

論文

J. of The Korean Society for Aeronautical and Space Sciences 45(5), 401-408(2017)

DOI:https://doi.org/10.5139/JKSAS.2017.45.5.401

ISSN 1225-1348(print), 2287-6871(online)

저가 하드웨어 기반 멀티로터 비행제어 컴퓨터 설계 및 검증

이다솔*, 심현철**

Design and Validation of Low-cost Flight Control Computer for Multi-rotor UAVs

Dasol Lee* and David Hyunchul Shim**

Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology* **

ABSTRACT

This paper describes development and validation processes of a low-cost hardware based flight control computer designed for multi-rotor UAVs. The developed flight control computer controls multi-rotors stable and can handle complex flight missions using an integrated high-performance Linux computer. A complementary filter generates a navigation solution with 500 Hz, and a proposed observer significantly reduces measurement noise. A control algorithm utilizes a feed-forward term computed by a three-dimensional curve fitting method, and it increases tracking performance. The developed flight control system has been fully tested through several test flights, and it can apply to real flight environments.

초 록

본 논문에서는 저가 하드웨어 기반의 멀티로터용 비행제어 컴퓨터의 하드웨어 개발과 방법 및 제어 알고리즘의 설계 및 구현, 그리고 실제 비행실험을 통한 검증 과정을 서술하였다. 개발된 비행제어 컴퓨터는 마이크로컨트롤러를 통하여 멀티로터를 안정적으로 제어하며 통합된 Linux 컴퓨터를 활용하여 복잡한 임무에 대응이 가능하도록 설계되었다. 항법 해는 Complementary Filter를 통하여 500 Hz의 속도로 계산을 수행하고, 멀티로터의 운동모델을 기반으로 Observer를 설계, 측정 잡음을 크게 줄였다. 제어 알고리즘은 3차원 Curve Fitting을 통하여 얻은 Feed-forward Term을 사용하여 반응속도를 크게 향상시켰으며, 다수의 비행실험을 통하여 실제 상황에서 효과적으로 동작함을 확인하였다.

Key Words : Low-cost Hardware(저가 하드웨어), Flight Control Computer(비행 제어 컴퓨터), Multi-rotor(멀티로터), Complementary Filter(보상필터)

1. 서 론

현재에 들어 멀티로터는 다양한 임무수행이 가능하여 많은 분야에서 주목을 받고 있다. 제자

리 비행이 가능한 멀티로터는 항공촬영에 널리 활용되고 있으며, 특정 지역의 감시 및 경찰의 임무를 수행하기도 한다.

이러한 멀티로터는 구조적으로 단순하여 설계

† Received : November 4, 2016

Revised : April 11, 2017

Accepted : April 14, 2017

** Corresponding author, E-mail : hcshim@kaist.ac.kr

및 제작이 간단하나, 동역학적으로는 불안정한 시스템에 해당한다. 따라서 안정적인 비행제어를 위해서는 제어루프의 시간지연 현상을 최소화하여야 하는데, 시중에 판매되는 마이크로컨트롤러는 다양한 주변회로를 내장하여 계측, 제어, 동작(Actuation)을 하나의 칩 안에서 수행이 가능하기 때문에 불안정한 시스템인 멀티로터를 효과적으로 제어할 수 있다. 이러한 마이크로컨트롤러와 저가의 센서를 결합한 오픈소스 기반 비행제어 컴퓨터는 멀티로터를 안정적으로 제어하고 있으며, Paparazzi, Ardupilot, PixHawk, Open Pilot 등이 대표적인 예라 할 수 있다[1-2].

국내의 경우 비행제어 컴퓨터의 개발사례는 많이 존재하나[3-4] 소형 멀티로터에 특화된 저가의 비행제어 컴퓨터의 경우는 그렇지 않다. 본 연구에서는 소형 멀티로터에 적합한 비행제어 컴퓨터를 개발하여 마이크로컨트롤러에서는 기본적인 항법 및 제어를 담당하고, 추가적으로 통합된 Linux 컴퓨터에서는 복잡한 임무를 수행하기 위한 High-level 알고리즘을 실행하는 통합형 비행제어 컴퓨터를 설계하고 검증하였다.

본문에서는 비행제어 컴퓨터의 하드웨어 개발 과정 그리고 마이크로컨트롤러에서 실행되는 항법 및 제어 알고리즘 또한 멀티로터에 탑재하여 수행한 실제 비행실험 결과를 서술하였다.

