



## **A Design and Implementation of UAV Control Environment using Ardupilot**

**Gi-Weon Kim<sup>\*</sup>, Jin-Young Yang**

*School of IT Convergence, Chodang University*

### **A B S T R A C T**

The basic control for the application of unmanned aerial vehicles(UAVs), commonly known as a drone, is through automatic pilot and navigation guidance. In order for this control to function, one must deal with the terrestrial magnetism and acceleration sensors first. In this paper, the ArduPilot board and the Arduino are combined to structure UAV control system. The flight control method was examined, in a processing IDE, while each sensor was incorporated to design and develop an UAV as a quadcopter. With PID control, control parameter  $K_p$ ,  $K_i$ , and  $K_d$  for balancing the flight were calculated by experimental method. Next, the optimal profit value was applied to an UAV with Multiwii Config program. P control could approach the object value flexibly throughout the experiment. Additionally, when it came to the value, it tilted to the heavy direction. After solving the problems in the PI control, it was utilized to achieve a balance through raising the output in the motor of the heavy part in said UAV. When the UAV experiences sudden change in tilt, D control helps to reach the object value quickly. As a result, by using the Multiwii board is a low-cost, it is possible to construct a stable UAV flight environment.

© 2014 KKITS All rights reserved

**K E Y W O R D S** : ArduPilots, UAV, Processing IDE, Multiwii, PID Control, Multiwii Config

**ARTICLE INFO:** Received 14 November 2014, Revised 12 December 2014, Accepted 12 December 2014.

### **1. 서론**

<sup>\*</sup>Corresponding author is with the Department of Smart Aviation Robotics at Chodang University, 419, Muan-cup, Muan-gun, Jeollanam-do, 534-701, KOREA.  
*E-mail address:* kwkim@edu.ac.kr

무인항공기(UAV)는 유인 항공기가 수행하기 어려운 장시간 정찰과 위험지역의 탐사, 재난·재해

감시와 방재 등이 가능해짐에 따라 과거 단순한 군 작전용에서 벗어나 민간용으로 그 수요가 차츰 증가되고 있다. 그러나 무인항공기를 설계하고 제작하기 위해서는 복잡한 비행원리를 이해하고 이를 제어해야 한다. 무인항공기의 운용을 위한 가장 기본적인 제어는 자동조종과 항법유도이다. 자동조종은 비행체의 안전비행을 보장하는 조종 안정성 장치이다. 항법유도는 원거리 목표지점까지 주어진 경로를 따라 오차 없이 비행을 가능하게 하는 임무 제어기이다. 이와 같은 제어가 가능하도록 하기 위해서는 지자기센서, 가속도센서 등과 같은 센서 처리가 우선되어야 한다. 본 논문에서는 아두이노 장비를 결합하여 무인항공기를 제어하는 환경을 구축한다. 이를 위해 아두파일럿을 기반으로 다양한 무인항공기 센서들을 효율적으로 처리하는 환경을 구축하고 UAV를 설계, 제작한 뒤 그 후 PID 제어기를 구축하여 안정적인 비행 환경을 구축하고자 한다.

## 2. 시스템 구현 기술

### 2.1 Processing IDE

Processing은 MIT 미디어 연구소의 Casey Reas와 Ben Fry가 공개한 프로그래밍 언어이다. 컴퓨터를 이용하는 예술가들에게 사용의 편의성을 제공하기 위한 목적으로 제작되었다. 언어 자체의 문법은 자바에서 가져왔지만 사용은 훨씬 단순화되어 있다. Processing의 가장 큰 장점은 간단한 명령으로 시각적 표현이 가능해 3D 그래픽스나 영상 제어 등도 그다지 어렵지 않게 구현 할 수 있다는 것이다. 또한 Wii 리모콘이나 키넥트, 미디어아트 등 다양한 라이브러리를 제공해 하드웨어에 대한 확장성이 좋다. 아두이노 스케치 프로그램 개발 중에, 아두이노 IDE에서 제공하는 시리얼 모니터를

이용하여 디버깅을 할 수도 있지만, Processing IDE를 이용하여 동작 상태를 PC 화면에 그림 형태로 나타내 디버깅을 할 수도 있다.

