Novel Experiment Design for Unmanned Aerial Vehicle Controller Performance Testing

Cherub Dim, Francis Nabor, Giancarlo Santos, Martin Schoeler, Alvin Chua*
Mechatronics Research Laboratory, Mechanical Engineering Department, De La Salle University, Manila, Philippines

*corresponding author: alvin.chua@dlsu.edu.ph

Abstract. Unmanned Aerial Vehicles (UAV) are becoming popular to the innovating world. This paper presents a novel experiment design to determine the performance of UAV control system. The hardware platform developed is composed of a quadcopter with a multipurpose autopilot system. While the experimental design considers both quantitative and qualitative aspects of the controller. The qualitative components dealt with the hardware compatibility, acceptability and license. On the other hand, the quantitative component measured the accuracy of waypoints, altitude hold and loiter performance of the UAV. The results shows that the new experiment design platform was able to compare two popular autopilot software in terms of performance and acceptability.

1. Introduction
Unmanned Aerial Vehicles (UAV), also known as drones, are becoming popular more and more. They provide humans with different applications similar but not limited to: mapping, delivery, and entertainment. The UAV could be driven remotely, or be programmed on autopilot. These UAV/drones vary through the hardware configuration in the platform: Multi-Rotor, Fixed-Wing, Single-Rotor, and Fixed-Wing Hybrid. Multi-Rotor drone contains multiple motor which drives the platform, an advantage would be high-precision flight, quick throttle changes, and heavy-weight lifting capability as compared with other types of drones. The disadvantage of multi-rotor drone is flight endurance-due to multiple motors, and speed limitation due to the rotors configuration[1][2][3]. Drone control software is in the heart of any type of UAV. The implemented controller to drive the drone will define its performance and capability to perform the specified tasks. Currently, most drones are already equipped with control software. However, some drones are driven by closed source or proprietary control software. These software are exclusively developed and distributed by the manufacturer. Without the ability to modify the drone's software, the drone's application will be limited only to what was pre-programmed by the manufacturer.[4][5][6] Furthermore, whenever there are occurrences of bugs and errors on the software, the patches and updates will be dependent on the software's manufacturer. Hence, open source UAV software are progressing wherein the user can modify and develop the program to suit the vehicle’s application. One of the advantage of open source software is the development of the software is dependent on the community wherein innovative ideas, progresses and bug fixes are accomplished for the application of the autopilot software and is available for further improvement. Moreover, open source UAV software are freely distributed, and are available at the user’s dispersal. There are many variations of open sourced UAV software; each equipped with its own capability and methods of implementations[7]
This paper presents a new experimental design that is able to analyze the performance of an autopilot software. A quadrotor platform together with a multipurpose controller was used as the testbed of the experimental design. The experiment design focuses on the qualitative and quantitative aspects of the flight control system. Two well-known open source autopilot software in the market were chosen as test case to validate the experimental design.

2. Experimental Platform

The control system of a quadcopter plays a key role in its performance. The precise control of the attitude and altitude states of the aerial vehicle is very critical in different applications. This section describe the frame and the electronic components chosen for the experimental platform (see Figure 1). The frame of the quadcopter chosen is the DJI F450 Frame due to its proven design and functionality. The following are the other components: ReadyToSky 2212 920KV, ReadyToSky 30A Brushless ESC, Propellers, Dinogy 11.1V 5000mAh, RadioLink AT 10 II, 3DRobotics Radio Telemetry, and Ublox Neo-M8N GPS.

![Experimental Platform](http://example.com/figure1)

**Figure 1.** Experimental Platform

Two autopilot softwares are chosen in the testing namely: PX4[8][9] and Ardupilot[10]. These are the most widely used open source flight controller available in the market. They use different ground stations as shown below. For Ardupilot, they use Mission Planner (see Figure 2) while PX4 uses QGroundControl (see Figure 3).

![Mission Planner](http://example.com/figure2)

**Figure 2.** Mission Planner (http://ardupilot.org/planner/)
3. Methodology

The methodology flowchart was shown in Figure 4. This starts with the design and assembly of the quadcopter hardware as shown in the previous section. This was then followed by the software integration of Ardupilot and PX4 into the quadcopter flight controller. The experiment design developed considers both the qualitative and quantitative aspects of the flight control system. Finally, actual testing was done.