II. 본 론

2.1 멀티로터용 비행제어 컴퓨터 개발

2.1.1 비행제어 컴퓨터 장점 및 상세 사양

개발된 비행제어 컴퓨터는 마이크로컨트롤러 TM4C129x 및 Linux 컴퓨터 DuoVero CoM을 통합한다. 실시간성이 보장되어야 하는 연산(각종 센서 데이터 수신 및 Low-Level 제어)은 마이크로컨트롤러가 담당하며 실시간성이 필요하지는 않으나 상대적으로 연산부하가 심한 임무 할당 혹은 경로계획 등은 Linux 컴퓨터가 담당하도록 설계하였다. 마이크로컨트롤러는 다양한 주변회로를 내장하여 각종 MEMS 센서에서 직접적으로 데이터를 수신하고, 전자변속기 제어를 위한 PWM 회로 또한 내장하고 있어 실시간 제어에서 시간지연 현상을 최소화 시킬 수 있는 장점이 있다. 본 마이크로컨트롤러는 Linux 컴퓨터와 UART 통신을 통하여 정보를 주고받으며, Linux 컴퓨터는 Bluetooth 및 WiFi 통신기능을 내장하고 있어 별다른 통신모듈 없이 지상국 프로그램과 통신이 가능하다. 또한 Dual-core

Table 1. Low-cost hardware of the FCC

| Item | Name | Per price \$ |
|-----------------|-------------|--------------|
| Microcontroller | TM4C129x | 13.21 |
| 9 Axis IMU | MPU9250 | 4.24 |
| Barometer | MS5611 | 7.08 |
| Linux Computer | DuoVero CoM | 179 |

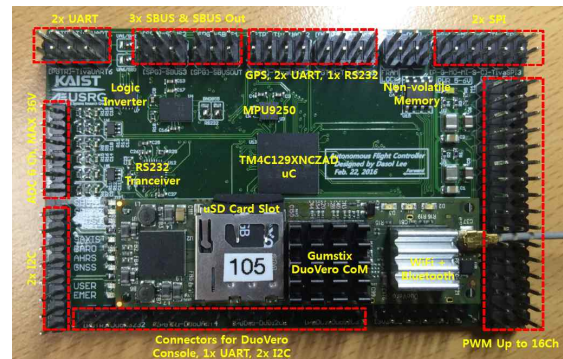


Fig. 1. Developed flight control computer

Table 2. Flight control computer specification

| | |
|--------------------------------|---|
| Size, Weight | 50 x 80 mm, 30 g |
| Peripherals (Micro controller) | Futaba SBUS Input x2, SBUS Output x1, UART x4, RS-232 x1, SPI x2, I2C x2, ADC x6 (up to 36V), PWM x16 |
| Peripherals (DuoVero) | RS-232 x1, UART x1, I2C x2, WiFi, Bluetooth |

1Ghz ARM Cortex-A9 기반 프로세서를 통하여 강력한 연산성을 제공하므로 각종 High-level 알고리즘의 실행이 가능하다. Table 1에서 비행제어 컴퓨터에 사용된 주요 저가 하드웨어 부품 및 가격을 정리하였다. 항법에 사용되는 저가 MEMS 센서로는 MPU 9250 및 MS5611이 있으며, 이를 통하여 항법 해를 도출한다. 비행제어 컴퓨터의 PCB 설계는 EAGLE CAD를 활용하여 설계를 진행하였으며 완성된 비행제어 컴퓨터의 그림은 Fig. 1에 나타내었고, 상세사양은 Table 2에 정리하였다.

개발된 비행제어 컴퓨터는 넉넉한 UART, I2C, SPI, RS-232 인터페이스를 통하여 여러 센서를 연결할 수 있고, ADC를 통해 여러 배터리의 전압을 측정할 수 있다. 또한 크기 및 무게가 50 x 80mm, 30g으로 매우 작고 가벼워 소형 멀티로터에 대응이 가능하다. 그리고 기존의 방식인 FCC와 Single board computer (RaspberryPi,

Odroid 등)를 같이 탑재하는 번거로운 방식에서 벗어나 하나의 소형/경량의 비행제어 컴퓨터로 대체함으로써 비행제어 시스템이 차지하는 임무 장비의 공간 및 무게를 크게 줄일 수 있다.

