Task 4A : Position control of drone

blogpost-style

Smit 1 February 6, 2023, 7:42am

Task 4A : Position control of drone

Aim

The aim of this task is to control the drone in real life. The task is more or less the same as task 1A that was done in simulator. The difference is now it is real drone in real world.

Prerequisites

- It is mandatory that you have completed task 3C (drone assembly) with WhyCon sticker as a fully functional drone is required to perform this task.
- **IMPORTANT:** Tie a string on to the drone (eg. white string used in making garlands) and use this string to control the drone while tuning the PID controller of drone. If you do not do this, you will crash the drone and break/damage parts. e-Yantra will not be providing replacements for damaged parts and parts are not available in the market, which means you will not be able to continue further in the competition.

Installations

- Update the sentinel_drone ros package from github

```
cd ~/catkin_ws/src/sentinel_drone
git pull origin main
```

- Open ~/.bashrc of your **laptop/pc** using any editor. Add these lines in the last line as your PC/laptop is a ROS master.

```
export ROS_MASTER_URI=http://"<ip_address_of_laptop/pc>":11311
export ROS_IP="<ip_address_of_laptop/pc>"
```

- Open ~/.bashrc of **Bananna Pi** using ssh and vim editor. Find the ip address of your laptop/computer. If you are using laptop/pc 's WiFi hotspot, then the usual ip is `10.42.0.1` . Else, you can check the laptop/pc ip address by typing `ifconfig` command.
  Add these lines in the last line if not present in the .bashrc file.

```
export ROS_MASTER_URI=http://"<ip_address_of_laptop/pc>":11311
export ROS_IP="<ip_address_of_bananapi>"
```

- Once this is set, connect the Banana Pi and laptop/pc on the same network and run `roscore` on **laptop/pc**. SSH into Banana Pi and check `rostopic list` without running **roscore** on Banana Pi. If you get the output of `rostopic list` then there is successful connection between the Banana

Pi and laptop/pc using ROS. This means the ROS master is running on laptop/pc and the Banana Pi is able to communicate to the laptop/pc using ROS.

- Run command on Banana Pi

```
rosrun sentinel_drone_driver driver.py
```

This will create a list of rosservices and rostopics which will be used to send commands to drone via ROS from any remote system

> Note: You will need to run this command on Banana Pi every time you restart the drone and want to fly the drone using ROS

- Open a new terminal on **laptop/pc** and run `rosservice list`. There should be a service named `/sentnel_drone/cmd/arming`. Call this service to arm the drone.

```
rosservice call /sentinel_drone/cmd/arming value = "true"
```

Great! you have armed the drone remotely using ROS from another system !

Problem Statement

- The drone has to be placed approximately at the center of arena.
- Connect the usb camera used in task 3B to laptop/pc. Launch the usb camera, WhyCon node, plotjuggler, pidtune gui - all together using the launch file

```
roslaunch sentinel_drone task_4a.launch
```

- Complete the boiler plate script `controller.py` provided in the scripts folder of sentinel_drone.
- After running the `task_4a.launch` and `controller.py` and tune the PID values, the drone should go to setpoint [0,0,22] in the WhyCon coordinates.

> Note: DO NOT change the settings of marker dimensions in the launch file, this will change the coordinate units of WhyCon.

- The allowable errors in x,y,z axes are ±0.8 in all the axes. For eg, the drone is said to be at the setpoint if it is at [0.2 -0.32 22.7] and is not said to be at the setpoint if it is at [0.3 0.9 22.5], as the y coordinate is out of the allowable limits.
- After running the `controller.py`, the drone should takeoff and go to the setpoint and stay there for 10 seconds. After 10 seconds in the allowable range, drone should come down and land **slowly**. For landing, you can give setpoint [0 0 base_z] where base_z is the value of z coordinate when you keep the drone on ground (This will be different for everyone).

Recording and submission instructions

Recording instructions

- Once the tuning of PID values is done, hardcode them into the script and use the launch file

```
roslaunch task_4a_submission.launch
```

  This will start the usb camera, WhyCon node and rosbag recording for 60 seconds. This is the maximum time given to complete the task and recording will end after 60 seconds.

- Record the screen of pc/laptop before launching the above launch file. Split the windows on the screen such that half of the screen should have terminals and other half should have the camera feed from the ceiling. Record the video until the drone lands.

Submission instructions

- Rename the `controller.py` to SD_<team_id>*controller.py . Rename the .bag file to SD*<team_id>*controller.bag. Zip both the files and name the zip file as SD*<team_id>task_4a.zip. Submit the .zip file on the portal in place of task 4A.

- Upload the video recorded on YouTube as **unlisted** video with the title `eYRC 2022-23 Sentinel Drone Task 4A`. Submit the link of unlisted video on the portal.

Deadline

Deadline for submitting task 4A is 16th February 2023 23:59

All the best !

2 Likes

Smit Closed 2 February 6, 2023, 7:42am

Smit Unlisted 3 February 7, 2023, 8:09am

Smit Listed 4 February 7, 2023, 8:11am