Set up record for this project:

1. Set up the ROS kinetic, this is based on the ROS-kinetic-desktop-full. using link: http://wiki.ros.org/kinetic/Installation/Ubuntu

2. Then create a ROS workspace: http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment
   For me, I did not use the official recommended command `catkin_make`. I replaced it by `catkin build`. For using this, install python-catkin-tools first.
   ```
   kjaebye@kjaebye-XPS-15-7590:~/ws$ sudo apt-get install python-catkin-tools
   ```
   Then, do source.
   ```
   source devel/setup.bash
   ```
   make sure ROS_PACKAGE_PATH environment variable includes the directory you're in.
   ```
   $ echo $ROS_PACKAGE_PATH
   /home/youruser/catkin_ws/src:/opt/ros/kinetic/share
   ```
   if the workspace directory is not in included, then add it manually:
   1) open a terminal and tap
      ```
      kjaebye@kjaebye-XPS-15-7590:~/ws$ sudo gedit ~/.bashrc
      ```
   2) add at the end
      ```
      export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:~/ws/src
      ```
   3) open a new terminal and check env path
      ```
      $ echo $ROS_PACKAGE_PATH
      ```

3. Build the project in workspace.
   Firstly install some dependencies by:
   ```
   $ sudo apt-get install ros-kinetic-geographic-msgs
   $ sudo apt-get install ros-kinetic-octomap-msgs
   $ sudo apt-get install ros-kinetic-mavlink
   $ sudo apt-get install autoconf
   $ sudo apt-get install ros-kinetic-tf2-sensor-msgs
   $ sudo apt-get install libgeographic-dev
   $ sudo apt-get install ros-kinetic-control-toolbox
   ```

check ubuntu has Eigen3 or not, if there are eigen3 directories which means eigen3 has been installed but cannot detected, because the Eigen lib is installed to /usr/include/eigen3/Eigen by default. Thus we need to use command to map to /usr/include.

```
$ sudo updatedb
$ locate eigen3
```

```
$ sudo ln -s /usr/include/eigen3/Eigen /usr/include/Eigen
```

if Eigen lib has not been installed, then install it by:

```
$ sudo apt-get install libeigen3-dev
```

After installing all the required packages, download packages from github and copy to src directory, and build it.

```
$ cd /ws
$ catkin build
```

4. Set up the CoppeliaSim
   Download the CoppeliaSim from https://coppeliarobotics.com/
   Then follow the CoppeliaSim User Manual
   https://www.coppeliarobotics.com/helpFiles/index.html, and see the ROS tutorial for connecting simulator with ROS.
   Just follow the tutorial and connection can be established successfully.
   Before build this package, add a specified message `mav_msgs/CommandMotorSpeed`, edit files located in `SimExtROSInterface/meta/`. Do not forget also edit `CMakeLists.txt` and `package.xml` to add `mav_msgs`
   Then add scenes, models and add-on script into the CoppeliaSim.