International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 12, December-2015

An Automatic Thrust Measurement System for Multi-rotor Helicopters

Myunggon Yoon
Department of Precision Mechanical Engineering
Gangneung-Wonju National University,
South Korea

***Abstract*—In this paper we introduce a microprocessor-based automatic thrust measurement system with which one can easily characterize the static thrust force of a propeller widely used for small battery-operated multi-rotor helicopters. Our measurement system consists of a microprocessor, signal amplifiers and external sensors; a load-cell for a thrust measurement, a photo sensor for a rotational speed measurement. A test run has shown that the static thrust and the speed of a propeller can be automatically and precisely measured only in a couple of minutes.**

Keywords— Unmanned Aerial Vehicle, Thrust, Transfer Function, Multi-rotor Helicopter

I. INTRODUCTION

Recent interest on multi-rotor helicopters from both general public and academic community seems to come from promising future applications of those helicopters in various fields such as search and rescue operation, mapping, aerial photograph, surveillance and so on.

However, within the author's knowledge, there are still only a little systematic and rigorous study performed in academic community on multi-rotor helicopters, even though one can easily find huge experience-based case reports and casual writings by amateurs on the internet. It is also true that an increasing number of researchers in various academic fields are getting interested in small multi-rotor helicopters, commonly called as *drones*, including the authors of [1, 2, 3].

For systematic analysis and developments of drone systems, it is essential to have a reliable mathematical model of a drone system. A key difficulty in obtaining a mathematical model is how to precisely describe the thrust force of propellers. A mathematical modeling of a trust force is difficult in general because an ESC (electronic speed controller) driving a propeller is a microprocessor-based digital system. In fact the driving algorithms and various parameters of commercial ESC systems can be changed but those settings are mostly unknown to end users. Furthermore, the aerodynamic relation between a propeller speed and a thrust force is highly nonlinear and heavily depends on particular operating conditions.

In order to circumvent those difficulties, the author of [4] proposed a *back-box* approach in which the dynamic relation between an ESC command and a thrust force, as a whole, is modelled as an unknown transfer function. In an actual application of the identification procedures proposed in [4], however, there are several difficulties. First of all, most procedures require manual work. For instances, the angular velocity of a propeller was measured from a manual reading of the pulse period of an optical sensor signal on an oscilloscope. In addition, the PWM (pulse width modulation) command signal for an ESC was generated and manually modified using an external waveform generator.

Motivated from the fact that most of those manual work in the identification procedures of [4] can be easily automatized with a MCU (microprocessor control unit), we have developed an automatic thrust measurement system which will be explained below.

II. MEASUREMENT SYSTEM

A. Thrust Measurement Electronic Board

Our thrust measurement board is made with a commercial Arduino Due © board and a shield board [5]. The Arduino Due board is based on the Atmel© SAM3X8E ARM Cortex-M3 CPU and has 54 digital input/output, 12 analog inputs, 84 MHz clock.

One of remarkable advantages of choosing Arduino Due for our system is that it has two independent 12 bits DAC's (digital analog converters). Thanks to those DAC's, we can easily monitor digital signals inside the Due board on an oscilloscope. A shortcoming however is that the Due board runs at 3.3V and therefore a level matching is essential between the Due board and other sensor boards that run at 5.0 V.

