e-Yantra Robotics Competition (eYRC 2019-20)

Task 1.1 – Survey & Rescue

Position Hold of the drone within Gazebo

Problem Statement

- The drone should hold its position in the given simulation scene at the given point **[2, 2, 20]** using the PID control algorithm.

Procedure

1. Before going through this procedure, make sure you have gone through the Tutorials thoroughly as there some packages that are to be installed in the tutorials section.
2. Clone the packages in your **'src'** folder using the following:

```
>>    cd ~/catkin_ws/src

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

>>    git clone https://github.com/simmubhangu/pid_tune  _

>>    sudo apt-get install ros-kinetic-plotjuggler

>>    catkin_make

>>    source devel/setup.bash

>>    source ~/.bashrc
```

NOTE: Visit and implement tutorials section before proceeding from this point.

3. Then you can launch the Gazebo world containing the drone model and the WhyCon package at the same time by running the launch file *launch_gazebo_and_whycon.launch* as follows (This script contains already mapped whycon node, which was one of the solution of task 0, do not make any changes to this file):

```
>>    roslaunch survey_and_rescue launch_gazebo_and_whycon.launch
```

NOTE: This file launches models from edrone_rotors packages which is quite heavy in terms of computing, it will take time to launch as well. And it is recommended to close it, using CTRL+C (kill command) rather than CTRL+Z (hibernate command) and wait patiently.

4. Write a PID script **after** going through the tutorials provided to you. A boiler plate code titled *position_hold_gazebo.py* is provided to you, you can find the script in the scripts folder of survey & rescue package in src folder. You need to complete the python script and then launch your code and the drone must hold position at the point **[2, 2, 20]** as per the WhyCon coordinate system. You can launch your code by typing (before that make your python script executable)
5. To make script executable ,

```
>>    roscd survey_and_rescue/scripts

>>    chmod +x position_hold_gazebo.py
```

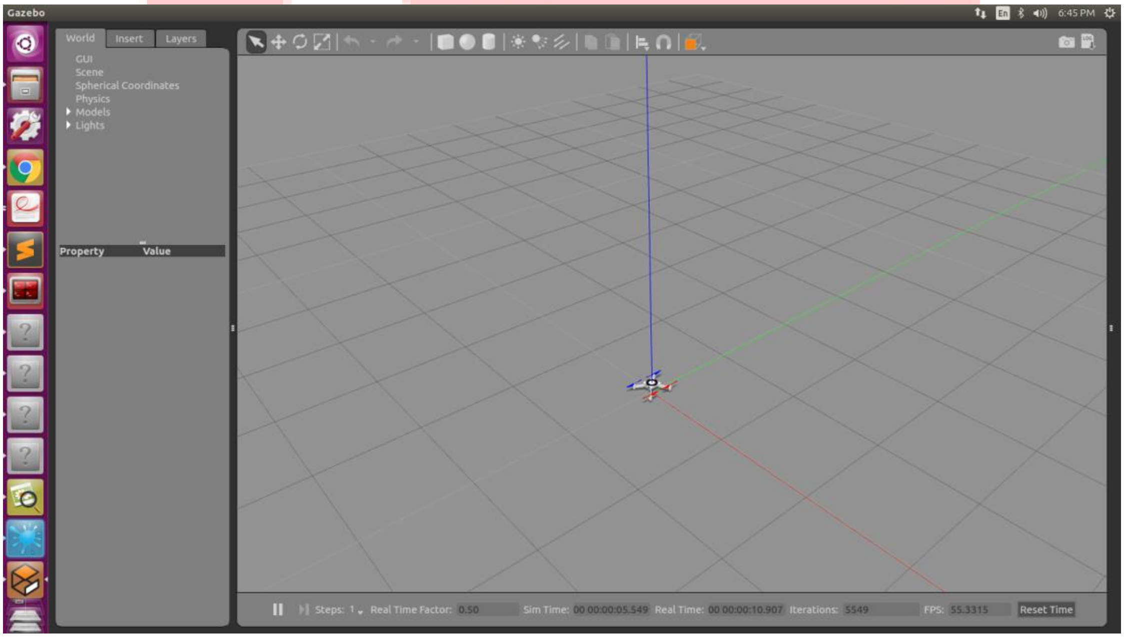
6. To run your python script

```
>>    rosrun survey_and_rescue position_hold_gazebo.py
                                 OR
Visit the location of your script , to do that enter
>>    roscd survey_and_rescue/scripts
>>    python position_hold_gazebo.py
```

From the tutorials, you have learned

- Basics of PID
- eDrone and the edrone_msgs type

World Description



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

Figure 1: World

Following are the various objects in the world:

edrone: There is one drone model attached with a whycon marker on top.
overhead_camera: This gives an image with a resolution 1280x1280, its field of view (FOV) is in the shape of a
pyramid (as shown in figure 2), with camera facing downwards.

