

Open-Source Project (OSPs) Platform for Outdoor Quadcopter

S. Sabikan¹ and S. W. Nawawi^{*,2}

¹Electrical Engineering Technology Department, Fakulti Teknologi Kejuruteraan, Kampus Teknologi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.

²Control and Mechatronic Engineering Department, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, 81300 Johor, Malaysia.

¹sulaiman@utem.edu.my, ^{*,2}sophan@fke.utm.my

Abstract – In recent years, there has been an increasing interest in quadcopter technology implementation in the real world; for instance in real estate photography, aerial surveying, periodic forest monitoring, and search/rescue missions. Generally, each quadcopter implementation required different sensors which are needed to attach and integrate into quadcopter system. However, the most critical part in almost cases is preparing the quadcopter flight performance and capability to be suited in any outdoor applications. Because of that reason, this paper has proposed an implementation of Open-Source Project (OSPs) platform as autonomous Unmanned Aerial Vehicle (UAV) quadcopter development that can be fitted for any outdoor applications or even in research experimental purposes. We started out with an explanation about the general approach that has been used in the development of a quadcopter testbed, and then followed with detail explanations in the OSP platform approach. The OSP platform is the most popular approach. The main reason is because of their flexibility in both hardware and software. The basic quadcopter configuration for autonomous flight also presented and applied. This paper also provided several outdoor experiments results in uncontrolled environment that have been executed using our developed testbed to evaluate their performance, such as attitude and altitude stabilization, interference and vibration effect, and trajectory mapping generation. Finally, throughout this project, we realized that the OSs quadcopter platform has offered almost complete frameworks in the development of quadcopter for any outdoor applications or even as a research testbed system. **Copyright © 2016 Penerbit Akademia Baru - All rights reserved.**

Keywords: UAV, quadcopter, open-source

1.0 INTRODUCTION

A quadcopter is one kind of autonomous aerial vehicles. It is most regularly called as a drone by the public. Quadcopter has attracted a considerable amount of interest from researchers or hobbyist because of a wide area of applications and it has a lot of advantages. The advantages of a quadcopter such as the capability of the vertical takeoff and landing (VTOL) - enables launch and landing with minimum space, hover capability, high manoeuvrability (i.e. performing longitudinal and lateral flight), and agility. Comparison of a standard helicopter with a single rotor, however quadcopter has four symmetry equivalent sizes of the rotor. Because of that configuration, quadcopter possess more advantages in terms of small size, efficiency, and safety. At the same time, quadcopter is possible to flight in tight indoor space

and in outdoor environment as well [1]. Due to these advantages, the quadcopter is eligible for many applications; like military services, surveillance, search and rescue, remote inspection, and photography [2].

Development of a quadcopter testbed is undoubtedly important in order to verify our method (e.g. proposed control strategy) in a real system and environment. It is also very useful for studying the dynamics, performances, and parameters tuning. Therefore, there are quite many researchers around the world were successfully developed a quadcopter testbed to assist their researches and developments. By having a quadcopter testbed, any developed new control strategy can be implemented and then verify the performance and effectiveness in the real system. Thus, the comparison of results can differentiate between simulation and real experiment.

There are many issues in the design and development of a quadcopter to perform as expected; such as research requirements, the scope of target applications, and overall costs. For instance, in the swam quadcopter research, they should prefer to use lots of small and low cost platforms [3]. To design the search and rescue operation using quadcopter UAV, for example, researchers are preferred to use commercial quadcopter platform, which has capability to provide a quality sensors information, satisfactory flying duration, supported with collision free navigation and information sharing capability [4]. Furthermore, several researchers need a quadcopter testbed to be used only for specific purpose experiments in the indoor laboratory environments [5]. Commonly, indoor environment is lack of GPS signal for quadcopter localization, which is very important positioning system to realize autonomous navigation and path planning. Because of that, some researchers had utilized a vision-based tracking system in order to tracking, instead of using the GPS sensor [6]. Other than that, time constraint in the quadcopter testbed development process also needs to be considered carefully. The longer time is required in the testbed development, which will definitely increase the cost.

This paper will start with defining the development quadcopter categories in the next section. Section 0 gives a brief overview of the Arducopter quadcopter OSPs, which includes Ground Control Station (GCS) software and the quadcopter main hardware components. It will then go to the explanation on our quadcopter UAV configuration, in Section 0. Section 0 presents several experiments results from flying quadcopter in the outdoor environment, and in the end, we summarize our findings and experiences in this project.

2.0 QUADCOPTER DEVELOPMENT APPROACH

As illustrated in Figure 1, development of the quadcopter testbed can be categorized into three different approaches, instead of two in [2].

The first approach is by constructing their own quadcopter controller to suit with their dedicated purpose or as a ground test bench. Successes development has been reported from academia such as OS4, STARMAC [7], X4-flyer to mention a few. Such quadcopter prototypes are useful for studying the dynamics of the quadcopters at the early stage, and in this way the safety for parameter tuning and experimental tests can always be guaranteed [2].

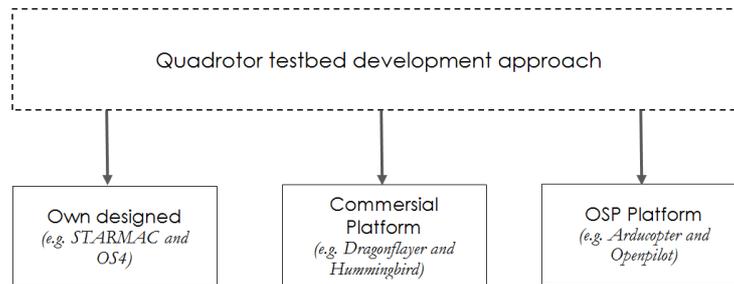


Figure 1: Quadcopter testbed development approach

Second approach is by purchasing commercial quadcopters platform to build as an experiment testbed, such as the Dragonflyer [8], QBall X4 [5] and several models produced by Asctec Hummingbird company [2]. In these categories, the controllers for a quadcopter are allowed and can be directly implemented to the real systems, so that real-time flight experiments can provide a convincing way to verify the effectiveness of the developed controllers. Basically, commercial product needs higher initial cost, but providing complete solution to our research purpose.

