Design and Development of ReCOPTER: An Open source ROS-based Multi-rotor Platform for Research

Dinuka Abeywardena [1*], Paul Pounds [2], David Hunt [1] and Gamini Dissanayake [1]
[1] Centre for Autonomous Systems, University of Technology, Sydney, Australia
[2] Robotics Design Lab, University of Queensland, Brisbane, Australia
{Dinuka.Abeywardena, David.Hunt, Gamini.Dissanayake}@uts.edu.au
Paul.Pounds@uq.edu.au

Abstract

Selection of multi-rotor aircraft systems for robotics research is a trade-off between competing objectives. While Commercial Off The Shelf systems are fast to set up and provide a ready-made platform, they often lack complete documentation and have limited extensibility for allowing researchers to modify them for scientific work. Conversely, developing an aircraft from the ground up is labour intensive and time consuming, and requires substantial experience to ensure a satisfactory result. This paper ranks common robotic multi-rotor aircraft used in research against several criteria for openness, extensibility and performance. We propose a standard platform using open components and an open-source design, specifically geared to the needs of the research community.

1 Introduction

Multi-rotor Unmanned Aerial Vehicles (UAV) have gained significant popularity among both the research and hobbyist communities due to their manoeuvrability, simplicity and low cost. Research conducted using multi-rotor aerial vehicles belong to two main categories: research that advances the abstract capabilities of the platform itself, and research that employs those capabilities to perform a specific task. Examples of first category include developing multi-rotors capable of aggressive flying and perching [Mellinger *et al.*, 2012], grasping [Mellinger *et al.*, 2013], robustness to wind disturbances [Waslander and Wang, 2009] and many instances of novel state estimation algorithms [Abeywardena *et al.*, 2013], [Mahony *et al.*, 2012]. Examples of the second category include the use of multi-rotor aerial vehicles for indoor exploration [Achtelik *et al.*, 2008], infrastructure inspection [Sa and Corke, 2014] and robotic construction [Lindsey *et al.*, 2011].

The principal features of a UAV are its structure, propulsion, stability and guidance [Pounds and Singh, 2013] — all of which are essential for conducting research using a multi-rotor platform. While the specific hardware and software needs may vary depending on the exact research questions being addressed, it is possible to identify a common set of desirable features for multi-rotor aerial vehicles, if they are to be useful as research platforms. These features include open-source and easily extensible hardware and software, low cost, minimum set-up time, ease of repair and reasonable flight time.

Figure 1: Proposed open-source multi-copter with autopilot, on-board computer and smart camera.

There are two options available to a robotic researcher wanting to employ a multi-rotor aircraft as a research platform. First is to purchase one of many Commercial Off The Shelf (COTS) aircraft available in the market today. However, only a few of them, such as the research line of multi-rotor UAVs from Ascending Technologies GmbH, have been designed with the needs of robotics researchers in mind. Most COTS multi-rotor platforms focus on ease of use and payload characteristics, with little or no attention to features such as openness and extensibility. The second option is to design and build multi-rotor platforms in-house to suit the specific needs of the research being conducted, either from scratch or from using COTS parts. While this approach is appealing due the flexibility it offers it is also time consuming and expensive due to the number of iterations and prototypes required to perfect the design.

Our goal is to assist the robotic researcher with both of the above options. Specifically, we aim to provide a framework for multi-rotor platform selection, and a blueprint for an open-source alternative platform specifically targeting the needs of the robotics research community. This is achieved through two key contributions that are detailed in this paper. First, we identify a set of evaluation criteria based on the requirement of aerial vehicle researchers and then use the said criteria to compare three of the most commonly used COTS multi-rotor research platforms. Second, we use the same evaluation criteria as design specification to propose a multi-rotor configuration that can be constructed in a straightforward manner using easily sourced hardware and software components. We have built and tested several multi-rotor platforms according to this configuration and report on one such instantiation so that other researchers are able to use the same configuration to assemble platforms according to their specific needs, while minimising the design iterations and costs. Whenever possible, we employ open-source hardware and software components to ensure that other researchers interested in building in-house multi-rotor platforms can replicate our designs with minimum effort.

This paper is organized as follows: In section 2 we present our evaluation criteria and in section 3 we use this criteria to compare three of the most common COTS multi-rotor research platforms. In section 4 we propose our novel configuration based on the same evaluation criteria. Section 5 provides details on experimental flights of a multi-rotor platform constructed according to the proposed design and section 6 details an example application of the platform to demonstrate the utility of openness and extensibility of the design. Finally section 7 concludes the paper with a summary of the contributions and future research directions.

2 Evaluation Criteria

The key utility aspects of a multi-rotor unmanned aerial system can be decomposed into:

- Platform performance
- Economics and logistics
- Openness and extensibility.

This section details the importance of these aspects for research. Note however that the criteria identified here are by no means an exhaustive set. Depending on the specific sub domain of multi-rotor research, there may be other features that are also important; we aim to identify a generic set of criteria that are relevant to a majority of multi-rotor researchers.