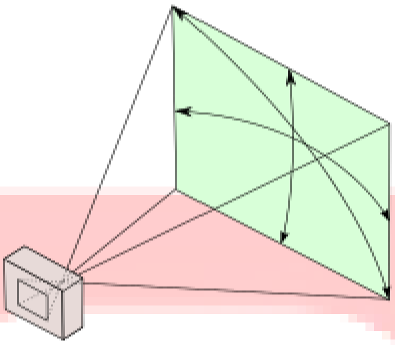


Figure 2: Camera's FOV

<u>Points to remember</u>

- **Please do not change any of the templates**
- **Create a PID script or edit the given example script by adding your PID code. Finally, run it to control the drone and make it hover at point [2, 2, 20]**
- **Most of the code in the position_hold_gazebo.py is already written. Read the comments within the script and fill out the empty functions. Study the PID tutorial in order to understand how to implement PID.**
- **Do NOT change the inner and outer diameters of whycon markers, keep it default else the grading will affect**

<u>Submission Instructions</u>

Follow the instructions below to submit your Task.

1. **Bag File:**

   a. Next when you want to record the bag file for submission run the launch file record_rosbag.launch . This file will launch both your position_hold_gazebo.py and the rosbag recording.

```
>> roslaunch survey_and_rescue record_rosbag.launch
```

This will record a bag file (titled *markers.bag* by default) for a duration of 20 seconds of <u>Simulation Time</u> as shown in Figure 3. Note that this time may be lagging with real-time due to load on the simulator. Your aim is to stabilize the drone at the given setpoint within that time. This bag will be stored in the survey_and_rescue package folder in the src folder of your catkin_workspace.

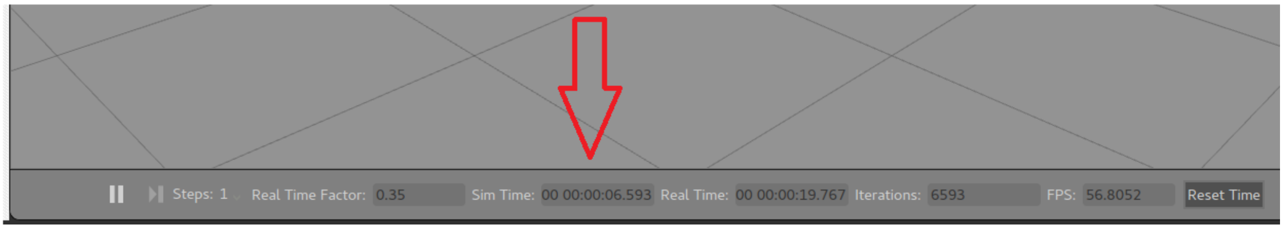


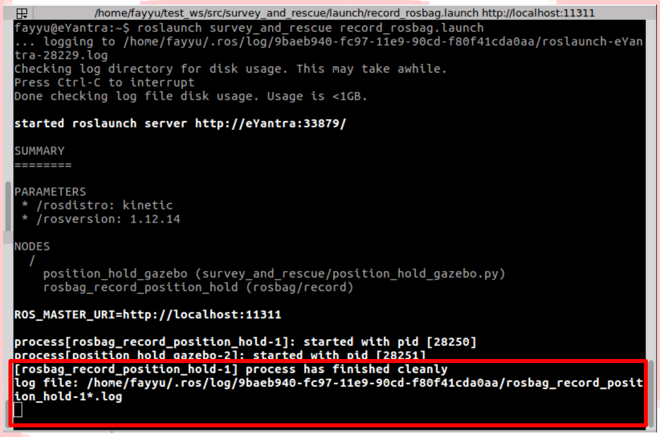
Figure 3: Simulation Time

NOTE: Repeated running of this launch file will overwrite the markers.bag file, please make sure you have a backup if you so wish. Alternatively, you can use the argument rec_name as follows to have a custom name.

```
>> roslaunch survey_and_rescue record_rosbag.launch rec_name:=hello.bag
```

This will create a bag file of the name *hello.bag* instead of the default *markers.bag*

b. The rosbag will record for 20 seconds of Simulation Time and stop recording, the output will look like Figure 4 when this has happened. But you will need to close the terminal running the launch file manually afterwards, since the PID script will still be running.


/home/fayyu/test_ws/src/survey_and_rescue/launch/record_rosbag.launch http://localhost:11311

```
fayyu@eYantra:~$ roslaunch survey_and_rescue record_rosbag.launch
... logging to /home/fayyu/.ros/log/9baeb940-fc97-11e9-90cd-f80f41cda0aa/roslaunch-eYan
tra-28229.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://eYantra:33879/

SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.14

NODES
  /
    position_hold_gazebo (survey_and_rescue/position_hold_gazebo.py)
    rosbag_record_position_hold (rosbag/record)

ROS_MASTER_URI=http://localhost:11311

process[rosbag_record_position_hold-1]: started with pid [28250]
process[position_hold_gazebo-2]: started with pid [28251]
[rosbag_record_position_hold-1] process has finished cleanly
log file: /home/fayyu/.ros/log/9baeb940-fc97-11e9-90cd-f80f41cda0aa/rosbag_record_posit
ion_hold-1*.log
```

Figure 4: rosbag process has finished cleanly

c. Next step is to compress the .bag file that is created before you can upload it. Run:

```
>> rosbag compress -j bag_file_name.bag
```

d. Rename the compressed bag file as SR_**<team_id>_1_1.bag**
Hence, say your team ID is 105, rename it as SR_105_1_1.bag

2. **Python Code:**
   a. You must submit your PID script that you developed.
   b. Rename the python script as **SR_<team_id>_1_1.py**

Store the files mentioned above into .zip file and rename the zip as <team_id>.