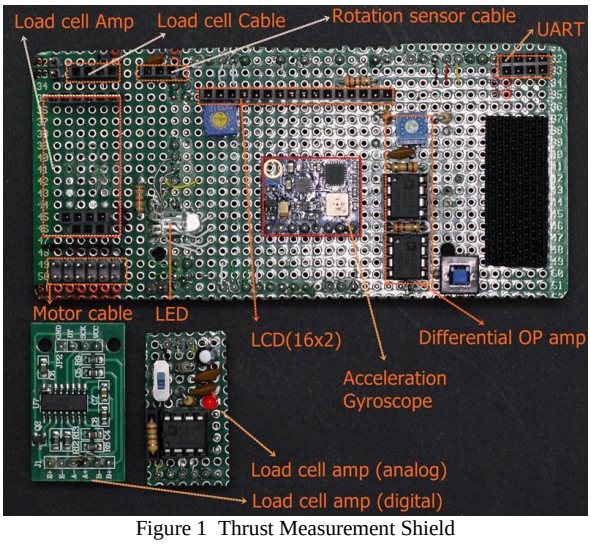

Figure 1 Thrust Measurement Shield

A photo of our shield board and signal amplifiers is shown in Fig. 1. The shield has cable sockets for a load cell, an external analog load cell amplifier and a photo (rotor speed) sensor. It also includes sockets for both analog and digital load cell amplifiers, both are shown in Fig. 1. The analog load cell amplifier is made with an instrumental amplifier (AD 623 ©) from *Analog Devices* [6] and the digital amplifier is a commercial one which is based on the *HX711* scale amplifier from *Avia Semiconductor* [7]. We have found unfortunately that the low-cost digital amplifier is robust to electrical noises but rather slow (around 80 Hz). In addition, the analog amplifier is very sensitive to external noises and absolutely needs a low-pass analog filter which also limits the bandwidth of a measurement. Depending on applications, those two amplifiers can be used for convenience but for a precise measurement we have also used an external load cell amplifier *P-3500 Strain Indicator* © from *Vishay* [8]. The analog output of P-3500 is connected to the same analog output socket for the analog amplifier.

Our shield also has two independent instrumental differential amplifiers (AD 623 ©) which convert the output range 0.55~2.75 V of the DAC in Arduino Due to a full voltage range 0~3.3 V. Furthermore, our shield is equipped with a standard 16x2 text LCD display and an acceleration/gyroscope sensor MPU6050© from *InvenSense* [9] for future use as a flight controller board. Fig. 2 shows a photo of our measurement boards and a load cell amplifier P-3500.

B. External Sensors

Two external sensors are used for our thrust measurement system. A strain gauge type load cell is mechanically connected to the stationary part of a BLDC (Brushless DC) motor and an optical sensor is installed below a propeller as shown in Fig. 2. For better responses of optical sensor, a reflective tape is attached to the back of a propeller.