2.2 항법 알고리즘 설계

2.2.1 자세 추정 알고리즘

멀티로터의 자세를 추정하기 위한 항법필터로는 보상필터를 활용하였다. 보상필터는 다른 항법필터에 비하여 구조가 간단하여 구현이 쉽고, 연산부하가 적기 때문에 마이크로컨트롤러에서 구현될 때 많은 장점을 지니게 된다[5]. 또한 필터의 Tuning parameter가 비교적 적기 때문에 다양한 상황 하에서 적용이 빠른 장점이 있다.

자세추정을 위한 보상필터의 구조는 Fig. 2에 나타내었다. 롤 및 피치각의 추정은 가속도 센서를 통하여 얻은 자세 값에 2차 저주파 통과필터를 적용한 값과 Euler angular rate 값을 적분하고 이를 다시 2차 고주파 통과필터를 적용한 값을 서로 합하여 최종적으로 추정된다. 2차 저주파 및 고주파 통과필터는 식 (1)과 같다.

$$HPF = \frac{s^2}{s^2 + K_{ps} + K_I}, LPF = 1 - HPF \quad (1)$$

가속도 센서를 활용한 롤각 ϕ_{acc} 및 피치각 θ_{acc} 은 식 (2)를 통하여 구할 수 있다.

$$\theta_{acc} = \sin^{-1}\left(\frac{x_{acc}}{g}\right), \phi_{acc} = \sin^{-1}\left(\frac{-y_{acc}}{g \cos\theta_{acc}}\right) \quad (2)$$

여기에서 x_{acc} 및 y_{acc} 는 Specific acceleration의 x, y축 성분이며, Euler angular rate는 식 (3)을 통하여 계산된다.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \cos\theta & \cos\phi / \cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3)$$

위 식에서 p, q, r은 자이로 센서의 x, y, z축 값에 해당한다. 요각의 추정은 롤, 피치각의 추

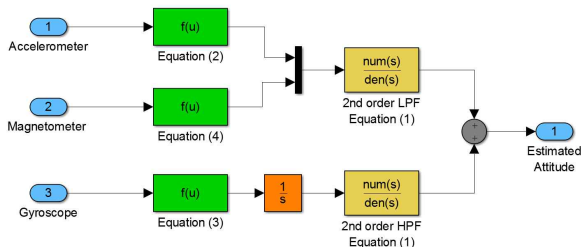


Fig. 2. Complementary filter for attitude estimation

정과 비슷하나, 가속도 센서 대신 지자기 센서를 사용한다. 지자기 센서를 활용한 요각은 식 (4)를 통하여 계산되며 최종적인 요각은 저주파 및 고주파 통과필터를 통과한 값들의 합과 같다.

$$\psi_{mag} = -\arctan2(y_{mag}, x_{mag}) + \delta_{declination} \quad (4)$$

위에서 x_{mag} , y_{mag} 는 지자기 센서의 x, y축의 값이며 $\delta_{declination}$ 은 편차각으로서 대한민국의 경우 약 8°이다.

2.2.2 수평 위치 및 속도 추정 알고리즘

수평축 위치와 속도를 추정하는 알고리즘 또한 보상필터를 활용하였고, GPS 센서와 가속도 센서를 융합한다[6]. 가속도 센서는 짧은 시간에 대해 정확한 움직임을 측정할 수 있으나, 적분오차가 존재하므로 고주파 통과필터를 적용하고 GPS 센서는 누적오차가 없으나 빠른 움직임을 정밀하게 측정할 수 없으므로, 저주파 통과필터를 적용하였다. 최종적으로 각각의 필터를 통과한 값들을 서로 합하여 위치 및 속도를 추정하게 되며 그 구조는 Fig. 3과 같다.

본 알고리즘에 활용되는 가속도는 중력가속도를 보정한 선가속도를 사용하며 이를 적분하고 고주파 통과필터를 통과시킨 값은 기체의 빠른 움직임을 잘 반영하게 된다. 그리고 GPS 속도에 저주파 통과필터를 적용한 값과 서로 합하여 최종적인 속도를 추정하며, 수평축 위치를 추정하는 것 또한 같은 방식으로 수행한다. 수평 축 좌표계는 North-East-Down (NED)좌표계를 활용하였으며 ECEF 좌표계에서 기준으로 삼는 위도 및 경도를 설정하여 좌표변환 후 얻을 수 있다.

