Understanding Autonomous Drone Maneuverability for Internet of Things Applications

Azade Fotouhi*†, Ming Ding† and Mahbub Hassan*†
*School of Computer Science and Engineering
University of New South Wales (UNSW), Sydney, Australia
Email: {a.fotouhi, mahbub.hassan}@unsw.edu.au
†Data61, CSIRO, Australia
Email: {azade.fotouhi, ming.ding, mahbub.hassan}@data61.csiro.au

***Abstract*—Increasing sensing and communication capabilities combined with falling prices have made drones very attractive for Internet of Things applications. A key requirement of these applications is that the drones should be autonomously maneuvered by computer programs. It is therefore important to understand the practical limitations of autonomous drone maneuverability to ensure that target application performance is met. In this paper, we first analyze drone maneuverability using theory to shed light on the tradeoff between the flying speed and the turning agility of the drone. To investigate the practical maneuverability performance, we then emulate as well as fly a commercial drone under the control of an Android program. We reveal some practical maneuverability factors that must be considered for the applications that require frequent changes of direction for the drone.**

I. Introduction

According to Federal Aviation Administration, drone sales are expected to grow from 2.5 million in 2016 to 7 million in 2020. This massive popularity of drones is driven by the increasing sensing and communication capabilities combined with their falling prices. Modern drones are now equipped with a large number of sensors, such as temperature, humidity, air pollution, GPS, 4K camera, etc., as well as an expanding range of wireless communication capabilities including WiFi, Bluetooth, 4G, RFID, and so on. Finally, drones have also improved in mechanical performance, such as flying speed, degree of autonomy, agility and maneuverability, and are available in a variety of form factors.

These developments have made drones a very attractive platform to launch many Internet of Things (IoT) applications. For example, a drone can be instrumented with the latest IoT gateway technologies, such as LoRa [1], to read gas, water, and power meters from the sky, giving an unprecedented advantage to IoT operators. A drone equipped with the right sensors can be used to collect specific air quality data from hard to reach places in a smart city and transmit that data to an IoT server in real-time using LoRa or even 4G depending on the availability. Indeed, researchers have already started visionary projects that use drones as the main IoT sensing and data delivery platform [2], [3].

In most of the envisaged IoT applications, the drone has to be flown autonomously using computer algorithms that can quickly and efficiently maneuver it to the right location at the right time. Any small maneuverability error at any stage of a complicated flight can propagate quickly through the rest of the flight causing significant performance issues for the target application. It is therefore important to understand the real effect of a particular command specified in the application program interface (API) of the drone.

The goal of this study is to experiment with a real commercial drone to understand these API commands for drone maneuverability and identify any factors that must be considered when designing autonomous drone maneuvering. In particular, we want to answer questions like: what is the trade off between flying speed and turning agility, what is the impact of flying speed on the battery life, how frequently we can make a drone change its direction, and so on. We have attempted to answer these questions using both theory as well as experiments with a DJI Phantom drone. Our experiments reveal some key practical maneuverability factors that have not been captured in the basic theoretical formula available to study drone trajectories.

The remainder of this paper is organized as follows. Section II reviews the related work on drone applications in IoT. The theoretical analysis of the trade off between drone flying speed and turning agility is presented in Section III. We present our test-bed in Section IV, followed by experimental results in section V. Finally, the conclusion and the future research direction are discussed in Section VI.

II. Related Work

In this section we briefly review the recent research on IoT application of drones and unmanned aerial vehicles (UAVs) and their challenges.