![Methodology Flowchart](image)

**Figure 4. Methodology Flowchart**

The design and testing of the research focuses on the test plan in qualitative and quantitative analysis. Qualitative Analysis accounts for the immeasurable data that couldn’t be gathered from the testing. While the Quantitative analysis will account for the actual data that could be acquired using the experimental platform for actual data analysis.

3.1. Qualitative Analysis

Qualitative analysis includes the following: hardware compatibility, user experience, licensing, and partner companies. First, the hardware compatibility accounts for various particular operating system or device management software for the unmanned aerial vehicle in both ArduPilot and PX4 applications. Second, the user experience accounts for the different aspects of the interface which includes the calibration process, parameters guideline, PID tuning, and user interface (UI). These open source software have their own unique UI that help distinguish one from the other by having features and modes that each software is capable of doing within the interface. Also, the licensing plays a crucial role to the researcher so that it can be modified and redistributed. Finally, partner companies
shows the support of different companies to the software adding value to its existence and improvements in the future.

3.2. Quantitative Analysis
On the other hand, the Quantitative Analysis relies on mathematical and statistical method to evaluate data points. Quantitative Analysis focuses on the Pre-Test such as GPS and Barometer Calibration, Flight Path in terms of time and accuracy, Loiter, Return to Launch, Altitude Control, and Battery Management.

Calibrating the barometer and GPS are needed to ensure that the data gathered with respect to altitude was accurate. The accuracy of the flight path for each software was tested and how fast the software can complete the run of the desired flight path. This is to determine which software has the better capabilities of following configured paths that will have low percentage errors while being the fastest to complete the run. Considering that all the devices and instruments attached to the drone are correctly calibrated. Figure 5 shows the flight path accuracy test points.

![Figure 5. Flight Path (Accuracy and Time)](image)

The stability of the drone during Loiter mode at a certain altitude was tested. This is to determine which software has the better capabilities of holding the latitude and longitude at a certain altitude with minimal errors. Figure 6 show the Loiter flight path.

![Figure 6. Loiter Flight Path](image)

The return to launch (RTL) experimental procedure below was used to test the efficiency of the quadcopter to go back to its home position. The test will focus on the duration and deviation of the GPS coordinates the drone landed on to the original location. Figure 7 shows the RTL flight path.
The altitude elevation accuracy of each software was tested. This is to determine which software has the better capabilities of following desired altitude elevations. Figure 8 shows the altitude control flight path.

![Figure 8. Altitude Hold Flight Path](image)

The battery management of the software will be tested to determine how the two software consume battery in order to accomplish a certain task. Figure 9 shows the flight path.

![Figure 9. Battery management flight path](image)
4. Results
After performing the experiment design and testing of the flight paths, the data gathered from the test was constructed and analyzed. Both ArduPilot and PX4 were compared qualitatively and quantitatively.

4.1. Qualitative Analysis
As seen from the items listed in their respective websites, it can be observed that ArduPilot is compatible to more hardware as compared to PX4 especially with closed source hardware with 11 different types of off-the-shelf drones to PX4's 3 compatible closed source hardware. Based on Table 1, it can be concluded that ArduPilot focuses more on their software being compatible to different kinds of open and closed source

| Hardware Compatibility Tally |
|-----------------------------|
| ArduPilot                   |
| Open Source Hardware        | 10 |
| Closed Source Hardware      | 11 |
| Total                       | 21 |
| PX4                         |
| Open Source Hardware        | 13 |
| Closed Source Hardware      | 3  |
| Total                       | 16 |

ArduPilot has more partner companies for each software developer compared to PX4 comparing 39 to 24. Numerically, ArduPilot is greater than PX4 in number of partner companies, but the partners of PX4 are bigger companies weighing over the 39 partner companies of ArduPilot.

The licensing of the two software is very different from one another wherein these licenses focuses on two different scenarios. ArduPilot licensing, under the GNU General Public Version 3, has a focus on user experience and distribution of businesses and closed source hardware in which they must notify the developers or beta testers of any modifications or bugs in the software as well as notifying the end users that the software is open source. PX4, on the other hand, focuses more on the proper distribution of the software in which the copyright notice, list of conditions, and disclaimer must first be assented along with prior copyright holder and contributors.