2.1 Size, payload, flight time and cost

Flight experiments for multi-rotor research are predominantly conducted in an indoor lab environment with limited flight space. Safe operation with sufficient actuation inside small flight spaces requires small aerial vehicles. Therefore, we postulate that smaller multi-rotor aerial vehicles have a higher utility for conducting research unless the requirement is to specifically analyse the dynamics of large UAV platforms [Pounds and Mahony, 2009].

For most forms of aerial robotic research, the vehicle needs to carry additional sensing, computational or actuation payloads. We define payload as the maximum weight of the components removable from the aerial vehicle while retaining the ability to fly in a useful manner. Strictly speaking, the payload capacity of a given aerial vehicle is only dependent on the maximum thrust it can produce. However, maximum payload capacity can be a misleading indication of the capability of an aerial vehicle; it is necessary to analyse the inverse relationship between the payload and the flight time. To simplify the analysis, here we chose the flight time at a given anticipated payload as our evaluation criteria.

Considering the sensing and computational payloads commonly used in multi-rotor research, we define two payload categories as representative. For research involving inertial or monocular vision sensing we define a payload class of 0.1 kg, which could include a small embedded computer and storage for data logging in addition to the sensors. For research involving other bulkier sensing modalities such as LiDAR or depth cameras we define a payload class of 0.5 kg. The evaluation criteria are thus the flight time at 0.1 kg and 0.5 kg payloads.

Cost of the aerial platform is obviously a key concern for researchers with limited resources. To enable a fair comparison of the different COTS platforms, cost of the not just the aerial platform itself, but also the complete system including the radio transmitter, receiver, telemetry, spare battery and a reasonable set of other spare parts should be considered[1].

2.2 Open and extensible hardware and software

Conducting multi-rotor research often requires modification or extensions to the hardware and software components of the aerial vehicle. For example, evaluating novel control or estimation algorithms require software modifications but adding new sensors require both hardware and software extensions. An effective way to facilitate such modifications is to only employ open source hardware and software components. Software has a long

[1]While dependent on application, we consider a reasonable set of spare parts to include a pair of extra motors, a pair of ESCs and a replacement set of propellers

Figure 2: Common platforms: (a) Parrot AR Drone 2.0 (b) Ascending Technologies Hummingbird (c) 3D Robotics Iris+.

culture of open source licensing that enables a third parties to use and modify it for non-commercial purposes. More recently, hardware source files have also been published under permissive licenses, including frame components and also the schematics and board files of electronic modules being used. While silicon designs of integrated circuits and other discrete components could also be available under open source licenses, the resources required to fabricate them put them beyond the reach of most researchers. Instead, their utility is measured by the availability of technical data sheets and APIs that enable a researcher to exploit their functionality.

Openness of the hardware and software does not guarantee the ability for a third party to extend and modify the platform to suit their needs. For example, some platforms may provide the source code of their control algorithms but not an interface to reprogram the autopilot executing the said code. Some other platforms may be accompanied with the design files for the frame components but might not have the necessary modularity to modify a specific part of the frame. Also related to extensibility is the availability of sufficient documentation for both the hardware and software.

To facilitate a quick comparison between different COTS platforms, we grade their openness and extensibility in hardware and software separately on a scale of 0-5, where 5 is the most open or extensible. We recognise that grading openness and extensibility is inherently subjective. We provide information on specific components of each platform that are not open-sourced or not easily extensible in the following section.

3 Comparison of COTS multi-rotor research platforms

Numerous COTS multi-rotor platforms have been used by various aerial vehicle research groups around the world. Of these, we selected two of the most common platforms for comparison, based on an informal survey of the 100 most-cited multi-rotor research papers published between 2010 and 2015 on IEEE Xplore: the AR Drone by Parrot Inc. and the Hummingbird by Ascending Technologies GmbH. We also selected the Iris+ by 3D Robotics Inc. which features many characteristics required of a multi-rotor research platform. Even though the Iris+ has not been popular among robotics researchers, its Pixhawk autopilot is extremely popular and including Iris+ in the comparison enables us to broaden the spectrum of multi-rotor features being evaluated.

Table 1 presents an overview of the COTS platforms features, along with the open source design proposed in section 4. A detailed discussion of these features are presented next.

3.1 AR Drone

AR Drone[2] is a relatively inexpensive quadrotor platform intended as an easy to use toy aerial vehicle targeting the hobbyist community (see Fig. 2 a). Its tip-to-tip size is 373 mm and weight is 0.455kg. It has a hover flight time of about 9 minutes with 0.1kg of payload but is unable to take-off with a 0.5kg payload. The total cost for a the system is about 500USD.

The AR Drone is pre-equipped with two low resolution cameras and a downward pointing sonar in addition to the standard IMU sensor package. The auto-pilot and sensor driver binaries exist on an on-board Linux computer that can execute custom binary files as well. However, the source code for the auto-pilot and sensor drivers are closed source and therefore accessing sensor measurements or modifying the auto-pilot behaviour is not trivial. AR Drone makes the sensor data available on a separate Ground Station Computer (GSC) via an API that also enables sending high-level navigation commands to the auto-pilot. However, the data connection between the GSC and the AR Drone is wireless (WiFi, IEEE 802.11b/n) and the limited range and delay associated with such networks limits the usefulness of the API for robotic research where real-time sensing and control is a requirement. Design files for neither the frame components nor the avionics are open-source so extending

[2]There are two available versions of the AR Drone and here we refer to AR Drone 2.0

Evaluation Criteria	AR Drone	Hummingbird	Iris+	Proposed
Total mass (kg)	0.420	0.6	1.4	1.03
Tip-to-tip Size (mm)	373	360	526	500
Flight time @0.1kg (minutes)	9	20	15	30
Flight time @0.5kg (minutes)	-	-	7	20
Cost (USD)	500	5000	1000	1000
Hardware/Software openness	1/2	2/3	4/5	5/5
Hardware/Software extensibility	1/3	3/2	3/3	5/5
On-board camera	720p @ 30Hz	-	-	VGA @ 60Hz
On-board computer	1GHz	-	-	1.7GHz Quad-Core
Indoor stability	Optical flow based	-	-	AR Tag based

Table 1: Comparison of the COTS and proposed multi-rotor platforms. Openness and extensibility measures range from 1 - 5 where 5 is the most open/ extensible.

the functionality of AR Drone is also non-trivial.

Apart from its low cost, the main advantage of using the AR Drone for robotic research is its ability to perform stable indoor flights with minimum user input. This is achieved by combining the optical flow data from the downward pointing camera and the distance to the ground measurements from the sonar to construct a velocity estimate of the platform which is then used in a closed-loop controller.

3.2 Hummingbird

The Hummingbird is part of a research specific line-up of multi-rotor platforms from Ascending Technologies GmbH (see Fig. 2 b). Its tip-to-tip size is 360 mm and weighs 0.6kg. It has a hover flight time of approximately 20 minutes with 0.1kg payload but is unable to carry a 0.5kg payload. Other higher-end Ascending Technologies platforms are advertised as being capable of up to 14 minutes of flight time with 0.6kg payload. The total cost of the full system (with no additional sensors or computing apart from the basic configuration) is approximately 5000USD.

The basic version of the Hummingbird has two on-board 32-bit ARM processors out of which one is reserved for real-time low-level control of the platform. The firmware on the low-level processor is closed source and moreover it cannot be reprogrammed. The high-level processor can be programmed by the user and has access to all in-built sensor data streams and command interfaces of the low-level processor. Adequate documentation on programming the high-level processor and its communication interfaces are provided as well as the ability to add a selected set of new sensors. However, design files for neither the frame components nor the avionics are open-source so adding custom functionality beyond the options provided by the manufacturer is non-trivial. Also, the software architecture adopted for the high-level processor is not that of a modular real-time operating system, thus complicating the implementation of multiple firmware modules on-board the platform.

For on-board implementations of more complex algorithms, the Hummingbird can be equipped with an optional Intel Atom computing board. Various additional sensing modalities such as cameras and LiDARs can be connected and used with the computing board.

3.3 Iris+

Iris+ is a ready-to-fly quadrotor platform by 3D Robotics Inc. featuring the open-source Pixhawk flight controller (see Fig. 2 c). Its tip-to-tip size is 526 mm and weighs 1.4kg. It has a hover flight time of about 15 minutes with 0.1kg payload and about 7 minutes with 0.5kg. The total cost of the full system is about 1000USD.

The key differentiator between the Iris+ and the other COTS platforms discussed above is the openness and extensibility of the Pixhawk auto-pilot hardware and the Pixhawk flight stack. Pixhawk hardware is based on a 168 MHz Cortex-M4F processor and features multiple connectivity options and a full suite of in-built inertial sensors. The hardware is fully open-source (including the schematic and board layout) and multiple additional sensors (also open-source) are supported by default. The Pixhawk processor can be easily programmed with custom firmware but two mature open-source flight stacks are available to free the user from the burden of developing auto-pilot firmware from scratch. These flight stacks — the native Pixhawk flight stack and the APM flight stack — have been tested and verified by thousands of users. The Pixhawk flight stack features a real-time operating system and a modular software architecture [Meier *et al.*, 2015] making it straightforward to extend the functionality of the auto-pilot.

One disadvantage of the Iris+ compared to the AR Drone and the Hummingbird is the lack of a readily

available on-board computer to perform high-level tasks such as sensor fusion, localization and mapping. Even though the Pixhawk and APM flight stacks both support MAVLink [Meier *et al.*, 2013] - a standardized protocol for auto-pilots to communicate with the external world - an off-the-shelf on-board computer that can interface with the Pixhawk via MAVLink protocol is not readily available. Also, even though the Iris+ frame provides many options for mounting and interfacing with other sensors, the design files for the fame components are not available, thus reducing the extensibility of the platform.

4 Proposed Design

It can be seen from Table 1 that none of the commonly used platforms excel in all areas. For this reason, we seek to outline the structure of a multi-rotor configuration particularly targeting aerial vehicle research. We specify the major features of the design including the frame, thrusters, avionics and software.

4.1 Design specifications

We aim for a minimum flight time of 30 minutes with 0.1 kg payload and 20 minutes with 0.5 kg payload. We specify the maximum tip-to-tip size to be 500 mm. The upper limit on total cost is to be 1500 USD. Additionally, we impose a Maximum Take-off Weight (MTOW) of 2 kg, motivated by the impending weight-based Unmanned Aerial Systems (UAS) classification by the Australian regulator, CASA, in which it is proposed to deregulate UAS with a MTOW less than 2kg. Similar classifications are now being adopted or proposed in many other countries including the U.S. and Canada.

Perhaps the most important of the evaluation criteria of section 2 were the openness and extensibility. For the proposed design, we specified that all non-trivial hardware and software components to be open-sourced under a license that is at least as permissive as to allow share and modify for non-commercial purposes. This specified that where possible, all non-trivial off-the-shelf modules be sourced with a license that is at-least as permissive as the Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license and be released to the public without violating their current licenses. This also specified that any hardware or software modules that were developed in-house to be released under the same CC BY-NC-SA license.

Openness is a necessary, but not sufficient, condition for extensibility. Moreover, extensibility is a more difficult criteria to objectify than openness and depends largely on the skill level and experience of the researcher. In general, extensibility stems from modularity and adherence to standards. For the proposed design, we specified that hardware and software be preferably modularized and that all software components should make use of standard packages that are commonly used among robotics researchers.

4.2 Frame, thrusters and battery

The design specifications which were set forth previously called for a tip-to-tip size of 500mm and a minimum flight time of 30 minutes. This is a significant design challenge especially considering the fact that COTS multi-rotor platforms (for both research and commercial purposes) of similar scale have flight times in the range of 10 - 15 minutes. To achieve this specification, we needed to identify the most efficient commercially available motor - propeller - battery combination that is also within the frame design parameters and to evaluate whether that combination was able to provide the required flight time. If not, then a redesigning of one or more components of the motor - propeller - battery combination would be required.

The first question to answer in the thruster selection process is the number propellers to use and in which configuration. The available options are 3 propellers in tricopter configuration [Zou *et al.*, 2012], 4 propellers in a quadcopter configuration [Pounds *et al.*, 2006] or triangular quadrotor configuration [Driessens *et al.*, 2013], 6 propellers in a hexacopter configuration [Baranek and Šolc, 2012] or twin Y configuration [Czyba *et al.*, 2015] and 8 propellers in a octocopter configuration or twin quadcopter configuration. Out of these, we omit the tricopter and triangular quadrotor configurations due to their added mechanical complexity. The 500mm upper limit on tip-to-tip size means that designed multi-rotor should be fully contained within a circle of 500mm radius. The most efficient configuration will be the one which employ most of the area inside that circle for thrust production, with the least amount of propeller overlap. This can be solved by posing it as a variation of the "circle packing problem", with the objective of optimizing the area of non-overlapping, constant size, n number of circles inside a unit circle with $n \in [3, 4, 6, 8]$. Optimum solution is reached with $n = 4$ [Kravitz, 1967], making the quadrotor configuration the most efficient — a hexacopter and octocopter would require 1.5 per cent and 5.5 per cent more power to hover, respectively.

The 500mm tip-to-tip constraint also allows us to identify a suitable size range for propellers. Griffeths and Leishman reasoned that to be free of inter-propeller turbulence, the tip-to-shaft clearance of small scale propellers should be at least $\sqrt{2}r$ where r is the propeller radius [Griffiths and Leishman, 2002]. This constraint limits the available COST propeller diameter sizes to 6, 7, 8 inches[3]. At this size range, both Carbon Fiber (CF) and plastic propellers are commonly available in

[3] 8″ propellers results in slightly larger tip-to-tip dimension than 500mm.

the market. We decided to make use of the CF propellers as their stiffness lends to added efficiency due to less propeller twist under load. We selected CF propellers from T-Motor to evaluate, in three different sizes: $6 \times 2''$, $7 \times 2.4''$ and $8 \times 2.7''$.

Given the requirement for a quadcopter configuration with a MTOW of 2kg, each propeller should produce at least 4.9N of thrust. This thrust requirement, coupled with the selected propellers, limits the number of available COTS motors options. We identified two T-Motor motors that are able to match the thrust requirement with at least one of the 6, 7, 8 inch propellers: the MN2206 and MT2208, with K_V values of 2000 rpm/V and 1100 rpm/V respectively. Motors with significantly higher K_V values than that have insufficient torque to be able to drive the selected propellers. Those with significantly lower K_V values are unable to maintain sufficiently speed for the required thrust.

There are two suitable battery technologies available for the energy density and discharge rate requirements of multi-rotors: Lithium-Polymer and Lithium-Ion. Lithium-Polymer batteries have less energy density than Lithium-Ion batteries but are the most popular in the multi-rotor community because of their higher discharge rates. A new category of Lithium-Ion batteries that have discharge rates on par with Lithium-Polymer batteries have been introduced to the market by Samsung and LG. These include the 2.5Ah Samsung INR18650-25R which is rated for 22A continuous discharge current and the 3Ah LG 18650HG2 which is rated for 20A continuous discharge current with energy densities of 209WHr/kg and 234WHr/kg respectively.[4]

The 3Ah LG 18650HG2 is proposed for the configuration due to its higher energy density. These batteries come in individual cells, each with a cell voltage of 3.6v, and need to be stacked together to be able to drive the selected motor-rotor combinations. We assessed the suitability of several different battery configurations to identify the most suitable for each motor-rotor pair. A given battery configuration is denoted here by the xSyP notation, which indicates a battery pack consisting of x individual cells in series and y number of such configurations in parallel, for a total of $x \times y$ cells in the pack.

We assessed several combinations of the three rotors and two motors considered, with different combinations of battery arrangements. Much of the data required for this empirical analysis are readily available from the manufacturer websites. Here we combine that data with other parameters of our design and present them in a unique way that facilitates an informed selection of the motor, propeller and battery combination capable of providing best flight times at different payloads. The results of the analysis are illustrated in Fig. 3. Details of how the results were generated are given in Appendix A.

[4]In comparison, the Thunder Power TP2100-3SP+25, which is a high energy density Lithium-Polymer battery rated for 52.5A continuous current and 2.1Ah capacity has an energy density of 156WHr/kg

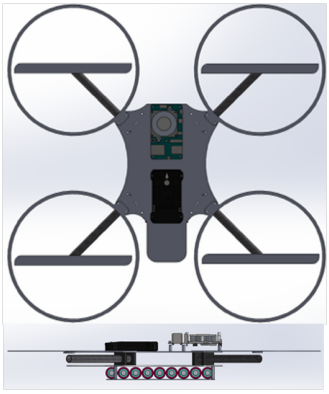

Figure 4: Proposed frame design. Top: top view. Bottom: side view.

Fig. 3 illustrates that out of the 11 configurations analysed, only the MT2208 motor equipped with a $8 \times 2.7''$ propeller and powered by a 4S3P battery configuration is capable of exceeding the design specification of 30 and 20 minutes flight times at 0.1kg and 0.5kg payloads, respectively. However, the $8 \times 2.7''$ propellers were not available for purchase at the time and therefore, the next configuration that was closest to the design specification needed to be chosen. Both the 3S3P-7 inch (2206) and the 3S3P-6 inch (2206) configurations had performances slightly lower than the design specifications. Out of these two, the 3S3P-7 inch (2206) configuration was chosen as it had about twice the payload capacity as the 3S3P-6 inch (2206) configuration.

4.3 Frame Design

The selected motor-propeller-battery combination, rotor configuration and the tip-to-tip size constraint determines the key physical parameters of the frame design. Other main considerations included the ease of sourcing raw material, ease of assembly and repairability. Considering these, we opted for a simple frame design which makes use of ubiquitous Carbon Fiber (CF) tubes and sheets, as illustrated in Fig. 4. The most complex process of the build is machining of the CF plates to the required shapes, which can be performed using a conventional CNC router. The design files for the frame are released on-line under CC BY-NC-SA.[5]

4.4 Avionics

Pixhawk auto-pilot was chosen for the proposed design to satisfy the openness and extensibility design specifi-

[5]github.com/thedinuka/ReCOPTER

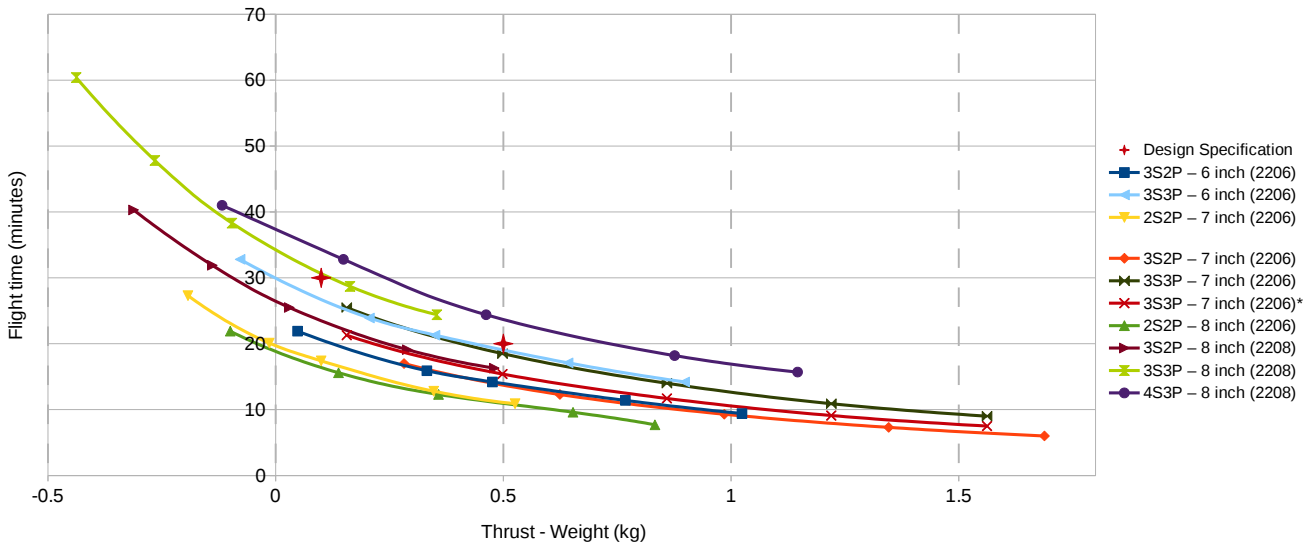

Figure 3: Payload versus flight time curves. Each curve is labelled using <battery_configuration>-<propeller_diameter> (<motor_used>) notation. 3Ah single cell capacity was assumed for each curve except for the one marked with $*$ for which 2.5Ah capacity was assumed. X axis represents the thrust (in kg) over and above that which is required for the given configuration to hover. Hover flight time is given by the Y axis intercept of each curve. The right most data point of each curve represents the maximum thrust producible from that configuration.

cation. Schematics and board files of the Pixhawk auto-pilot hardware are available on-line[6], licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0).

To improve the hardware and software extensibility, we also incorporated an on-board computer to the proposed design. There are many COTS single board computing platforms available to chose from and considering the power, weight and ease of use, the Odroid U3 by Hardkernel was chosen as the on-board computer. Odroid U3 features a Cortex-A9 Quad-core processor with 2GB main memory and a host of i/o interfaces. Although the actual schematics and board files of the U3 are not provided, documentation diagrams in PDF and JPEG format are available online.

To be useful, the on-board computer needs to communicate with the auto-pilot to receive sensor data and to issue control commands. It is possible for the Odroid U3 to communicate with the Pixhawk via the USB interface. However, communications over USB interfaces incur significant and variable delays that can adversely affect the performance of real-time estimation and control tasks performed on the on-board computer. For this reason, it was decided to make use of the hardware UART interfaces of the U3 and the Pixhawk for communicating between them. To achieve the required voltage level conversion between the two devices a separate PCB was designed, the schematics and board files of which are released under the CC BY-NC-SA license. Apart from the level conversion, this board was also used to generate a hardware trigger for the on-board camera.

The proposed design also includes an on-board machine vision camera to facilitate computer vision related research using multi-rotor platforms. In choosing a specific camera model, we focused on three main features; high frame rate, global shutter and external trigger capability. These features are essential for accurate synchronization of the camera images with IMU measurements originating from the Pixhawk. The model selected for the on-board camera was the FMVU-03MTM-CS by Point Grey. This camera features a USB 2 interface for image transfer, a monocular 0.3 Mega pixel sensor and global shutter. The camera supports a maximum frame rate of 60fps, but this maximum can only be reached in continuous trigger mode. The speed at which the camera can be externally triggered depends on the exposure time and through experimentation it was found that a frame rate of about 50fps can be achieved for external triggering while maintaining a suitable exposure for an indoor scene.

Avionics also includes the Electronic Speed Controllers (ESC) required for controlling the BLDC motors. Again, there are many available COTS ESC options and in choosing one we focused on the continuous current rating and the PWM refresh rate. Continuous current rating of the ESC should be above the maximum current drain supported by the motor and as a rule of thumb the

[6]https://github.com/PX4/Hardware

PWM refresh rate should be at least 400Hz. For the proposed design we have chosen the Hobywing XRotor-20A ESC but there are many others that satisfy there criteria. However, it should be noted that the hardware for most COTS ESCs, including the XRotor-20A, are not open source. Usually their firmware is also not open source but some can be programmed with open source ESC firmware (for example see https://github.com/sim-/tgy). If fully open source ESCs are required, then one can make use of the Pixhawk ESC, which is planned to be made available commercially towards the end of 2015.

4.5 Software

The software for the propose design consists of two main components: the firmware of the auto-pilot and the software for the on-board computer. In this section, we discuss the available choices given the selected hardware and make use of the design specifications to identify the most appropriate option for the proposed design.

There are two options available for the auto-pilot firmware of the Pixhawk: the APM flight stack based on the ArduPilot code base and the native Pixhawk flight stack. Out of the two options, Pixhawk flight stack is more modular and robust as it is based on a real-time operating system known as NuttX. Actual auto-pilot code is implemented as modules that communicate with each other through a subscriber-publisher architecture. This abstracts away the specific details of reading from or writing to of information to peripheral devices such sensors and actuators. A researcher with the need to extend the functionality of the auto-pilot - for example with a new sensing or control algorithm - needs only to subscribe to the topics that carry the required input information and to publish the results to the relevant topics without having to know in detail how the input information is captured or how the output data is made use by other modules. Due to these benefits the Pixhawk flight stack was chosen for the proposed design. Pixhawk flight stack also supports MAVLink, which is a standardized protocol for auto-pilots to communicate with the external world. In this case MAVLink is used to communicate both with the ground station computer and the on-board computer.

