On Implementing a Low Cost Quadcopter Drone with Smartphone Control

Conference Paper · May 2016

4 authors, including:

Mohammad zakaria Masoud
Al-Zaytoonah University of Jordan

Yousef Jaradat
Al-Zaytoonah University of Jordan

On Implementing a Low Cost Quadcopter Drone with Smartphone Control

Mohammad Masoud, Yousef Jaradat, Mohammad Farhoud, Ammar Almdallaleh
Computer and Communication Engineering Department
Al-Zaytoonah University of Jordan
Amman, Jordan
{m.zakaria,y.jaradart}@zuj.edu.jo

***Abstract*- Duo to their small size, performance and simple design, quadcopters proliferated in the past few years. Nowadays, quadcopters have been utilized in different environmental, military and health applications. In this work, we attempt to design and implement quadcopter drone using inexpensive commercial off-the-shelf (COTS) electronics. Arduino kit will be utilized to control the drone flying process. Accelerometer and gyroscope will be used for drone flight stabilization. A smartphone with Bluetooth connection will be used to control the drone in short range communication. A proportional–integral–derivative (PID) controller with complementary filter will be programmed in Arduino kit to control the speed of motors. We also show how PID and complementary filter can be utilized and programed with these inexpensive COTS components. A backtracking algorithm was proposed to allow the drone to track back its path to the starting point without using any GPS information whenever it crosses the Bluetooth transmission range of the smartphone controller.**

***Keywords*- Drones, quadcopter, (PID), complementary filter, Accelerometer, gyroscope**

I. INTRODUCTION

A quadcopter is type of copters with four arms each with propeller and engine on the end of each one arm. It is similar to helicopters in some way, though helicopters have a tail that helps stabilize the craft, whereas quadcopter do not. In a quadcopter, two of the propellers spin in one direction (clockwise); the other two spin the opposite direction (counterclockwise); this setting enables the machine to hover in a stable formation. In general, Drones have a rich history and they improved a lot in the last few decades with the help of advanced computer technology and engineering.

In the last decades, drones, especially small form ones, were used in many applications. They were implemented for military purposes. In that time military agencies were the only one to use them due of their high costs. After that companies start build small drones for commercial uses. Nowadays, drones have a wide spectrum of applications. These applications varies from surveillance and reconnaissance tasks by military; agriculture and law enforcement agencies [1, 2]; search and rescue missions in urban environments; aerial imagery because of their autonomous nature and huge cost savings; medical, live video streaming and security. Moreover, people use them for entertainment and social networking in sharing videos and tacking pictures [3]. However, drones systems used in these applications setups are expensive since they are using expensive components, such as, quadcopter's remote controller and balancers. Moreover, the controlling and intelligence are part of the ground station in the remote controller. This paper answers the following questions: *Can we implement a quadcopter drone with low cost components? Can we write real time control algorithms using microcontrollers?*

In this project, a general purpose quadcopter will be designed and assembled. Light weighted *Lipo* battery will be used as power source. An Arduino kit will be used to control the flying mechanism of the quadcopter. A proportional–integral–derivative controller (PID controller) will be programmed for drone's flight stabilization. Accelerometer and gyroscope sensor will be used to obtain a feedback from the flying process. Ultrasonic sensors will be used to protect the drone from collisions with other objects, such as, walls. The drone will be controlled by commands sent from a smartphone using Bluetooth (controller). Moreover, the drone will have the ability to come back to its starting location if it is out of controller range without GPS equipment.

Our contribution in this work can be summarized as following:

- Designing and Implementing a low cost quadcopter utilizing commercial off-the-shelf (COTS) low cost electronics
- Implementing PID controller using Accelerometer and gyroscope signals for stabilizing the drone in the air.
- Proposing a backtracking algorithm to allow the drone to return back to its starting point when it is out of the controller range without using GPS information.

The rest of this paper is organized as follows; section II introduces some of the related works that have been conducted in this area. Section III overviews our design and demonstrated the implementation of PID. Section IV shows the results of stabilizing the quadcopter utilizing Accelerometer, gyroscope and complementary filter. We conclude this paper in section V.