Furthermore, third approach is develop a quadcopter testbed by using open-source project (OSPs) platform (see this paper [9]). In the case of quadcopter OSPs, one of the main reasons to use them is flexibility in both hardware and software, which makes modification easier to meet the specific requirements of a user. Several OSPs quadcopter autopilot hardwares are available in the market, such as ArduPilot, OpenPilot Mikrocopter and e.t.c. Almost OSP architectures are freely to use, easy to modify the software codes, firmware codes, providing free technical support from the internet community, and offer flexible ready-made peripheral devices. Pixhawk [10]

After considering the issues on the development of a quadcopter testbed in the research activities, this paper will present among the easy, fast, and effective approach to develop a quadcopter testbed for any research purposes. Our main considerations are flexibility and low cost at the same time. For these reasons, this project will present and focus on utilizing open-source platform.

Nowadays, there are many open-source projects for quadcopter UAVs that are available freely in the market, such as Arducopter, Openpilot, Multiwii, and Pixhawk. For more details, this [9] magazine paper presents in general on the Open Source Project (OSP's) available in the market for quadcopter UAV up to year 2012. There are eight platforms were reviewed. From our literature, Arducopter platform is the most popular (based on user volume), fully established hardware, software and community support [9]. Therefore, we decided to implement Arducopter platform in this project.

3.0 INSIDE OPEN-SOURCE PROJECT FOR QUADCOPTER UAV

This section will introduce about the basic functions of each component for both software and hardware of the Arducopter OSPs platform. Arducopter autopilot is an open-source project for aerial vehicle designed from the Arduino framework. Arducopter is a developed project by

community from all over the world and had used internet for sharing all resources together. The Arducopter quadcopter platform consists of software, hardware and firmware. In other words, Arducopter not only provides an open-source in term of software, but also provides an open-source for hardware in reasonable price with free firmware. Those are interesting parts of an open source.

3.1 Mandatory hardware for outdoor implementation

3.1.1 Flight Controller

Flight controller or flight avionic is a main board hardware for processing and operations of the quadcopter UAV system, for example; controlling the motor, interfacing with internal or external sensors, communicate with ground control, and implementing the control law. The performance of flight controller depends on the microcontroller used.

The flight controller that we have selected for the quadcopter UAV platform is the ArduPilotMega (APM) 2.6, an open-source embedded autopilot platform designed by 3DRobotics. This autopilot is based on the Arduino Mega 2560 and features an integrated inertial measurement unit (IMU) that measures acceleration, rotational velocity, and the direction of the magnetic field in 3D. APM autopilot is equipped with 6-DoF IMU in a single chip, MPU6000. The APM processes the IMU data, serial commands, and user input from a 2.4GHz telemetry radio control signal, and sends speed commands to the ESCs for control and stabilization of the aircraft. It also provides ready port for telemetry and RC receiver interfacing. The firmware of this board is uploaded via USB using the APM planner software.

Instead of APM2.6 autopilot, Arducopter can support many options of flight controller such as Pixhawk, PX4, Pixracer and Erle-Brain Linux Autopilot. Currently, the latest stable firmware for APM2.6 is version 3.2.1 and updated firmware only support for Pixhawk autopilot.

3.1.2 Propulsion system

The propulsion system is composed of four Brushless DC motors (BLDC), four Electronic Speed Controllers (ESC), and four propellers. Each component has its own performance characteristics and region of maximum efficiency. The selection and combination of each component is crucial in achieving an overall propulsion system which is well suited to the application and efficient [11].

Almost all quadcopters are actuated by BLDC motor. The SunnySky X2212 with 980 KV (RPM/Volt) BLDC motor is selected for our quadcopter. The motor weight is 58.5g, required 20A current from ESC and can be driven by using 2 or 4 cell LiPo battery.

A quadcopter requires two sets of counter propeller pairs for balancing its aerodynamics, but the selection of the most efficient propeller requires finding out the propeller with the highest thrust to current ratio. However, from the BLDC motor's datasheet, the suited propeller for our motor Kv size is APC 8038 with size: 8 x 3.8" (Length x Pitch), Diameter of Shaft: 6.0mm, Recommended Wheelbase: 300-500mm, Recommended Motor: 650KV - 1150KV, Adopts APC material, and super impact resistance.

3.1.3 External Sensors

GPS provides a practical, easy-to-use and the most suitable sensor for outdoor vehicle positioning, such as car, rover, and aeroplane. However, for accurate position estimation and mapping, researcher are turning to vision sensor systems [12] for indoor application, or combined the GPS reading with IMU measurements[13].

GPS/compass module sensor is the general external sensor for an autonomous quadcopter for outdoor environment. It will be used for positioning and autonomous flight. Basically, external compass comes together with GPS module. The standard GPS receiver's position accuracy is within 2-4 meters. We are surely recommended to use a separate compass sensor to reduce interference from high current around the quadcopter, which coming from electronic devices.

