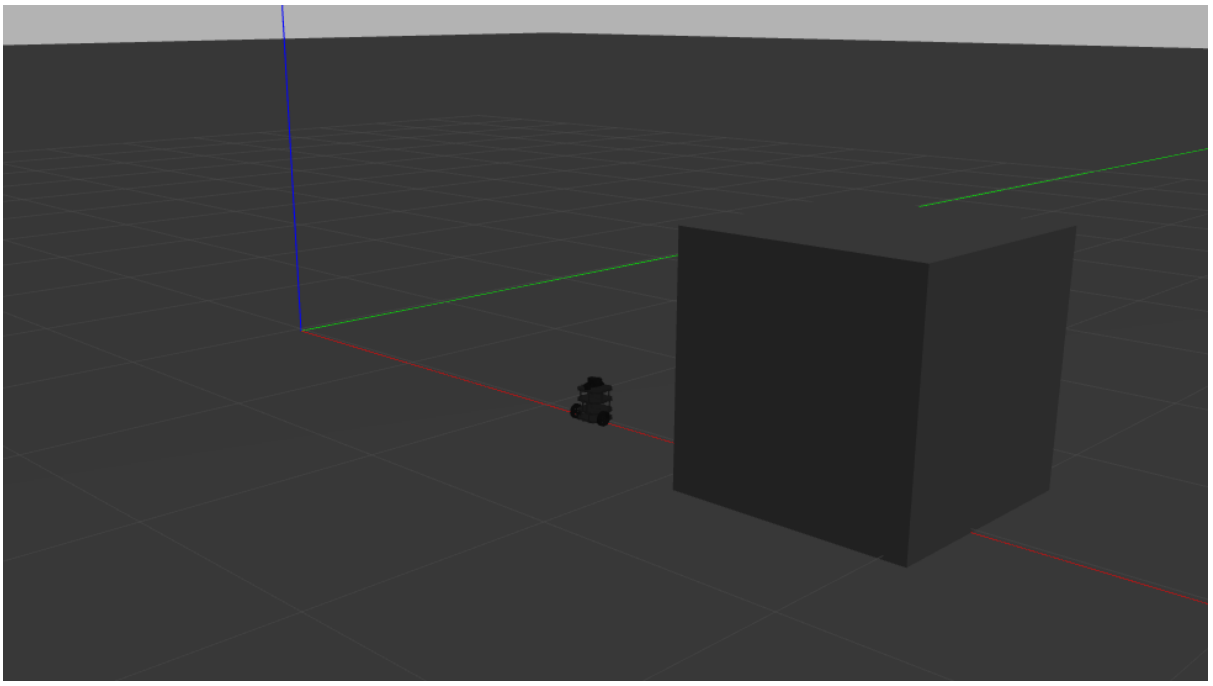
CODE

```
#!/usr/bin/python3

import rospy
import numpy as np
from numpy import inf
from geometry_msgs.msg import Twist
from sensor_msgs.msg import LaserScan
import sys

class object_irritation_robot:
```

```
    def __init__(self):
        rospy.Subscriber("/scan", LaserScan, self.laserData_cb)
        self.pub = rospy.Publisher('/cmd_vel', Twist, queue_size=10)
        self.robot_velocity=Twist()

    def laserData_cb(self,data):
        laser_data=np.array(data.ranges)
        laser_data[laser_data == inf] = 0
        laser_data=max(laser_data)
        rospy.loginfo(laser_data)

        if(laser_data > 0.5):
            self.irritated()
        else :
            self.move_forward()

        self.pub.publish(self.robot_velocity)

    def irritated(self):
        rospy.loginfo("I am Irritated")
        self.robot_velocity.linear.x=0.0
        self.robot_velocity.angular.z=2.0

    def move_forward(self):
        rospy.loginfo("Lets Moving ON ")
        self.robot_velocity.linear.x=0.5
        self.robot_velocity.angular.z=0.0




if __name__ == '__main__':
    rospy.init_node('object_irritation_robot', anonymous=True)
```

object_irritation_robot()
rospy.spin()

OUTPUT



Exercise 2:

From the above example, make the irritated robot to a diplomat robot where the robot moves away from the obstacle and move forward with a nominal speed.

CODE

```
#!/usr/bin/python3

import rospy
```

```
import numpy as np
from numpy import inf
from geometry_msgs.msg import Twist
from sensor_msgs.msg import LaserScan
import sys

class object_irritation_robot:

    def __init__(self):
        rospy.Subscriber("/scan", LaserScan, self.laserData_cb)
        self.pub = rospy.Publisher('/cmd_vel', Twist,
queue_size=10)
        self.robot_velocity=Twist()

    def laserData_cb(self,data):
        laser_data=np.array(data.ranges)
        laser_data[laser_data == inf] = 0
        laser_data=max(laser_data)
        rospy.loginfo(laser_data)

        if(laser_data > 0.5):
            self.irritated()
        else :
            self.move_forward()

        self.pub.publish(self.robot_velocity)

    def irritated(self):
        rospy.loginfo("I am Diplomatic & Diverted!")
        self.robot_velocity.linear.x=0.5
        self.robot_velocity.angular.z=0.785

    def move_forward(self):
```

```
        rospy.loginfo("Lets Moving ON ")
        self.robot_velocity.linear.x=0.5
        self.robot_velocity.angular.z=0.0
```

```
if __name__ == '__main__':
    rospy.init_node('object_irritation_robot', anonymous=True)
    object_irritation_robot()
    rospy.spin()
```

OUTPUT