II. RELATED WORK

Constructing and assembling low cost and simple quadcopters attracted designer and researchers for decades [4-5]. Flight Stability is the main concern in controlling quadcopters. Many methods and works have been conducted in this area [6-7]. Among all of them, PID controller dominated [8, 9]. However, other control theories can be used, such as, fuzzy control which is very hard to program and to implement [10]. To use PID controller, a feedback signal should be obtained. Accelerometer and gyroscope sensor can be used for this purpose [11]. In our work we utilized the same sensor.

Our work defers from others works in three main points. First, our drone will use Arduino kit to control it without the use of commercial control systems. Second, multi-feedback data will be fed into the controller for obstacle avoidance, stabilizing and height control. Finally, a proposed mechanism will be used to let the drone return to its starting point if it flies out of the controller range.

III. Quadcopter Design

In this section we will demonstrate quadcopter design. We will start by examine the accelerometer, gyroscope, complementary filter and PID controller. Subsequently, quadcopter design process and parts will be demonstrated. Finally, we will demonstrate the simple implementation of the backtracking algorithm when the drone is out of the controller range. Table 1 list all the variables used in the upcoming analysis

Table 1: Variables Definitions

Variable	Description
ang(.)	Angle in a given direction (x, y and z)
accel(.)	Accelerometer data in a given direction (x, y and z)
A	Angle of the drone
α	Optimization variable
Gyro	Measured gyroscope sensor data
Acc	Measured angle of accelerometer sensor
Kp, Ki, Kd	PID controller optimization variables
Err	Error in measured angle
β	Optimization variable for motor response time
Ctrl_Sig	Output of the PID controller used to find duty cycle of the PWM

A. Accelerometer and Gyroscope

Accelerometer and gyroscope are special type of sensors used to get the position and the orientation of objects. Gyroscope uses the gravity of the earth to measure the orientation of an object. On the other hand, accelerometer is a non-gravitational device. It utilizes the vibration occurs after a movement to measure the orientation of an object. To stabilize an object, these two sensors should be used. However, why not to use only one of them?

Accelerometer measures linear acceleration of an object without the rotational velocity. The output of this sensor is noisy. In other words, it changes very quickly. On the other hand, gyroscope measures the rotational velocity of an object over an axis. However, gyroscopes measurement may drift and not return to the initialization point. According to this, a combination of the values measured of these two sensors must be used. To measure their values, Kalman [12, 13] or complementary filters [14] must be used. In the next section we will introduces complementary filter.

One thing to be mentioned is that accelerometer and gyroscope measured values are relative to the axis. Three different values are measured for each one of them. To obtain slope angle relative to the axis we use the following formulas.

$$ang(x) = \text{atan}\left(\frac{accel(x)}{\sqrt{(Accel(y))^2+(accel(z))^2}}\right) \quad (1)$$

$$ang(y) = \text{atan}\left(\frac{accel(y)}{\sqrt{(accel(x))^2+(accel(z))^2}}\right) \quad (2)$$

$$ang(z) = \text{atan}\left(\frac{accel(z)}{\sqrt{(accel(y))^2+(ccel(x))^2}}\right) \quad (3)$$

Each one of these angles has a special name as we can observe from figure 1. For example *ang(x)* is called *Roll*, *ang(y)* is called *Pitch* and *ang(z)* is called *Yaw*.

B. Complementary Filter

Complementary filter has been selected in this work rather than Kalman filter for two reasons. First, it is simpler than the Kalman filter in mathematical operations and algorithm programming. Second, the memory size of Atmega microcontroller in Arduino kit is small to program Kalman algorithm with drone controller and sensors.

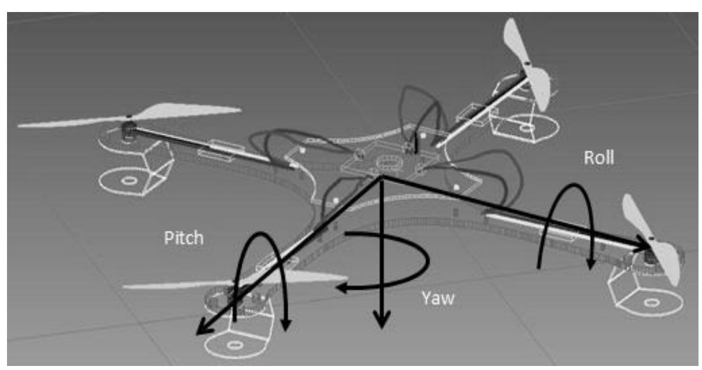

Figure 1: Angle names

Complementary filter is a simple equation consists of two parts; a rapid changing part and a slower changing part. The first part will be used for accelerometer value and the second part will demonstrate gyroscope measured value. Equation 4 shows the general formula that has been utilized in our control system for complementary filter. See Table 1 for variables definitions.