Delivering IoT services using UAVs from great height has been addressed in [4]. In this paper, the high level design for the UAV-based platform, with the focus on crowd surveillance use case is discussed. The captured video by mounted UAV camera is processed to recognize suspicious activities or people. Processing the collected video either by onboard UAVs or offloading to an edge cloud is a challenging issue in terms of processing time and UAV battery consumption. A test-bed is developed to compare the advantages and drawbacks of these two alternatives. Experiment results show that offloading the recorded video to a mobile edge cloud outperforms local

978-1-5386-2723-5/17/$31.00

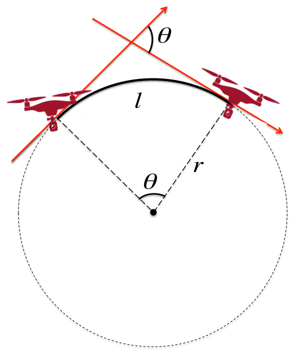


Figure 1: Drone path while taking a turn

onboard processing. Both energy consumption and processing time reduce significantly by offloading. The limited resource of UAVs for disaster surveillance application is also considered in [5] by proposing a cloud supported framework. The collected information by UAVs is pre-processed on UAV onboard, and storing, retrieving, and detection algorithm are processed in cloud. Moreover, an energy efficient framework for deployment and mobility of UAVs to collect uplink data from ground IoT devices is proposed in [6], [7]. Results show that the transmit power of IoT devices reduce hugely by using the proposed framework.

An open source smart IoT platform using UAVs to provide a range of smart city services is investigated in [3]. The proposed platform consists of four different physical components namely the cloud server, the operator, the UAV and the client. One of the possibilities the author considered for the platform architecture, is the scattered stations in the city provided for maintenance and recharging of UAVs.

Motlagh et al. [8] studied another important aspect of designing a UAV-based IoT platform. An efficient UAV selection mechanism for a particular IoT task is proposed in this paper, considering different criteria such as energy, geographic location and UAV equipment. Two optimization problems are formulated for UAV selection to minimize the energy consumption and the time to handle the event, respectively.

Although there exist considerable benefits in using UAVs in IoT applications, significant challenges need to be overcome before they can be fully embraced in commercial products. A comprehensive study on UAV related challenges such as regulation, privacy and safety, physical collision, and sky pollution is presented in [9]. However, our survey shows that detailed study on drone maneuverability is still rare in the literature. Given the importance of precise drone maneuverability in carrying out many emerging IoT services, we have provided an experimental analysis of some of the drone maneuverability limitations in this paper. We believe that our findings will help design more efficient drone-based IoT platforms.

III. Flying Speed vs. Turning Agility Trade off

For the envisaged IoT applications we discussed earlier, the drone needs to turn frequently, especially in urban environments with many high rise structures. These turns may be part of a pre-planned trajectory that the drone must follow to achieve the desired IoT outcome, such as reading all the home meters in a given neighborhood before the battery runs out. The trajectory may also be driven by certain conditions that arise during the flight. For example, in case of data delivery to mobile users, the drone may dynamically detect active users and follow the user to achieve high data rates [10], [11]. In either of these situations, we want to find out how fast the drone can change its direction when travelling at a certain speed.

When a flying path (for pre-planned trajectory) or flying algorithm (for dynamic trajectory) is designed for the drone, the path must meet the kinematic constraints of the drones. Figure 1 shows how the drone, or any moving object, would take a turn given that the object flies at a constant speed of v. First, the drone will travel in an arc for some time t before turning by an angle θ. Following equations define the maneuverability constraints and tradeoffs for the drone [12], [13]:

$$l = v.t; \quad l = r.\theta; \quad r = \frac{v^2}{a} \tag{1}$$

where l is the arc length, r is the circle radius, and a is the lateral acceleration. Thus, we have

$$v.\theta = a.t; \quad t = \frac{v.\theta}{a} \tag{2}$$

Equation (2) reveals some interesting tradeoff for flying speed and turning agility for the drone, which must be carefully considered by the IoT application developers involving drones. We notice that the higher the flying speed, the longer it will take the drone to make a turn given that the acceleration is something that is constrained by the hardware as an agility capability of the product. Therefore, it may not be always better to select a higher flying speed to minimise the flight time or maximise the application performance if the drone is expected to turn frequently. A lower speed helps the drone to make a quicker turn, which may help achieve particular application goals.

