Understanding the e-Drone Model

In this tutorial you will learn how to use the **e-Drone model** for Task 1.

Steps to understand the drone model:

NOTE: This task requires eyantra_drone package, make sure you have eyantra_drone package in the src folder of catkin workspace**, if <u>not</u> present clone it by typing, and if <u>already present</u>, skip to step 1**

```
>>      cd ~/catkin_ws/src

>>      git clone https://github.com/simmubhangu/eyantra_drone.git
```

1. Type the following commands in a terminal

```
>>      cd ~/catkin_ws/src

>>      git clone https://github.com/rishikeshrmadan/edrone_rotors.git

>>      cd ..

>>      catkin_make
```

   NOTE: This step will take a sometime to be execute. Chances are that your system might look suspended. But make sure to not interrupt the terminal after this step and wait patiently

```
>>      source ~/.bashrc
```

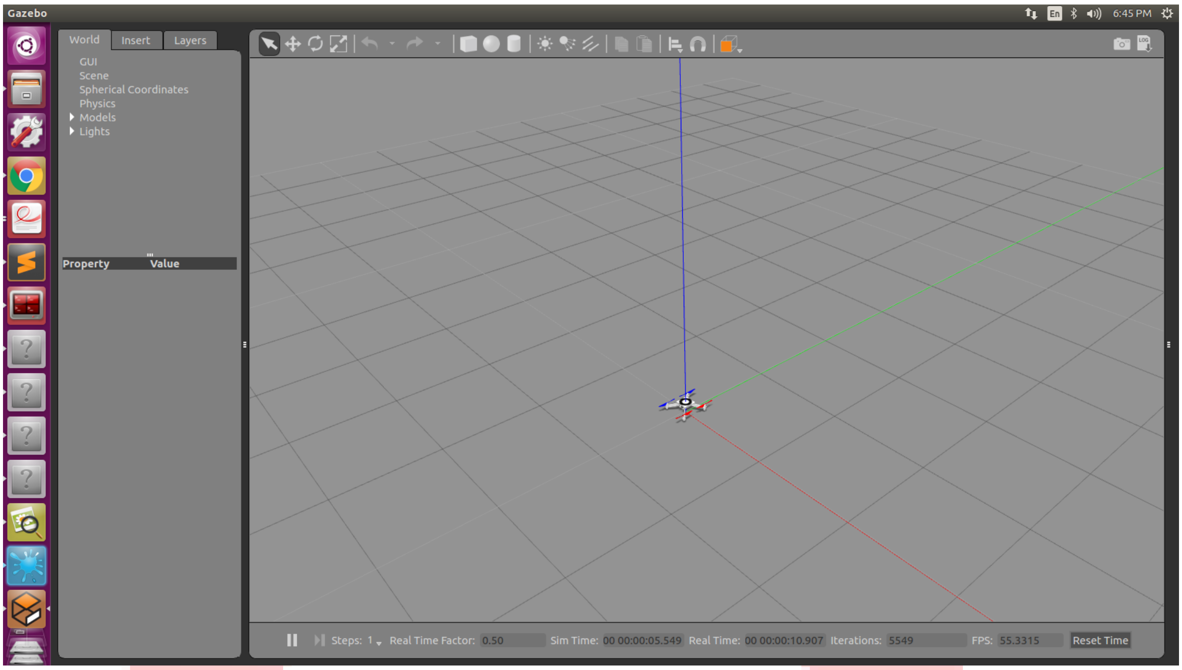
2. Run roscore by typing the following command in your terminal:

```
>>      roscore
```

3. Now launch the following command to see the gazebo world. Make sure there are no multiple gazebo windows opens.

```
>>      roslaunch rotors_gazebo edrone_with_edrone_msg.launch
```

   NOTE: To kill it use CTRL+C (Kill command) rather than CTRL+Z (hibernating command), it will take some time but wait patiently.



