

# AI 3603 Artificial Intelligence: Principles and Techniques

## A Quick-Start Tutorial on CoppeliaSim and DR20API

This is a quick-start tutorial that teaches you how to use CoppeliaSim and APIs for DR20 robot to complete your homework. It contains CoppeliaSim installation and API use, as well as some common problems.

### 1 Software Installation

#### 1.1 Programming Environment

The APIs for DR20 has been successfully tested in the following environment:

- Python 3.6;
- Numpy 1.19.5;
- matplotlib 3.1.1.

#### 1.2 CoppeliaSim

Please download the installer package of **CoppeliaSim EDU V4.2.0 rev5** from following jBox link or official website of CoppeliaSim, <https://www.coppeliarobotics.com/downloads>, for your operating system, available for Windows, Linux and MacOS.

- Download the installer package

Installer package for Windows: <https://jbox.sjtu.edu.cn/l/21unMw>

Installer package for Ubuntu 16.04: <https://jbox.sjtu.edu.cn/l/F1RpKk>

Installer package for Ubuntu 18.04: <https://jbox.sjtu.edu.cn/l/P11kNv>

Installer package for Ubuntu 20.04: <https://jbox.sjtu.edu.cn/l/86Eh2f>

Installer package for MacOS 10.13: <https://jbox.sjtu.edu.cn/l/LFkJvZ>

Installer package for MacOS 10.15: <https://jbox.sjtu.edu.cn/l/J11ZRb>

- Installation For Windows

Follow the guide of the installer package to install and open CoppeliaSim.

- Installation For Linux (Recommended)

Unzip the package;

Open a terminal and change the directory to the path of CoppeliaSim: **cd CoppeliaSim\_Edu;**

Enter the command in the terminal to start CoppeliaSim: **./coppeliaSim**

- Installation For MacOS

Follow the guide of the installer package to install and open CoppeliaSim.

## 2 Basic Usage of CoppeliaSim

Python Remote API is one of the ways to connect with CoppeliaSim. It allows communication between CoppeliaSim and your Python program. As an example, a Python remote API client is shown as:

```
import sim
import time

if __name__ == '__main__':

    # Get clientID to communicate with CoppeliaSim through Python Remote API.
    clientID = sim.simxStart("127.0.0.1", 19997, True, True, 5000, 5)
    # Start the simulation.
    sim.simxStartSimulation(clientID, sim.simx_opmode_blocking)
    # Get the handle of DR20's left wheel.
    handle_left_wheel = sim.simxGetObjectHandle(clientID, "dr20_leftWheelJoint_", sim.
                                                simx_opmode_blocking)
    # Send joint target velocity of DR20's left wheel through the handle.
    sim.simxSetJointTargetVelocity(clientID, handle_left_wheel[1], 1, sim.
                                    simx_opmode_streaming)
    time.sleep(2)
    # Stop the simulation.
    sim.simxStopSimulation(clientID, sim.simx_opmode_blocking)
```

The above description may help you understand the usage of Remote API to communicate with CoppeliaSim. However, learning Remote API of CoppeliaSim is not indispensable for the homework, because a controller of the robot has been designed for you, which is introduced in Section 3. If you are interested in more details about Remote API of CoppeliaSim, please refer to <https://www.coppeliarobotics.com> and <https://www.coppeliarobotics.com/helpFiles/index.html>.

## 3 DR20API

### 3.1 Introduction to APIs

DR20API provides a quick-start interface to obtain the information of the robot and sensor, move the robot, and update the world map. The introduction of APIs is as follow.

```
def __init__(self, port = 19997):
    """
    Initialize the controller of DR20 robot, and connect and start the simulation in
    CoppeliaSim.

    Arguments:
    port -- The port used to connect to coppeliaSim, default 19997.
    """
```

```
def connect_simulation(self, port):
    """
    Connect and start simulation.

    Arguments:
    port -- The port used to connect to CoppeliaSim, default 19997.

    Return:
    clientID -- Client ID to communicate with CoppeliaSim.
    """
```

```
def stop_simulation(self):
    """
    Stop the simulation.
    """
```

```

def update_map(self):
    """
    Update the map based on the current information of laser scanner. The obstacles are
    inflated to avoid collision.

    Return:
    current_map -- A 120*120 array indicating current map, where 0 indicating traversable
    and 1 indicating obstacles.
    """

```

```

def move_robot(self, path):
    """
    Given planned path of the robot,
    control the robot track a part of path, with a maximum of 3 meters from the start
    position.

    Arguments:
    path -- A N*2 array indicating the planned path.
    """

```

```

def get_robot_pos(self):
    """
    Get current position of the robot.

    Return:
    robot_pos -- A 2D vector indicating the coordinate of robot's current position in the
    grid map.
    """