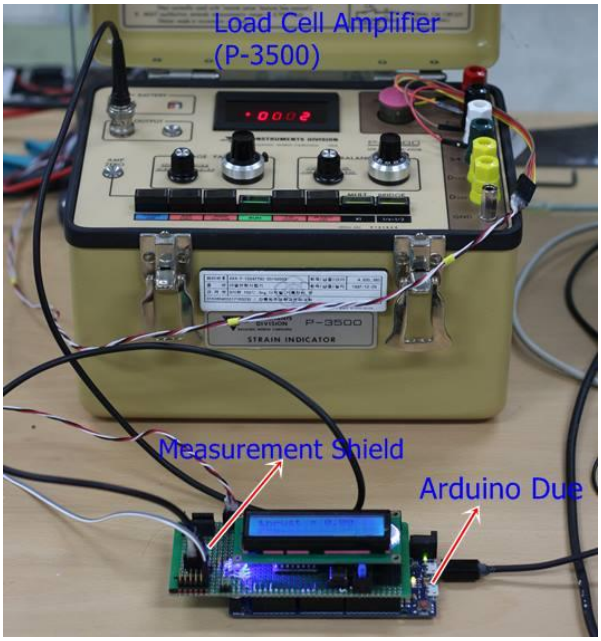

Figure 2 Thrust Measurement System Setup

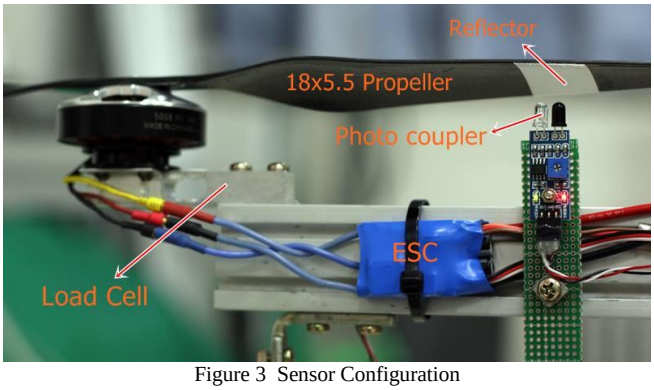

Figure 3 Sensor Configuration

C. MCU Firmware

An Arduino firmware was developed for our thrust measurement system. The analog (thrust force) sensor signal from a load cell amplifier is captured by a 12 bits analog-digital converter inside a Due board. The digital output of an optical (rotor speed) sensor served as an external interrupt for a Due board and the period of a pulse train generated by a propeller rotation was measured with an internal timer of a Due board and then converted to the rotor RPM (rotation per minute).

The factory firmware of our ESC (electrical speed controller) was replaced by a custom one provided by *BLheliSuite 14.2.0.1* [11]. This allowed us to use a 4 kHz PWM signal as a driving signal for the ESC. More details on the driving signal of ESC can be found in [12].

According to PWM duty ratios varying from a given lower bound toward an upper bound, both the static thrust force and rotor speed were simultaneously measured. Those data were sent to both a computer and the internal DAC for a monitoring at an oscilloscope. Fig. 4 gives a screenshot of a computer. Moreover the source code for our firmware can be found in Appendix A.

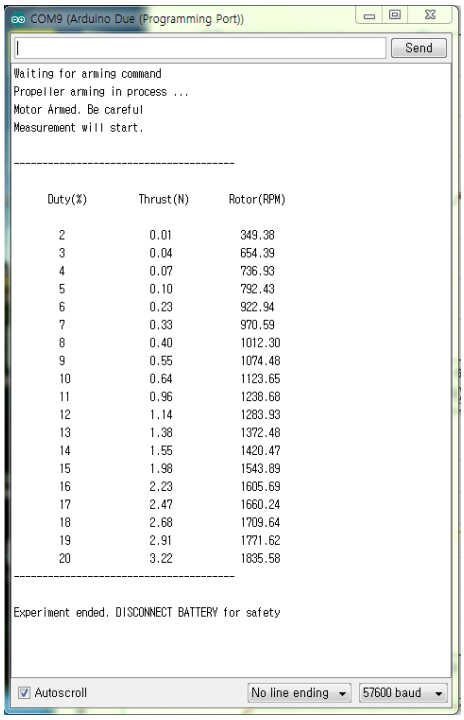

Figure 4 Screenshot of Computer

TABLE I. COMPONENTS SPECIFICATION [4]

BLDC Motor	Motor Outer Diameter	58.5 mm
	Stator Diameter	50.0 mm
	Speed per Volt	340 RPM /V
	Stator Number	12
	Motor Poles	14
	Weight	168 g
Propeller	Length	18 inches
	Pitch	5.5 inches
	Material	carbon fiber
	Blade Root Thickness	3.3 mm
Load Cell	Capacity	5 kg
	Resistance	1000 Ω
	Material	Aluminum
	Nonlinearity	0.05 %
ESC	Output (continuous)	40 A
Battery	Type	LiPo
	Capacity	10000 mAh
	Nominal Voltage	22.2 V
	Discharging Rate	25C

III. VERIFICATION

For a verification, we tested our measurement system for the same propeller actuator system studied in [4]. Several technical specifications of components used in this experiment are given in Table 1 cited from [4].

The relation of a PWM duty ratio and a thrust force was found to be as in Fig. 5. As our system can run without interventions of human being, the measurement data (blue line with 99 data points marked with small filled dots) in Fig. 5 could be obtained only in a couple of minutes. In contrast, the manual data (red line with 10 data points marked with large circle) from [4] required much more time and effort. In overall our new data are very close to the old data.

An interesting result in Fig. 5 is that when a duty ratio is more than 90 %, the thrust force does not increase anymore and even decreases sometimes. This phenomenon also appeared in the manual data in [4] but two data points therein are obviously insufficient to make a meaningful observation.