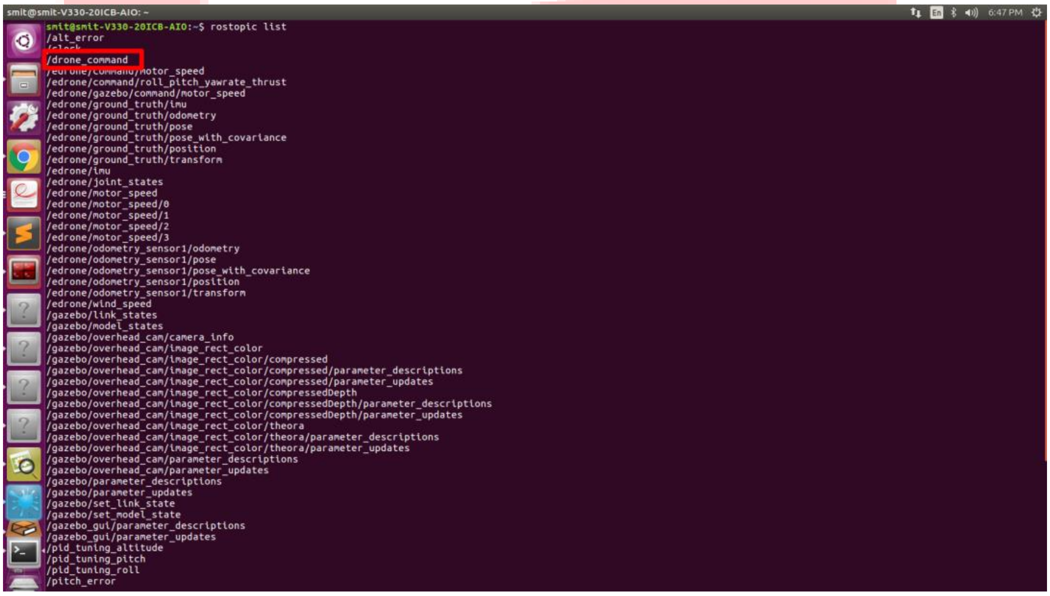
Steps: 1 Real Time Factor: 0.50 Sim Time: 00 00:00:05.549 Real Time: 00 00:00:10.907 Iterations: 5549 FPS: 55.3315 Reset Time

Figure 1: World

4. Check all the topics published by gazebo. Run the following command in the terminal to check all the topics published by Gazebo

   ```
   >>      rostopic list
   ```

   You should find the topic **“/drone_command”**



```
smit@smit-V330-20ICB-AIO: ~
smit@smit-V330-20ICB-AIO:~$ rostopic list
/alt_error
/clock
/drone_command
/edrone/command/motor_speed
/edrone/command/roll_pitch_yawrate_thrust
/edrone/gazebo/command/motor_speed
/edrone/ground_truth/imu
/edrone/ground_truth/odometry
/edrone/ground_truth/pose
/edrone/ground_truth/pose_with_covariance
/edrone/ground_truth/position
/edrone/ground_truth/transform
/edrone/imu
/edrone/joint_states
/edrone/motor_speed
/edrone/motor_speed/0
/edrone/motor_speed/1
/edrone/motor_speed/2
/edrone/motor_speed/3
/edrone/odometry_sensor1/odometry
/edrone/odometry_sensor1/pose
/edrone/odometry_sensor1/pose_with_covariance
/edrone/odometry_sensor1/position
/edrone/odometry_sensor1/transform
/edrone/wind_speed
/gazebo/link_states
/gazebo/model_states
/gazebo/overhead_cam/camera_info
/gazebo/overhead_cam/image_rect_color
/gazebo/overhead_cam/image_rect_color/compressed
/gazebo/overhead_cam/image_rect_color/compressed/parameter_descriptions
/gazebo/overhead_cam/image_rect_color/compressed/parameter_updates
/gazebo/overhead_cam/image_rect_color/compressedDepth
/gazebo/overhead_cam/image_rect_color/compressedDepth/parameter_descriptions
/gazebo/overhead_cam/image_rect_color/compressedDepth/parameter_updates
/gazebo/overhead_cam/image_rect_color/theora
/gazebo/overhead_cam/image_rect_color/theora/parameter_descriptions
/gazebo/overhead_cam/image_rect_color/theora/parameter_updates
/gazebo/overhead_cam/parameter_descriptions
/gazebo/overhead_cam/parameter_updates
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/set_link_state
/gazebo/set_model_state
/gazebo_gui/parameter_descriptions
/gazebo_gui/parameter_updates
/pid_tuning_altitude
/pid_tuning_pitch
/pid_tuning_roll
/pitch_error
```

Figure 2: rostopic list output

“/drone_command” is a topic subscribed by the **e-Drone model.** It commands the drone’s motion in terms of **roll, pitch, yaw and throttle.**