$$A = \alpha * (A * \int Gyro) + (1 - \alpha) * Acc \qquad (4)$$

The measured angle can be fed as the error value to the PID controller. One thing to be mentioned is that gyroscope and accelerometer measured values must be converted into angular value. The received data from these sensors does not demonstrate or show the angle or the slope of the object. To obtain the angle of the slope, equations (1 and 2) should be used.

C. *PID Controller*

PID controller is one of the most old control tools. It consists of three parts proportional, integral and derivative. Equation (5) shows the general formula of a PID controller.

$$Ctrl\_Sig = Kp * Err + Ki * \int Err * dt + Kd * \frac{dErr}{dt} \qquad (5)$$

The error value in our work is the measured angle obtained from equation (4). In this work we utilized two angles only; *pitch* and *roll*. *Yaw* has not been used. Two angles are calculated. Moreover, two error values are used with two PID controllers. The two obtained control signals are used to change the speed of the four motors.

In our design, four brushless motors have been used. To control these motors, electronic speed controller driver (ESC) must be used. To connect this driver to our microcontroller, pulse width modulation (PWM) is used. A higher duty cycle gains higher speed. The duty cycle value has been calculated according to equation (6).

$$duty\_cycle(t) = duty\_cycle(t-1) + \beta * Ctrl\_Sig \qquad (6)$$

D. *Quadcopter Design and Parts*

To design the quadcopter, carbon fiber utilized to construct the body. Four brushless motors with 30A ESC have been connected to Arduino NANO board. Power distributer board has been connected to 5000 mAh LiPO battery. ITG/MPU kit which consists of 6 axis accelerometer and gyroscope has been connected to the Arduino board. Five different ultrasonic sensors have been connected to avoid obstacles. Finally, a Bluetooth module has been connected to create a connection with the android application implemented to send control command over Bluetooth connection. Our battery provides 25 minutes of flying time. Figure 2 shows the designed drone.

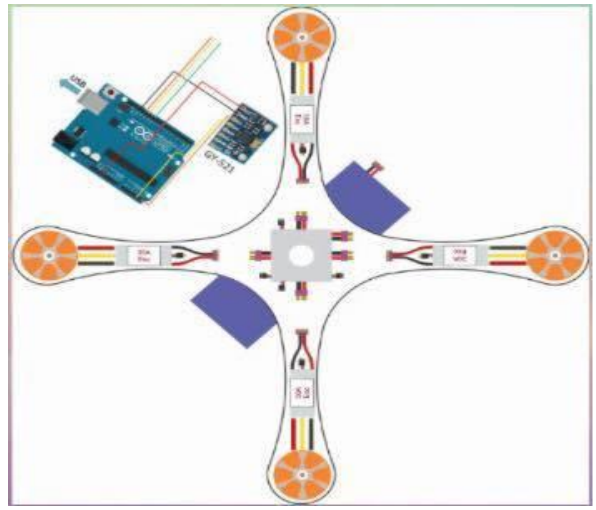

Figure 2: The Implemented Drone

E. *Backtracking Algorithm*

For the drone to find its way home, the Bluetooth transmission range has been utilized as boundary to trigger the algorithm. The controller tracks the way the drone follows all the time by recording the *pitch* and *roll* angles over the time. If the values of these angles have changed more than a threshold value, the old value is recorded with its period of time. Subsequently, the timer is initialized and the new angle will be used for comparison again. When the drone goes out of the transmission range of Bluetooth, the controller inverse the recorded values of all angles and start to follow them in reverse order. Algorithm 1 shows pseudo code steps for implementing the backtrack algorithm.