본 알고리즘을 통하여 얻은 위치 및 속도는 가속도 센서를 활용하여 빠른 움직임을 정밀하게 측정할 수 있으며, GPS 센서를 활용하여 적분으로 인한 누적오차를 제거할 수 있다. 또한 GPS 센서의 낮은 업데이트 속도 (약 5~10 Hz)를 충분히 보상하면서 효과적으로 기체의 위치와 속도를 추정할 수 있다.

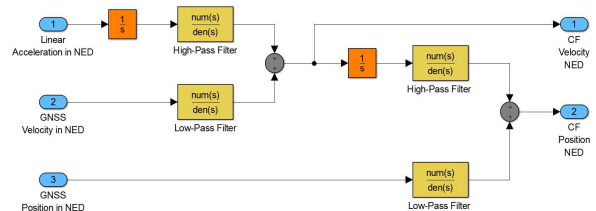


Fig. 3. Complementary filter for horizontal position and velocity estimation

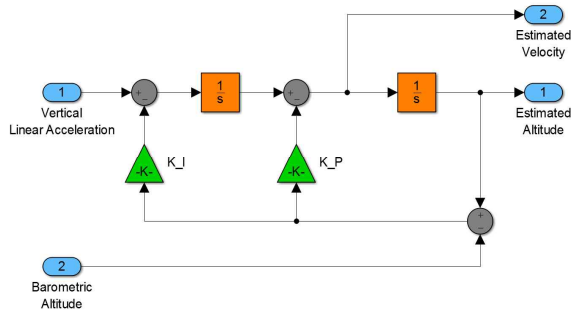


Fig. 4. Complementary filter for vertical position and velocity estimation

2.2.3 수직 위치 및 속도 추정 알고리즘

수직축의 위치와 속도는 기압고도계와 가속도 센서를 결합한 보상필터로 설계하였다[7]. 기압고도계는 노이즈가 심하나 누적오차가 없으므로 저주파 통과필터를 적용하여 효과적으로 노이즈를 제거할 수 있으며, 가속도 센서에 존재하는 노이즈는 적분을 통하여 제거하고 이때의 적분으로 인한 누적오차는 고주파 통과필터를 적용하여 효과적으로 제거할 수 있다. 이 둘을 합하여 최종적으로 수직축의 위치 및 속도를 추정하게 되며, 본 알고리즘의 구조는 Fig. 4와 같고 식 (5)와 같이 표현된다.

$$h_e = \frac{s^2}{s^2 + K_P s + K_I} \left(\ddot{h}_a \right) + \frac{K_P s + K_I}{s^2 + K_P s + K_I} h_b \quad (5)$$

여기에서 h_e , \ddot{h}_a , h_b 는 각각 추정된 수직축 고도, 수직축 선가속도, 기압고도에 해당하며 추정된 수직축 속도 \dot{h}_e 는 식 (5)를 변형하여 식 (6)과 같이 표현할 수 있다.

$$\dot{h}_e = \frac{1}{s} \ddot{h}_a - \frac{1}{s} K_I (h_e - h_b) - K_P (h_e - h_b) \quad (6)$$

위와 같이 식 (5-6)을 활용하면 GPS가 가용하지 않은 실내에서도 수직축의 위치 및 속도를 얻을 수 있게 된다.

2.2.4 항법 알고리즘 실험결과

항법 알고리즘의 실제 성능을 평가하기 위하여 실험을 수행하였다. Fig. 5는 롤, 피치각의 추정결과를 나타내는데 전체적으로 잘 추정됨을 확인할 수 있고, 중간에 가속도의 교란에도 불구하고 이를 효과적으로 필터링 하면서 추정된 각도에는 큰 변화가 없음을 확인하였다. Fig. 6은 요각의 추정결과를 보여주는데 자기장의 교란이 없는 초반에는 지자기 센서 단독으로 구한 요각