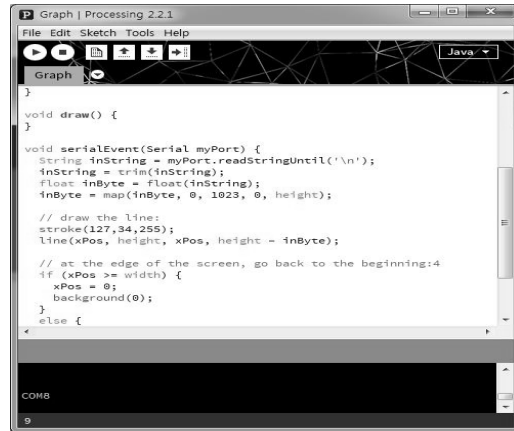


그림 1. Processing IDE  
Figure 1. Processing IDE




즉, Processing은 PC에서 동작하는 IDE로서 아두이노 보드와 시리얼 통신을 통해 전송된 데이터를 화면에 2D나 3D로 쉽게 나타낼 수 있도록 지원한다. Processing은 Mode Manager를 통하여 Android, CoffeeScript, JavaScript, PDE X, Python, Tweak 등 다양한 개발 환경을 제공한다. Processing IDE로 개발된 스케치는 .pde로, 아두이노 IDE로 개발된 스케치는 .ino 파일 타입으로 저장된다[1]. 본 논문에서는 UAV 센서들의 제어 결과를 다양한 시각적인 표현으로 결과를 얻기 위해 Processing IDE 환경을 사용한다.

### 2.2 아두파일럿

아두이노(arduino)는 오픈소스를 기반으로 한 단일 보드 마이크로컨트롤러이다. 아두이노는 다수의 센서로부터 값을 받아들이며, LED나 모터와 같은 외부 액츄에이터들을 제어함으로써 주변 환경과 상

호작용이 가능한 전자 장비들을 만들어낼 수 있다. 아두이노의 가장 큰 장점은 마이크로컨트롤러를 쉽게 동작시킬 수 있다는 것이다. 일반적으로 AVR 프로그래밍이 WinAVR로 컴파일하여, ISP장치를 통해 업로드를 해야 하는 등 번거로운 과정을 거쳐야 하는데 비해, 아두이노는 컴파일된 펌웨어를 USB를 통해 업로드를 쉽게 할 수 있다. 아두이노 보드의 회로도가 CCL에 따라 공개되어 있으므로, 누구나 직접 보드를 만들고 수정할 수 있다[2,8,9]. 아두이노 보드 중에서 UAV 제작에 사용되는 보드를 아두파일럿(Ardupilot)이라 한다. 주로 사용되는 아두파일럿 제품군은 다음의 <표 1>과 같다.

표 1. 아두파일럿 제품군  
Table 1. The Ardupilot Family

| Board            | Multiwii   | Ardupilot mega(2560)   | Ardupilot mega(APM2)   |
|------------------|--|--|--|
| Figure           |  |  |  |
| Processor        | atmega 328P  | atmega 2560  | atmega 32u2  |
| On board sensors | 6 axis gyro/accel<br>3-axis digital magnetometer<br>pressure sensor                | 3-axis gyro,<br>3-axis accel,<br>baro,<br>optional mag                             | 3-axis gyro,<br>3-axis accel,<br>baro, mag   |
| Datalog memory   | 2M   | 2M   | 4M   |
| Size             | 40*40*11   | 40*72*20   | 40*65*10   |

멀티위(Multiwii) 보드는 UAV를 제어하는 비행 제어보드(Flight Control Board)중의 하나이며, 이 제어보드를 구동하기 위한 아두이노 IDE에 기반을 둔 공개 소프트웨어를 Multii라고 한다. Multiwii 소프트웨어 즉, 펌웨어는 아두이노 보드인 uno, Pro Mini, Pro Micro 또는 Mega를 이용하여 업로드할

수 있다. 또한 다양한 센서들이 제공되는데, 3축 자이로와 3축 가속도 센서는 기체의 자기 안정화를 위해 사용되며, 기압 센서는 기체의 고도를 고정할 때에 사용된다. 3축 지자기 센서는 기체의 진행방향을 고정시키기 위해 사용된다.