The formula in Equation (2), however, does not answer some of the practical constraints a programmer is going to face. For example, the formula does not say anything about how frequently the drone can be instructed to take turns. Can we write a program that attempts to optimize drone movement for a highly dynamic application by constantly working out the best direction to fly and instruct the drone to do that, say every 500 ms? Does the flying speed have any impact on the battery life of the drone? How to find out the lateral acceleration capability of the drone if it is not mentioned in the data sheet? Can we simply use Equation (2) to track a drone which is going through frequent turns, or are there other factors that may affect the trajectory of the drone? These are some of the questions we attempt to answer next by actual experiments.

Table I: Specifications of drones used in our experiments [14]

Name	Phantom 3 Advanced	Phantom 4
Release Date	April 2015	March 2016
Weight	1280 g	1380 g
Max Speed	16 m/s	20 m/s
Battery	6000 mAh LiPo 2S	6000 mAh LiPo 2S
Max Ascent Speed	5 m/s	6 m/s
Max Descent Speed	3 m/s	4 m/s
Max Flight Time	23 min	28 min

IV. Experimental Test-bed

In this section, the required tools and software for analyzing drone maneuverability are described.

A. Drone Hardware

In order to identify any practical limitations of drone maneuverability, we conducted both simulations and real field experiments using DJI Phantom drones [14]. The mechanical capabilities of the drones that we used are summarized in Table I. We also developed an Android app to autonomously maneuver the drone according to some pre-designed experiments and collect data for later analysis.

B. Android App

We develop an Android application that utilized the DJI's software development kit (SDK) [15] to control the drone to perform actions such as going forward and turning with different speeds and angels. Currently, there is no open-source Android application that allow to instruct the drone to maneuver as precisely as we need in our experiments. Although a large portion of this application was built off the existing DJI demo application [16], we had to develop several key components including controlling the drone via the virtual joystick and logging the flight telemetry data.

Figure 2 shows the developed application installed on an Android based phone. There are many control buttons, but the ones we used for this experiment are described below (the rest will be used for our future experiments with more advanced applications):

- Locate: Pressing this button will show the location of the drone on the Google map. Any movement of drones can be tracked by its movement on Google map.
- Enable Virtual: This button will enable the virtual stickers of drone. Once enabled, the drone will not respond to any of the remote control stickers. The drone will be controlled by the Android application.
- Disable Virtual: Return the control of the drone to remote control stickers.
- Start Timer: This button starts the timer for the flight information to be recorded at the minimum sampling rate (100ms) allowed by the drone.
- Stop Timer: This will stop the timer.
- Program: By pressing this button, the drone will follow the programmed instructions.
- Export Data: All recorded flight information will be exported to Android phone storage.

Figure 2: A screenshot of the developed Android app

```
//for zigzag movement
height = 10; //meter
interval = 1; //sec
speed = 4; //m/s
turnAngle = 1.55; //in radians
velocityX = speed * Math.sin(turnAngle/2);
velocityY = speed * Math.cos(turnAngle/2);

FlightInstructionsWithTime toLeft =
        new FlightInstructionsWithTime(generateFlightInstructions(
            velocityX, velocityY, 0, height), interval);
FlightInstructionsWithTime toRight =
        new FlightInstructionsWithTime(generateFlightInstructions(
            -velocityX, velocityY, 0, height), interval);
for(int i=1;i<=10;i++) {
    programInstructions.add(toLeft);
    programInstructions.add(toRight);

}
```

Figure 3: A set of flight instructions in our developed Android application to force the drone follow a zigzag path with 1.55 radians turn and speed of 4 m/s

To control drone maneuverability, we write a set of flight instructions in the application. The drone follows these instructions sequentially. Generally, the velocity in X and Y direction, drone height and time to fly are set up in each flight instruction. Different flying path, such as zigzag vs circular, can be designed by a combination of flight instructions. Flight data is recorded at the sampling rate (one record every 100 ms), and the entire recorded data at the end of the flight is saved on the Android storage for later analysis. Altitude, latitude, longitude, velocity in X, Y and Z direction, battery voltage, time and date are among the recorded information. An example of the instruction codes to generate a zigzag movement is illustrated in Figure 3. In this Figure, the speed and turning angle is set to 4 m/s and 1.55 rad, respectively. Then the value of v_x and v_y are assigned according to Equation (3). Two flight instructions are used to simulate the turn to right and left for 1 second. The turnings repeated 10 times to create a zigzag pattern.