On the other hand, lots of external sensor or devices can be integrated with ArduPilotMega autopilot, such as power modules, rangefinders, optical flow sensor, cameras & gimbals, OSD for FPV, servo, relay, switch, UAVCAN ESCs, EPM gripper (EPM688), PPM encoder, received signal strength indication (RSSI), external LEDs, parachute, joystick or gamepad, buzzer, landing gear, precision landing (IRLock), ADS-B receiver, and companion computers.

3.1.4 Wireless Communication Systems

Several options are available for quadcopter wireless communication system, however, with slightly different protocol and purpose. In this project, a radio telemetry unit and RC transmitter also used to support real time digital communications between the quadcopter and ground control station.

Telemetry wireless implements MAVlink protocol serial connection [29]. This protocol is specially designed to share information with a ground station. For example, attitude (Euler angles and Euler angle speeds), VFRhub (ground speed, air speed, altitude and climb rate), GPS raw data, and RC channel (the values of the 8 radio frequency channels). This serial connection also provides heartbeat (systems can use this message to track if the system is alive), navigation controller output (with information about the goals and errors), hardware status (board voltage), raw IMU (acceleration, gyroscope and compass values), pressure, and some more flight parameters. In other words, MAVlink telemetry communication will collect and record all flight information we need into ground control station.

The second communication that can be used with Arducopter is Radio Control (RC). RC transmitters are used to minimally control vehicle movement (throttle) and orientation (pitch, roll, and yaw). Each of these control signals is mapped to transmitter stick/switch(s) and in turn to autopilot channels from the connected receiver. Unused channels can be mapped to control additional peripherals. Default channel mappings as following;

- Channel 1: Roll
- Channel 2: Pitch
- Channel 3: Throttle
- Channel 4: Yaw

- Channel 5: Flight modes
- Channel 6: (Optional)

Calibrating each of the transmitter controls/channels can be done from Mission Planner software. It is a straightforward process.

In addition, Radio Control System is prepared to control quadcopter in manual mode, especially in emergency cases or any situation needed to manually control. To communicate in real time between our flying quadcopter with ground control, we have used wireless MAVlink telemetry communication. We called Mission Planner software that is equipped with telemetry communication and RC transmitter device as a Ground Control System.

3.2 Ground Control Station (GCS) Software

The Ground Control Station (GCS) is an application software that runs on a ground-based computer and communicates wirelessly with the UAV. Generally, GCS is used to configure vehicle parameter, controlling, and continuously monitoring the UAV status at a real time. Here are at least eight different options ground control stations, as shown in Table 1; that can be used to communicate with our quadcopter Arducopter platform based. In addition, all softwares are supported by the MAVLink communication protocol and they are free to use.

Table 1: Ground Control Stations Option for arducopter platform based

GCS	Userbase	Supported OS
QGroundControl	Basic and Advanced Users, Developers	Win,Mac,Linux
Mission Planner	Basic and Advanced Users, Developers	Win only, runs on Linux using Mono and OS X using VM
APM Planner	Basic and Advanced Users, Developers	Win, Linux, OS X
MavProxy	Advanced Users, Developers	Win, Linux, OS X, Android
Tower	Mobile/tablet users	Android
AndroPilot	Mobile/tablet users	Android
UgCS	Mobile/tablet, Basic and Advanced Users	Win, Linux, OSX, Android (PV+ app)
Drone Deploy	Mobile/tablet users	Web UI

Mission Planner Version 1.3.30 was selected in this project for our ground control station. It was widely used for autopilot project. Mission Planner is a fully featured GCS running on the Windows Operating System. Its features include: Point-and-click waypoint entry (e.g. Using Google Maps), provide mission commands from drop-down menus, able to download mission log files and analyse them directly, easy to configure APM settings for any airframe design, interface with a PC flight simulator, and in flight HUD view.

4.0 QUADCOPTER TEST BED PLATFORM CONFIGURATION

Basically, assembling quadcopter arducopter platform is a simple process, but you need to do it very carefully. This development process can be divided into two major parts, which are hardware assembling and followed by hardware tuning using ground control software.

4.1 Hardware setup

Our autonomous quadcopter UAV system configuration for this project is shown in Figure 2. That was designed for a very minimum requirement in order to implement autonomous flight mission. This configuration composed of two elements; an on-board quadcopter and a ground control system.