최근에 Multiwii는 Autopilot-Mega(APM)와 함께 UAV 제어에 널리 사용되고 있는데, Multii는 가격 경쟁력 면에서 APM 보다 우수한 비행 제어보드로 알려져 있다. 즉, APM은 매우 우수한 비행 제어기이지만 고가인 것이 단점이다[11,12]. 따라서 APM에 필적하는 성능을 가지고 있으면서 비교적 저렴한 Multiwii 비행제어기가 인기를 얻고 있다. 본 논문에서도 APM의 성능을 유지하면서 비교적 저렴한 Multiwii 비행 제어보드를 기반으로 UAV에 대한 안정적인 비행 환경을 구축하였다.

### 3. 시스템 설계 및 구현

#### 3.1 UAV 비행제어

UAV의 구조는 <그림 2>와 같이 서로 반대되는 두 쌍의 로터가 서로 반대 방향으로 회전하는 구조를 가지고 있다. 기계적 구조는 십자형의 프레임의 끝에 4개의 모터가 부착되어 있으며, 배터리와 RF 하드웨어가 붙어있다. 대각선 방향의 2개의 로터가 쌍을 이루는데, 로터의 각 쌍은 동일한 방향으로 회전한다. 한 쌍이 시계 방향으로 회전하고 다른 쌍은 반 시계 방향으로 회전을 한다. 이러한 구조의 특징은 회전하는 두 쌍의 모터 속도를 적절하게 제어하면 프로펠러의 회전에서 생기는 반토크가 서로 상쇄되는 특징이 있어 방향 전환과 이착륙과 같은 다양한 운동 방식을 제어할 수 있게 된다[3].

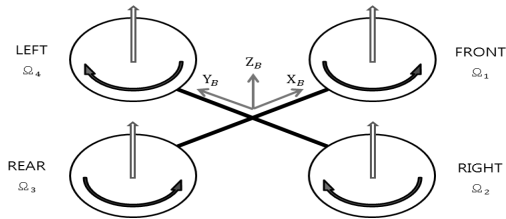


그림 2. UAV의 로터 구조  
Figure 2. Roter Structure of UAV

<그림 2>에서와 같이 에어드론은 4개의 로터를 갖게 되는데, left 및 right 로터 속도를 서로 다르게 하면 roll이 변화되어 에어드론은 전진과 후진을 한다. front 및 rear 로터 속도를 서로 다르게 하면 pitch가 변화되어 좌측 또는 우측으로 이동한다. 각 로터 쌍의 속도를 서로 다르게 하면 yaw 값이 변화되어 에어드론은 좌, 우 회전을 한다[3,4]. pitch, roll, yaw의 변화에 따른 UAV의 비행 제어 방법은 <표 2>와 같다.

표 2. UAV의 비행 제어  
Table 2. Flight Control of UAV

| 로터 동작 | left       | right      | front      | rear       |
|-------|------------|------------|------------|------------|
| 전진    | speed up   | speed down | -          | -          |
| 후진    | speed down | speed up   | -          | -          |
| 좌측 이동 | -          | -          | speed up   | speed down |
| 우측 이동 | -          | -          | speed down | speed up   |
| 좌회전   | speed up   | speed up   | speed down | speed down |
| 우회전   | speed down | speed down | speed up   | speed up   |
| 이륙    | speed up   | speed up   | speed up   | speed up   |
| 착륙    | speed down | speed down | speed down | speed down |

### 3.2 UAV의 센서 연결

본 논문에서는 Multiwii 보드를 이용하여 UAV를 설계, 제작하였다. 세부적인 UAV 제작 방법은 다음의 <그림 3>과 같다. 그림에서 수신기(Receiver)는 무선 조종기(Transmitter)의 수신 장치로 2.4GHz 대역을 사용함으로써 채널당 정보량이 증가하여 양방향 수신까지도 가능하다. 논문에서 사용한 수신기는 6채널로 각 채널에 서보(Servo)가 할당된다면 독립적으로 6개의 기계적 서보를 동작시킬 수 있다.