Algorithm 1: Backtrack algorithm

```
T: Array of time durations
PWM_ara: 2D array of PWM of the four motors
Data: data received from Bluetooth
PWM(i): PWM value of motor i
Num: Number of motors 4
temp(num): Array of temporary data

start=time;
FOR i DO
        temp(i)=PWM(i);
WHILE Data DO
        FOR i DO
                IF abs(temp(i)-PWM(i))>Threshold THEN
                        PWM_ara=temp;
                        temp(i)=PWM(i);
                        T(i)=start-time;
                        start=time;
FOR i DO
        PWM(i)=PWM(i-4);
FOR T DO
        Motors(PWM);
```

IV. Experiment and Results

To measure the stability of PID controller, different β values have been used. Moreover, wind impact on the quadcopter body has been measured. To generate wind effect, a fan has been used. To measure its stability in the air, the angles and the slope of the quadcopter has been utilized. These angles have been used to measure the error height in the drone. Figure 4, shows the measured angle and the calculated error. The error angles are four angles. Each two of them are opposite to each other. Two *roll* angles and two *pitch* angles. With two PID controllers, the control signals of these two PIDs are used to control the PWM of the four motors.

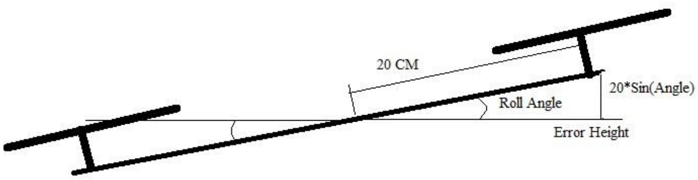

Figure 4: Error Height

Figure 5 shows the theoretical value of PID controller and the measured height value from the quadcopter. We can observe the similarity between the two values. We also can observe how Equ.6 can be implemented in easy way to obtain an acceptable performance. The y-axis in this figure ‘slope’ is the error height of the drone which equals *20*sin(ang)*. We utilized *roll* angle in our measurements only since the *pitch* angle will generate the same results under the same PID controller.

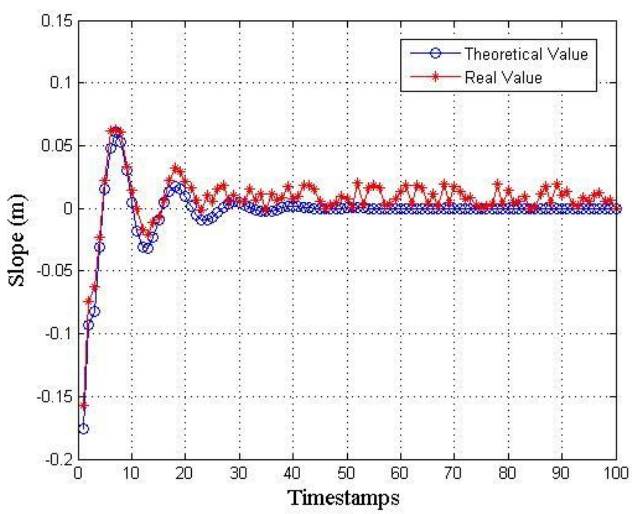

Figure 5: PID theoretical and real values

Figure 6 shows the impact of fan on the PID real and theoretical values. We can observe that even under this condition, the controller reported an acceptable performance

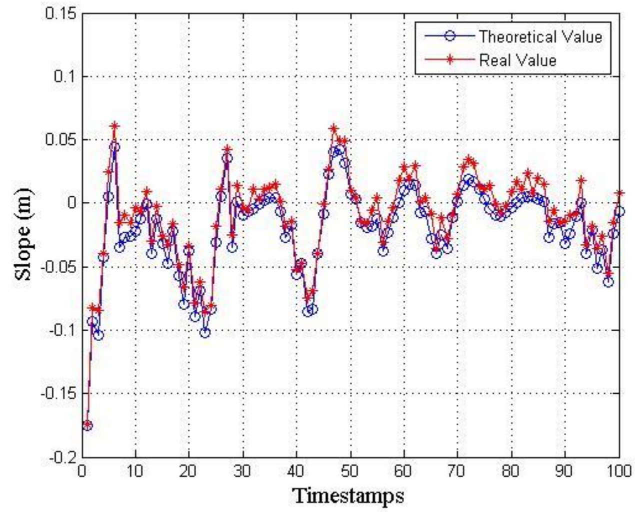

Figure 6: Fan Impact on PID controller

Finally, to obtain a value of β, we utilized a fan to generate a wind around the drone. We secure the drone from its four arms with cords. Subsequently, different β values have been utilized. Figure 7 shows how a higher β value allows the drone to stabilize faster. However, in normal air condition, higher β value may generate an instable control. This opens the door for an adaptive controller. However, we did not implement such a controller in this work.

