USING DRONE TESTBED TO PERFORM AERIAL FLIGHTS

Tips and guidelines

Hugo Cabrita

NOVA School of Science and Technology
23th Jan 2021

Contents

Introduction

The objective of these instructions is to allow and facilitate users to quickly interact with the testbed that has been designed in the Faculty of Sciences and Technology from NOVA University of Lisbon, inside the Department of Electrical Engineering.

You will be able to understand the structure of the testbed as well be capable of interacting with it, giving the opportunity to perform experimental trials of aerial vehicles inside its perimeter while maintaining the security parameters that are crucial in these situations. These instructions are considering you are using a PX4 controller and the Marvelmind motion capture system that has been installed on the testbed.

The instructions are separated in phases. In first place, the programs you must install and the files you must download. Afterwards, some configurations of PX4 and Marvelmind will be shown. Finally, some tips that might be handy before starting an experimental trial.

Note that these instructions are not fully detailed, the ideas of how everything should work is here but this is just a summary of all the procedures.

If you encounter any problems during the process you should take a look at the Master's thesis entitled "Design and control of a rotary wing drone testbed" that provides more detailed explanations on some matters.

Setting up the environment

In first place, you should prepare your environment, for that you will need:

- Linux environment on your machine (Tested with Ubuntu 18.04)
- Robotic Operating System (ROS) and MAVROS must be installed. Script in https://docs.px4.io/master/en/dev_setup/dev_env_linux_ubuntu.html#rosgazebo facilitates this process. This script will install some other dependencies alongside that will also be needed.
- Download and install Marvelmind's software (Dashboard) which can be found from https://marvelmind.com/download/ which will be responsible by configuring Marvelmid's settings.
- Install QGroundControl, this program will simulate a ground control station and can be downloaded from the official website. http://qgroundcontrol.com/downloads/
- Access https://bitbucket.org/bguerreirostudents/hugo-cabrita-code/src/master/src, download "hedge_rcv_bin.cpp" and Marvelmind's submap.
- Clone PX4's firmware from https://github.com/PX4/PX4-Autopilot.

Note that when trying to run the simulations, there might be the need to install some other dependencies (Python3, geographiclib-tools, libgstreamer, …) that are not yet installed on your machine. The missing dependencies should be shown in the console as well as the command to install them.

Preparing PX4's autopilot

Install the autopilot module in the drone and connect it via USB into your computer. Open QGroundControl (QGC), make sure you flash the firmware corresponding your autopilot module and follow the instructions needed to proceed with the necessary sensor calibrations. In QGC you can open the MAVLink Inspector and make sure the data is being read from the autopilot. Also, in QGC, a couple of parameters will have to be changed, allowing the autopilot to use an external position instead of calculating its position based on its sensors. Go to Parameters and change the following:

SYS_MC_EST_GROUP	Set the estimator to ekf2 (Recommended)
EKF2_AID_MASK	Enable *vision position fusion* and disable *inhibit IMU bias estimation.*
EKF2_HGT_MODE	Set to *Vision* (Ensure that the altitude in use is the one from Marvelmind).
EKF2_EV_DELAY	Set to *400ms*, which is roughly the position's delay between Marvelmind and PX4).
COM_RC_IN_MODE	Set to "1" if the objective is to perform an autonomous flight.
CBRK_IO_SAFETY	Set to "22027" if you did not install the GPS module (Recommended).
MAV_0_MODE	Set it to *External Vision*, ensuring the maximum transmission data of MAVLink messages is achieved.

Note that there might be the need to change a couple more parameters depending on the type of flight you'll be performing.

In terms of the communication between PX4 and QGC, it is recommended the usage of the ESP2866 Wi-Fi module to ensure the rate of the transmission of MAVLink messages is the best possible. This module should be connected to one of the telemetry ports of PX4. After configuring this module, it might be needed to create a manual connection on QGC, introducing the IP of the Wi-Fi module (192.168.4.1) and its port (14550). After the configuration of this module is done, ensure the transmission is being done wirelessly by using the MAVLink inspector in QGC.

Preparing Marvelmind's system

Install Marvelmind's beacon (the hedgehog) on top of the drone, making sure it is not covered in any direction. Apart from that, make sure the hedgehog is charged, and the stationary beacons are being charged whilst performing any flight. Some steps to prepare Marvelmind's environment are the following:

- Connect Marvelmind's modem to your computer via USB.
- Open *Dashboard* and load the submap that you downloaded in the beginning.
- Ensure the following parameters are set:

Protocol on UART/USB output	Set to Marvelmind (This parameter is under Interfaces tab).
Location update rate	Set to 16+ Hz.
Movement filtering	Enabled with a window of averaging of 5.

- You should be able to see the location of each beacon. Also make sure that the submap is frozen.
- Move the hedgehog and check if it's moving in *Dashboard*.
- If it's moving, close the program (yes, you can close it! The coordinates of the position will still be streamed)

Ensure that you have the file "hedge_rcv_bin.cpp" in your workspace and compile it. Afterwards, you can run the command "rosrun [your_package] hedge_rcv_bin [usb_port]", where usb_port should be /dev/ttyACM0 or /dev/ttyACM1 depending on which USB port you have Marvelmind's beacon connected to.

Make sure to listen to the topic, ensuring that the position's coordinates are being published by running the command: "rostopic echo /mavros/vision_pose/pose" which should output messages containing the position of the hedgehog.

Pre-flight tips

Before running any experimental trial, it is obligatory to check if the safety measures are ensured. The usage of both ropes attached to the drone is recommended. Make sure you firmly attach the rope that is attached on the ceiling on the upper part of the drone's frame. The other rope should also be firmly attached to the lower part of the airframe of the drone.

Ensure that the position of the hedgehog is being published in the respective topic. It is advised to try arming and disarming the rotors without blades, so that you know that the communication is well established. Certify that the battery you will use in the drone is charged and that the blades are securely attached.

If you want to perform a takeoff maneuver using QGC you can set the altitude you desire to achieve using parameter MIS_TAKEOFF_ALT.

After that, you must connect PX4 to QGC, for you can run px4.launch using the IP and port of the Wi-Fi module using the following command:

```
roslaunch mavros px4.launch fcu_url:="udp://:14550@192.168.4.2:14555"
```

Make sure you can access to the data of the autopilot in QGC, including the position from Marvelmind and then you can start your flight, whether by controlling it via QGC or using your code.