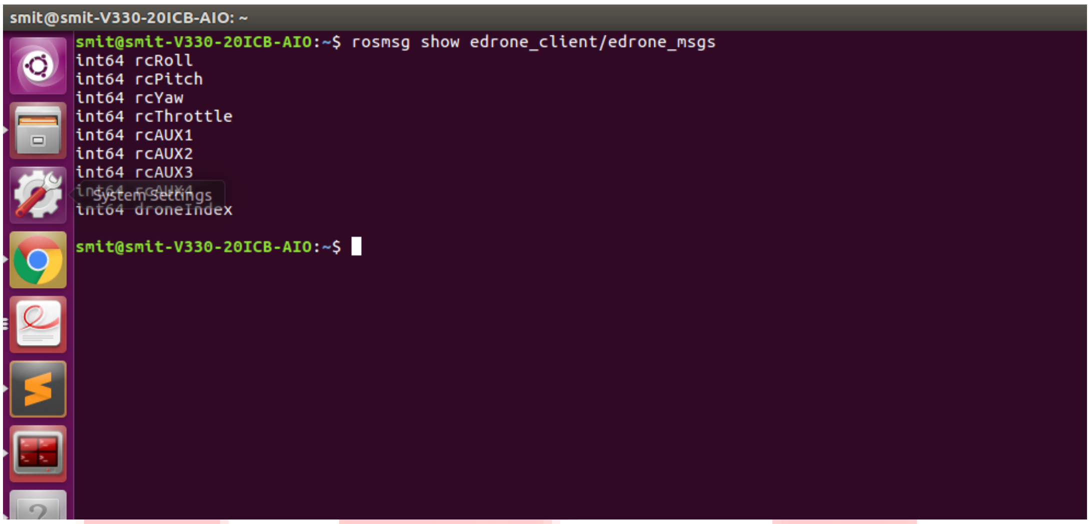
5. Check the type of messages accepted by the **“/drone_command”** topic. Run the following command in the terminal to check:

```
>>      rostopic info /drone_command
```

Your output will display the topic type as **“edrone_client/edrone_msgs”**.

Check the structure of the message by typing the following command in the terminal:

```
>>      rosmsg show edrone_client/edrone_msgs
```



```
smit@smit-V330-20ICB-AIO: ~
smit@smit-V330-20ICB-AIO:~$ rosmsg show edrone_client/edrone_msgs
int64 rcRoll
int64 rcPitch
int64 rcYaw
int64 rcThrottle
int64 rcAUX1
int64 rcAUX2
int64 rcAUX3
int64 rcAUX4
int64 droneIndex

smit@smit-V330-20ICB-AIO:~$ 
```

Figure 3: drone_command message structure

The values for *rcRoll, rcPitch, rcYaw* and *rcThrottle* range from 1000 to 2000.

Arming the Drone:

An armed drone means the drone is ready to take commands from a user or software to fly.

The condition to arm the drone is rcThrottle = 1500 and rcAUX4 = 1500. To test arming the drone model, publish the following message to the topic **“/drone_command”** by typing the command (do not copy-paste, see video tutorial for better alternative):

```
>>    rostopic pub /drone_command edrone_client/edrone_msgs "{rcRoll:
      1500,  rcPitch:  1500,  rcYaw:  1500,  rcThrottle:  1500,  rcAUX1:  0,
      rcAUX2:  0,  rcAUX3:  0,  rcAUX4:  1500}"
```

The propellers should start rotating.

NOTE: After publishing any command to the drone, before entering new command press `Ctrl+C` **and then enter next command.**

Flight (Take-Off):

The condition for the drone to take-off is rcThrottle ≥ 1500, after arming. To test the drone’s take-off, publish the following message to increase the throttle:

```
>>  rostopic pub /drone_command edrone_client/edrone_msgs "{rcRoll:
    1500, rcPitch: 1500, rcYaw: 1500, rcThrottle: 1510, rcAUX1:  0,
    rcAUX2: 0, rcAUX3: 0, rcAUX4: 1500}"
```

The drone should now steadily rise until a new command is given, as shown in figure 4.