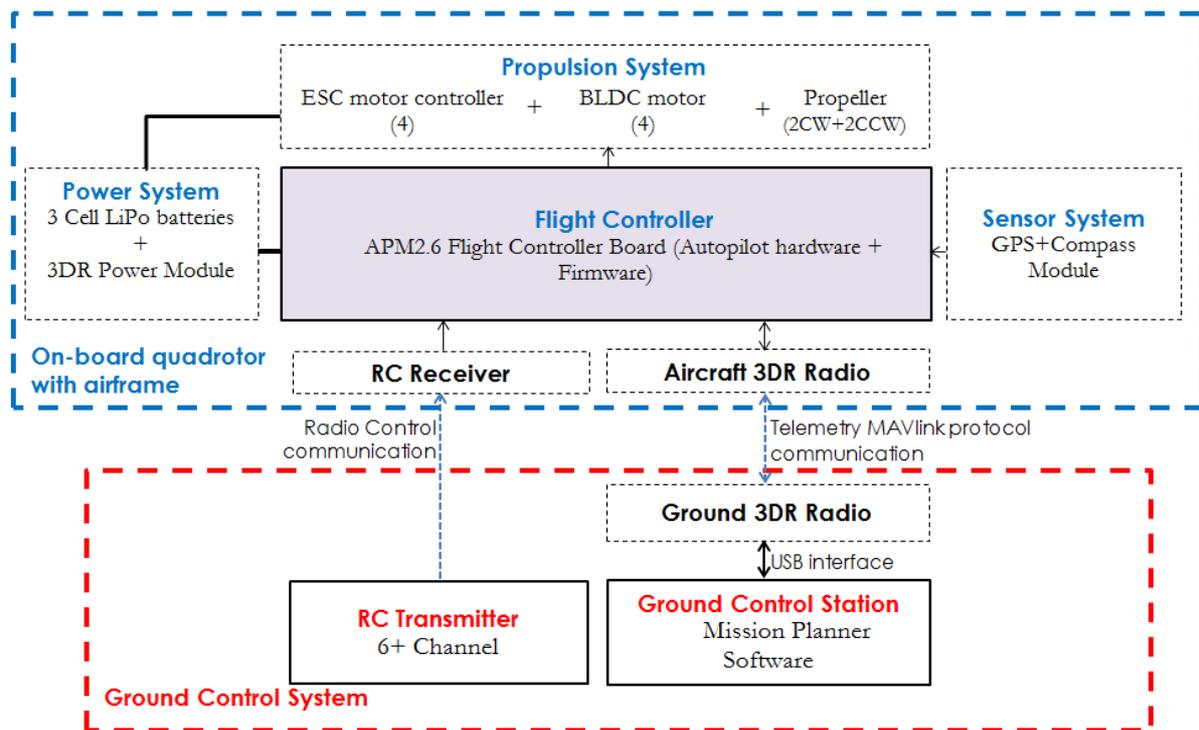


Figure 2: Configuration of an autonomous quadcopter UAV

In the on-board quadcopter (blue box), it shows the connections of the essential electrical components (e.g. propulsion, power, sensor, and communication) of a quadcopter on flight controller board. A quadcopter structure airframe consists of the main center plate (allocated for assembling the flight controller, battery, sensor, and communication devices) and four arms to support the propulsion system. We used the x configuration quadcopter frame to mount all electrical and mechanical quadcopter components. We need to do carefully fitting (i.e. placed in the center of gravity frame and in the correct direction) the flight controller board and we recommended to support it by a bumper system. It is helpful to reduce vibration effects from the motors. The propulsion system is composed of four Brushless DC motor (BLDC), four Electronic Speed Controller (ESC), and four propellers.

We prepared two types of communication system to communicate between an on-board quadcopter and the ground control system, there are a Radio Control (RC) system (i.e. include RC transmitter and RC Receiver) and Telemetry System (i.e. Ground 3DR Radio and Aircraft 3DR Radio). The RC receiver device and aircraft telemetry radio device were connected with the flight controller board, while 3DR radio device were directly connected into the USB port of the ground control station. Basically, both communications are used for initial flight

configuration, real-time flight monitoring and to manually control quadcopter movement and orientation.

4.2 Software Setup

Software setup of the autopilot includes downloading and installing the Mission Planner GCS, loading the firmware into autopilot and then performing initial configuration and calibration. The most critical step is performed controller configuration.

Mission Planner is an open source software, free to customize and available for Windows OS. The installation process is straight forward. After the Mission Planner installed onto the computer, the appropriate firmware can be downloaded to the autopilot via USB cable.

As part of software setup, quadcopter hardware components were required to calibrate using the Mission Planner. That component includes frame configuring, the RC transmitter/receiver, compass, accelerometer, and the ESCs. This is a mandatory step before the first flight and it is recommended to first starting to use quadcopter in several different flight modes before progressing to the next steps.

The next step is performing quadcopter controller configuration in order to get best respond and optimal flying performances. It involved Roll, Pitch and Yaw controller tuning parameter. There are two ways to do this configuration step; there are by using auto tune feature or manual tuning. Autotune is more critical and need an experienced operator, that is why we preferred to implement manual methods. As we know, Arducopter had implemented default PI+P controller configuration based on an inner-/outer-loop structure. From the Mission Planner, we were needed to adjust basic Roll and pitch and throttle tuning firstly, before changing the PID autopilot parameters.

PID autopilot parameter adjustment is to make sure our quarter can fly and reliable performance in the wind. Although there are many gains that can be tuned in quadcopter to get optimal performance, the most critical is the Rate Roll and Pitch P values which convert the desired rotation rate into a motor output. Rate Roll P and Rate Pitch P can be tuned in flight using your transmitter's channel 6. After that, to get optimal performance, we need to make an adjustment from the default PID gain setting, such as Roll/Pitch, Yaw, Altitude hold, Loiter, and Waypoint navigation. It is because the default PID gain is best for the 3DR quadcopter frame.

5.0 OUTDOOR FLIGHT EXERIMENTS RESULTS

In this section, we will present only two outdoor flight experiments results. They are quadcopter flight performances and multi-waypoint flight trajectory. These experiments were conducted in sequence and all analysis with excerpts from the quadcopter MAVlink telemetry raw data (explained below). In the first experiment, some basics that related to quadcopter performance will be examined; such as attitude and altitude stability, compass sensor magnetic interference effect, and vibration effect. After we had satisfied with first experimental results (i.e. by tuning PID parameters), next experiment will be conducted that aims to examine the flight route accuracy and generate a 3D trajectory mapping. To perform

this experiment, we have utilized the *Mission Planner* features of the ground control software. We have created an automated mission planning with five waypoints which were selected randomly before launching the flight mission for several times. All waypoint locations were designated from the GPS coordinate system.

Arducopter platform has offered two ways to collect flight data from quadcopter. These two methods are able to record very similar data, but in different ways.

Firstly, it is recorded by on-board dataflash memory on the flight controller, which is preferred as “dataflash logs”. On a quadcopter, dataflash logs are created soon after their first arm. This dataflash logs can be extracted from an autopilot via USB connection after a flight. For instance, APM2.6 autopilot has only 2MByte on-board dataflash memory size. However, with that amount of memory, it is very limited and not sufficient to record all flight data, especially for continuous or repeated missions. Despite that, it is generally recommended method for mechanical failure diagnosis because of the higher data stream rate.

