Project 1a Setup ArduPlane Test Environment

ArduPilot is an open source autopilot and is found onboard many hobby UAVs. Further information can be found on the ArduPilot webpage (<http://ardupilot.com/>). For this project, the ArduPlane autopilot will be used; be careful not to follow directions for other types of vehicles, such as ArduCopter and ArduRover. ArduPilot is free to use and comes with a simulation Software in the Loop (SITL) capability which we will use for this project. Additional information regarding the simulation software can be found at <http://dev.ardupilot.com/wiki/sitl-simulator-software-in-the-loop/>. It is strongly recommended that the test environment is constructed within an Ubuntu Virtualbox or Vmware virtual machine. Once you have a default/clean VM up and running, installation instructions for install the SITL environment on Linux can be found at <http://dev.ardupilot.com/wiki/setting-up-sitl-on-linux/>. I have also included a setup.sh script which should (as of 7/23/18) install the test environment for you. Use it at your own risk, but I do recommend reading it regardless. You will also need to download the mavlink utilities, found at https://github.com/Parrot-Developers/mavlink, it contains the mavlogdump.py file under pymavlink/tools.

Make sure you are able to run very basic flight tests, experimenting with options found in *simvehicle.py.* You can also install a ground control station (“Mission Planner”) for easier control and experimentation.

Additional information for connecting a ground control station:

http://ardupilot.org/dev/docs/using-sitl-for-ardupilot-testing.html

1. Run the CMAC-toff-loop.txt mission, the directions can be found here: (<http://ardupilot.org/dev/docs/setting-up-sitl-on-linux.html>). Make sure you fly for at least 10-15 minutes. Note that you can always speed up the simulation time using simvehicle.py parameters.

2. Use mavlogdump.py to parse the collected logs into a human readable format. Mavproxy will need to be installed, giving you access to the tools. Check out additional log analysis tools and command line options here:

<https://erlerobotics.gitbooks.io/erle-robotics-mav-tools-free/content/en/tools/mavlogdump.html>.

You may use the following command line:

python <path to mavlogdump>/mavlogdump.py <path to flight file>/*flight.tlog > ~/out.txt.*

3. Again use mavlogdump.py but now export the log as a csv file. Try experimenting with different header types, which can be found as message types within the output file from part 2. More information on the messages can be found here (https://mavlink.io/en/messages/common.html).

You may use the following command line:   
python mavlogdump.py --format=csv --*types=ATTITUDE,SERVO\_OUTPUT\_RAW,NAV\_CONTROLLER\_OUTPUT ~ flight.tlog > ~/out.csv.*

Submission directions:

Attach your submission as a compressed file with the following name “<lastname>\_project\_1a”.

Submit the following in a zip file:

1. The original flight tlog file.
2. Commands used in step 2 and 3.
3. Output files from step 2 and 3.