The Pixhawk hardware has limited computing power and is not suitable for more high-level sensor fusion, control and planning tasks that are of interest to robotic researchers. Such tasks should be offloaded to the on-board computer. In-order to perform these tasks, the on-board computer needs to provide an easily extensible interface to the researchers to receive and process the data from the Pixhawk and to send commands back to it. To achieve this, the proposed design makes use of the Robotic Operating System (ROS) middleware for the Odroid U3. Architecture of ROS is similar to that of the Pixhawk firmware in that it is based on a subscriber-publisher framework which allows individual modules that makes use of data from other modules to be developed easily. The proposed design makes use of MAVROS, which is a standard ROS package to communicate with devices supporting MAVLink, to establish seamless bi-directional communication with the Pixhawk firmware. The proposed design also makes use of the time synchronization feature of MAVROS to synchronise the Pixhawk and Odroid U3 clocks, thereby enabling precisely timestamped sensor and command messages. With the MAVROS-MAVLink connection and with the time synchronization, the distinction between the firmware modules on Pixhawk and ROS packages on the Odroid U3 disappears, allowing the robotics research to implement their algorithms on the much familiar ROS framework instead of on the Pixhawk framework with same level of performance for all but the strict delay sensitive tasks.

The source code of the core components of ROS including the MAVROS package and the entire Pixhawk flight stack are available under the BSD license. Other third party ROS packages may have different licenses.

5 Final Specifications and Flight tests

A quadcopter platform incorporating all of the components of the proposed design is illustrated in Fig. 1. The key parameters of the platform are presented in Table 2. This platform with different payloads was employed to perform flight tests in an indoor arena to evaluate how it conformed to the flight time estimates presented in Fig. 3. For all such flights, only the hover flight time was measured, with the quadcopter hovering approximately 1.5m above ground. Each flight was initiated with fully charged battery cells at 4.1V cell voltage and terminated when the cell voltage reached 3V, which is the recommended maximum discharge voltage for the batteries used.

Flight tests conducted with $7 \times 2.4''$ propellers and 2.5Ah Lithium-Ion battery in 3S3P configuration demonstrated a 24 minute flight time with 0.1kg payload and 13.5 minute with 0.5kg payload which are in close agreement with the results presented in Fig. 3 (consider the curve for 3S3P-7 inch (2206)*). This indicates that a planned future design iteration with $8 \times 2.7''$ inch propellers and 3Ah batteries in 4S3P configuration would be capable of surpassing the flight time specifications identified in section 4.1.

6 Application Example - Low Cost Motion Capture System

This section details how the openness and extensibility of the proposed design can be used effectively in conducting robotics research, taking as example one of the

Parameter	Value
Total mass (kg)	1.03
Propeller radius (m)	0.1016
Motor flux linkage coefficient	0.00478
Propeller thrust coefficient	0.01079
Propeller torque coefficient	0.00098
Rotational inertia I_{xx} (kgm^2)	0.0426
Rotational inertia I_{yy} (kgm^2)	0.0523
Rotational inertia I_{zz} (kgm^2)	0.0885

Table 2: Frame and aerodynamic parameters of the quadcopter platform constructed according to proposed design.

most common requirements for research involving novel control and estimation techniques in robotics: obtaining ground truth trajectory estimates. We demonstrate how the components of the proposed design can be easily extended to achieve this task without having to resort to expensive external motion tracking systems such as Vicon.

The ground truth trajectory estimation system is based on the open-source an open "ar_track_alvar" ROS package by Scott Niekum which makes use of Augmented Reality (AR) tags affixed rigidly to the environment. This package was installed on the Odroid U3 and was configured to subscribe to the images from the on-board camera, operating at 30fps. One wall of the flight space was affixed with 63 unique AR tags. The 3D coordinates of tags with respect to the centre of the wall was included in an XML file which the ar_track_alvar package uses to estimate the coordinates of the body mounted camera with respect to a coordinate frame positioned at the centre of the said wall. These estimates can either be used to derive ground truth states for evaluating novel state estimation algorithms or can be used in a feedback loop to control the position of the aerial vehicle within the flight volume. To achieve the latter, the trajectory estimates are transmitted via MAVROS-MAVLink to the Pixhawk framework, where they are published as "local position" estimates. A position controller module on the Pixhawk then makes use of those position estimates to control the vehicle position. The flight time experiments described in section 5 were conducted in this manner.

7 Conclusion and Future Work

This paper first identified a set of desirable features of multi-rotor aerial platforms for the robotic researcher and evaluated three commonly used off-the-shelf platforms using those features. Having identified that none of the platforms under consideration excelled in all of the areas, we then proposed an open-source multi-rotor configuration which combines the desirable characteristics of the off-the-shelf platforms. This proposed design can be used as a blueprint by the robotic research community for assembling multi-rotor aerial vehicles with minimum design iterations and prototyping. We also presented details of one such instantiation of the proposed design demonstrating its ability to achieve the specifications set forth for the design.

In our future work we intend to improve proposed design, mainly for increased hardware extensibility and endurance. To increase extensibility, we aim to improve the modularity of the frame design so that the "arms" of the design (including the thrusters) can be easily added, removed or replaced with minimum effort. To increase endurance, we aim to reduce the platform weight and make use of the best propeller-motor-battery combination identified in section 4. Our aim in this regard is to achieve 1 hour flight time with 0.1kg payload while still maintaining the tip-to-tip size below 500mm.