Overall, the calibration process of PX4 is better than the calibration process of ArduPilot as it is more interactive and easier to understand as it uses pictures and graphics in order for the user to know what should be done next. PX4 also has a better calibration process because all components that need to be calibrated could be done within the UI.

For the Parameters changed, the use of ArduPilot is easier because unlike PX4 wherein the PID and other parameters had to be manually tuned, ArduPilot could be used immediately after uploading the software on to the drone as it already works smoothly right out of the box. The only parameter that had to be modified in ArduPilot was the function wherein the motors would not activate when the drone was armed before the flight.

Moreover, Ardupilot, again has a better user interface even if it has a cluttered interface and a not so streamlined interface. Ardupilot has more features available at its disposal than PX4 which is less refined with only limited features and flight modes. Another important feature that ArduPilot has is that the user can download and analyze data within the software and export it into a Matlab file whereas PX4 requires the use of external software.

4.2. Quantitative Analysis

The barometer and the GPS module were tested. It was found that the average error of the barometer reading is 6.7% for ArduPilot and 2.6% for PX4. For the GPS module the average error for ArduPilot is 0.87% while PX4 is 0.24%.
The flight path accuracy tests shows that PX4 is better in checkpoint accuracy as compared to Ardupilot to which this difference could be attributed to the response of the software as the mission is being performed. Ardupilot registered an average error of 0.7% while PX4 had a 0.55% error. Figure 10 shows a sample run of the flight path accuracy data while Figure 11 show the flight path accuracy t-test. The t-test done showed that the resulting displacements of both software controllers are significantly different from one another. This would attributed to the difference in response and its respective performance in coordinates calculations.

The flight path time test shows that PX4 performs the mission faster than Ardupilot by an average of 0.476 seconds. Voltage measurement were also checked and were observed to have insignificant effect on time completion of the mission. Table 2 shows that flight path time data summary.
Table 2. Flight Path Time Data Summary

| Run | ArduPilot | PX4 |
|-----|-----------|-----|
|     | Time      | Voltage | Time      | Voltage |
| 1   | 17.09 s   | 12.2 V  | 17.30 s   | 12.2 V  |
| 2   | 17.34 s   | 12.0 V  | 16.08 s   | 12.1 V  |
| 3   | 17.04 s   | 11.8 V  | 16.85 s   | 11.9 V  |
| 4   | 17.06 s   | 11.7 V  | 16.56 s   | 11.8 V  |
| 5   | 17.02 s   | 11.6 V  | 16.38 s   | 11.7 V  |
| Average | 17.11 s | 16.634 s |

From Figure 10, ArduPilot have points wherein there are notably large displacements. In terms of speed, PX4 is also faster than ArduPilot in accomplishing the RTL mode as by an average of 9.3 seconds, at a set height of 15 meters. ArduPilot slows down the decent at the 10 meter mark unlike PX4 wherein the speed of the decent is consistent throughout the RTL mode.

Figure 12. Return to Launch Graph Summary

Based from the data generated for the battery test, ArduPilot consumes more battery in-flight than PX4. This means that PX4 can have a longer flight time than ArduPilot due to its battery consumption in-flight. This also means that there is a distinctive difference between the two software in terms of how the sensors communicate with each other and how it accomplishes missions.

The PX4 has a better performance in altitude control as compared to the capabilities of ArduPilot. The data gathered in Figure 11 shows that PX4 can elevate to different heights as accurately as possible unlike ArduPilot which elevates at heights with a large discrepancy to the desired altitude. It can also be said that ArduPilot has the better capability of holding the height it has elevated to as it is shown in the graph that it is more stable unlike PX4 which has spikes in the graphs below.

Figure 13. Altitude Control Test Run 1
From the final tally in Table 3 done for the quantitative tests, PX4 is better between the two-autopilot software. The tests conducted shows that PX4 tends to be more accurate in attaining a desired value especially on runs that concern altitude control and flight path accuracy. On battery management test, PX4 was only marginally better than ArduPilot. While on RTL test Ardupilot has more displacement error. However, ArduPilot performed better in loiter test. On the other hand, deducing from the qualitative findings of both software, ArduPilot is shown to be better in hardware compatibility and user experience. While PX4 is better in licensing and partner companies.