Secondly, the platform is recorded directly to the ground control station computer, which is preferred as “telemetry logs” or in short form as known as “tlogs”. Telemetry logs can collect similar information as well as dataflash logs. The MAVlink wireless telemetry communication protocol is used to transfer flight data between the on-board flight controller and the ground station computer. Without telemetry devices, we cannot communicate in real-time with our flying quadcopter as well as transferring flight data. The “tlogs” file is automatically created and stored in the Mission Planner’s logs folder. Flight recording will start from connecting autopilot until disconnected from the ground control software. Beside that, this second method has the ability to record all flight data throughout the mission with desired stream rates without limited memory, as long as our computer has a memory space allocation. Furthermore, several advantages of using telemetry method are; recorded data that can be easily played back flight missions, able to convert the “tlogs” file to the KMZ file format, and also directly generate graphs of any flight parameters. All of these can be done from the ground control software. Therefore, we have used the second method to collect flight data throughout these experiments for analysis purpose.

Moreover, all experiments were conducted without adding any payload and did not take into account of the current environmental situation, such as wind gust conditions.

5.1 Quadcopter flight performance

The great advantages of Mission Planner software are their capabilities to generate an automated report in the form of visual graph from both recorded datalogs (dataflash and telemetry), in order to diagnose any problems or to evaluate quadcopter performances. From this quadcopter experiment, we present the three commonly flight data analyses that have been used to evaluate their performance, which are; quadcopter’s attitudes and altitude stability, magnetometer interference effect, and quadcopter’s vibrations.

We have applied manually setting into PID parameters in the interest of achieving the optimal tuning of our quadcopter PID controller instead of Auto Tune. However, these parameters are very depending on our mechanical design. For example, by changing the on-board battery

positioning will be affected the quadcopter performance. The applied tuning PID parameters stated as follows;

Roll tuning parameters

Stabilize Roll = 6.500, P = 0.080, I = 0.100 and D = 0.003

Pitch tuning parameters

Stabilize Pitch = 6.500, P = 0.080, I = 0.100 and D = 0.004

Yaw tuning parameters

Stabilize Yaw = 4.500, P = 0.200, I = 0.020 and D = 0.000

5.1.1 Quadcopter's attitudes and altitude stability

An emphasis of quadcopter performance analysis, especially for attitude and altitude is very important to ensure quadcopter stability in an uncertain outdoor flying environment. Attitude controller quadcopter comprise of stabilization of three Euler angles (roll, pitch and yaw). Figure 3 shows the result of attitude stability of our quadcopter in autonomous flight in three linear waypoints. It represents time history of three angles and has represented in different colours. From that figure, roll and yaw control are more stable along the flight, but for pitch control, it looks gradually increases while leaving from each waypoint. However, from our experimental, this result is one of the best after various changing in PID control parameter.

Altitude is measured using either mean sea level (MSL) or local ground level (above ground level, or AGL) as the reference datum. From our experience, the relative altitude above ground is given a more stabile reading compared with altitude above sea level from GPS raw data. It is because of the relative altitude above the ground is the filtered global position (e.g. fused from GPS and accelerometers). In our conducted experimental place, altitude from the sea is about 20 meters and our flight mission is set for 30 meters from the ground level. Figure 4 provides the experimental data on the relative altitude above the ground and altitude error output from navigation controller. It is given the fluctuated ranges below than 0.65 meters after passing the takeoff position. While on the takeoff and land mode, it has given error around 3.5 meters. That means, our flight altitude control still can be acceptable for unpredicted wind gust magnitude in an outdoor environment.

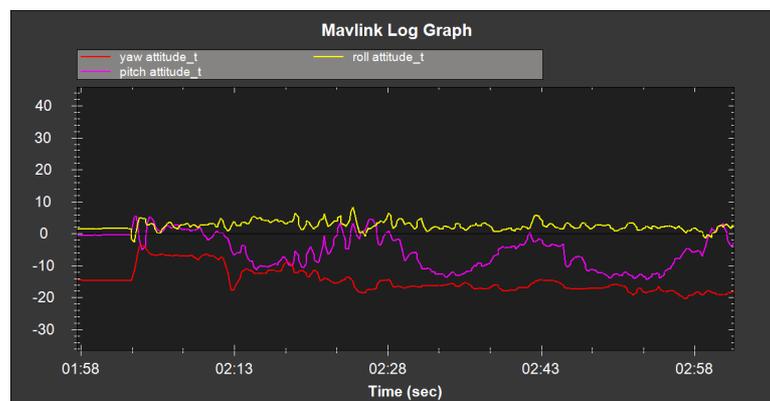


Figure 3: Time history of Euler angles [attitude – roll (yellow), yaw (red) and pitch (magenta)] collected from MAVlink telemetry