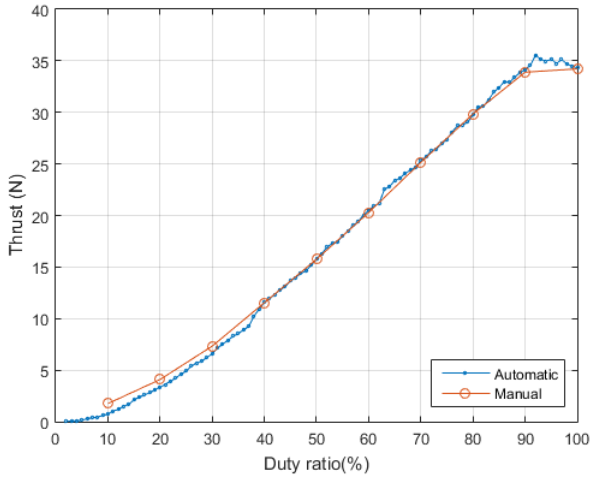

Figure 5 Thrust versus Duty Ratio

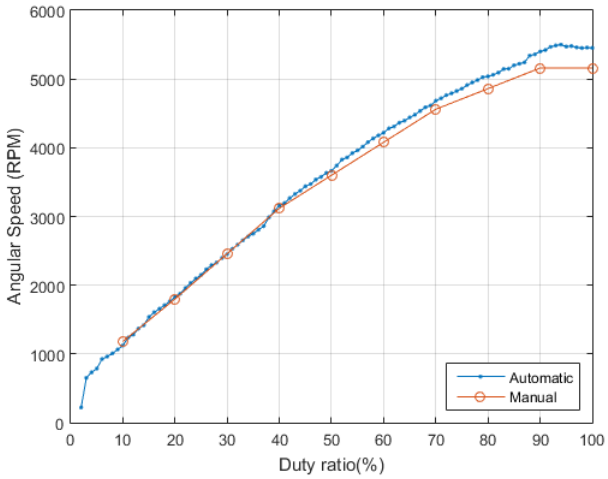

Figure 6 Rotor Speed versus Duty Ratio

Fig. 6 shows a relation between a PWM duty ratio and a rotor speed. Compared to the case of Fig. 5, one can see a relatively large discrepancy between new and old data from [4]. The difference is especially apparent at high rotor speed. This is because, as rotor speed increases, rotational period gets smaller and smaller and thus a manual reading is vulnerable to errors. This result shows an advantage of employing an automatic measurement system.

From data in Fig.5 and Fig. 6, a relation between a rotor speed and a thrust force could be obtained as given in Fig. 7. Because of the aforementioned inaccuracy in manual reading of rotor speed, we can also see some differences between new and old data.

From a quadratic interpolation, it was found in [4] that the quadratic relation between a rotor speed and a thrust force could be approximately given as