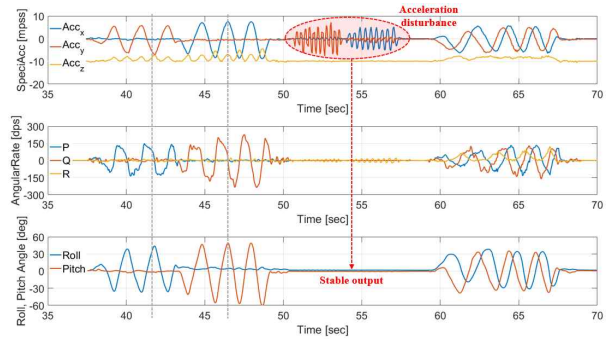


Fig. 5. Roll and pitch angle estimation results

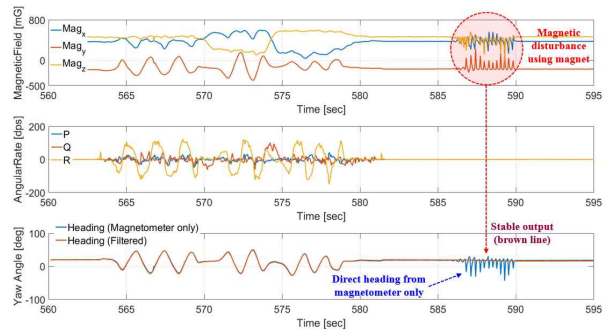


Fig. 6. Yaw angle estimation results

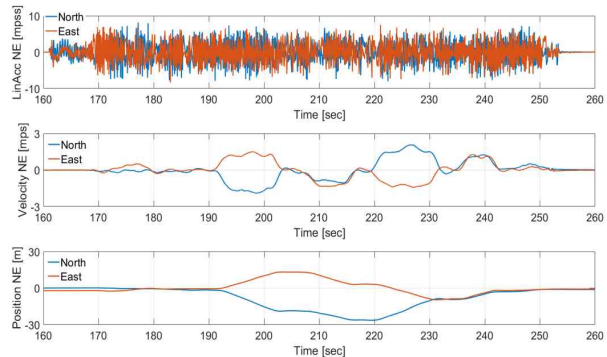


Fig. 7. Horizontal position and velocity estimation results

과 필터를 통하여 추정된 요각의 차이가 거의 없으나, 후반에 자기장의 교란이 존재할 때에는 확실한 차이가 있음을 확인하였다. 따라서 보상필터 기반의 추정된 요각은 주변의 자기장 교란에 강건한 특성을 지니게 된다.

다음으로 Fig. 7은 비행제어 컴퓨터를 멀티로터에 탑재하여 사각형 궤적을 그렸을 때의 추정된 수평축 위치와 속도를 나타낸다. 선가속도는 멀티로터의 모터 회전으로 인해 비행 중 노이즈가 매우 심하게 측정되지만 GPS와 결합하여 항법 해를 도출하기 때문에 추정된 속도와 위치는 노이즈 없이 매우 잘 추정됨을 확인할 수 있다.

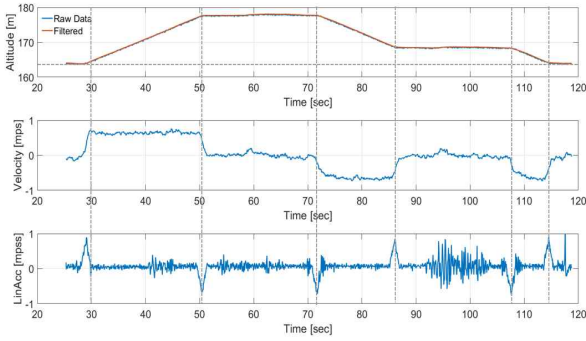


Fig. 8. Vertical position and velocity estimation results

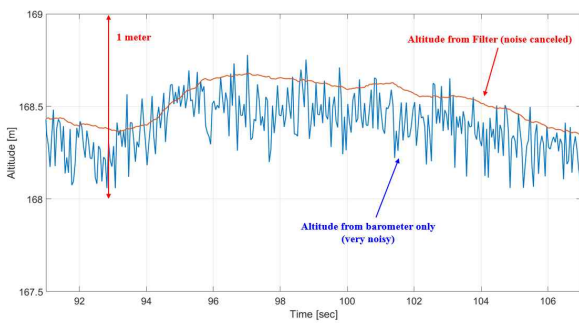


Fig. 9. Vertical position (zoom in)

수직축 속도와 위치의 추정결과는 Fig. 8 및 9에 나타내었고, 엘리베이터 안에서 실험을 수행하였다. 3층에서 출발하여 6층에 도달 후, 다시 4층을 거쳐 3층으로 돌아왔는데, Fig. 8-(a)에서 볼 수 있듯이 수직축 위치가 잘 추정되었음을 확인할 수 있다. Fig. 8-(b)에서는 추정된 속도, Fig. 8-(c)에서는 수직축 선가속도를 보여준다. 한편 Fig. 9는 수직축 고도의 확대된 그래프이며, 기압고도계만을 사용해서 구한 고도(파란 선)에 비해 추정된 결과(갈색 선)는 노이즈가 효과적으로 억제되었음을 확인할 수 있다.

