Task 3.1 - Survey & Rescue

1 Problem Statement

- The Drone should start at [0, 0, z] w.r.t. to WhyCon frame, where z is your ground value.
- The Drone should then take-off hold its position in the real world at the given point [2.0, 2.0, 20.0] as per the WhyCon frame of reference using the PID control algorithm.
- Drone should reach this position as **quickly and precisely** as possible.

2 Procedure

- Open a terminal and change directory to the survey_and_rescue package by typing the following command

  ```
  cd ~/catkin_ws/src/survey_and_rescue
  ```

  If you have not already committed your local changes yet follow this link to do so. In summary you have to run the following commands:

  1. To know the files you have changed and added

     ```
     git status
     ```

  2. To add all these files to be tracked.

     ```
     git add .
     ```

  3. To commit (log) these changes with an optional message.

     ```
     git commit -m "My local changes in Stage 1"
     ```

  4. Update the survey and rescue package by typing the following line. This will both update the package and resolve the merge conflicts if they exist.

     ```
     git pull origin master && git checkout HEAD launch
     ```

- After updating the package, the *usb_cam_SR.launch* file will have the required changes in it to launch the WhyCon node and the image_proc node (which implements the undistortion based on your calibration). You need to use this launch file for detecting the WhyCon marker in real world.
- Paste the provided WhyCon marker on the canopy of the Drone firmly. Refer to this video for tips and suggestions for attaching the WhyCon marker to the quad.

- Refer to this video to patch a bug in the WhyCon package and for additional tips in implementation.

- You may continue using the *position_hold.py*, to hold position of the drone at [2.0, 2.0, 20.0] of the WhyCon frame of reference. Before running the PID script, make sure you connect to the Drone's WiFi and run the edroneclient node, to do that type the following command

```
rosrun edrone_client edroneclient
```

- The maximum allowed errors are $\pm 0.5, \pm 0.5 \& \pm 1.0$ in x, y & z axis respectively.

- Try to achieve minimal error.

- On execution of the node, your script should arm the drone, take off from and hold at the positions specified in the Problem Statement.

3 Expected Output

- Refer to this video to get an idea regarding position hold with quad.

4 Points to Remember

- Please do not change any of the templates.

- Do NOT change the inner and outer diameters of whycon markers, keep it default else the grading will affect.

- The real Drone provided is not exactly same in terms of response w.r.t to the Drone in Gazebo.

- Thus, even when just throttle is being tuned, it should ideally just move in z axis; that is not the case. The drone tends to move in both x & y axis also. One way to tackle this problem is to tie a 20-30cm thread at the bottom of the Drone so as to:
  - Keep it within the WhyCon frame while tuning.
  - Protect it from crashing.

5 Submission Instructions

Follow the instructions below to submit your Task.

5.1 Bag File:

- Next, after tuning, when you want to record the bag file for submission, run the launch file name record_rosbag. This file will launch both your position_hold.py and the rosbag recording at the same time.

  ```
  roslaunch survey_and_rescue record_rosbag.launch
  ```

  WARNING: For recording/creating the bag file, use ONLY record_rosbag launch file. Recording (using record_rosbag) should cover from the Arming of the drone till stabilising at the setpoint within mentioned time limit.

- This will record a bag file (titled markers.bag by default) for a duration of 20 seconds of real time. 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.

- Before submitting make sure to verify your bag file, by Check the number of messages, which should be in between 590-610, which can be done by following line

  ```
  rosbag info markers.bag
  ```

- Further verification can be done by *plotjuggler* package, to see errors while tuning or for performance verification, after the bag file has been created.
  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

5.2 Python Script:

- You must submit your PID script that you developed.

- Rename the python script as SR_<team_id>_3_1.py

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

5.3 Video:

- Upon verifying that your task is complete, record a maximum 2-minute video using a screen recorder like simplescreen recorder or kazam.
- The video must be as follows:
  1. Screen Recording starts
  2. Team Slide –All member's details in a slide , included in this folder.
  3. Any One member of the team, running the necessary scripts or launch files in terminal.
  4. After the script begins, make fullscreen the whycon image_out window.
  5. Assume a proper video, when the drone reaches the setpoint as quickly and precisely as possible and within 20 sec.
- Please refer this video a sample recorded video.