Table 3. Final Tally Results

| Qualitative Test          | ArduPilot | PX4 |
|--------------------------|-----------|-----|
| Hardware Compatibility   | X         |     |
| Licensing                |           | X   |
| Partner Companies        |           | X   |
| User Experience          |           | X   |

| Quantitative Test         | ArduPilot | PX4 |
|--------------------------|-----------|-----|
| Flight Path Accuracy     |           | X   |
| Flight Path (Time)       |           | X   |
| Altitude Control         |           | X   |
| Loiter                   |           | X   |
| Return to Launch (RTL)   |           | X   |
| Battery Management       |           | X   |

Conclusions
The paper was able to present a novel experiment design for UAV performance testing. A test case was done comparing two popular autopilot software namely: ArduPilot and PX4. The selection of the components of the experimental platform used in the study was based on availability, costing, and compatibility to different autopilot software. The integration and implementation of each software were done in accordance to each specifications. Hence, it became possible for the designed flight test to be uniformly conducted to each software.

The qualitative comparison of both software shows that the two software are better in some aspects. Ardupilot was better in terms of hardware compatibility and user interface. While PX4 in terms of licensing and partner companies.

On the other hand, the quantitative test results show that PX4 is better than Ardupilot. PX4 faired better in most of the tests which involved accuracy in attaining a desired value, such as altitude and checkpoints. However, in loiter tests ArduPilot performed better. These results were obtained by extraction of flight data log from each software that were statistically analyzed and used to plot different graphs.

The researchers recommend that further performance testing of other UAV controllers be done using the novel experimental method as a strategy. This would provide a benchmark to compare future controllers for its performance.

Acknowledgments
The authors would like to acknowledge the funding support of DOST-PCIEERD and the Mechanical Engineering Department of De La Salle University for the assistance in the conduct of the study.

References
[1] Aarons, T. D. (2011). Development and Implementation of a Flight Test Program for a Geometrically Scaled Joined Wing SensorCraft Remotely Piloted Vehicle. Masters Theses [17872],
[2] Choi, H., Geeves, M., Alsalam, B., & Gonzales, F. (2016). Open source computer-vision based guidance system for UAVs on-board decision making. 2016 IEEE Aerospace Conference, Pages: 1-5, DOI: 10.1109/AERO.2016.7500600.

[3] DiCesare, A., Gustafson, K., & Lindenfelzer, P. (n.d.). Design Optimization of a Quad-Rotor Capable of Autonomous Flight. Massachusetts: Worcester Polytechnic Institute.

[4] Liang, O. (2017). Quadcopter PID Explained. OscarLiang.com.

[5] Garcia Carillo, L., Rondon, E., Sanchez, A., Dzul, A., & Lozano, R. (2011). Stabilization and Trajectory Tracking of a Quad-Rotor Using Vision. Journal of Intelligent & Robotic Systems, Volume 61, pp 103-118.

[6] Guido, G., Gallelli, V., Rogano, D., & Vitale, A. (2017). Evaluating the accuracy of vehicle tracking data obtained from. International Journal of Transportation Science and Technology 5 (2016) 136–151, https://doi.org/10.1016/j.ijtst.2016.12.001.

[7] Hoffmann, G., Rajnarayan, D., Waslander, S., Dostal, D., Jang, J., & Tomlin, C. (November 2004). The Stanford Testbed of Autonomous Rotorcraft For Multi Agent Control (Starmac). In In The Proceedings Of The 23rd Digital Avionics System Conference (pp. Pp. 12.E.4/1-10). Salt Lake City.

[8] PX4 autopilot. (2017). Retrieved from Pixhawk Autopilot: https://pixhawk.org/modules/pixhawk

[9] Team, P. D. (2017). Architectural Overview. Retrieved from PX4 Development Guide: https://dev.px4.io/en/

[10] Team, A. D. (2016). Learning Ardupilot. Retrieved from Ardupilot: http://ardupilot.org/dev/docs/learning-ardupilot-introduction.html