Fuzzy Logic Controller

M28COM Evolutionary and Fuzzy Systems

Robot Simulation Tutorial

# Intro

This tutorial will show you how to simulate a virtual robot using V-Rep and control it through a Fuzzy Logic Controller implemented in Matlab. The robot used in this tutorial is a Pioneer PD3X robot. For more information on the robot visit: “http://www.inf.ufrgs.br/~prestes/Courses/Robotics/manual\_pioneer.pdf”.

Although Matlab is used in this example, the concepts covered in this tutorial can be applied to any other programming language supported by V-Rep (C++, Lua, Python, Java, etc.). The following link contains a list of the API methods supported by V-Rep “[http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab. htm](http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.%20htm)”.

# Setting up V-Rep

## Creating a scene

If you already have a scene, simply load it by clicking File Open scene.

1. Click Start All programs V-Rep Educational V-REP PRO EDU.

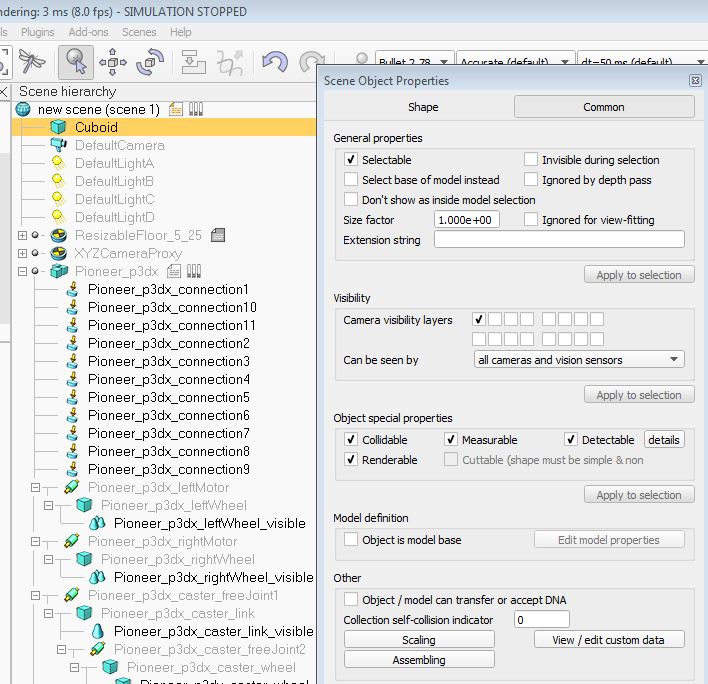
* Do not search for V-Rep through the search bar, scroll down and find it, else it will not load all the required libraries.

1. Find the Model browser menu robots mobile. Scroll down and find the pioneer p3dx robot. Click on the robot and drag it onto the scene.
2. Under the “Scene hierarchy” menu you will see the “Pioneer\_p3dx” object and if you expand it you will be able to see all the components of the robot such as the ultrasonic sensors and their default name.
3. Click on the play button and you should see the robot moving (of course it will go off the scene). If the pd3x has fallen off the scene, simply stop the simulation and it should restart the scene.

## Adding objects to the scene

1. Add Primitive shape cuboid. Modify the X/Y/Z-size parameters as desired. You can adjust these later by clicking on the object icon.
2. Before doing anything else, click on the new object icon common collidable renderable detectable. This will allow you to detect the object using the sensors of the robot.

Once this is done you can adjust the size of the object by clicking (same window) Shape adjust geometry and setting the X, Y, and Z values as desired.



1. Repositioning the object is done by clicking on the object/item shift button: C:\Users\ac1753\Desktop\FuzzyLogic\shiftButton.JPG

From here you can simply drag the object to the desired position or adjust the X/Y/Z-coord values on the menu that popped up when clicking the shift button.

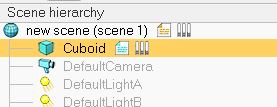
The object can also be rotated by clicking on the rotate button: C:\Users\ac1753\Desktop\FuzzyLogic\rotateButton.JPG

1. C:\Users\ac1753\Desktop\FuzzyLogic\ScriptsButton.JPGOnce you have finished building your scene/environment click on the scripts menu Non-threaded child script (pioneer\_p3dx) disabled. This will stop the sample script from running and controlling the robot. You can now play the simulation and notice that this time the robot will not move. From here on you need to make sure that before running a Matlab script the simulation is running.

hint: if you can't see some of the menus or properties of an object, stop the simulation.

