Towards Optimization of Energy Consumption of Drones with Software-Based Flight Analysis

Luis Corral
ITESM / UAQ
Queretaro, Mexico
lrcorralv@itesm.mx

Ilenia Fronza, Nabil El Ioini, Aristea Ibershimi
Free University of Bozen/Bolzano
Bolzano, Italy
{ilenia.fronza, nabil.elioni, aristea.ibershimi}@unibz.it

Abstract—**The capabilities and applications of new drones require a high demand of power that traditional batteries cannot sustain, triggering the need of developing strategies to promote efficient energy usage on drone platforms. However, to reduce the energy consumption it is required to have reliable means to measure the device's behavior and its relationship to the battery discharge. This paper introduces Green Flight, a software system that acquires data during a drone mission, featuring an online battery discharge analyzer and a real-time suggestions mode to help to optimize the overall power consumption of the drone, allowing for longer, more autonomous missions.**

Keywords- airworthiness, battery, drone, energy, power;

I. INTRODUCTION

The fast development and diffusion of the new generation of unmanned aerial vehicles, more popularly known as drones, has brought a number of needs and opportunities for different research paths. The drone industry has been a hot topic for the past several years, and the focus of many discussions in the technology sector in different aspects: applications, handling, management, airworthiness, autonomy, and energy efficiency. In this regard, the capabilities of new generation drones boost a high demand of power that traditional batteries cannot sustain. This situation triggers the need of strategies to promote efficient energy usage to reduce the power demand on drone platforms, with the goal of guaranteeing the capacity of the drone to complete a mission safely. In the past, several research and practitioner work has focused towards designing autonomous, energy efficient targets (for instance, smartphones, tablets, or wearables) [1]; however, there is little research and experience applying these approaches in drone platforms. Moreover, an important need that arises when designing energy aware systems is the ability to understand the way in which the energy is used and spent. This opens the doors to design, implement and validate reliable means to measure the battery discharge for drone targets.

In this paper, we outline a strategy to measure the energy consumption of commercial quad drones. Our strategy includes a systematic data collection of drone missions, and online data analysis to characterize and report out the actions that relate the behavior of the drone to the overall consumption of energy. In this way, users may identify what are the maneuvers, movements and general aspects of the flight profile (load, altitude, speed) that can involve more energy investment, and as a consequence faster battery discharge that leads to shorter drone autonomy. This knowledge can be of utmost importance for strategic actions, such as: identifying and monitoring power-relevant characteristics of drones; proposing metrics to evaluate the energy performance of the drone; and establishing baselines to manage energy consumption and identify energy relevant faults.

The rest of the paper is organized as follows: Section II covers the related work on energy aware software design for non-stationary platforms; Section III shows the project environment and solution approach; Section IV introduces Green Flight, a tool that acquires and reports drone flight data; Section V discusses the outcome of a series of executions in a real drone platform; Section VI sets directions for future work and draws conclusions.

II. RELATED WORK

The analysis of the state of the art in the field of non-stationary devices shows that the existing strategies are mostly based on common, autonomous devices such as smartphones, tablets or wearables. As battery technology and other hardware-based innovations run at a slow pace, a common approach is the design and implementation of software-based energy aware techniques [2]. These techniques include strategies to save energy at operating system level, implementing extensions for economic profiles [3], distributing the computing power to stationary resources [4], and in general, trying to reduce the energy footprint of software executed away from a power-loading source [5].

Many of these strategies have been implemented and validated in targets like smartphones and tablets. Nonetheless, just like smartphones or wearables, drones have to rely on their battery capacity since it represents one of the most important limitations to their operation; similarly, drones have to perform much of their function far from a stationary power-loading source. However, this constraint becomes much more relevant as a key operative attribute is airworthiness, which includes "sufficient power to maintain movement, to implement the controls, and to operate sensors and data-feeds, for the duration of the flight" [6]. Moreover, the potential usefulness of drones in many mission critical applications (such as emergency relief, surveillance, defense) is prevented by the battery life constrain. This limitation is increased in those cases where power resources are needed not only to fly successfully, but also to power on-board facilities such as a built-in videocamera.