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#Task3_1#<TeamID>
  (For example: If your team ID is 1234 then, save it as eYRC#SR#Task3_1#1234)
- Please note that while uploading the video on YouTube select the privacy setting option as Unlisted as shown in Figure 1. You need to upload the video as instructed on the portal.

NOTE: You must upload all of the following files:

- bag file
- 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.

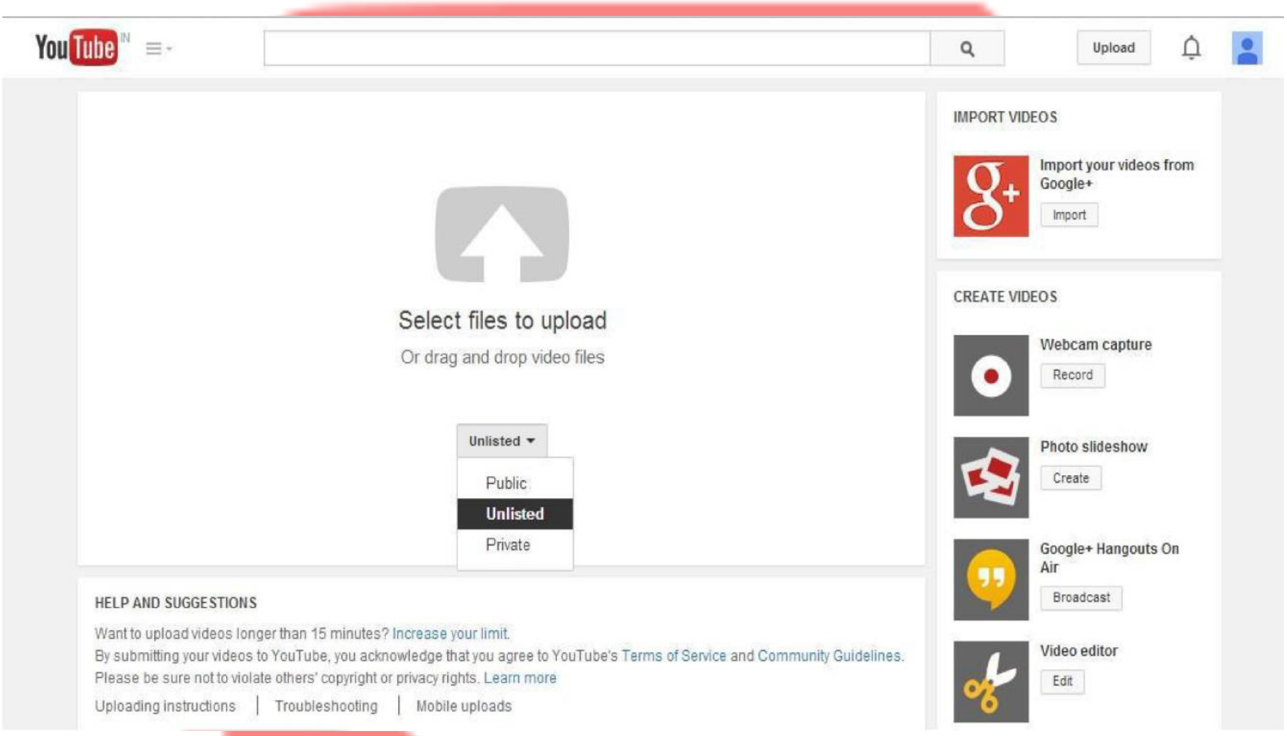
Please follow the naming convention strictly as specified in each step. Failure to do so may lead to repercussions. The deadline for Task_3.1 is 8th January, 2020.

Your final .zip output must be of the following structure:
<team_id>_3_1.zip

- SR_<team_id>_3_1.bag
- SR_<team_id>_3_1.py

Instructions for uploading the folder will be provided on portal.

Good Luck!!!



Select files to upload

Or drag and drop video files

HELP AND SUGGESTIONS

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

Uploading instructions | Troubleshooting | Mobile uploads

IMPORT VIDEOS

Import your videos from Google+

CREATE VIDEOS

Figure 1: Uploading unlisted video on YouTube