Running Drone Bug Algorithm using ROS

Bishop Clark

University of South Florida

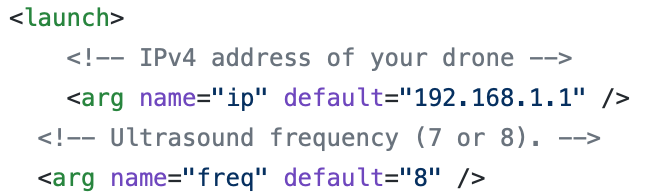
# Introduction

This walkthrough explains how to run the bug algorithm program for the AR Drone 2.0 using the ROS software. It is assumed at this point that the reader has an operable Ubuntu Linux device with ROS installed properly on it. As well as have already cloned the ‘ardrone’ repository from the USF GitHub Bio robotics page.

* Ubuntu Linux 16.04 Melodic (tutorials.ubuntu.com)
* ROS Kinetic version (wiki.ros.org)
* Files (github.com/biorobaw)
* AR Drone 2.0
* If not already in the bash file, add the line “source /opt/ros/kinetic/setup.bash” to enable ROS commands within the terminal; you can open this file by inputting “gedit /.bashrc”

## Initial Configuration

1. AR Drone 2.0
   1. Verify a fully charged lithium ion battery and equip it to the AR Drone
   2. Equip the wing propeller hull
2. Personalize course
   1. Design a course: attach a fixed-size pink colored cardboard paper on to the ending location of your choice, this will be where the drone lands once it has reached the pink paper within a certain distance. Attach fixed-sized neon green colored carboard papers to various objects. These obstacles are what the drone will try to avoid during its path to the goal. Note: these components must be fixed-sized because the drone measures distance by area of the object. As such for a small sized object the drone will think its farther away than it actually is and conversely for a large sized object, the drone will think it is very close when in actuality, it is far away.
   2. The default ending goal pink HSV color used in the majority of tests is lower bound: (90, 90, 200) and upper bound: (180, 255, 255)
   3. The default obstacle neon green HSV color used in the majority of tests is lower bound: (10, 110, 100) and upper bound: (100, 255, 255)
   4. Change these values by preference within the fly.py script
3. Linux machine
   1. Turn machine on and open Network Connection Interface
   2. Connect to the AR Drone 2.0 access point (no security)
   3. Open a terminal and input the command “roscore” this will connect to the ROS Master node (do not close)
   4. Open a second terminal and navigate to the directory in which the ‘drone.launch’ file is held, input command “roslaunch drone.launch”, this file determines what initialized parameters to set when running the program. It returns to the machine various data on the current state of the drone as well, such as battery power, camera usability, etc.



* 1. Note: If “roscore” does not successfully connect, change the IP address in the launch file to “192.168.1.2” instead, if it still does not connect you must change your hostname to this.

## Running Fly Program

1. Verify the above steps are functional
2. On a third terminal, navigate to the directory where ‘fly.py’ is held
   1. Place the drone in a safe location away from objects
   2. To begin flying, run the command “python fly.py” on the command line

Warning: the drone will takeoff approximately two seconds after the command is entered. If for any reason the drone has malfunctioned stop the program using CTRL+C.

1. If the drone has red lights among any rotors, disconnect and reconnect the battery, the launch file will reconnect with the drone and repeat steps starting with Running Fly Program step 1.