# Linking Matlab to V-Rep

1. Stop the simulation and right click on the icon of one of the objects you have added to the scene (for example a cuboid). Select Add Associated child script Threated.



1. Click on the child scrip you just created and scroll down to the line “simSetThread SwitchTiming(2)” and paste the following line after it:

simExtRemoteApiStart(19999)

This line specifies the port on which the remote API server will run. It is important that you don’t do this on the main script to avoid leaving the robot in an unusable state.

1. Start the simulation again.
2. Navigate to the following folder and copy all the files into the folder in which you will be writing your Matlab scripts (for example “Desktop/FLC/”).

“C:\Program Files (x86)\V-REP3\V-REP\_PRO\_EDU\programming\remoteApi Bindings\ matlab\matlab”

Your path may differ according to your installation but it should be similar. In any case the “programming\remoteApi Bindings\ matlab\matlab” bit should be the same.

1. Also copy the file “remoteApi.dll” into the same folder. This is located at the following path:

“C:\Program Files (x86)\V-REP3\V-REP\_PRO\_EDU\programming\remoteApi Bindings\lib\lib\ 64Bit“

It is important to make sure that you copy the 64bit file and not the 32bit one if you will be using a 64bit Matlab version.

1. Open up Matlab2015 and locate the folder where you copied the V-Rep files into.
2. Open the SampleTest.m file and run it. If you’re V-Rep simulation is running and you have followed all the steps you should be able to see a message on V-Rep saying “Simulation started, Hello V-REP!”. Moreover, Matlab will be displaying “Mouse position x: ####” for a number of iterations. This means that you have successfully connected Matlab and V-Rep.
3. Get familiar with the sample script and observe what is happening, how data is being sent to and from V-Rep. Here are some of the lines of code that you must include in your script:

vrep = remApi('remoteApi'); % To access vrep methods

vrep.simxFinish(-1); % close all exiting opened connections

%connect to the robot

clientID = vrep.simxStart('127.0.0.1',19999,true,true,5000,5);

%To access sensors you need to create handles. The following line will create a handle for the ultrasonic sensor 1 (the robot has 16)

[errCode, sensorHandle] = vrep.simxGetObjectHandle(clientID, strcat('Pioneer\_p3dx\_ultrasonicSensor1'),vrep.simx\_opmode\_bloc king);

%Make the left motor of the robot move forward with a velocity of 1. First create a handle and then set the velocity:

[errCode, motorLeft] = vrep.simxGetObjectHandle(clientID, 'Pioneer\_p3dx\_leftMotor', vrep.simx\_opmode\_oneshot\_wait);

vrep.simxSetJointTargetVelocity(clientID, motorLeft, 1, vrep.simx\_opmode\_streaming);

% Close the connection to V-REP

vrep.simxFinish(clientID);

vrep.delete(); % call the destructor!

# Using the sample FLC

1. Load the “FLC.m ” file into Matlab.
2. Load the “FLC\_VREP\_Scene” file into V-Rep.
3. Assuming that the “SugenoFLC.fis” file is in the same directory, run the FLC.m file and observe the robot moving around and turning left every time it gets close to an obstacle.
4. Get familiar with the code in the FLC.m file, the comments explain what each line of code does.
5. You can also open the Fuzzy Logic GUI on Matlab by tying fuzzy on the command line and then open the SugenoFLC.fis file to observe what is happening. You can try and add rules or modify the parameters, save it again and run the FLC.m file again to see the results.

You can now implement your FLC and save it as a .fis file. You can then load it to your Matlab script and run it by setting it to evaluation mode as explained in the FLC.m file. The following line, for example, will return the output of a FLC that takes 3 inputs, each one being an ultrasonic sensor reading (the distance at which an object is). The output is a table with two values: the speed for the right and left motors of the robot.

speed = evalfis([flc\_input1, flc\_input2, flc\_input3],flc)

The basic idea is to decide what sensors and parameters you want to use as input for your FLC and what you want as an output. In the example provided only the values of four sensors are used (two of them combined) and the FLC only allows the robot to turn right. You can improve this by allowing the robot to go left and back as well. When making the robot reverse (by giving it a negative speed value) you can use the rear ultrasonic sensors to see if there is an obstacle or not.

You can also manipulate the way the ultrasonic sensors work: click on the ultrasonic sensor show detection parameters and modify accordingly. In the scene provided they have a reach of 1meter in Ray mode. But this can be changed to get a reading over and angle according to XYZ coordinates.

Check out the following documentation to see what Remote API functions you can call from Matlab. It is important that you collect all the return values of each function (even if you don’t use them) or your code may not behave appropriately.

http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm