Crash Course Paparazzi 2018
Using optitrack and programming your own flightplan

Lesson 2... Your first autonomous flight

Roland Meertens, Christophe de Wagter, Guido de Croon, Tom van Dijk

February 2018

Introduction

This week is a very exciting week: you will have your first manual flight and you will perform your first autonomous flight. Although there is a written document in front of you, you will mainly use our video tutorials on Youtube. This document will merely guide you through the videos[1], consider the videos as your main resource if you have questions.

Goals of this today

- Download and install Paparazzi
- Upload a program to your drone
- Fly manually with your drone
- Get a 3D fix on your drone
- Perform your first autonomous flight
- Edit your flightplan
- Create a safety rule

Downloading and installing Paparazzi

Paparazzi only runs on Linux. If you Linux installed as the native OS or as a dual boot you can follow the instructions on this site: `http://wiki.paparazziuav.org/wiki/Installation` and watch this video: `https://www.youtube.com/watch?v=eW0PCSjrP78`. If not, you can download a pre-made virtual box that has everything installed here: `https://www.openuas.org/pub/mirrors/mc/Ubuntu%20MAVLab%20Course%202017.vdi`. The password for this virtual box is ”mavlabcourse” (the same as the username).

Please checkout the branch mavlabcourse on this repository: `https://github.com/tudelft/paparazzi`. When starting paparazzi with the start.py program `/paparazzi$python start.py`, make sure you select as Conf: `userconf/TUDELFT/tudelft_course2018_conf.xml` and as Controlpanel: `userconf/TUDELFT/tudelft_course2018_control_panel.xml`.

[1] The movies are somewhat outdated, yet still highly relevant. Please also refer to the installation manual, which is part of this document.

Uploading your first program

You will now upload your first program to the Bebop. There is only a tutorial video for another Parrot drone, the ARdrone 2. As these drones are very similar, you can learn how to upload your first program by watching this video: `https://www.youtube.com/watch?v=eojAPZvT1Ck`.

Fly manually with a joystick

Now it's time to test if your drone works by flying it. Plug in your Hobbyking joystick and check that:

- You started the paparazzi ground station (and have a datalink with the drone)
- You started the joystick program and use the file hobbyking.xml
- When you switch the mode switch (upper left switchon the hobbyking joystick), the mode changes in the Paparazzi ground station.

We start by checking if the drone works by holding the 'cross' of the drone hand very firmly (with your hands away from the propellors) while wearing safety glasses. Put the drone in ATT mode and arm the motors by putting the left stick to the lower-right position. If everything went well the motors should now spin slowly. Put the left stick up to give more throttle and verify that this works. Now try if pitch and roll work with the right stick. Also test if the drone steers in the correct direction when turning the drone with your hand. If the drone is tilted to the left it should give more thrust with its leftmost propellors.
If all these checks worked it is time to start your first manual drone-flight!

Using the Optitrack system

To use the Optitrack system, view this video: `https://www.youtube.com/watch?v=7t6oqgIWGMc`[2][3].
To make sure you don't lose GPS while flying, check that:

- You applied the markers correctly
- The markers are visible to the software (no blinking spots in the Optitrack software)
- Optitrack is callibrated

As soon as you have a 3D fix it is time to test your drone manually by carrying your drone through the arena. Verify that:

- The position of your drone is correct: if you walk to the left the drone on your GCS will go to the left.
- The heading of your drone is correct: point your drone in different directions and verify the drone is looking in the right direction on your ground station (if not: did you calibrate the rigid body in optitrack with the nose pointing in the direction of the arrow?).
- The positions of your waypoints are correct: walk with your drone to the waypoints in your flight plan and verify that they are inside the arena (not too close to the nets) and safely reachable for the drone.

[2] The field Rigid Body ID is called user_data in later versions of motive, and due to a bug in the current version this number cannot be changed.

[3] Due to the installment of grass in the cyberzoo there is currently no arrow, register your drone in the optitrack pointing the same way as the video.

Your first autonomous flight

First test your flightplan in the Paparazzi simulator. Make sure you know what every button in your groundstation does. If and only if you verified that everything mentioned above is correctly working (is the joystick still working so you can take over?) you can start your first autonomous flight. Set the drone in navigation mode (if you don't know what this is, watch the videos again) and select start engines in the ground control station. If your flight plan is correct the drone will now hover in the arena. Congratulations: you are now flying autonomously! Please don't get distracted now: keep monitoring your drone and be ready to take over. Also remember to check your battery voltage as it might drop very fast.
You already checked your waypoints before, let the drone fly to one of them. To do so, select a block in your flight plan which has a certain waypoint as goal.

Edit your flightplan

Now the power of Paparazzi is in your hands: you can now create a totally autonomous drone with the power of a flightplan. Read this page to understand what you can do: `http://wiki.paparazziuav.org/wiki/Flight_Plans`. Make sure you understand:

- What a waypoint is, and what a sector is.
- What exception, and while can do.
- What the vertical controls are that you can use (alt, climb, throttle).
- What the navigation modes are that you can use (attitude, heading, go, path).
- How you can set a variable and call a function.

Try to create a simple flightplan that performs something of your choice.

Create a safety rule

Last week we discussed several problems that can make your drone crash, such as an empty battery, losing GPS or coming too close to the wall of the arena. When flying your drone you want your drone to do something as soon as these dangerous situations occur:

- If your battery is empty, you want to perform a normal landing
- If your GPS is lost, you want to perform a normal landing
- If you fly to the wall of the arena you want to stay at the last safe point you found.

The Paparazzi flight plan allows you to create exceptions: when the check of that exception becomes true the drone will execute a certain block. Look at the flight plan file
flight_plans/TUDELFT/tudelft_course2018_avoid_orange.xml to see how these exceptions are implemented.

Installation manual

This manual is only required if you aren't using the supplied VM.

To get the TUDELFT/mavlabCourse branch working there are some instruction you'll have to follow. The instructions are straightforward and should be easy enough to follow. Try to understand why every step is neccesary, this will improve your understanding of how everything works.

1. Install Paparazzi UAV as mentioned in `http://wiki.paparazziuav.org/wiki/Installation`. It is recommended for Ubuntu users to use the one-liner.
2. Open a terminal window and navigate to the paparazzi folder "$ cd paparazzi". Add the mavlabCourse remote "$ git remote add mavlabCourse https://github.com/tudelft/paparazzi", and update using "$ git fetch mavlabCourse mavlabCourse"
3. Checkout the mavlabCourse branch using "$ git checkout mavlabCourse"
4. Sync, initialize and update the submodules using:
   (a) "$ git submodule sync"
   (b) "$ git submodule init"
   (c) "$ git submodule update"
5. Build paparazzi by using "$ make"
6. Select the right conf and control_panel files using "$ python start.py"
7. Install the toolchain used by Parrot to build software for the bebop.
   (a) Go to the bebop tools folder using "$ cd ./sw/tools/parrot/bebop"
   (b) Install gdebi using "$ sudo apt-get install gdebi"
   (c) Install the toolchain using "$ gdebi ./parrot-tools-linuxgnutools-2012.03_1.0.0-1_i386.deb"
8. Prepare OpenGL ES
   (a) Go to the libopengles directory using "$ cd ../../../ext/libopengles/lib/arm"
   (b) Make OpenGL ES using "$ make"
9. Install ffmpeg, vlc and java using "$ sudo apt-get install ffmpeg vlc default-jre"
10. Install eclipse to easilly navigate through the paparazzi source code.
    (a) Download the eclipse-installer from `https://www.eclipse.org/downloads/download.php?file=/oomph/epp/neon/R2a/eclipse-inst-linux64.tar.gz`
    (b) Navigate to the Downloads directory and double click the eclipse installer
    (c) Select the C/C++ version
    (d) It is recommended to use the default installation directory
    (e) After eclipse is installed start eclipse
    (f) Navigate to "File - New - Makefile Project with Existing Code
    (g) Name the project paparazzi select the paparazzi installation location and keep the default options
11. Make openCV.
    (a) Navigate to sw/ext/opencv_bebop using "$ cd ./paparazzi/sw/ext/opencv_bebop
    (b) Set up your git Email using "$ git config –global user.email 'your_email@host.com'"
    (c) Set up your git Name using "$ git config –global user.name 'your_name'"
    (d) Make openCV using "$ make"
12. You can now enjoy running paparazzi and developing paparazzi in eclipse. Run paparazzi at any time by either the .desktop file located in "./conf/system/launcher/Paparazzi.desktop" or by using "$ ./paparazzi" from the paparazzi folder.