We use the Android app for two different kinds of experiments. The first kinds of experiments, which we call emulations, are done entirely within the laboratory by DJI

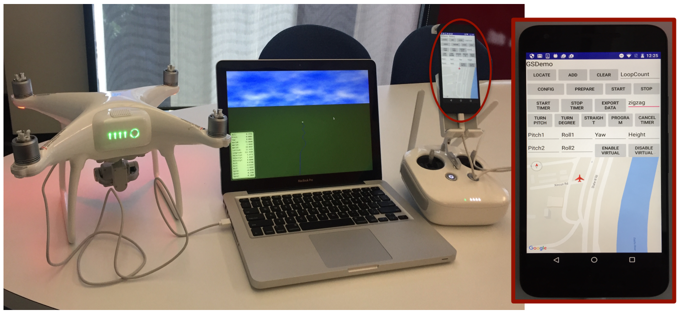


Figure 4: Emulated Phantom 4 (propellers-less) in DJI assistant 2 using our developed Android application

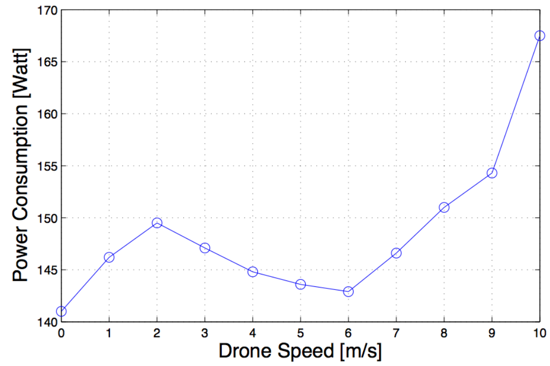


Figure 5: Drone power consumption vs. speed at 10 m height

Simulator and our app. The second kinds of experiments are conducted in an open field where we actually fly the drone using the Android app.

C. DJI Simulator

DJI provides a flight simulator that allows to connect a propellers-less drone to the simulator, which simulates drone flight by taking all control commands from the drone. The drone therefore actually does not fly (propellers are taken off), but the simulator can show the drone movements on a map and record all flight data. This allows experimenting with a real drone without leaving the laboratory and without being affected by wind. We have found that this type of experiment is more reproducible due to lack of unpredictable influence from the weather. DJI PC Simulator and DJI Assistant 2 are used for phantom 3 and phantom 4, respectively.

We used DJI simulator together with our developed Android application in order to force the drone to follow the designed movements and collect flight records. Figure 4 shows the set up of our experimental test-bed with simulated phantom 4 in DJI Assistant 2, which is following the programmed instructions from the Android app wirelessly. The propellers-less drone is connected to the laptop (simulator) using a USB cable.

V. Experiments and Results

In this section, we first explore the impact of drone flying speed on its battery life by conducting a set of real flying experiments in an open field. Then, we use the DJI simulator to study various aspects of drone maneuverability in a windless condition inside our laboratory.

A. Battery Life Experiment

Like most IoT devices, drones are battery powered. Unlike other IoT devices, however, drone battery life is mostly affected by the mechanical movement of the drone than the sensors or the communications. The mechanical power consumption of drones depends on its speed, mass, and the design [17]–[19]. Since other hardware elements cannot be controlled, it is important to understand how the flying speed affects the battery life.

To observe the impact of speed on battery life, we fully charged the battery at the start of each experiment. Then we flew the drone in a *way point format*, i.e., between two specified points going back and forward continuously, until the battery reached 20%, which is the minimum the drone can fly on, while keeping the drone altitude at 10 m. We repeated this experiment for 11 different speeds, from 0 m/s to 10 m/s with increments of 1 m/s. For each speed, we repeated the experiment five times and reported the average power consumption in Figure 5.

