Anirudh Topiwala UID: 115192386, Homework #3, ENPM 808F- Robot Learning

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.

- **Software's used:** Matlab, V-REP

I have simulated a differential drive robot in VREP using commands from Matlab. The model being used here is that of Pioneer_p3dx. It is an object avoidance robot equipped with ultrasonic sensors. The environment is created in vrep. It consists of cuboids and cylinders of various sized and a bounded wall so that the robot does not fall off.

Sensor Input: Ultrasonic Sensors

Control Outputs:

- Control of Robot using Arrow Keys
- Video output to Matlab using a camera mounted on the robot
- Linear and Angular velocities of both the left and the right wheels.

As we can see in the attached video the robot is moving according to the keys pressed on the keyboard. A keywait function is used to interrupt the code if a keyboard key is pressed. The control keys are as follows:

W= Forward   A= Left   D= Right   S= Reverse   E= Stop

The camera attached on the robot transmits the video to Matlab. The resolution rate is set at 512x512.

The code is attached at the end for reference.

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

- **Software's used:** Matlab, V-REP

The programmed behaviour added here is that of object avoidance. Three sensors have been used to estimate the distance of obstacle from the robot. The sensors selected are at the front, left and right side of the robot. The sensors at the back are not used yet. (it is assumed for now, that the robot will only move in forward direction).

Sensor Input: Ultrasonic Sensors

Control Outputs:

- Control of Robot using object avoidance algorithm
- Distance readings of the three ultrasonic sensors.
- Video output to Matlab using a camera mounted on the robot
- Linear and Angular velocities of both the left and the right wheels.

A minimum distance of 0.35m is used as the avoidance limit. That is if the distance is less than 0.35m the robot will turn.

As seen in the simulation the robot is avoiding obstacles and keeps on wandering in the enclosed space.

Future Goals: Try to smoothen out the object avoidance. As seen in the video the turns are jerky as opposed to the output of the inbuilt scripts of the object avoidance robot. Also, I would want to implement the gradual change in speeds during motion as seen in the default script.

The code is attached at the end for reference.

The YouTube link for the Video is: https://youtu.be/V2szTzmPurU

Code: Paste the code in Matlab and configure VREP to Execute.

1) Pioneer:

```
vrep=remApi('remoteApi');
 vrep.simxFinish(-1);
 clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);

 if (clientID>-1)
        disp('Connected');
     motion= input('To move the Robot, Press W,A,S,D for Forward, Left, Reverse, Right, and Press e to end','s');
    while(motion~='e')
   if (motion=='W' ||motion=='w' || motion=='A' ||motion=='a' || motion=='S' ||motion=='s' || motion=='D' ||motion=='d')

%         code here
%         Handle
    [returnCode,left_Motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_leftMotor',vrep.simx_opmode_blocking );
    [returnCode,Right_Motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_rightMotor',vrep.simx_opmode_blocking );
    [returnCode,front_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSensor5',vrep.simx_opmode_blocking );
    [returnCode,camera]=vrep.simxGetObjectHandle(clientID,'Vision_sensor',vrep.simx_opmode_blocking );

%     Other Code
   if( motion=='W' ||motion=='w')
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,2,vrep.simx_opmode_blocking);
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,2,vrep.simx_opmode_blocking);
%     pause(1)
%     [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
%     [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
   elseif( motion=='A' ||motion=='a')
   [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,2,vrep.simx_opmode_blocking);
   [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-2,vrep.simx_opmode_blocking);

%    pause(1)
%    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
   elseif( motion=='S' ||motion=='s')
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-2,vrep.simx_opmode_blocking);
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,-2,vrep.simx_opmode_blocking);
%      pause(1)
%     [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
%     [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
   elseif( motion=='D' ||motion=='d')
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,2,vrep.simx_opmode_blocking);
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,-2,vrep.simx_opmode_blocking);

%       pause(1)
%     [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);

   end
    [returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,front_Sensor,vrep.simx_opmode_streaming);
    [returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.simx_opmode_streaming);
    [returnCode,leftlinearvelocity, leftangularvelocity]=vrep.simxGetObjectVelocity(clientID,left_Motor,vrep.simx_opmode_streaming);
    [returnCode,rightlinearvelocity, rightangularvelocity]=vrep.simxGetObjectVelocity(clientID,Right_Motor,vrep.simx_opmode_streaming);

    [returnCode,leftlinearvelocity,leftangularvelocity]=vrep.simxGetObjectVelocity(clientID,left_Motor,vrep.simx_opmode_buffer);
    [returnCode,rightlinearvelocity, rightangularvelocity]=vrep.simxGetObjectVelocity(clientID,Right_Motor,vrep.simx_opmode_buffer);
    [returnCode,detectionState,detectedPoint,~,~]=vrep.simxReadProximitySensor(clientID,front_Sensor,vrep.simx_opmode_buffer );
    [returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.simx_opmode_buffer);
    imshow(image)
    fprintf("Obstacle Distance from Robot= %d m \n",norm(detectedPoint));
    fprintf("Linear Velocity of Left Wheel= %d m/s \n",leftlinearvelocity);
    fprintf("Linear Velocity of Right Wheel= %d m/s \n",rightlinearvelocity);
    fprintf("Angular Velocity of Left Wheel= %d rad/s \n",leftangularvelocity);
    fprintf("Angular Velocity of Right Wheel= %d rad/s \n",rightangularvelocity);


%     disp(leftlinearvelocity);
%     disp(rightlinearvelocity);
%     disp(leftangularvelocity);
%     disp(rightangularvelocity);

     pause(0.1);

  else
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);

   end
  motion = getkeywait(2);
  continue



    end
    vrep.simxFinish(-1);
 end

 vrep.delete();
```

2) Object avoidance:

```
vrep=remApi('remoteApi');
 vrep.simxFinish(-1);
 clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5);
 dpcenter = 1;i=0;
 if (clientID>-1)
        disp('Connected');

%          code here
while( dpcenter~='null' )
i=i+1;
%          Handle
    [returnCode,left_Motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_leftMotor',vrep.simx_opmode_blocking );
    [returnCode,Right_Motor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_rightMotor',vrep.simx_opmode_blocking );
    [returnCode,front_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSensor5',vrep.simx_opmode_blocking );
    [returnCode,left_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSensor3',vrep.simx_opmode_blocking );
    [returnCode,right_Sensor]=vrep.simxGetObjectHandle(clientID,'Pioneer_p3dx_ultrasonicSensor7',vrep.simx_opmode_blocking );

    [returnCode,camera]=vrep.simxGetObjectHandle(clientID,'Vision_sensor',vrep.simx_opmode_blocking );

%      Other Code

% Reading sensor readings
    [returnCode,detectionStatecenter,dpcenter,~,~]=vrep.simxReadProximitySensor(clientID,front_Sensor,vrep.simx_opmode_streaming);
    [returnCode,detectionStateleft,dpleft,~,~]=vrep.simxReadProximitySensor(clientID,left_Sensor,vrep.simx_opmode_streaming);
    [returnCode,detectionStateright,dpright,~,~]=vrep.simxReadProximitySensor(clientID,right_Sensor,vrep.simx_opmode_streaming);

    dpcenter= norm(dpcenter);
    dpleft= norm(dpleft);
    dpright= norm(dpright);

    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,3,vrep.simx_opmode_blocking);
    [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,3,vrep.simx_opmode_blocking);


    if ( dpcenter<0.35 && dpcenter>0.001 || dpleft<0.35 && dpleft>0.001 || dpright<0.35 && dpright>0.001)
        if ( dpleft<0.35 && dpleft>0.001 && dpright< 0.01 && dpcenter< 0.01)
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0.2,vrep.simx_opmode_blocking);
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,-0.2,vrep.simx_opmode_blocking);
             pause(0.05)
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);

        elseif (dpright<0.35 && dpright>0.001 && dpleft< 0.01 && dpcenter< 0.01)
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-0.2,vrep.simx_opmode_blocking);
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0.2,vrep.simx_opmode_blocking);
             pause(0.05)
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
            [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
%         else
%             a= power(-1,i);
%             if(a==1)
%                 [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-0.2,vrep.simx_opmode_blocking);
%                 [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0.4,vrep.simx_opmode_blocking);
%                  pause(2.5)
%                 [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
%                 [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
%             else
%             [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-0.2,vrep.simx_opmode_blocking);
%             [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0.4,vrep.simx_opmode_blocking);
%              pause(2.5)
%             [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
%             [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);
            end

                [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,-0.5,vrep.simx_opmode_blocking);
                [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0.5,vrep.simx_opmode_blocking);
                 pause(0.2)
                [returnCode]=vrep.simxSetJointTargetVelocity( clientID,left_Motor,0,vrep.simx_opmode_blocking);
                [returnCode]=vrep.simxSetJointTargetVelocity( clientID,Right_Motor,0,vrep.simx_opmode_blocking);


    end




    [returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.simx_opmode_streaming);
    [returnCode,leftlinearvelocity, leftangularvelocity]=vrep.simxGetObjectVelocity(clientID,left_Motor,vrep.simx_opmode_streaming);
    [returnCode,rightlinearvelocity, rightangularvelocity]=vrep.simxGetObjectVelocity(clientID,Right_Motor,vrep.simx_opmode_streaming);

    [returnCode,leftlinearvelocity,leftangularvelocity]=vrep.simxGetObjectVelocity(clientID,left_Motor,vrep.simx_opmode_buffer);
    [returnCode,rightlinearvelocity, rightangularvelocity]=vrep.simxGetObjectVelocity(clientID,Right_Motor,vrep.simx_opmode_buffer);
    [returnCode,resolution,image]=vrep.simxGetVisionSensorImage2(clientID,camera,0,vrep.simx_opmode_buffer);
    imshow(image)
    disp(norm(dpcenter));
    disp(norm(dpleft));
    disp(norm(dpright));
%     disp(leftlinearvelocity);
%     disp(rightlinearvelocity);
%     disp(leftangularvelocity);
%     disp(rightangularvelocity);

   end
    vrep.simxFinish(-1);
 end
 vrep.delete();
```