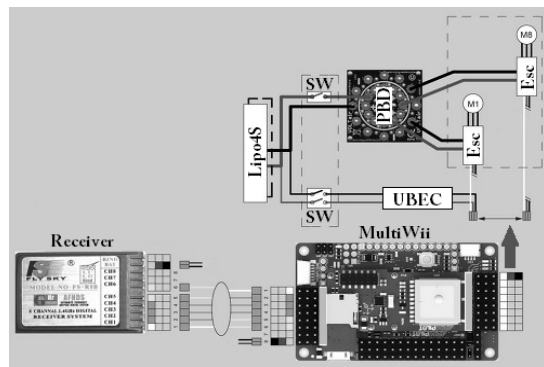


그림 3. UAV 구성  
Figure 3. UAV Configuration

UAV에 전원을 공급하는 전원으로는 LiPo 4S를 사용하였다. 과거에는 니켈-카드뮴 충전지를 사용하였으나 최근에는 리튬이온(LiIon)이나 리튬폴리머(LiPo)를 사용한다. 이는 이러한 충전지가 메모리 효과가 없으며 충전용량과 방전용량이 크게 향상되었기 때문이다. 변속기(ESC)는 'Electronic Speed Controller'의 약어로 Multiwii 보드로부터 오는 신호의 크기에 따라 모터의 속도를 제어하는 장치로 최대 8개 까지 제어가 가능하다. 모터(Motor)는 주로 Brushless DC Motor를 사용하는데 기계적 접점이 없어 소음이 적고 수명이 반영구적이지만, 반면에 제어가 복잡하여 전용 제어기가 필요하고 비싸

다는 단점이 있다. 본 논문에서는 4개의 로터를 갖는 쿼드콥터를 제작하므로 변속기와 모터는 각각 4개씩 장착하였다.

PDB(Power Distribution Board)는 균일하게 전류가 흐르도록 하기 위해 사용하였다. 모터들은 비행 시 수십 혹은 수백 암페어의 전류를 소비하게 된다. 이때 PDB는 배선 사이의 저항을 최소화하면서 4개의 로터로 균일한 전류가 흐르도록 하는 장치이다. 로터 사이에 전류의 차이가 발생하면 추력의 차이로 나타날 수 있는데 이는 안정된 비행을 방해할 수 있다. 따라서 본 논문에서는 PDB를 사용하였다. UBEC(Ultimate Battery Elimination Circuit)는 배터리로부터 공급된 전압을 비행 제어 보드 정격전압으로 맞추어 공급하기 위한 일종의 레귤레이터이다. 본 연구에서는 배터리가 4셀 14.8V이므로 이를 UBEC를 이용하여 5V로 변환하여 APM2.5 보드에 공급하였다. 이전에는 선형 레귤레이터를 사용하였지만 감압되어지는 만큼 전력 손실이 발생하는 문제점이 있어 이를 최소화하기 위하여 Ultimate BEC를 사용하였다.

### 3.2 PID 제어기 구현

UAV 비행 제어보드의 궁극적인 목표는 최대한 짧은 시간에 UAV를 수평자세로 만드는 데에 있다. 자이로 센서, 가속도센서 등에서 보내주는 현재 상태와 수평상태의 오차를 최대한 짧은 시간 내에 없애도록 각각의 모터에 명령을 주는 것이 비행 제어보드의 역할이다. 이를 위해 사용되는 것이 PID(Proportional-Integral-Derivative controller) 제어기이다[6,7]. PID 제어기는 실제 제어 응용분야에서 가장 많이 사용되는 대표적인 형태의 제어기법이다. PID 제어기는 기본적으로 피드백 제어기의 형태를 가지고 있으며, 제어하고자 하는 대상의 출력값(output)을 측정

하여 이를 얻고자 하는 참조값(reference value) 혹은 설정값(setpoint)과 비교하여 오차를 계산하고, 이 오차값을 이용하여 제어에 필요한 값을 계산하는 구조로 되어 있다.