3. Video:

i. Upon verifying that your task is complete, record a maximum 2-minute video using a screen recorder like simplescreen recorder or kazam.

ii. The video must be as follows:

1. Team Slide –All members' details in a slide , included in this folder.
2. Any One member of the team, running the scripts and launch files in terminal. Output of gazebo and whycon image_out window on the PC screen captured clearly.

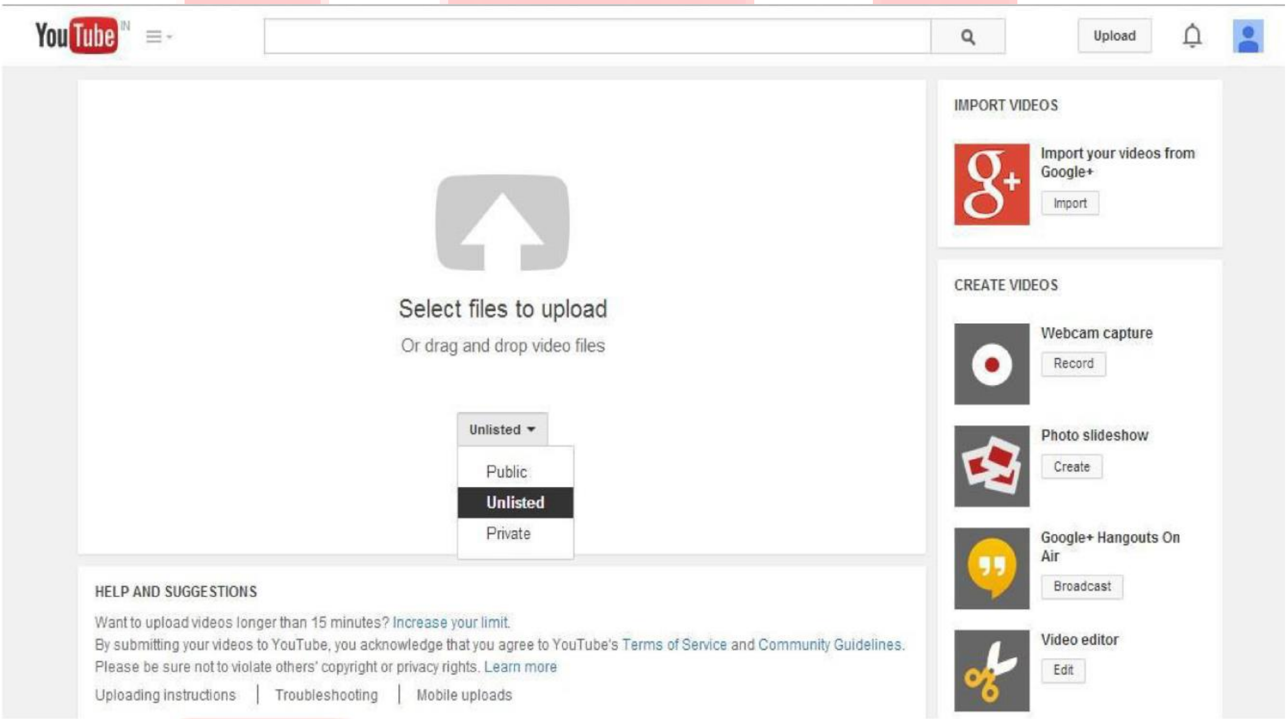
iii. Please record the video as shown here

The video should not be edited in any manner. Teams uploading an edited video will be disqualified from the competition. e-Yantra reserves the rights to disqualify any team if any foul play is suspected.

Uploading video/s on YouTube:

● Upload a one-shot continuous video with the title eYRC#SR#Task1_1#<TeamID>
(For example: If your team ID is 1234 then, save it as eYRC#SR#Task1_1#1234)

● Please note that while uploading the video on YouTube select the privacy setting option as Unlisted as shown in Figure 2. You need to upload the video as instructed on the portal.



Want to upload videos longer than 15 minutes? Increase your limit.
By submitting your videos to YouTube, you acknowledge that you agree to YouTube's Terms of Service and Community Guidelines.
Please be sure not to violate others' copyright or privacy rights. Learn more

Figure: Uploading unlisted video on YouTube

NOTE: You must upload all of the following: (i) bag file and (ii) Python code. Please place all these files inside a .zip file before uploading. You must also upload the video to YouTube with the instructions above. You will have to submit the video link on the portal.

Please follow the naming convention strictly as specified in each step. Failure to do so may lead to repercussions. The deadline for Task_1.1 is 19th November 2019

Your final .zip output must be of the following structure:

<team_id>_1_1.zip

- **SR_<team_id>_1_1.bag**
- **SR_<team_id>_1_1.py**

Instructions for uploading the folder will be provided on portal

Good Luck!!!