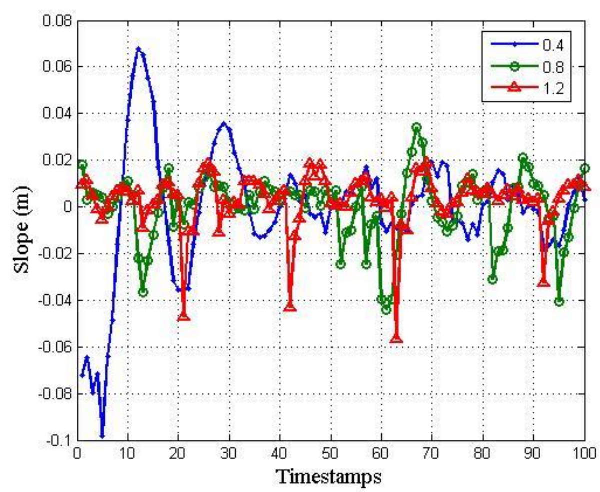

Figure 7: Different β values

V. CONCLUSION

In this work, a low cost Bluetooth controlled quadcopter drone has been designed and implemented. A PID controller has been implemented in Arduino kit to stabilize the drone. We examined the stabilization controlling process of the drone under wind effect. Our results show the stability of the drone in this condition. However, a more advance adaptive controller should be implemented for different scenarios. In our future work we will attempt to implement such a controller.

VI. ACKNOWLEDGMENT

This research was supported in part by Al-Zaytoonah University of Jordan fund (2/11/2014). We would like to thank the University for the equipments and tools they provided for this work.

Reference

[1] Gupte, Shweta, Paul Infant Teenu Mohandas, and James M. Conrad. "A survey of quadrotor unmanned aerial vehicles." *Southeastcon, 2012 Proceedings of IEEE*. IEEE, 2012.

[2] Patel, Parth N., et al. "Quadcopter for agricultural surveillance." *Advance in Electronic and Electric Engineering* 3.4 (2013): 427-432.

[3] NIXI Project http://flynixie.com/

[4] Leong, Bernard Tat Meng, Sew Ming Low, and Melanie Po-Leen Ooi. "Low-cost microcontroller-based hover control design of a quadcopter." *Procedia Engineering* 41 (2012): 458-464.

[5] Sa, Inkyu, and Peter Corke. "Vertical infrastructure inspection using a quadcopter and shared autonomy control." *Field and Service Robotics*. Springer Berlin Heidelberg, 2014.

[6] 3- Argentim, Lucas M., et al. "PID, LQR and LQR-PID on a quadcopter platform." *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*. IEEE, 2013

[7] Luukkonen, Teppo. "Modelling and control of quadcopter." *Independent research project in applied mathematics, Espoo* (2011).

[8] Argentim, Lucas M., et al. "PID, LQR and LQR-PID on a quadcopter platform." *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*. IEEE, 2013.

[9] Jones, Omar. "DESIGN AND DEVELOPMENT OF AN EMBEDDED DC MOTOR CONTROLLER USING A PID ALGORITHM." (2010).

[10] Sa, Inkyu, and Peter Corke. "System identification, estimation and control for a cost effective open-source quadcopter." *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012

[11] V.P. Kodgirwar , Vivek Kumar "Design of Control System for Quadcopter using Complementary Filter and PID Controller", International Journal of Engineering Research & Technology (IJERT), 2014

[12] Becerra, Victor Manuel, P. D. Roberts, and G. W. Griffiths. "Applying the extended Kalman filter to systems described by nonlinear differential-algebraic equations." *Control Engineering Practice* 9.3 (2001): 267-281.

[13] Ahn, Kyoung Kwan, and Dinh Quang Truong. "Online tuning fuzzy PID controller using robust extended Kalman filter." *Journal of Process Control* 19.6 (2009): 1011-1023.

[14] Euston, Mark, et al. "A complementary filter for attitude estimation of a fixed-wing UAV." *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008.