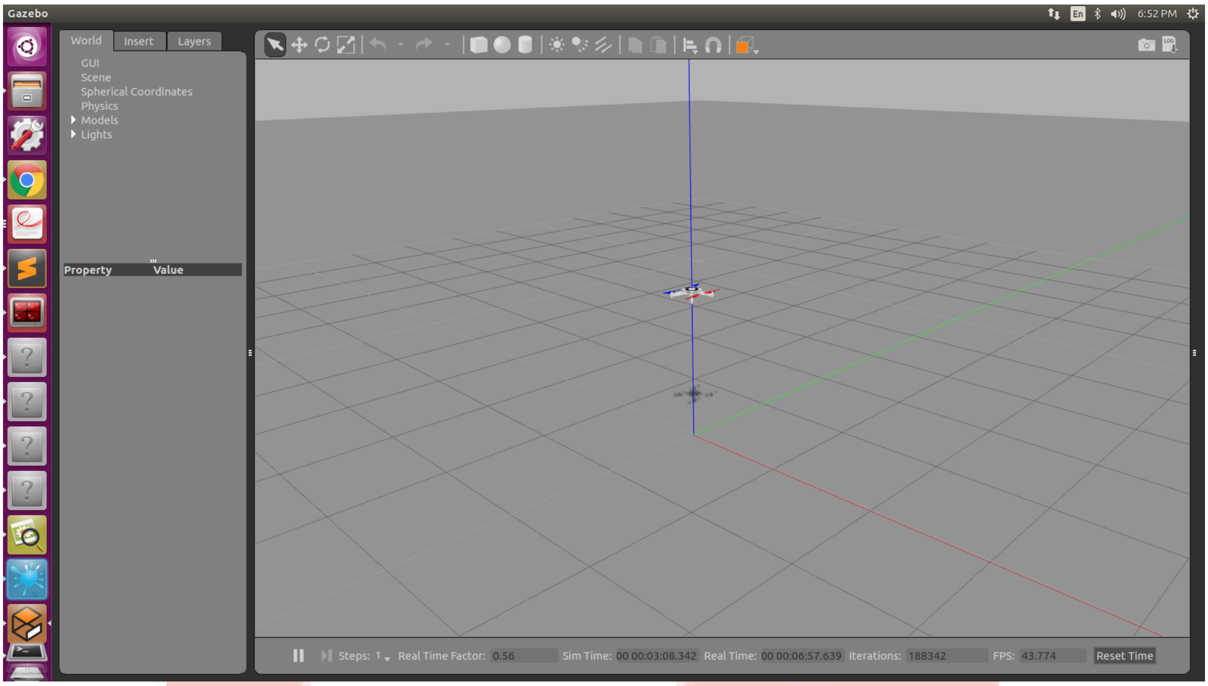


Figure 4: Flight

NOTE: If the drone goes up and gets stuck to the camera, you can bring it down by resetting the world, to do that go to Edit and click on Reset Pose or you can also press `Ctrl+Shift+R`

Disarming the Drone:

A disarmed drone means the drone is in stop state.

The condition to disarm the drone is rcThrottle = 1000. To test disarming the drone model, publish the following message to the topic **“/drone_command”** by typing the command:

```
>>  rostopic pub /drone_command edrone_client/PlutoMsg "{rcRoll:
    1500, rcPitch: 1500, rcYaw: 1500, rcThrottle: 1000, rcAUX1: 0,
    rcAUX2: 0, rcAUX3: 0, rcAUX4: 0}"
```

The drone should now be disarmed. And come back to the ground

Heading if the Drone:

It is important to understand the heading direction of the drone *with respect to world*. Refer to Figure 5 to check the heading of drone.

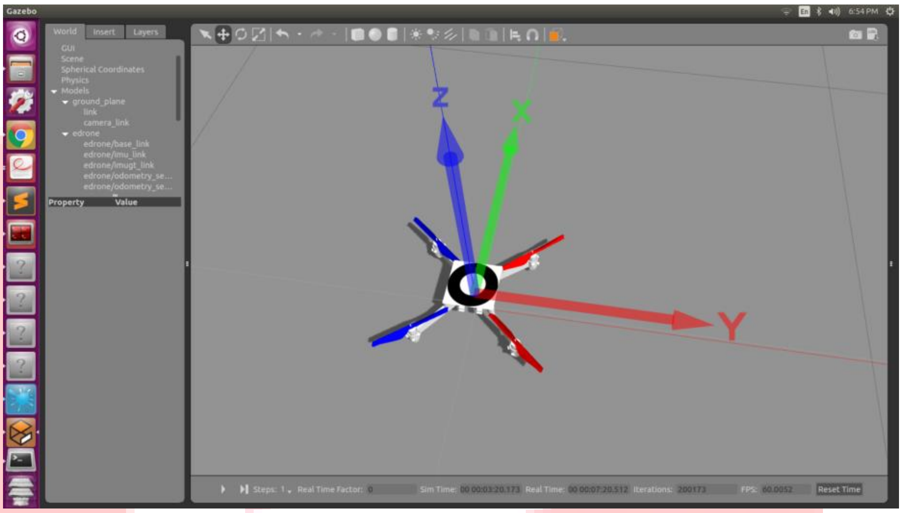


Figure 5: Heading of the Drone

- Red Arrow: Positive Y-axis (Roll)
- Green Arrow: Positive X-Axis (Pitch)
- Blue Arrow: Positive Z-axis (Throttle)