We observe a very interesting result. The power consumption remains below 150 W until 8 m/s, after that power consumption starts to increase rapidly. For example, at 10 m/s, the power consumption is 167 W, which is 11% higher than what it consumes at 8 m/s. This means that although trying to fly the drone at the full speed allowed by the manufacturer is detrimental to battery life, we can still fly close to the maximum speed without affecting the battery life any further than if it would have been flown at a lower speed.

We also observe some fluctuations in power consumption before 8 m/s, but our conjecture is that the wind may actually help reducing power consumption sometimes if the drone is flying in the same direction of wind flow (tail wind). Unfortunately, the DJI did not have a mechanism to record wind data for us to further analyse power consumption fluctuations below 8 m/s. Wind analysis remains part of our future study.

The remaining experiments are conducted inside our lab using the DJI simulator.

B. Maximum Turning Frequency

The objective of this experiment is to figure out what is the maximum frequency at which we can instruct the drone to change its direction (take turns). For many typical applications, the drone direction may not have to change for several seconds or so. However, if drones are expected to be used in applications that involve dynamic elements, such as a mobile network trying to deliver data to moving devices, then the application could increase its data rate performance if high frequency turning was possible for the drone.

To find out the maximum turning frequency, we designed an experiment where we change the command value of the X direction every t sec, while keeping Y direction fixed. The velocity in X direction changes to positive and negative values

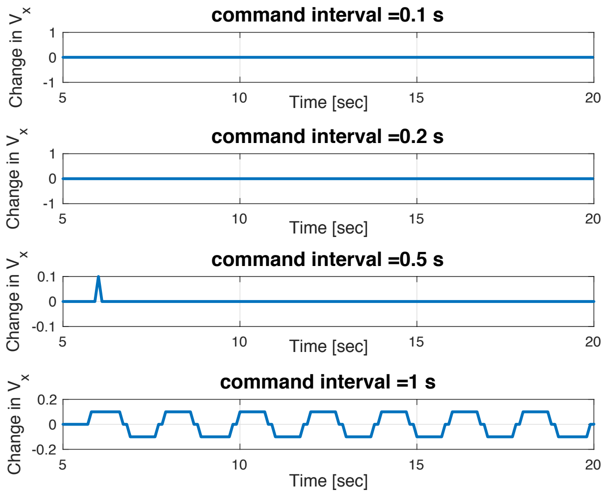


Figure 6: Variation of drone Velocity in X direction with different command intervals

repeatedly to simulate a zigzag movement. Both the velocity in X and Y direction are selected based on the cruising speed and the target turning angle as presented in Equation (3), where $v = \sqrt{v_x^2 + v_y^2}$ is the cruising speed, and θ is the issued turning angle in radians. We choose a very low flying speed and a small turn command to avoid hardware restrictions play a part. The drone is moving with a speed of 2 m/s and the turning angle command is 0.09 rad (5 degree). We keep the drone height fixed to 10 m, so the velocity in Z direction is zero. This selected height is recommended by recent studies (e.g., [20] and 3GPP standards). With this setting, the only drone variable that should change due to the turning commands is its velocity in X direction.

$$v_x = \pm v.sin(\theta/2); \quad v_y = v.cos(\theta/2) \tag{3}$$

We recorded velocity in X direction (v_x) every 100 ms and plotted the *change* in Figure 6 for four different command intervals, 0.1 sec to 1 sec, with decreasing frequency. For example, a command interval of 500ms means a frequency of two turning instructions per second (2 Hz) and so on. As we can see in Figure 6, the velocity in X direction does not change during the flight if the command intervals are less than 1 sec (frequency is greater than 1 Hz). However, the velocity in X direction changes according to the commands when the commands are issued at 1 Hz frequency. This experiment provides an important finding that there is a maximum turning frequency for the drones, which must be taken into account when developing a drone IoT application. Currently, this value is not given in the data sheets, but at least this can be obtained using experiments like this.

A possible explanation for the existence of a maximum turning frequency is because of the existence of multiple latency in this drone system, such as “the signal reception time", “the command processing time",“the hardware response time", etc. Hence, we can say that there is a control “overhead time", which is the minimum time interval that the drone needs to perform any action from the moment a command is issued to it. Having more than one command within this “overhead time" will lead to no action of the drone, as shown in the first 3 rows of Figure 6. In our future work, we will investigate further to quantify the “overhead time", as knowing its value enables more precise application of the time variable in Equation (2).