2.3 제어 알고리즘 설계

본 연구에서는 멀티로터를 제어하기 위하여 다중 루프 기반 제어구조를 활용하였고, 가장 바깥쪽 루프에는 선속도(수평/수직) 및 요각 컨트롤러가 존재한다. 선속도 제어는 선가속도 제어를 통해 제어되며, 수평축 선가속도 제어기는 롤, 피치각 제어명령을 생성하고, 수직축의 경우는 스스로 신호를 생성한다. 요각 제어는 요 각속도 제어를 통해 제어되며, 상위루프에서 하위루프를 거쳐 각각의 축에 대한 제어출력은 δ_{ail} , δ_{ele} , δ_{thr} , δ_{rud} 로서 총 4개에 해당한다. 이는 각속도 피드백 제어신호와 더해진 후 최종적으로

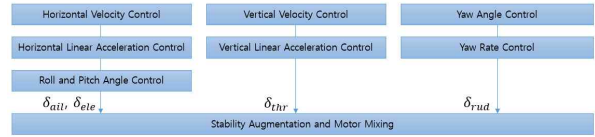


Fig. 10. Multirotor control architecture

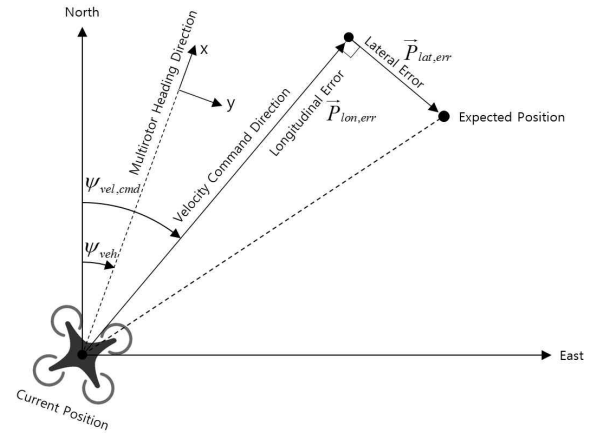


Fig. 11. Velocity tracking algorithm

Motor mixing을 통하여 각각의 모터에 대한 제어신호를 생성한다(Fig. 10 참조).

2.3.1 외란에 강건한 속도추종 알고리즘 설계

외란에 강건한 속도추종을 위해 수평위치를 활용한 횡방향 보상 속도명령 생성 알고리즘을 Fig. 11과 같이 설계하였다. 우선 매 주기마다 예상위치 \vec{p}_{expect} 와 현재위치 \vec{p}_{cur} 를 비교하여 횡방향 위치오차 $\vec{p}_{lat, err}$ 를 계산한다. 이를 보상하는 추가적인 속도명령 $\vec{v}_{lat, comp}$ 을 식 (7)과 같이 생성하며, 이를 기존의 속도명령 \vec{v}_{cmd} 에 더하여 최종적인 속도명령 \vec{v}_{final} 을 식 (8)과 같이 생성한다. 예상위치는 식 (9)를 이용하여 매 주기마다 갱신되며, \vec{v}_{cur} 는 현재속도, $\vec{e}_{vel, cmd}$ 는 속도명령 단위 벡터, Δt 는 제어주기에 해당한다.

$$\vec{v}_{lat, comp} = \vec{p}_{lat, err} K_p \quad (7)$$

$$\vec{v}_{final} = \vec{v}_{cmd} + \vec{v}_{lat, comp} \quad (8)$$

$$\vec{p}_{expect} = \vec{p}_{cur} + \vec{p}_{lat, err} + (\vec{v}_{cur} \cdot \vec{e}_{vel, cmd}) \vec{e}_{vel, cmd} \Delta t \quad (9)$$

횡방향 보상 속도명령은 외란 등으로 발생할 수 있는 속도추종 오차를 효과적으로 상쇄하여 보다 정밀한 속도제어를 가능하게 한다. 속도명령은 PI 컨트롤러를 통하여 선가속도 명령을 생성하고 선가속도 제어기에 전달된다.