```

```

def get_robot_ori(self):
    """
    Get current orientation of the robot.

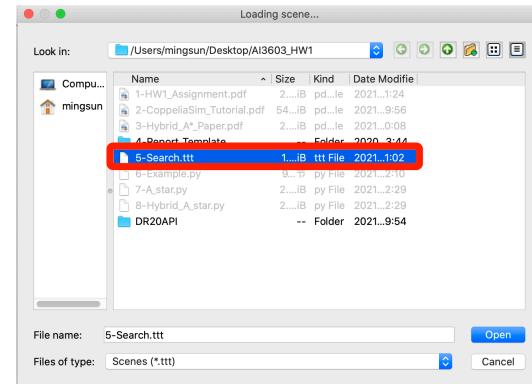
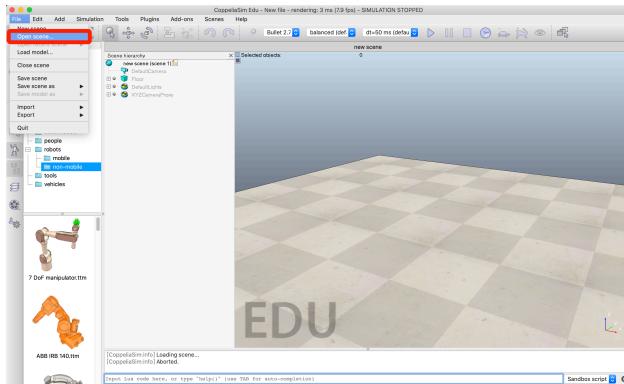
    Return:
    robot_ori -- A float number indicating current orientation of the robot in radian.
    """

```

## 3.2 DR20API Example

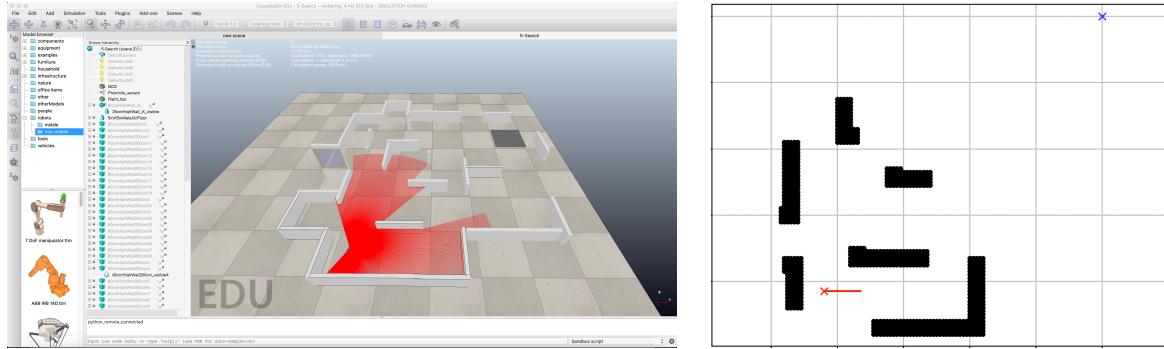
This subsection describes the usage of CoppeliaSim and DR20API through the example code “6-Example.py”.

- Open CoppeliaSim.
- Open the scene saved in “5-Search.ttt”.



- Open a terminal and change the directory to the folder “AI3603\_HW1”: **cd AI3603\_HW1**.

- Test the DR20API: **python 6-Example.py**. In this example, A test path is defined, and the map, path, goal, and robot's position are visualized. The robot will move along the path from start position to target position and then update the world map based on the information of laser scanner.



## 4 Trouble-Shooting

### 4.1 Import DR20API Failed

If you are failed to import DR20API, check your running path, especially when you are using an IDE rather than a terminal. If you are sure that the running path is correct, try to find the following two files in your installation path of CoppeliaSim and replace them in the folder “DR20API”.

- *simConst.py*
- *remoteApi.dll* for Windows, *remoteApi.so* for Linux, and *remoteApi.dylib* for MacOS.

### 4.2 Connection Failed

If you are failed to connect CoppeliaSim, it is probably because the port is not open. To fix it, you can try to specify the port through using the following command to open CoppeliaSim

- For Windows: **coppeliaSim.exe -gREMOTEAPISERVERSERVICE\_19997**
- For Linux: **./coppeliaSim.sh -gREMOTEAPISERVERSERVICE\_19997**
- For MacOS: **./coppeliaSim.app/Contents/MacOS/coppeliaSim -gREMOTEAPISERVERSERVICE\_19997**

### 4.3 Slice Indices Error

For the error “`TypeError: slice indices must be integers or None or have an __index__ method`”, please update your Numpy to 1.19.5.

### 4.4 Others

Try to seek help from classmates and course staff on Canvas:

[https://oc.sjtu.edu.cn/courses/35320/discussion\\_topics/74830](https://oc.sjtu.edu.cn/courses/35320/discussion_topics/74830)