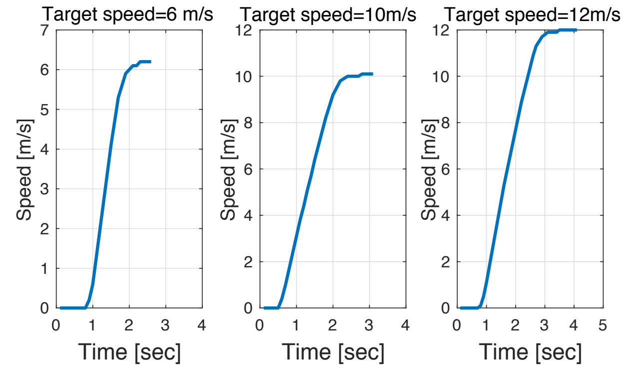


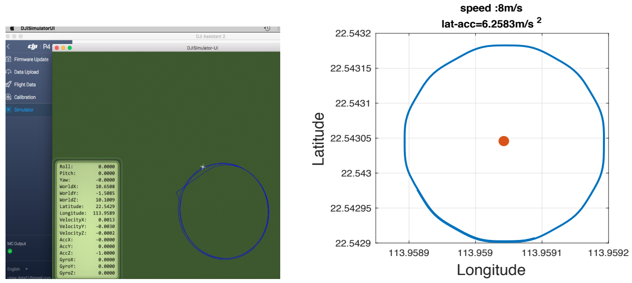
Figure 7: Drone speed as a function of time to measure linear acceleration

C. Deriving Drone Acceleration

Recall that, to be able to use Equation (2) for analysing the flying speed vs. turning agility of the drone, we need to know the acceleration capability of the hardware. Unfortunately, our survey revealed that the manufacturers are not providing this value in their data sheets. We therefore present some techniques to obtain the acceleration value of the drone used in our experiments.

Accelerations can be measured from the linear movement (linear acceleration) when the drone is moving at an increasing speed, or from the circular movement (lateral acceleration), i.e., when the drone moves in a circle with a constant speed. We therefore designed two different moving paths for the drone to follow in a simulated environment. The first one measures the linear acceleration, where the drone flies in a straight line from speed of zero to a larger target speed within the minimum time. The speed and the time is collected and the slope of speed variation with respect to time variation is calculated. We conducted several experiments using the Android application to measure such slopes. Figure 7 illustrates the variation of speed with time when the drone moves in a straight line until it reaches to the target speed of 6 m/s, 10 m/s and 12 m/s, respectively. According to these measurements, the maximum linear acceleration for Phantom 4 is estimated to be 6.2 m/s^2.

We also designed a circular path for the drone to measure its lateral acceleration as well. To find out the lateral acceleration, we use the equation of $a = \frac{v^2}{r}$, and the drone was instructed to turn regularly in a fixed turning angle to create a circular path. Given the recorded drone path (longitude and latitude), the



(a) (b)

Figure 8: (a) Drone is following a circular path in DJI simulator (b) The optimal center for the created circular path

radius of the created circle needs to be calculated. Therefore, an optimization problem is formulated to find the best center of the created circle and consequently the best corresponding radius. The formulated optimization problem tries to minimize the *mean square error* by comparing the distance between any point on the moving path and the candidate center C. The optimization problem to find the best candidate center is formally presented as follows:

$$\arg\min_{C} \frac{1}{N}\sum_{i=1}^{N}\Big(dis(P_i, C) - \frac{\sum_{i=1}^{N}(dis(P_i, C))}{N}\Big)^2 \tag{4}$$

where C is the candidate center, P_i denotes the $i-th$ sample point on the circular path, and N is the size of sample points on the circular path. The Euclidean distance between two points A and B is denoted by $dis(A, B)$.

