

CSIP5202– Lab 2 Handout

This is a quick and simple guide to setting up CoppeliaSim with MATLAB, along with the tasks for you to take a look at and complete.

For more CoppeliaSim Resources and help look at the following:

1. Functions List: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>
2. CoppeliaSim Remote API Guide: <http://www.coppeliarobotics.com/helpFiles/en/legacyRemoteApiOverview.htm>
3. CoppeliaSim Help Guide: <http://www.coppeliarobotics.com/helpFiles/>

Opening CoppeliaSim

CoppeliaSim has been installed on the lab machines, you'll find it in the CoppeliaSim EDU folder on the start menu. Open the simulator and the cloud page will be loaded.

Setting up MATLAB and CoppeliaSim

CoppeliaSim uses a remote API to connect to CoppeliaSim as a client, therefore we need a few files to connect MATLAB with CoppeliaSim. Taken from the Coppelia Robotics website:

To use the remote API functionality in your MATLAB program, you will need following 3 items:

- remoteApiProto.m
- remApi.m
- remoteApi.dll, remoteApi.dylib or remoteApi.so (depending on your target platform)
- simpleTest.m

Above files are located in CoppeliaSim installation directory, under

\programming\legacyRemoteApi\remoteApiBindings\matlab

The remoteApi.dll may be located at:

\CoppeliaSimEdu\programming\legacyRemoteApi\remoteApiBindings\lib\lib\Windows

Copy and paste these files under a directory of your choosing on your H:/ drive. Remember to make sure you are using your shared drive and not on the Documents folder on the local machine.

In MATLAB, either check your working directory is the folder containing those files or use "Set Path" and find the folder to add the files into your workspace.

Running a Script

CoppeliaSim needs to be told to look for incoming connections on a certain port using the remote API. To do so near the bottom of the window of CoppeliaSim you'll see a Lua command text box, enter the following command:

```
simRemoteApi.start(19999)
```

This starts the simulation on port 19999. Next, in MATLAB, open and run the simpleTest.m script and you should see no errors and that the mouse position in the CoppeliaSim window is returned.

CoppeliaSim Basics

Take a few minutes to look around CoppeliaSim, and find a few details such as:

- Moving the user camera
- The model browser
- Scene Hierarchy
- Simulation

CoppeliaSim has a large number of advanced features directed to its embedded scripting and building different types of robots, we need two things:

- A robot to operate
- A scene of objects for the robot to navigate around.

Using the model browser, find and open "robots > mobile" and scroll down to find the "pioneer_p3dx.ttm" model. Click and drag this into the scene.

This is the robot we will be operating this module, looking at the robot in the scene hierarchy you'll find the robot and its components. Take a look at how it is designed and what sensors and actuators are available.

Download 'simpleExample.m' from this week's learning materials and place the file in your working directory. Open this file and read through the comments to understand what's happening.

You'll notice that in CoppeliaSim the Pioneer robot has a script attached to the object which we need to disable so MATLAB can connect to the robot successfully. To do so, go to 'Tools > Scripts', select the Pioneer_p3dx script and check the 'Disabled' box.

To run the simulation, select the play button in CoppeliaSim, and run the MATLAB script. Does the script match up with what's happening to the robot? What is being displayed to the console in MATLAB?

TASK 1: Create a version of the example map that was handed out in Lab 1.

- Check to see if the floor is the right size, there might be a customizable UI that will help you adjust the size in the scene hierarchy?
- The model browser contains a large collection of objects.

TASK 2: Create a simple program that:

1. Move at a constant speed until a object is seen within X metres.
2. Stop when a object is less than Y metres away.
3. Turn randomly in a certain direction
4. Repeat

OPTIONAL TASK 3: Replicate Task 2 in using python.