Understanding V-REP simulator

Author: R Hariharan

Introduction

V-REP simulator is used to analyse the working of your robot. We used V-REP to simulate the “Circle formation algorithm” and the “Generic shape formation algorithm”. We found V-REP to be an easier simulator to use, compared other simulators like gazebo. V-REP provides the user with an easy to use GUI and different API’s for the convenience of the user. This document just gives a brief idea of V-REP, to know more go to V-REP official website.

Installation/Download

V-REP is available for Ubuntu, Mac and Windows. We used V-REP in Ubuntu 16.04 platform. Here’s a link to download V-REP @ http://www.coppeliarobotics.com/downloads.html. The Educational licence provides all features of V-REP.

After downloading V-REP for ubuntu, navigate to that folder and then open your terminal in that folder itself. Type ***./vrep.sh*** to run the simulator.

In case you are using Mac or Windows check out the V-REP forum for help.

GUI of the Simulator

When you run the simulator you can see that it has multiple windows open. One of them is the scene window which consist the scene hierarchy, this is where you models will be shown. There is model browser which shows all the available robots that are present in V-REP (check out the mobile robots present in the robot). Simulation window is where the models in the scene are show.

The simulator has a menu bar which gives access to all the available tools in the simulator. The tool bar gives clickable icons, to control the simulation environment (move objects around, rotate the objects, zoom in/out of the simulation window, move the simulation window around, rotate the view of the simulation window etc).

The tool bar also has the simulation tools, which allows the user to start, pause and stop the simulation. It has dropdown menus where you can set the physics engine you want to use and the time interval to run each simulation cycle. There are two button with icons as tortoise and rabbit to decrease and increase the simulation speed repectively.

For detailed explanation visit the **User Interfaces** section of the V-REP User Manual.

Models and Collections

Scenes have objects like cameras, shapes, sensors etc. These objects are nothing but models. So a robot has multiple models attached to it. A collection is a set of models. So a robot is not a model but a collection. In order to work with a robot we need to define it has a collection.

For detailed explanation visit the **Scenes and Models** section of the V-REP User Manual.

Writing Code

Any user can program into V-REP using these ways, 1) Embedded Scripts, 2) Plugins, 3) Remote API, 4) Add on’s and 5) ROS node.

I used embedded scripts to program my robot. Embedded scripts are written in a programming language called lua.

There are two types of embedded scripts, Main script – which takes care of initializing the simulator environment and running all the child scripts. Second is the Child Scripts which is embedded with are mobile/non-mobile robots. **Do not change or edit the Main Script**. The script is shown as a page beside the name of the model in the scene hierarchy. Double clicking on the icon opens the script in a text editor of lua. The Child Scripts are further divided into threaded and non-threaded scripts. To get a better knowledge about the threaded and non-threaded child scripts visit http://www.coppeliarobotics.com/helpFiles/en/childScript.htm.

The embedded scripts use the regular API’s.

To know more about other means of coding visit the **Writing code in and around V-REP** section of the V-REP User Manual.

Functioning of the Simulator

To run the simulator we should click the play button, we can set the simulation speed and the physics engine from the drop down menu. When the simulation starts it runs all the embedded scripts parallely. The embedded scripts runs in this fashion, first it runs the **intialisation** part of the child script once then it run the **actuation** part and the **sensing** part in a loop. When the simulation is stopped the **cleanup** part of the code is run.

This cycle keeps happening based on the simulation time interval you have set, hence you can’t run loops (while or for loops) which runs for more then the time interval set.

Remeber this is only for embedded child scripts. To know more about other means of coding visit the **Writing code in and around V-REP** section of the V-REP User Manual.

Building your own model

V-REP User Manual has a **tutorials** section. In this section the first tutorial is to build a bubbleRobot. Doing this tutorial will help in understanding the V-REP simulator better. After completion of this tutorial, any user will be able to use the simulator easily. This tutorial explains how to create shapes, add sensors to it, change sensor variable like range of the sensor, placing of the sensors (positioning) and even orientation, adding scripts to the robot, making a collection of the robot etc. This tutorial is really helpfull.

References

1) To understand the simulator to its depth read the V-REP User Manual.

2) If you have any question visit the V-REP Forum.

3) List of all regular API functions.