2.3.2 선가속도 Observer 설계

선가속도의 노이즈 제거와 멀티로터의 역학적 특성을 반영하기 위하여 식 (10)의 Luenberger observer [8]를 활용하였다. 멀티로터의 x축 선가속도 운동모델은 식 (11)과 같이 근사되며 (x1-y1-z1 좌표계의 x축), 이에 대한 비행데이터는 Fig. 12와 같이 3차원 평면으로 근사된다.

$$\dot{\hat{X}} = A\hat{X} + BU + L(X_{raw} - \hat{X}) \quad (10)$$

$$A_{x1} = c_1\theta_{cur} + c_2V_{x1} \quad (11)$$

A_{x1} 는 x축 선가속도, θ_{cur} 는 피치각, V_{x1} 는 x축 선속도이며, 식 (11)을 미분하면 식 (12)와 같다.

$$\dot{A}_{x1} = c_2A_{x1} + c_1\dot{\theta}_{cur} \quad (12)$$

여기에서 자세제어의 응답을 1차 전달함수와 같다고 하면 $\dot{\theta}_{cur}$ 는 식 (13)과 같이 나타낼 수 있고, θ_{cmd} 는 피치각 명령, a 와 b 는 상수이다.

$$\dot{\theta}_{cur} = b\theta_{cmd} - a\theta_{cur} \quad (13)$$

계속하여 식 (12-13)을 식 (10)에 대입하면 Luenberger observer 수식 식 (14)를 얻는다.

$$\dot{\hat{A}}_{x1} = c_2\hat{A}_{x1} + c_1b\theta_{cmd} - c_1a\theta_{cur} + L(A_{x1,raw} - \hat{A}_{x1}) \quad (14)$$

이렇게 설계된 Observer는 다른 축에 대해서도 같은 방식으로 적용 가능하고, 자세제어 특성 (상수 a , b) 및 저속에서의 항력특성 (상수 c_2)을 반영하기 때문에 선가속도 추정이 매우 정밀하며, 이를 통하여 선가속도 제어 성능을 한층 끌어올릴 수 있게 된다. Fig. 13은 설계된 Observer의 비행실험 데이터를 보여주는데, 추정된 선가속도는 Raw 데이터에 비하여 노이즈가 매우 감소되었음을 확인할 수 있다.

2.3.3 선가속도 제어기 설계

선속도 컨트롤러는 2.3.1에서 서술하였듯이 선

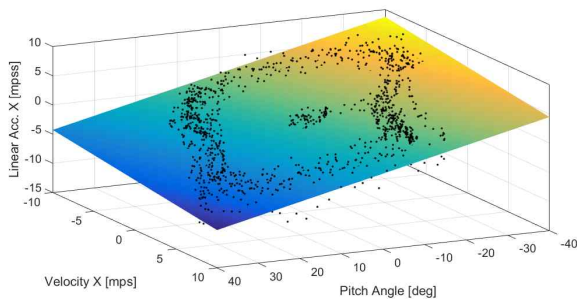


Fig. 12. Flight data according to Equation(11)

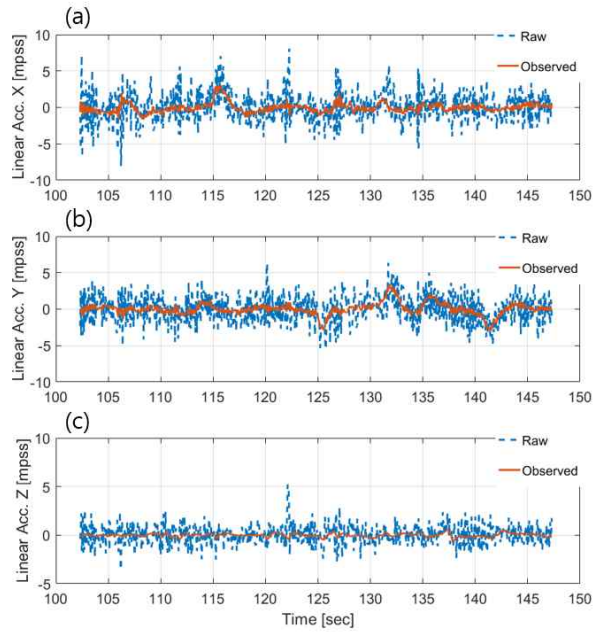


Fig. 13. Luenberger observer for linear acceleration filtering

가속도 명령을 생성한다. 선가속도 컨트롤러는 Feed-forward 및 Feed-back 제어를 동시에 활용하는데 식 (11)을 자세에 대하여 풀면 식 (15)와 같이 Feed-forward 제어를 위한 수식을 얻을 수 있게 된다.