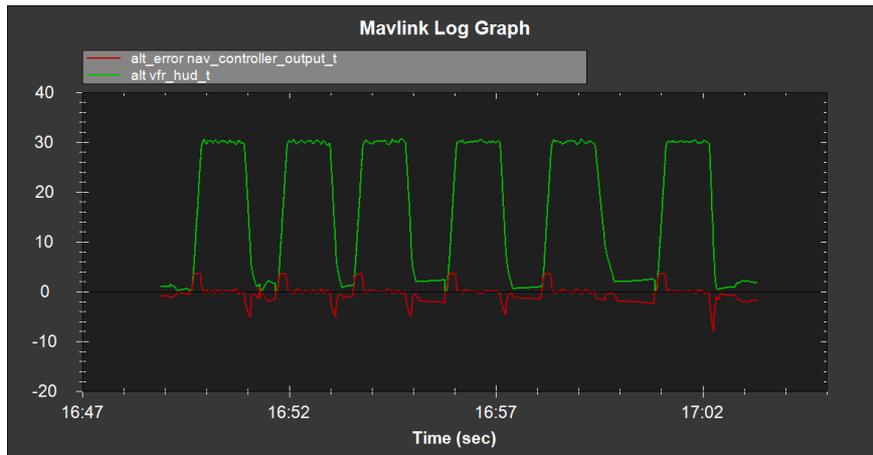


Figure 4: Relative altitude above the ground and altitude error

5.1.2 Magnetometer interference effect

Magnetometer or compass sensor is quite sensitive with magnetic field interference. Compass interference could be from the power distribution board, motors, battery, ESC, and other electrical devices near with the flight controller. It will cause interference on a compass, then giving the wrong direction. Because of that, we decided early in this project to not use the embedded IMU internal compass and prefer to utilize an external compass.

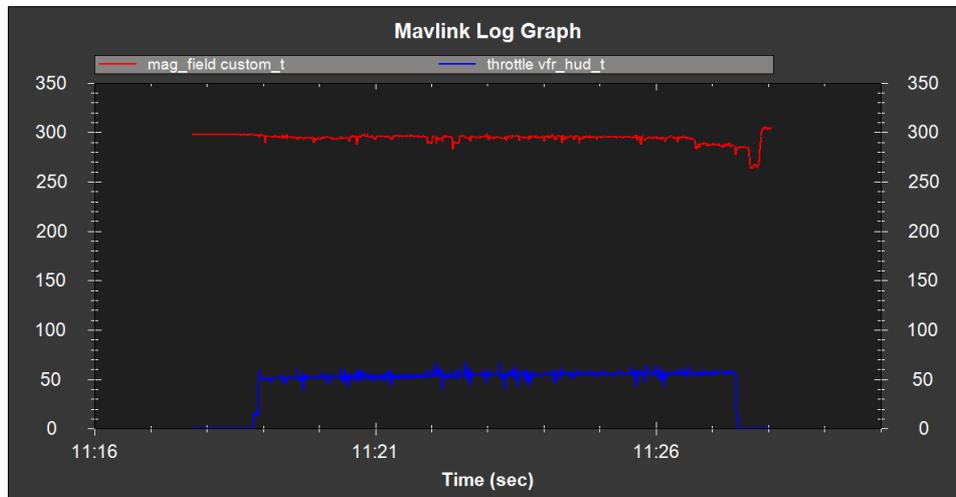


Figure 5: Magnetic field effect on quadcopter compass

Figure 5 shows the graph of magnetic field (red) and percentage of throttle (blue). The magnetic interference value (`mag_field`) will start to fluctuate when the throttle is raised. An acceptable amount of magnetic interference should be less than 30% of normal value. In the place of our conducted experiment, the normal magnetic interference reading is 297 milligauss. The results of this experiment show that the magnetic field only fluctuated between 10% and 20% and this result is acceptable.

5.1.3 Quadcopter's vibration results

High vibrations on the flight controller caused our quadcopter accelerometer based altitude and horizontal position estimates to drift far off from reality. Thus, it will create problems with altitude hold or loiter. Vibrations are best viewed by graphing the dataflash's IMU message's AccX, AccY and AccZ values because dataflash logs provide higher recording rate up to 50Hz. However, telemetry tlog's RAW_IMU x_{acc} , y_{acc} and z_{acc} can also be used to examine the vibration. The acceptable range for x_{acc} and y_{acc} is $-300 \sim +300$, and for z_{acc} is $-500 \sim -1500$.

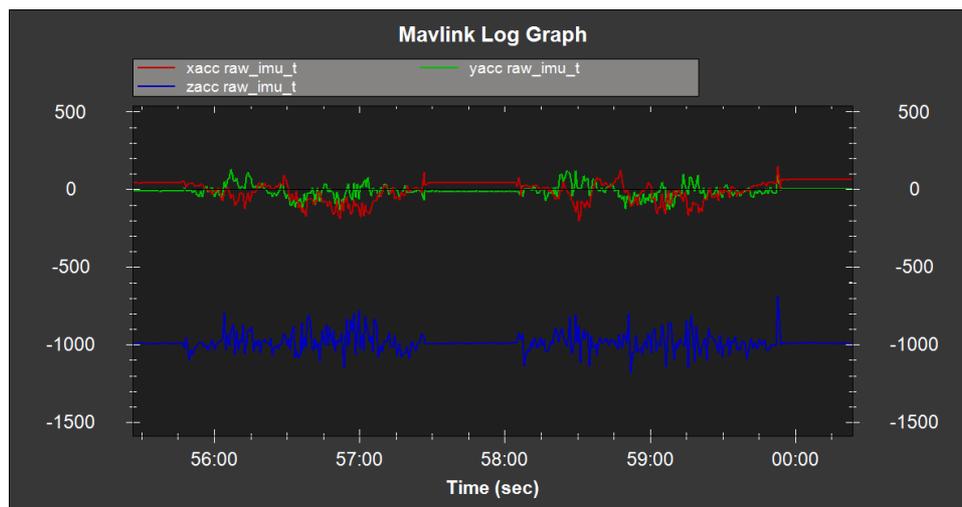


Figure 6: Horizontal and Vertical IMU raw acceleration graph

In this project, Figure 6 shows the graph of raw IMU data [i.e. horizontal; x_{acc} (red) and y_{acc} (green), vertical; z_{acc} (blue)] obtained from the MAVlink telemetry logs during AUTO mode flight. From the graph, we can see that both horizontal and vertical vibrations are within the acceptable range as mentioned above. Interestingly, the vibration ranges still in the acceptable range in all our conducted experiments. These findings suggest that by mounting the flight controller on the top of the Anti-vibration Plate on the quadcopter frame will reduce vibrations in IMU sensors. But, balancing propellers and motors are also very important to get better results.