A Appendix - Calculating flight times

This Appendix details how the data for Fig. 3 was generated by combining empirical data available from the propeller, motor, battery manufacturers and the parameters of the proposed design.

Thrust produced by the MN2206 and MT2208 motors with different propellers at several different levels of cell voltages and current draw are available from the manufacturer[7]. However, flight time of a given multi-rotor platform employing these motor-propeller combinations depends on other factors specific to that design such as the frame weight, battery configuration and battery capacity. Importantly, weight of the batteries account for approximately 50% of the platform weight and therefore, different battery configurations will have significantly different weights. To account for these variations, we focused on "additional thrust" instead of the actual thrust for generating data for Fig. 3. Additional thrust T_a was defined as the difference between the actual thrust and the hover thrust and was calculated as:

$$T_a = (T_M N_M) - (m_P + m_M N_M + (N_{Bp} \times N_{Bs}) m_B)$$

where T_M is the thrust generate by each motor (in kg), N_M is the number of motors in the considered configuration ($N_M = 4$ for a quadcopter), m_P is the weight of the platform including the frame and avionics, m_M is the weight of the motor being considered, N_{Bs}, N_{Bp} are the number of battery cells in series and parallel, respectively, and m_B is the weight of a single cell battery.

The flight time t_f for each configuration was calculated using the following equation:

$$t_f = \frac{C_B \eta_B N_{Bp}}{I_M N_M}$$

[7] www.rctigermotor.com

where C_B is the nominal capacity of a single cell battery, η_B is a factor that denotes how much of the battery can be drained without damaging it ($\eta_B = 0.85$ for the Lithium-Ion batteries used) and I_M is the current draw per motor.

References

[Abeywardena *et al.*, 2013] D. Abeywardena, S. Kodagoda, G. Dissanayake, and R. Munasinghe. Improved state estimation in quadrotor mavs: A novel drift-free velocity estimator. *Robotics Automation Magazine, IEEE*, 20(4):32–39, Dec 2013.

[Achtelik *et al.*, 2008] Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, and Nicholas Roy. Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In *Robotics: Science and Systems Conference*, June 2008.

[Baranek and Šolc, 2012] Radek Baranek and František Šolc. Modelling and control of a hexa-copter. In *Carpathian Control Conference (ICCC), 2012 13th International*, pages 19–23. IEEE, 2012.

[Czyba *et al.*, 2015] Roman Czyba, Grzegorz Szafranski, Marcin Janik, Krzysztof Pampuch, and Michal Hecel. Development of co-axial y6-rotor uav-design, mathematical modeling, rapid prototyping and experimental validation. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 1102–1111. IEEE, 2015.

[Driessens *et al.*, 2013] Scott Driessens, Paul E Pounds, et al. Towards a more efficient quadrotor configuration. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1386–1392. IEEE, 2013.

[Griffiths and Leishman, 2002] Daniel A Griffiths and J Gordon Leishman. Astudy of dual-rotor interference and ground effect using a free-vortex wake model. *Circulation*, 2:1, 2002.

[Kravitz, 1967] Sidney Kravitz. Packing cylinders into cylindrical containers. *Mathematics magazine*, pages 65–71, 1967.

[Lindsey *et al.*, 2011] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*, 2011.

[Mahony *et al.*, 2012] R. Mahony, V. Kumar, and P. Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.*, 19(3):20–32, 2012.

[Meier *et al.*, 2013] Lorenz Meier, JF Camacho, B Godbolt, J Goppert, L Heng, M Lizarraga, et al. Mavlink: Micro air vehicle communication protocol. *Online]. Tillgänglig: http://qgroundcontrol.org/mavlink/start.[Hämtad 2014-05-22]*, 2013.

[Meier *et al.*, 2015] L. Meier, D. Honegger, and M. Pollefeys. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, may 2015.

[Mellinger *et al.*, 2012] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, page 0278364911434236, 2012.

[Mellinger *et al.*, 2013] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. Cooperative grasping and transport using multiple quadrotors. In *Distributed autonomous robotic systems*, pages 545–558. Springer, 2013.

[Pounds and Mahony, 2009] Paul Pounds and Robert Mahony. Design principles of large quadrotors for practical applications. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3265–3270. IEEE, 2009.

[Pounds and Singh, 2013] Paul E Pounds and Surya PN Singh. Integrated electro-aeromechanical structures for low-cost, self-deploying environment sensors and disposable uavs. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4459–4466. IEEE, 2013.

[Pounds *et al.*, 2006] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a quad-rotor robot. *Australasian Conference on Robotics and Automation*, December 2006.

[Sa and Corke, 2014] Inkyu Sa and Peter Corke. Vertical infrastructure inspection using a quadcopter and shared autonomy control. In *Field and Service Robotics*, pages 219–232. Springer, 2014.

[Waslander and Wang, 2009] Steven L Waslander and Carlos Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference, Seattle, WA*, 2009.

[Zou *et al.*, 2012] Jie-Tong Zou, Kuo-Lan Su, and Haw Tso. The modeling and implementation of tri-rotor flying robot. *Artificial Life and Robotics*, 17(1):86–91, 2012.