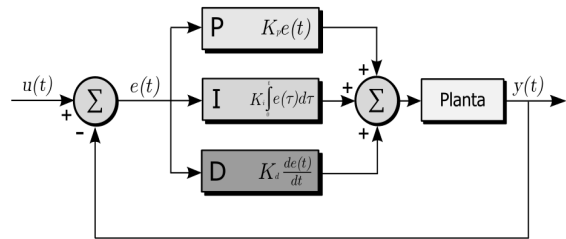


그림 4. PID 제어기  
Figure 4. PID Control

비례제어 P는 PID 제어기에서 반드시 사용해야 하는 가장 기본적인 제어이며 구현하기가 쉽다. 하지만 비례제어 만으로는 시스템에 적분기가 없을 경우에 정상 상태 오차가 발생할 수 있다. 적분제어 I는 정상상태 오차를 없애기 위해 사용된다. 미분제어 D는 안정성에 기여하고, 예측기능이 있어 응답속도를 빠르게 한다.

PID의 출력값은  $u(t) = P + I + D$  로 표현된다. P는 error에  $K_p$ 를 곱한 값이다. I는 처음 시작부터 error에 시간의 변화량과  $K_i$ 를 곱하여 계속 더한 값이다. 이 code에서는 1초마다 I를 0으로 초기화 시킨다. D는 error의 변화량 값을 시간의 변화량으로 나눈 후 high frequency noise를 제거하기 위한 과정을 거친 값에  $K_d$ 를 곱한 값이다.

본 논문에서는 아래의 식1을 이용하여 PID 제어 값을 계산하였다. 계산에 사용된 항들은 각각 오차값, 오차값의 적분(integral), 오차값의 미분(derivative)에 비례한다.

$$u(t) = K_p e(t) + K_i \int e(\tau) d\tau + K_d \frac{de}{dt} \quad (1)$$

비례, 적분, 미분, 이 세 개의 항들이 갖는 의미는 다음과 같다.

- 비례항 : 현재 상태에서의 오차값의 크기에 비례한 제어작용을 한다.
- 적분항 : 정상상태(steady-state) 오차를 없애는 작용을 한다.
- 미분항 : 출력값의 급격한 변화에 제동을 걸어 오버슈트(overshoot)을 줄이고 안정성(stability)을 향상시킨다.

### 3.3 지글러-니콜스 PID 튜닝 실험

위의 식1에서 제어 파라미터  $K_p$ ,  $K_i$ ,  $K_d$ 를 이득값(gain)이라고 하고, 적절한 이득값을 수학적 혹은 실험적/경험적 방법을 통해 계산하는 과정을 튜닝(tuning)이라고 한다. 최적의 PID 이득값을 선정하는 방법에는 시스템의 수학적 모델을 이용해 계산하는 방법이 있으며, 수학적 모델링이 어려워 직접 실험적으로 구하는 방법이 있다. 이와 같이 실험에 의한 무모델 PID 계수조정법 중에서 가장 널리 알려진 것은 지글러-니콜스 방법이 있다. 본 논문에서는 대상 시스템이 원점에 극점을 갖거나 불안정한 경우에도 적용할 수 있는 지글러-니콜스 방법인 임계진동법을 사용하여 PID 튜닝 실험을 하였다. 이 방법은 대상 플랜트에 비례제어기만을 적용하여 비례계수값부터 증가시키면서 출력이 지속진동(sustained oscillation) 나타날 때의 임계이득(critical gain)과 임계주기(critical period)로부터 PID 계수를 조정하는 방법이다[5].

P제어는 쿼드콥터의 목표치에 현재 기울기의 차와 비례해서 조작량을 변화시키는 제어로 PID제어에서 가장 기본이 되는 제어이다. 목표치에 유연하게 접근이 가능하지만 목표치에 도달하면 UAV는 무거운 쪽으로 기울어진다. PI제어는 P제어의 문제점을 보완하여 UAV의 무거운 쪽의 모터출력을 높

여 중심을 잡을 수 있게 해준다. 하지만 안정화까지 시간이 오래 걸린다는 문제점이 있다. D제어는 UAV의 급격한 기울기 변화가 발생한 경우 신속하게 기울어진 쪽의 모터출력을 높여서 목표치에 도달할 수 있게 한다.