5.2 Multi waypoints autonomous flight mission

In this second experiment, quadcopter flight routes need to create beforehand by using the mission planner software. Figure 7 illustrates our mission planning and events setting in this experiment. We decide to use different position for home and land. Five waypoints were selected randomly with additional 1 second delay that was added in WP1 to WP4. The flight altitude is set at 30 m height throughout the missions. The waypoint navigation speed setting are; linear speed is 700 cm/s, radius speed 200 cm/s, speed up rate is 250 cm/s, speed down rate is 150 cm/s and loiter speed is 1000 cm/s. We run this flight mission for six times.

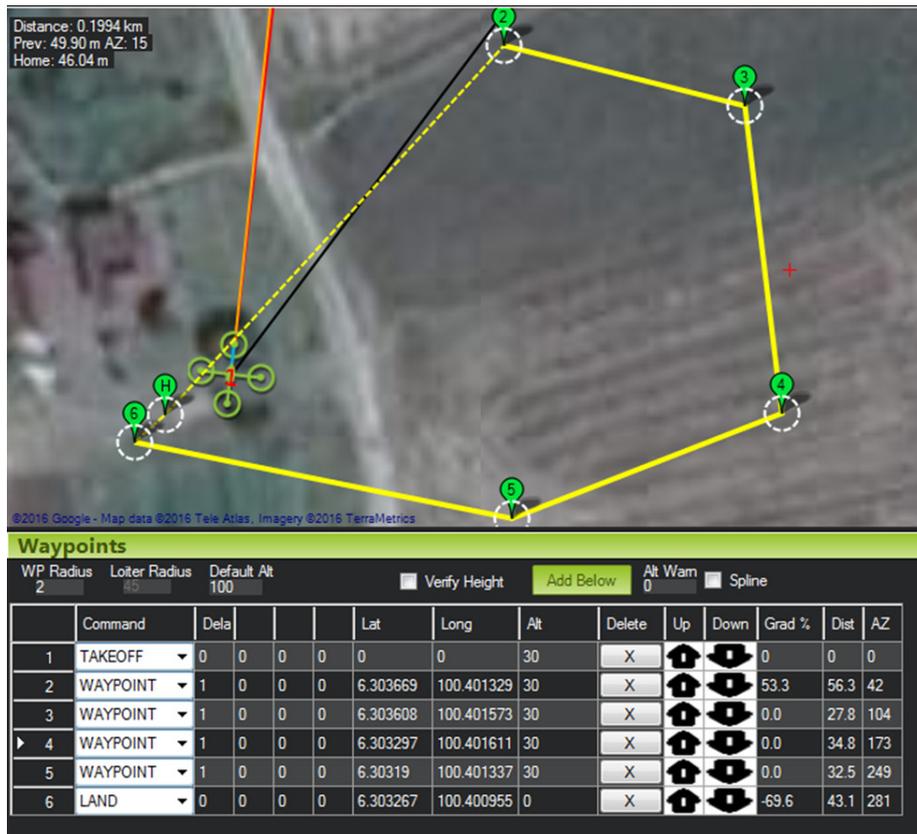


Figure 7: Flight Plan. Waypoints. 1 second delay was added for each Waypoint

In this experiment, the aim was to generate 3D mapping flight trajectory from telemetry logs. After complete six flying missions, telemetry logs file can be converted directly into the KMZ file format. A *.kmz and *.kml file will be created alongside the original .tlog and this can be opened in Google Earth (or any EarthViewer 3D) to interactively view the 3D flight path, as shown in Figure 8. In that figure, we present one route instead of six flight route missions conducted in this experiment.

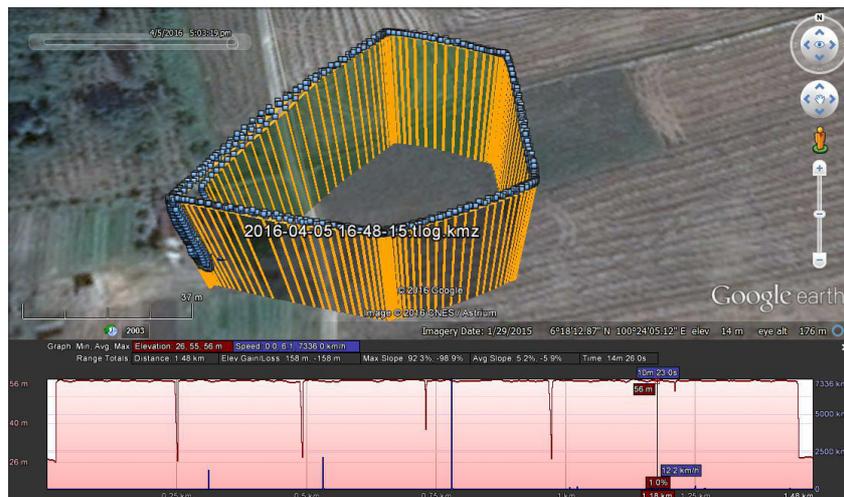


Figure 8: The 3D mapping for autonomous multi waypoint flight

In addition, to examine the flight trajectory accuracy, we extracted longitude and latitude raw data from telemetry data and plotting in the CAD drawing software. As a result, we present the original path (green line) and flight route (blue marks) in Figure 9. Overall, flight route has performed well to follow the original route on each determined waypoint. But, on the way between WP1 and WP2, as an example, we examined some drifting from original route. We quite believe, this happened because of the presence of wind disturbance. This finding provides additional evidence for the effect of wind disturbance that should not be neglected in designing quadcopter controller.