DOI reference number: 10.18293/SEKE2016-050

III. Project Environment and Approach

Drones are aircrafts that have no onboard, human pilot. The term "remote pilot" is commonly used because the person handling the device is at a certain distance, but in full control of it. The object of this experimentation is "small drones", since this kind of target is both energetically limited and technologically simple; moreover, small drones are commercially available for any user, so improvements on the aspects of energy consumption and increased autonomy may benefit a large number of users.

Small drones are subject to a range of limitations in relation to load capacity, flight duration (maximum autonomy of about 15 min), power supply, and other technical features. One of the most important limitations is the high power consumption which influences the flight duration of the drone mission. The power demand of drones is satisfied commonly using a battery, but the increasing intake of power and high requirement of autonomy on drones increases the need of a strategy towards efficient power consumption. The operational standard of drones should be aware and apply the most important quality requirement: *airworthiness*. This term is commonly used to refer to an aircraft's suitability for safe flight. A drone is considered airworthy if it is able to take off, conduct its mission and land safely [7]. Said this, *energy efficiency* towards *airworthiness* is the most important quality requirement to design a supporting solution.

A. Proposed approach

The main goal of this work is to design and build a software-driven solution that contributes to improve the energy consumption of drones by establishing tools, measurements and reports. This agenda was based in such way as to complete several tasks that we identified as "critical operations" to assess the energy status of the device, and provide a factual basis for suggesting strategies for energy consumption. These tasks are measurement and health management. As result, we developed a software system that allows to pilot a commercial small drone, which includes a data collection engine and a "real-time suggestions mode" that can be activated in order to optimize the overall power consumption of the drone. The suggestions provided are based on the data collected by the measurement engine, which tracks an approximation of the energy consumed by the execution of each maneuver during the flight, based on the readings of the battery level of the drone.

To achieve this goal, the first step is having the means for measuring the power consumption of the device. To handle this aspect, we designed a software tool which aims to measure with high granularity the power spent by the drone while it is flying, and to characterize it to the actions performed by the drone. The second step is to analyze the data collected acquired from the software tool, aiming to get some prognostics in order to manage the battery health. The third step is to use the analyzed data to build a software application with two main functionalities: (a) piloting the drone and show the current battery percentage as the flight goes on; and (2) activating the *"real-time suggestions mode"* (that is, a battery saving mode) once a critical battery percentage is reached. This mode consists of suggestions for the user in order to take fullest advantage of the remaining flight time.

This set of functionalities meet the critical operations identified above: measurement and health management. In summary, the software system, named "Green Flight", monitors the action of the drone, measures the consumed battery and the overall flight time while the drone is flying, delivers data about how the drone is consuming the energy and permits future analysis in the form of suggestions for the user.

B. Proposed Evaluation and Selected Target

To assess the effectiveness of the system, it is necessary to implement and evaluate the system in a real target, and to conduct a series of experiments that let us understand the impact of Green Flight in the accomplishment of our goal. To this end, we considered two different aspects of evaluation: the first stage of the evaluation should assess the measurement tool created to track battery consumption of each maneuver during the flight, while the second stage evaluated the impact of the "real-time suggestion mode" on the overall battery life.

The drone used for implementing Green Flight and collecting data throughout the necessary experiments is a Parrot Rolling Spider[1]. This drone is designed and commercialized for recreational and entertainment use only. The Rolling Spider is controlled by a smartphone or tablet that supports Bluetooth Low Energy (BLE). This means that the drone does not appear in the usual list of Bluetooth devices in smartphones, but it is only visible and able to connect to it through the host application [8]. The Rolling Spider weights 55 grams, and relies on a 550 mA, 1.1 V removable LiPo battery.

IV. System Development

A. System Design

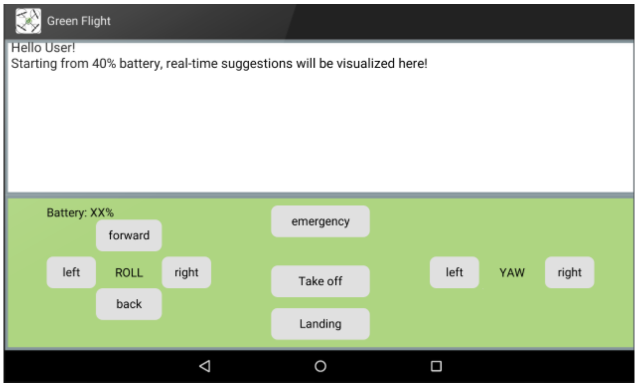
Green Flight is implemented as an Android application which connects to the Parrot Rolling Spider drone via Bluetooth Low Energy (BLE). This application allows piloting the RS drone and keeping track of the drone battery consumption during the flight; moreover, the "real-time suggestions mode" is activated starting from 40% until 7% in order to optimize the overall battery consumption. Suggestions are based on data collected by a measurement tool that we created to track battery consumption of each maneuver during the flight. Green Flight is built using the Parrot ARSDK3[2]. As the topic is still pretty new and innovations are continuously added, the Parrot's software development kit is not fully documented, and only some sample applications are provided on Github, which depend on the type of Parrot Drone and OS of the piloting device. Since the target is a Parrot Rolling Spider drone, Green Flight was built by customizing one of the sample Android applications that let pilot the device. The customizations included functionality to collect data about flights and battery usage.

B. System Implementation

The user interface displaying the piloting part of the Green Flight application is placed in the lower half of the screen of an Android-enabled tablet. In the upper half of the screen, suggestions will be displayed during the flight.

[1] http://www.parrot.com/usa/products/rolling-spider

[2] https://github.com/Parrot-Developers/ARSDKBuildUtils



Green Flight

Hello User!
Starting from 40% battery, real-time suggestions will be visualized here!

Battery: XX%

Figure 1: Green Flight GUI and available features.

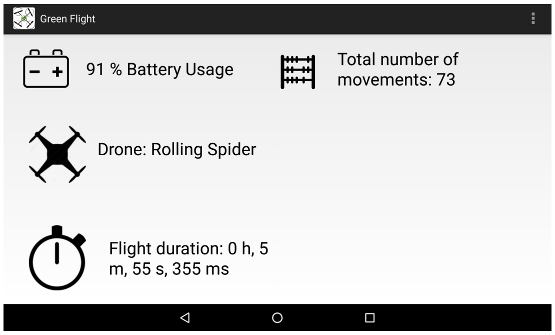
The operations permitted to pilot the drone are: taking off and landing; rolling left, right, forward and backward; and yawing, that is, rotating either to the left or to the right side (Figure 1).

The first customization implemented permits to keep track of the battery consumption and characterizing it to the drone's movements. This customization measures the energy spent by the drone and associates this consumption to the maneuver performed and to its duration. In order to improve accuracy, during the flight, whenever a button (which commands a maneuver) is pressed, the event is logged together with the current battery state and current timestamp. The same happens even when the button is released, which means that the maneuver is completed.

The second customization is the implementation of the "real-time suggestions mode". This feature is inspired by the battery saver mode of smartphones. It is known that, while making use of a smartphone there are certain activities that consume more energy than others: Wi-fi and Bluetooth connection, screen brightness, and Global Positioning System (GPS) are included among the most common battery drainer actions [9]. In order to optimize battery consumption, many manufactures provide the so-called power-saving mode which manages for the user the most power-exhausting features of the smartphone. This mode turns off almost all the features except the necessary ones for making and receiving phone calls and sending text messages. For drones, there is no such a knowledge.

As an effective approach towards drone energy consumption optimization, in this work we proposed a mode where some suggestions of maneuvers are given while the operator is piloting the drone. These suggestions include maneuvers that are still *safe* to perform even with low battery levels (for example: "suggested to land", "suggested to move down, and so on). Maneuvers that are regarded by the system as "energy hungry" are never suggested, as performing them will have a serious impact in the already low battery level.