$$T_{\exp(old)} = 1.29 \times 10^{-6}\, \omega_{RPM}^2 \qquad (1)$$

We repeated the same interpolation with new data and found a quadratic relation

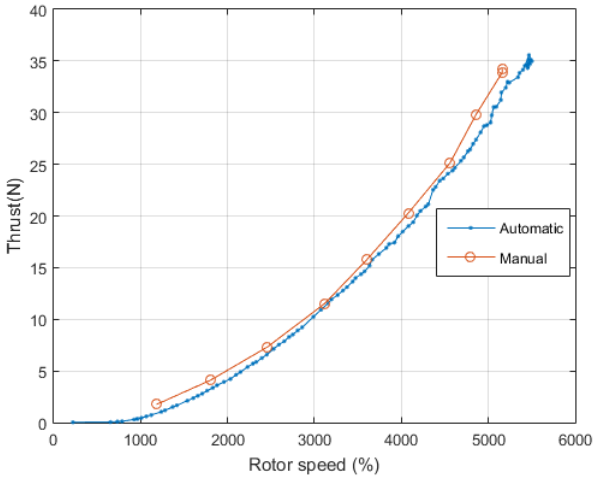

Figure 7 Thrust versus Rotor Speed

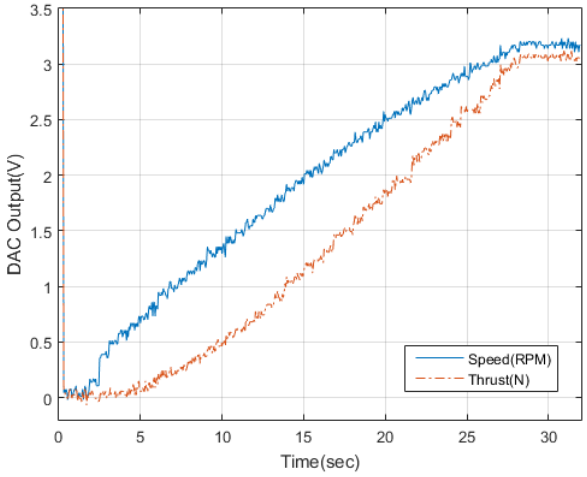

Figure 8 DAC Outputs

$$T_{\exp(new)} = 1.18 \times 10^{-6}\, \omega_{RPM}^2 \qquad (2)$$

which is closer to a theoretical value in [4]

$$T_{theory} = 1.09 \times 10^{-6}\, \omega_{RPM}^2. \qquad (3)$$

Fig. 8 shows analog outputs from the measurement board when a duty ratio varies from 2% to 90 % (step size 2%) regularly for about 30 seconds. The analog output corresponding to the rotational speed should be read by 1/1666.7 (Volt/RPM) and the thrust force by 1/10.61 (Volt/Newton). Details on this conversion can be found in the source code in Appendix A.,

IV. Conclusion

We have developed a microprocessor (Arduino Due)-based thrust measurement system with which one can easily characterize both the static thrust force and rotor speed of a propeller-driven multi-rotor helicopter. From a test run, it was found that our new automatic measurement system can provide more precise information only in a couple of minutes.