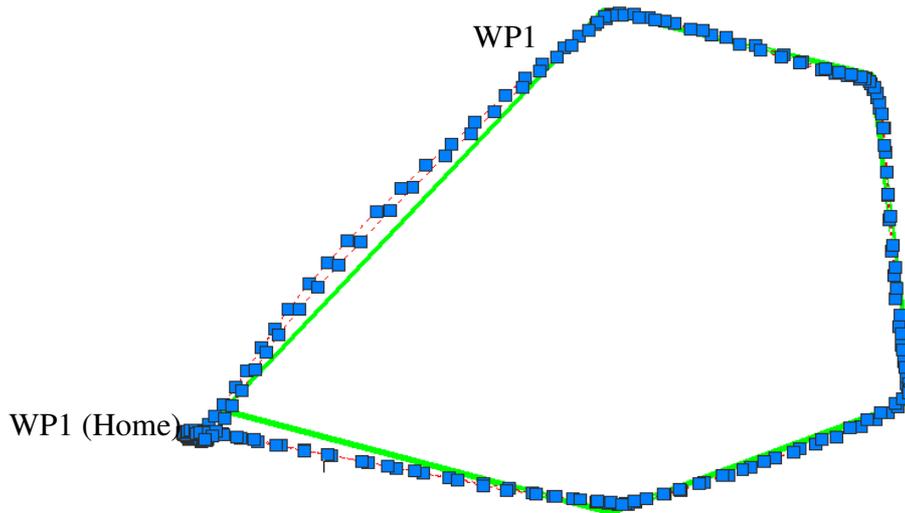


Figure 9: Details 2D trajectory mapping for autonomous multi waypoint flight

6.0 CONCLUSIONS

In this paper, we have present on general view on the implementation of the OSPs quadcopter platform to develop a quadcopter research testbed and should be easily extended to use in any application areas. An example of quadcopter testbed configuration, in terms of hardware and software, which have the capability to perform in autonomous flight mission are also presented. Then, several experiments were conducted in the outdoor real environment in order to investigate their elemental features of flight performances, such as attitude and altitude stabilization, compass interference, vehicle vibration, and multi-waypoint autonomous flight trajectory, also presented here.

As a conclusion, from our experiences and findings in this project, the OPSs quadcopter platform offers complete frameworks in both hardware and software aspects and they convey in designing flexibility to suit our research purpose or any project implementations.

REFERENCES

- [1] Toji, Junichi, Yuya Iwata, and Hiroyuki Ichihara. "Building quadrotors with Arduino for indoor environments." In Control Conference (ASCC), 2015 10th Asian, pp. 1-6. IEEE, 2015.
- [2] Dong, Wei, Guo-Ying Gu, Xiangyang Zhu, and Han Ding. "Development of a quadrotor test bed--modelling, parameter identification, controller design and trajectory generation." *International Journal of Advanced Robotic Systems* 12 (2015).
- [3] Davis, Edwin, Ben E. Nizette, and Changbin Yu. "Development of a low cost quadrotor platform for swarm experiments." In Control Conference (CCC), 2013 32nd Chinese, pp. 7072-7077. IEEE, 2013.
- [4] Waharte, Sonia, and Niki Trigoni. "Supporting search and rescue operations with UAVs." In Emerging Security Technologies (EST), 2010 International Conference on, pp. 142-147. IEEE, 2010.
- [5] Abdolhosseini, Mahyar, Y. M. Zhang, and Camille Alain Rabbath. "An efficient model predictive control scheme for an unmanned quadrotor helicopter." *Journal of intelligent & robotic systems* 70, no. 1-4 (2013): 27-38.
- [6] Liu, Chang, and Stephen D. Prior. "Design and implementation of a mini quadrotor control system in GPS denied environments." In Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, pp. 462-469. IEEE, 2015.
- [7] Hoffmann, Gabe, Dev Gorur Rajnarayan, Steven L. Waslander, David Dostal, Jung Soon Jang, and Claire J. Tomlin. "The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)." In Digital Avionics Systems Conference, 2004. DASC 04. The 23rd, vol. 2, pp. 12-E. IEEE, 2004.
- [8] Castillo, Pedro, Alejandro Dzul, and Rogelio Lozano. "Real-time stabilization and tracking of a four-rotor mini rotorcraft." *IEEE Transactions on control systems technology* 12, no. 4 (2004): 510-516.
- [9] Lim, Hyon, Jaemann Park, Daewon Lee, and H. Jin Kim. "Build your own quadrotor: Open-source projects on unmanned aerial vehicles." *IEEE Robotics & Automation Magazine* 19, no. 3 (2012): 33-45.
- [10] Allen, Ross, and Marco Pavone. "A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance." In AIAA Conf. on Guidance, Navigation and Control, San Diego, CA. 2016.
- [11] Gabriel, Darren Lance, Johan Meyer, and Francois Du Plessis. "Brushless DC motor characterisation and selection for a fixed wing UAV." In IEEE Africon Conference, pp. 978-1. 2011.
- [12] Gupte, Shweta, Paul Infant Teenu Mohandas, and James M. Conrad. "A survey of quadrotor unmanned aerial vehicles." In Southeastcon, 2012 Proceedings of IEEE, pp. 1-6. IEEE, 2012.
- [13] Roth, Stephan, and Sanjiv Singh. "Application of robust, high-accuracy positioning for autonomous ground vehicles." (2004).