Several experiments using different speeds and turning angles are conducted to compute the lateral acceleration. A sample of our experiments in the simulator environment with recorded path and the optimal center are shown in Figure 8a and 8b, respectively, when the drone is moving at the speed of 8 m/s. A lateral acceleration of 6.25 m/s^2 is obtained using this method, validating the earlier calculation using the linear movement. We should note however, that this acceleration is valid for windless condition. How to obtain the acceleration value in a windy environment will be investigated in our future work.

VI. Conclusion and Future Work

Drones can provide numerous IoT services because of their agility, maneuverability and speed. There are, however, practical limitations for drones' mobility. In this paper, we investigated drone maneuverability in the context of IoT applications that require frequent change of direction for the drone. We conducted various experiments to evaluate the limitations in terms of battery consumption, speed and turning angle. Our experiments considered different moving models such as straight line, circular pattern, and zigzag path at a fixed height. More field experiments and 3D drone movements are left for our future work.

Acknowledgment

The authors gratefully acknowledge Jay Gurnani's contributions in developing the Android application. Azade's research is supported by Australian Government Research Training Program Scholarship and Data61|CSIRO PhD enhancement scholarship.

References

[1] "LoRa Alliance: Wide Area network for Internet of Things," https://www.lora-alliance.org/, accessed: 2017-04-08.

[2] C. Snow, "Why Drones are the future of the internet of things," http://droneanalyst.com/2014/12/01/drones-are-the-future-of-iot/, 2017, accessed: 2017-03-23.

[3] A. Giyenko and Y. I. Cho, "Intelligent UAV in smart cities using IoT," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, Oct 2016, pp. 207–210.

[4] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-Based IoT platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, February 2017.

[5] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A UAV-Cloud system for disaster sensing applications," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–5.

[6] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications," *ArXiv e-prints*, Mar. 2017.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile internet of things: Can UAVs provide an energy-efficient mobile architecture?" in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[8] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV selection for a UAV-Based integrative IoT platform," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[9] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, Dec 2016.

[10] A. Fotouhi, M. Ding, and M. Hassan, "Dynamic Base Station Repositioning to Improve Spectral Efficiency of Drone Small Cells," in *2017 IEEE WoWMoM*, 2017.

[11] A. Fotouhi, M. Ding, and M. Hassan, "Dynamic base station repositioning to improve performance of Drone small cells," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–6.

[12] M. Shanmugavel, A. Tsourdos, B. White, and R. Żbikowski, "Co-operative path planning of multiple UAVs using dubins paths with clothoid arcs," *Control Engineering Practice*, vol. 18, no. 9, pp. 1084 – 1092, 2010.

[13] G. Avanzini, G. de Matteis, and L. M. de Socio, "Analysis of aircraft agility on maximum performance maneuvers," *Journal of Aircraft*, vol. 35, no. 4, pp. 529–535, 1998.

[14] D. Phantom, "DJI Phantom drones," http://www.dji.com/phantom, 2017, accessed: 2017-03-01.

[15] D. Developer, "DJI Android mobile sdk reference," https://developer.dji.com/iframe/mobile-sdk-doc/android/reference/packages.html, 2017, accessed: 2017-03-23.

[16] D. Phantom, "Creating a mapview and waypoint application," https://developer.dji.com/mobile-sdk/documentation/android-tutorials/GSDemo-Google-Map.html, 2017, accessed: 2017-03-23.

[17] E. M. Shakshuki, D. Zorbas, T. Razafindralambo, D. P. P. Luigi, and F. Guerriero, "Energy efficient mobile target tracking using flying drones," *Procedia Computer Science*, vol. 19, pp. 80 – 87, 2013.

[18] C. D. Franco and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference on*, April 2015, pp. 111–117.

[19] L. Di Puglia Pugliese, F. Guerriero, D. Zorbas, and T. Razafindralambo, "Modelling the mobile target covering problem using flying Drones," *Optimization Letters*, vol. 10, no. 5, pp. 1021–1052, 2016.

[20] M. Ding and D. L. Perez, "Please lower small cell antenna heights in 5g," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.