Appendix

A. *Arduino Source Code for Measurement Board*

```
// RPM meter
#define rpm_sensor_pin 35
volatile float rpm=200, old_rpm=200;
volatile unsigned long int new_usec=0,old_usec=0;
// UART input data
int incomingByte = 0;
// Motor & Duty Ratio
#define Motor_1 2
#define Arming_duty_ratio 1.5 // (note) Duty=61/4096=1.5 (%)
#define duty_max 90
#define duty_min 2
#define duty_step 2
#define step_delay 300 // perid for one duty ratio (milliseconds)
int duty_ratio=0, nominal_input=0, duty_percent=2;
// Emergency stop pin
#define emergency_stop_pin 7
// Averaging filter length
#define sum_number 500
float thrust=0, thrust_sum=0, rpm_sum=0, rpm_now=0;
// -------------------------- SETUP
void setup() {
  // UART baud rate setup
  Serial.begin(57600);
  // AD/DA Resolution setup
  analogWriteResolution(12);analogReadResolution(12);
  // RPM meter (external interupt setup)
  pinMode(rpm_sensor_pin, INPUT);
  attachInterrupt(rpm_sensor_pin, rpm_sensor,RISING);
  // Emergence STOP (external interupt setup)
  pinMode(emergency_stop_pin,INPUT_PULLUP);
  attachInterrupt(emergency_stop_pin, emergency_stop,FALLING);
  // Analog thrust sensor (AD pin setup)
  pinMode(A8,INPUT); // 2^12 bits
  // Wait arming command (UART0) "r (114)"
  Serial.println("Waiting for arming command");
  while(Serial.available()==0){} // endless waiting
  // Command received
  char inChar = (char)Serial.read();
  if (inChar==114){
        // Arming command run
        Serial.println("Propeller arming in process ...");
        pinMode(Motor_1, OUTPUT);

analogWrite(Motor_1,map(Arming_duty_ratio,0,100,0,4095));
        delay(3000);
        Serial.println("Motor Armed. Be careful");    }
  else {
        // Proceed without Motor arming
        Serial.println("Proceed without arming");delay(1000); }
      // data display
      Serial.println("Measurement will start.\n");
      Serial.println("----------------------------------------\n");
      Serial.println("    Duty(%)      Thrust(N)     Rotor(RPM) \n");
      delay(1000);
      // set minimum duty ratio
      duty_percent=duty_min;
}

void loop() {
  // Check if maximum duty ratio is reached.
  if (duty_percent>duty_max){
    analogWrite(Motor_1,map(0,0,100,0,4095)); // speed down to zero
    Serial.println("----------------------------------------\n");
    Serial.print("Experiment ended. DISCONNECT BATTERY for
safety\n");
    while(1){}; } // infinite loop
  // Change of Motor Speed
  analogWrite(Motor_1,map(duty_percent,0,100,0,4095));
  delay(step_delay);
  // Thrust & RPM averaging  (measurement data) ADC=21.73*Thrust
  thrust_sum=0;rpm_sum=0;
  for (int i=1; i<sum_number; i++){
   thrust_sum+=analogRead(A8);
   rpm_sum+=rpm;
   delay(1);  }
  thrust=thrust_sum/(sum_number*21.73);
  rpm_now=rpm_sum/sum_number;
  // (serial) data display
  Serial.print("\t");Serial.print(duty_percent);
  Serial.print("\t \t");Serial.print(thrust);
  Serial.print("\t \t");Serial.print(rpm_now); Serial.print("\n");
  //
  // DAC data display
  //(RPM)
  // rpm_max=5500 --> DAC_max=4095 -->  Analog_max=3.3 (V)
  //  rpm=1666.7 <--   DAC = 1241  <--   1.0 (V)
  //
  analogWrite(DAC0,0.7445*rpm_now);  //0.7445=4095/5500
  // (Thrust)
  // thrust_max=35 (N) --> DAC_max=4095 -->  Analog_max=3.3 (V)
  // thrust =10.61 (N) <--   DAC = 1241  <--   1.0 (V)
  //
  analogWrite(DAC1,117*thrust); // 117=4095/35
  // Increase duty ratio
  duty_percent+=duty_step;
}
```

```
// ----------------- rpm sensor external interrupt
void rpm_sensor(){
     new_usec=micros();
     if (new_usec!=old_usec){
       rpm=(float)30000000/(new_usec-old_usec);
       old_rpm=rpm;
     }
     old_usec=new_usec;
}
// ----------------- propeller stop external interrupt
void emergency_stop(){
  analogWrite(Motor_1,map(0,0,100,0,4095)); // speed down to zero
}
```

REFERENCES

[1] G. M. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed", Control Engineering Practice, 19(9), pp. 1023-1036, 2011

[2] S. Bouabdallah, P. Murrieri, R. Siegwart, "Design and control of an indoor micro quadrotor", Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, 26 April-1 May 2004.

[3] C. V. Junior Jose, Paula Julio C. De, Leandro Gideon V. and Bonfim Marlio C., "Stability Control of a Quad-Rotor Using a PID Controller", Brazilian Journal of Instrumentation and Control, Control 1.1,pp. 15-20, 2013.

[4] M. Yoon, "Experimental Identification of Thrust Dynamics for a Multi-rotor Helicopter", International Journal of Engineering Research and Technology, 4 (11), pp. 206-209, 2015

[5] "Arduino" Available at: https://www.arduino.cc/ Accessed 14 December 2015].

[6] "Analog Device" Available at: http://www.analog.com/en/index.html Accessed 10 December 2015] Accessed 14 December 2015].

[7] "Avia Semiconductors" Available at: http://www.aviaic.com/ Accessed 14 December 2015].

[8] "Vishay" Available at: http://www.vishay.com/ Accessed 14 December 2015].

[9] "InvenSense" Available at: http://www.invensense.com/ Accessed 14 December 2015].

[10] G. M. Hoffmann, H. Huang, S. L. Waslander, C. J. Tomlin, "Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment", In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, South Carolina, 20-23 Aug. 2007

[11] "BLHeliSuite" Available at: https://blhelisuite.wordpress.com/ [Accessed 16 November 15].

[12] M. Yoon, "On Driving Signal of Electronic Speed Controller for Small Multi-Rotor Helicopter", International Journal of Engineering Research and Technology, 4 (11), pp. 456-459, 2015