The knowledge to produce the suggestions is based upon flights analysis (the logs of the drone flight history), whose output identifies the most energy consuming drone maneuvers. To this end, the "real-time suggestions mode" uses the data collected by the first customization, that is, the energy measurement feature.



Green Flight

91 % Battery Usage

Total number of movements: 73

Drone: Rolling Spider

Flight duration: 0 h, 5 m, 55 s, 355 ms

Figure 2: List of drone flights and summary of information.

An additional feature was implemented: "My Piloting". This functionality consists in a list of all the flights performed by the user using a specific device. The content of the event log file is necessary to retrieve information in order to calculate the flight duration, the number of maneuvers performed during a single flight, and the battery usage. With this information, "My Piloting" is populated with a list of flights, and when a flight is selected, the above mentioned information are displayed in detail (Figure 2).

To roll out the application, tests were performed in two different environments: indoors (in an empty classroom) and outdoors (in a cleared University court). The implementation of the Android application permitted obtaining full control of the device and operating it using the features provided in the GUI. By reaching a battery level threshold (40% of remaining battery power), the real-time suggestions are visualized.

V. System Evaluation

Green Flight proved its ability to track in real time all the performed maneuvers, and the number of times they were executed throughout the flight's timespan. Test flights served as a validation of the customizations implemented. The validation of the customizations was successful as: (1) Green Flight allows to pilot the drone properly, (2) the "real-time suggestions mode" is launched and is operational, and (3) at the end of each flight, its details were saved and added to a list of flights accessible through "My Piloting". Additional experiments were run to evaluate the measurement tool created to track battery consumption of each maneuver, and to evaluate the impact of the real-time suggestion mode on the battery life.

A. Evaluation of the measurement tool

This evaluation has conducted 4 experiments of 4 combined movements: up-down, back-forward, roll left-roll right, yaw left-yaw right. All the performed experiments started with fully-charged battery and went on doing the combined movements (e.g., up - down) until the battery percentage went down to 3% (time when the drone automatically shuts down and forces landing). Data were collected as follows: (i) action (pressed/released), (ii) battery percentage (for action pressed and released) and (iii) time (for action pressed and released). Energy consumption was measured 75 times for each maneuver of the combination.

For each flight of the experiments, a log file was created. This evaluation has shown that (1) during all the experiments, the log file was generated successfully and their content was generated while piloting the drone. At the end of the flight, the generated log file was retrieved and further analyzed; (2) the measurement tool did not interfere with the already existent code; (3) Green Flight performed normally without crashing or losing the piloting control. Therefore, Green Flight can be defined as a tool that does not interfere with or diminish the conditions of airworthiness.

B. *Evaluation of the suggestions for optimizations*

After having implemented the "real-time suggestion mode", the next step is to evaluate its impact to the battery consumption and drone's lifetime. In order to perform this evaluation, there were performed four experiments without the suggestion mode of the Green Flight and other four experiments following the provided suggestions of the Green Flight. Considering all 8 experiments performed with and without the suggestion mode of the Green Flight, we present a summary of the experiments without (Table 1), and with suggestion mode (Table 2).

Table 1: Summary of 4 flights without suggestion mode.

Experiment	Duration of Flight
1	5 minutes, 44 seconds
2	5 minutes, 50 seconds
3	6 minutes, 5 seconds
4	6 minutes, 2 seconds

Table 2: Summary of 4 flights with suggestion mode.

Experiment	Duration of Flight
1	5 minutes, 53 seconds
2	5 minutes, 58 seconds
3	6 minutes, 3 seconds
4	6 minutes, 9 seconds

From this dataset, the boxplot in Figure 3 compares the flights with and without suggestions. We conclude that following the suggestions, the mean duration of the flight increases on about 5,6 seconds, showing a slight but noticeable contribution to the overall drone autonomy.

