

HOMEWORK 3

ENPM 690

ROBOT LEARNING

Prof. Don Sofge



UNIVERSITY OF MARYLAND

Kushagra Agrawal (116700191)

Problem 1

Program a simple robot vehicle in a simulated environment (robot simulation tools and libraries may be used). Your simulated robot should exhibit at least one sensor input (e.g., forward-looking range sensor that returns the distance to the nearest obstacle) and two control outputs (e.g., left and right wheels, or speed and direction of vehicle motion). Show that you can drive your robot around through mouse or keyboard inputs.

Solution:

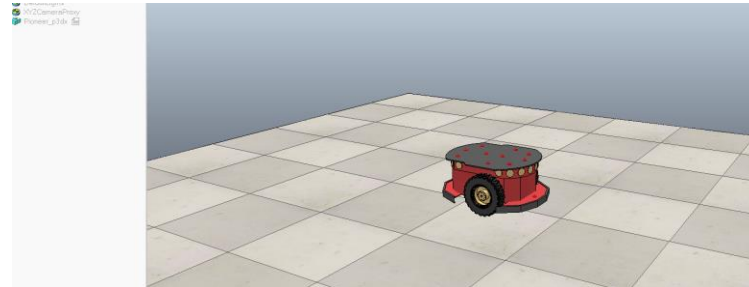


Fig 1

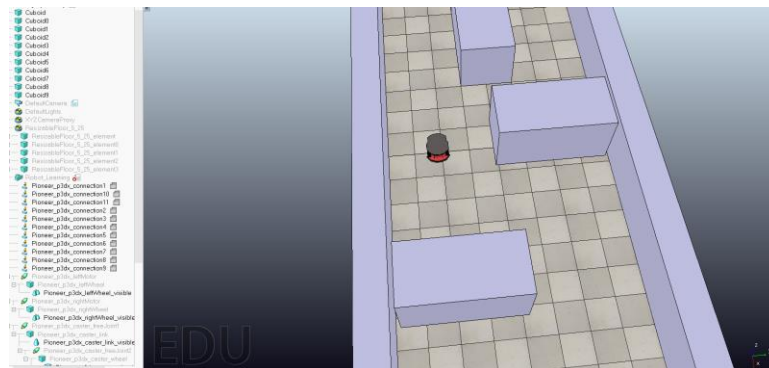


Fig 2

Approach:

- I have used V-rep to show the simulation and python for controlling my Robot through keyboard.
- To use the remote API functionality in a Python script we need 3 files in our source folder:
 1. vrep.py
 2. vrepConst.py
 3. remoteApi.dll
- I installed a library called “pynput” through which we can access our keyboard or mouse.

- I used a pre-existing model called as “Pioneer” for my simulation.
- The sensor input which I have taken or added are Front left sensor, Front right sensor, Left sensor and Right sensor.
- The two control outputs are Right and Left Motors.
- The biggest task was setting up a remote API between V-rep and Python as sometimes V-rep used to crash in the middle.
- Finding a correct remote API function to set the target velocity of the joint was one of the tasks.
- When different keys such as ‘up’, ‘down’, ‘left’, ‘right’ different combination of motors are activated.
- To set the desired velocity “vrep.simxSetJointTargetVelocity” function was used.
- First, I was using ASCII characters for controlling the robot with keyboard but finally I used “pynput” library through which we can directly access the keyboard.
- According to the action keys pressed, the motors get activated and the robot turns left, right and move forward or backward.

Problem 2

Add a programmed behavior to your robot, such as following (or avoiding) a light, or wandering, while avoiding collisions with obstacles.

Approach:

- Proximity sensor is used for detecting the collisions and avoiding it.
- The data from the proximity sensor is used to detect the object.
- If the sensor detects an object, then the distance between the object and the robot is calculated and if it is in a range which is set by us then the robot turns to avoid colliding with it.
- Four distances are being calculated which are front left, front right, side right and side left distance.
- To make the robot wander, sometimes it takes random turns even if there is no obstacle.
- The problem was selecting the appropriate range of distance which was calibrated by trial and error.
- The speed of the robot can be varied by changing the value of the speed variable in the script.

GITHUB link - https://github.com/kushagra7176/ENPM690_RobotLearning_Project3