UAV의 PID 제어 코드는 앞서 기술한 식1을 기초로 작성하였다. 실험에 사용된 제어코드는 다음과 같다.

- $kis = kis + K_i * 0.01 * angl$ ;
- $pid = K_p * angl + kis + K_d * (ang1 - ang2)$ ;
- $angles2 = angles1$ ;
- pid : 조작량
- kis : 오차 누적값
- $K_p$  : 비례상수
- $K_i$  : 적분상수
- $K_d$  : 미분상수
- $ang1$  : 현재 기울기
- $ang2$  : 이전 기울기

제어 파라미터  $K_p$ ,  $K_i$ ,  $K_d$ 는 각각 P, I, D의 상수값으로서 실험을 통해서 값을 변화시켜 최적의 이득값을 계산하였는데, 목표치 설정은 P, I, D 순으로 실행하였다.

표 3. PID 제어값  
Table 3. PID Control Value

| 파라미터       | 초기값   | 최종 이득값 |
|------------|-------|--------|
| 비례상수 $K_p$ | 4     | 4.50   |
| 적분상수 $K_i$ | 0.035 | 2.50   |
| 미분상수 $K_d$ | -15   | 50.00  |

<표 3>과 같이 본 실험에서는 튜닝의 초기값을 각각 P는 4, I는 0.035, D는 -15 값으로 설정한 뒤 실험을 하였다.

본 실험결과 제어 파라미터 상수값이 각각 P는 4.50, I는 2.50, D는 50.00에서 가장 안정적으로 비행 균형을 유지하였다.

### 3.4 PID 이득값의 설정

본 논문에서는 멀티위컨피그(Multiwii Config) 프로그램을 이용하여 UAV의 센서 설정이나 PID의 제어값 변경을 처리하였다. <그림 5>와 같이 멀티위컨피그와 멀티위 보드를 시리얼(UART) 통신으로 연결을 한 뒤, 멀티위컨피그에서 가상포트를 선택한다. 연결이 된 후 3.3절에서 계산한 이득값으로 멀티위컨피그 PID 설정값을 변경한 후 WRITE 버튼을 눌러 UAV에 저장을 한다. 그 후 안정적인 비행을 하는지 시험비행을 실시하는 과정을 반복하였다.

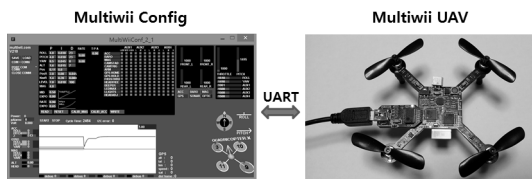


그림 5. PID 이득값 설정 환경  
Figure 5. Environment of PID Gain Setting

## 4. 결 론

무인항공기를 운용하기 위해서는 복잡한 비행원리를 이해하고 이를 제어해야 한다. 무인항공기의 기본적인 제어는 자동조종과 항법유도이다. 본 논문에서는 아두파일럿의 한 종류인 멀티위 보드를 사용하여 무인항공기 제어 환경을 구축하였다. 우선 Processing 통합개발환경에서 UAV의 비행제어 방법을 분석하였으며, UAV에서 사용되는 각각의 센서들을 결합하여 쿼드콕터 형태의 무인항공기를 설계, 제작하였다. 그 후 PID 제어를 사용하여

안정적인 비행 균형을 유지할 수 있는 제어 파라미터  $K_p$ ,  $K_i$ ,  $K_d$ 를 실험적 방법을 통해 계산하였다. 계산된 최적의 이득값은 Multiwii Config 프로그램을 사용하여 UAV에 적용하였다.

실험을 통해 P제어는 목표치에 유연하게 접근이 가능하지만 목표치에 도달하면 무거운 쪽으로 기울어짐을 알 수 있었다. P제어의 문제점을 보완하기 위해 UAV의 무거운 쪽의 모터출력을 높여 중심을 잡을 수 있도록 해주는 PI제어를 사용하였다. 하지만 안정화까지 시간이 오래 걸린다는 문제점이 있었다. 최종적으로 D제어를 사용하여 UAV의 급격한 기울기 변화가 발생한 경우 신속하게 목표치에 도달할 수 있게 하였다. 그 결과 APM의 성능을 유지하면서 비교적 저가인 Multiwii 비행 제어보드를 기반으로 UAV에 대한 안정적인 비행 환경을 구축할 수 있었다.