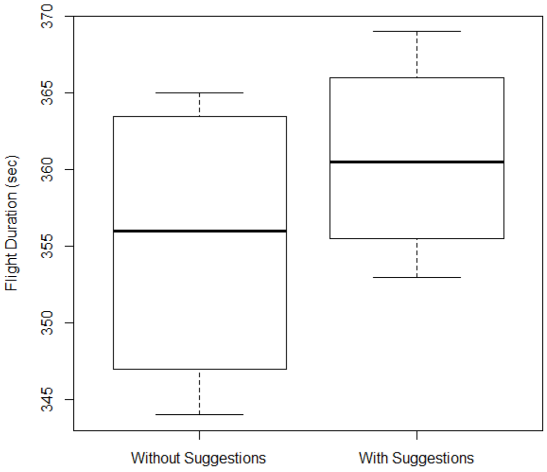


Figure 3: Box plot of the flight duration in seconds, with and without suggestions.

VI. FUTURE WORK AND CONCLUSIONS

In this paper we show the first software-based approach that aims to reduce the energy consumption of commercial quad drones: Green Flight, an Android application built on top of the Parrot ARSDK3. This software application has been evaluated by performing a series of experiments in a real target, the Parrot Rolling Spider drone.

An important constraint in the implementation of this project was imposed by the software development kit of the Parrot platform, since it is not well documented, and it does not allow APIs to access to the drone hardware. Without these constraints, we could have been able to implement a more accurate solution. This is an open issue that can be solved by new releases of the SDK, and by research works that follow up the present one. For reproducibility purposes, Green Flight should be tested using another kind of drone to strengthen the meaningfulness of the results obtained. Finally, a next step in the research roadmap is to use the data collected by Green Flight to develop tools that leverage this knowledge to suggest economic modes to be automatically inducted to the drone.

With the results obtained, we can conclude that having a tool that collects in real time relevant data from the drones, enables the user to identify the behavioral patterns that spend more energy, and as consequence, supply the necessary intelligence to avoid energy expensive maneuvers, leading to an operation oriented toward efficient energy consumption, illustrated in the real-time suggestion mode. By having this mode at hand, operators can extend the duration of missions, be aware of risky conditions created by the lack of battery, and promote the use of drones on critical missions or in scenarios that demand extended autonomy and high dependability.

REFERENCES

[1] Procaccianti, G.; (2015) The green lab: Experimentation in software energy efficiency. *Proceedings of the International Conference on Software Engineering, 2015*. ACM.

[2] Bornholt, J.; Mytkowicz, T.; & McKinley, K. S.; (2012) The model is not enough: Understanding energy consumption in mobile devices, *Power (watts)*, vol. 1, no. 2, p. 3.

[3] Corral, L.; Georgiev, A.B.; Janes, A.; & Kofler, S.; (2015) Energy-Aware Performance Evaluation of Android Custom Kernels. *4th International Workshop on Green and Sustainable* Software in connection with ICSE 2015. pp. 1-7. ACM.

[4] Lewis, G.; and Lago, P.; (2015) Architectural tactics for cyber-foraging: Results of a systematic literature review. *Journal of Systems and Software.* vol. 107 pp. 158-186. Elsevier.

[5] Balan, R. K., Satyanarayanan, M., Park, S. Y., & Okoshi, T. (2003). Tactics-based remote execution for mobile computing. *Proceedings of the 1st international conference on Mobile systems, applications and services* (pp. 273-286). ACM..

[6] Clarke, R; (2014) Understanding the drone epidemic. *Computer Law & Security Review*, vol. 30, pp. 230–246, June 2014.

[7] Corral, L.; Fronza, I.; & El Ioini, N.; (2015) The future of energy-aware software: The case of drones. *Cutter IT Journal, issue "What is Over the Technology Horizon"*. vol. 28 (8). pp. 19-23. Cutter Consortium.

[8] Parrot S.A.; (n.d.) Rolling Spider User Guide UK. www.parrot.com/support/parrot-rolling-spider. Retrieved Feb 5th, 2015.

[9] Corral, L.; Georgiev A.B.; Sillitti, A.; & Succi, G.; (2013) A method for characterizing energy consumption in Android smartphones and tablets. *2nd International Workshop on Green and Sustainable Software in connection with ICSE 2013*. pp. 38-45. ACM.