Intelligent Robotics MSc Summer Technical Challenge

Mobile Robot
Introduction to V-REP

Mark A Post (mark.post@york.ac.uk)

1. Aims and Objectives

This is the first step in the Intelligent Robotics MSc Summer Technical Challenge.

This challenge can be done from home, either by starting a PC or VM from a bootable Ubuntu image or by installing the needed software on your own computer. It is intended to set you up with a simulation and programming environment that you can use anywhere to complete robotics laboratory tasks.

2. Learning outcomes

By the end of this challenge, you should be able to:

- Create a robot simulation in V-REP and control a robot with Python
- Test algorithms on virtual robots in place of real ones in the laboratory

3. Software and hardware

- V-REP (CoppeliaSim) Software (https://www.coppeliarobotics.com/)
- Example simulation environment (v-rep-lab.ttt) and control scripts (v-rep-lab.py and v-rep-maze.py) provided in the Appendix.
- Python or iPython development environment (Spyder3 is recommended)

4. Pre-Lab Preparation

- Download the "Edu" version of the V-REP software for your PC platform
- Browse through the user manual at https://www.coppeliarobotics.com/helpFiles/
- Briefly have a look at the BubbleRob tutorial that is located at https://www.coppeliarobotics.com/helpFiles/en/bubbleRobTutorial.htm

5. Install and Start the V-REP Simulator

V-REP, the Virtual Robot Experimentation Platform. (recently badge-engineered into CoppeliaSim) is a fully-featured robot simulation software environment created by Coppelia Robotics. It includes an integrated development environment that allows you to model, edit, program and simulate any robot or robotic system. Although it is proprietary for industrial use, it is free and open-source to educational institutions and students, making it a relatively good trade-off between freedom and usability. It is installed on the lab PCs at the University of York and you are expected to install it on your own PC or run it from an Ubuntu Linux image for remote work.

Installing V-REP is quite simple, as it is distributed as a self-contained package. Download "V-REP/CoppeliaSim **Edu**" for your desired platform from the site https://www.coppeliarobotics.com/downloads

- On Linux (Ubuntu is best supported but most other distributions will work also), download the appropriate .xz archive file and unpack it into the location you want to run it from (there is no installer). Most Linux distributions would install it into /opt/ and you can unpack it there with: `sudo tar -xvJf <filename>.tar.xz -C /opt/`
- On Mac, you must also unpack it from the archive, and it can be moved into the Applications folder if you want it treated as a system application.
- On Windows, you can either unpack the binary archive, or run the downloadable installer that will take care of system installation for you.

Once you have installed V-REP/CoppeliaSim, you can start it either by finding it in the applications menu, or running the coppeliasim.sh or coppeliasim.exe executable.

If you are on your own PC and using Python for controlling your robots (recommended but not required), you will need to ensure you have Python 3 (Python 2 is officially depreciated...*finally*) and the libraries you need installed for your OS. It is recommended that you install the python3-numpy and python3-opencv libraries (see online documentation for your OS on how to do this), but you can use any libraries you would like in addition as there are a vast number of potentially useful ones available.

6. Get Familiar with the Simulator

When you open V-REP, the toolbar on the top of your window with  buttons is used for manipulating the camera, objects and the simulation itself. You can control the camera with the  buttons on the top toolbar, and you can use the "fit to view"  button to see the whole scene. The toolbar on the left is used to open up various windows that allow you to edit the properties of the scene.

You can open the Scene Hierarchy with the  button on the left toolbar if it is not open already. This shows a hierarchical display (what is connected/subordinate to what) of all the elements in your simulation. To resize the floor, select the “Floor” dummy element in the scene hierarchy and a “Floor Customizer” window will pop up with sliders to change the number of floor panels in use. **WARNING: do not use the “ESC” key or the “X” window close button to close this window or it will delete the script that resizes the floor and cannot be undone.** Instead, click elsewhere in the scene to close this window.

Try selecting various elements in the scene hierarchy and watch them as they are highlighted in the main window. With the buttons on the top toolbar, you can move , rotate , and zoom the camera, and select, move, and rotate elements in the scene (or directly transform them by editing numbers in the selection windows that pop up). You can change the various properties (mass, physical simulation, etc.) using the “Scene Object Properties” button on the left toolbar. You can also add additional elements to your scene, including other kinds of robots and static obstacles with the “Model Browser” icon.

Automation of scenes is done through scripts in the Lua language (which is more or less just a lightweight way to add dynamic scripting to C programs). You can double-click on the script icon next to the “Floor” element to see the code that creates the “Floor Customizer” dialog. Double click the script icon next to the “Robot” body element to see the script that starts the external API so that you can control the robot from outside V-REP. Also, an empty non-threaded script has been added to the “Maze” element in case you want to automate some changes to the maze on start-up using Lua code. (the scripts themselves can be closed safely using the “X” window close button)

To start the simulation itself, use the “play” button on the top toolbar. Use the adjacent “pause” and “stop” buttons for those functions. You will see time start to advance, and dynamic elements in the simulation start to move. The view from the camera will also be displayed in the “Camera” view window (if you do not see a camera view, you can open one by right-clicking anywhere in the simulation view and selecting “add -> floating view” and then right clicking on this view, selecting “view selector” and selecting “Camera”). You may also notice that the robot moves very slowly by itself without any input as the simulation progresses. This is because of small perturbation errors that occur within the physics engine in use, and can be decreased by increasing the mass of your robot. If physical movement does not seem correct, you can try different physics engines by selecting them under the “simulation” menu and comparing how well they work. Bullet is at the moment the standard real-time physics engine most often used for games and robot simulation.

7. Build a Simple “BubbleRob” Robot in Simulation

V-REP includes a graphical development environment where you can create your own robots. For project work, it will benefit you to go through the https://www.coppeliarobotics.com/helpFiles/ manual at and the relevant tutorials at https://www.coppeliarobotics.com/helpFiles/en/tutorials.htm.

You should start by building your own robot using the https://www.coppeliarobotics.com/helpFiles/en/bubbleRobTutorial.htm to help you understand robot building basics. This will be very helpful for you to be able to build simulated robots. However, to make scripting easier, the v-rep-lab.ttt scene has been created for you with a mobile robot that has additional sensors and actuators that you can use after.