## References

- [1] [http://en.wikipedia.org/wiki/Processing\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Processing_(programming_language))
- [2] <http://arduino.cc/en/Guide/Introduction>
- [3] G.-W. Kim, M.-S. Yang, and J.-Y. Yang, *A implementation of flight control environment for MAV*, Proceedings of the 12nd Autumn Conference of The Korean Knowledge Information Society, pp. 1-6, 2013.
- [4] <https://projects.drone.org/embedded/drone-api/index.html>
- [5] S.-V. Kim, and D.-Y. Lee, *An estimation method of critical oscillation parameters for ziegler-nichols the 2th PID tuning rule*, Proceedings of the 12nd Autumn Conference of Society of Mechanical Engineers, pp. 23-28, 2013.
- [6] H. M. Shin., *PID controller*, KAIST.

[7] Y. S. Kang, *UAV control system Developing trend*, KARI, pp. 55~60.

[8] "Arduino", Wiki, [Http://ko.wikipedia.org/wiki/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8](http://ko.wikipedia.org/wiki/%EC%95%84%EB%91%90%EC%9D%B4%EB%85%B8).

[9] "Ardupilot, All of requirement for Repor t", [Http://copter.ardupilot.com/wiki/introduction](http://copter.ardupilot.com/wiki/introduction)

[10] "Proximity sensor", <http://cafe.naver.com/chomdan11/130>

[11] "APM 2.6 Set", <https://store.3drobotics.com/products/apm-2-6-kit-1>.

[12] <https://projects.ardrone.org/embedded/ardrone-api/index.html>

보완하여 UAV의 무거운 쪽의 모터출력을 높여 중심을 잡을 수 있도록 P제어를 사용하였다. 최종적으로 D제어를 사용하여 UAV의 급격한 기울기 변화가 발생한 경우 신속하게 목표치에 도달할 수 있게 하였다. 그 결과 저가인 아두파일럿 Multiwii 보드를 이용하여 안정적인 UAV 비행 환경을 구축할 수 있었다.

## 아두파일럿을 이용한 UAV 제어 환경의 설계 및 구현

김기원, 양진영

초당대학교 IT융합학부 스마트항공로봇

### 요 약

무인항공기의 운용을 위한 가장 기본적인 제어는 자동조종과 항법유도이다. 이와 같은 제어가 가능하기 위해서는 지자기센서, 가속도센서 등과 같은 센서 처리가 우선되어야 한다. 본 논문에서는 아두파일럿 보드와 아두이노를 결합하여 무인항공기 제어 환경을 구축하였다. 우선 Processing 통합개발환경에서 UAV의 비행제어 방법을 분석하였으며, UAV에서 사용되는 각각의 센서들을 결합하여 쿼드콥터 형태의 무인항공기를 설계, 제작하였다. 그 후 PID 제어 방법으로 안정적인 비행 균형을 유지할 수 있는 제어 파라미터 Kp, Ki, Kd를 실험적 방법을 통해 계산하였다. 계산된 최적의 이득값은 Multiwii Config 프로그램을 사용하여 UAV에 적용하였다. 실험을 통해 P제어는 목표치에 유연하게 접근이 가능하지만 목표치에 도달하면 무거운 쪽으로 기울어짐을 알 수 있었다. P제어의 문제점을



**Giweon Kim** is an associate professor in the Department of Smart Aviation Robotics at Chodang University, Korea. His current research interests are UAV system and robot software. He is a life member of the KKITS.

E-mail address: kwkim@cdu.ac.kr



**JinYoung Yang** is an associate professor in the Department of Smart Aviation Robot at Chodang University, Korea. His current research interests are UAV system and traffic control. He is a life member of the KKITS.

E-mail address: jyayang@cdu.ac.kr