$$\theta_{cmd,FF} = (A_{x1,cmd} - c_2V_{x1})/c_1 \quad (15)$$

Feed-back 제어는 D 컨트롤을 제외한 PI 컨트롤러이며 anti-wind up으로서 forgetting factor를 적용하여 식 (16)과 같다. e_{acc} 는 선가속도 제어오차, K_P 및 K_I 는 제어이득, T_s 는 제어주기, F_f 는 forgetting factor, I_{term} 은 제어오차의 적분항에 해당한다.

$$\theta_{cmd,FB} = e_{acc}(K_P + K_I T_s) + I_{term}(1 - F_f T_s) \quad (16)$$

$$\theta_{cmd,final} = \theta_{cmd,FF} + \theta_{cmd,FB} \quad (17)$$

따라서 최종적인 제어기 Output은 식 (17)과 같이 Feed-forward 및 Feed-back 제어를 더하여 얻을 수 있으며 식 (16)에서 K_I 및 F_f 의 값을 상대적으로 크게 하여 선가속도 제어오차에 빠르게 대응할 수 있도록 설계하였다. 다른 축에 대해서도 위와 같은 방식으로 선가속도 제어기를 설계하여 적용하였다.

2.3.4 자세 제어기 설계

자세제어에서 롤, 피치제어는 수평축 선가속도 제어기의 하위루프에 해당하므로, 선가속도

제어를 위하여 제어된다. 한편 요각 제어는 특별한 상위루프 제어기가 없는 구조로 설계하였기 때문에 경로점 비행 혹은 기타 다른 임무 알고리즘에서 명령 값을 전달받아 제어를 수행한다.

롤, 피치각 제어를 위한 컨트롤러는 PID 컨트롤러를 활용하였고, 제어 결과로서 Fig. 10에서 보듯이 δ_{ail} 및 δ_{ele} 값을 생성한다. 한편 요각 제어는 요각 오차에 P 이득을 곱하여 요 각속도 명령을 생성하는데, 요 각속도 제어는 선가속도 제어기와 동일한 구조로서 식 (18)과 같다.

$$\delta_{rud} = R_{cmd}K_{FF} + e_R(K_P + K_I T_s) + I_{term}(1 - F_f T_s) \quad (18)$$

여기에서 R_{cmd} 는 요 각속도 명령, K_{FF} 는 Feed-forward 이득, e_R 은 요 각속도 오차, δ_{rud} 는 요 각속도 제어의 출력에 해당한다.

2.4 비행실험 결과

2.4.1 옥토로터(멀티로터) 무인항공기 구축

개발된 비행제어 컴퓨터에 설계된 항법, 제어 알고리즘을 구현하여 Fig. 14와 같이 옥토로터에 설치 후 비행실험을 수행하였다. 옥토로터의 자세한 사양은 Table 3에 정리하였으며 제자리 비행 및 경로점 비행실험을 수행하였다.

2.4.2 제자리 비행 실험

Figure 15와 같이 제자리 비행 실험을 수행하

Table 3. Octo-rotor UAV specification

| | |
|----------------|--|
| Size | Diameter : 1.05 m Height : 0.33 m |
| Weight | 4.5 kg (with Battery) |
| Prop Motor ESC | 10*4.5 inch Carbon Prop. dys 650Kv BLDC Castle 40A ESC |
| Battery | 5S1P 5200 mAh 30C |

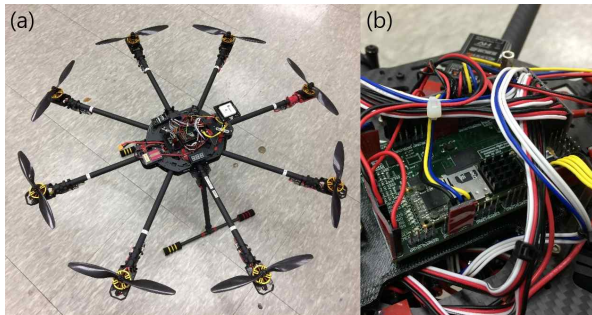


Fig. 14. (a) Octo-rotor UAV, (b) Developed flight control computer placed on the UAV



Fig. 15. Octo-rotor in hovering flight

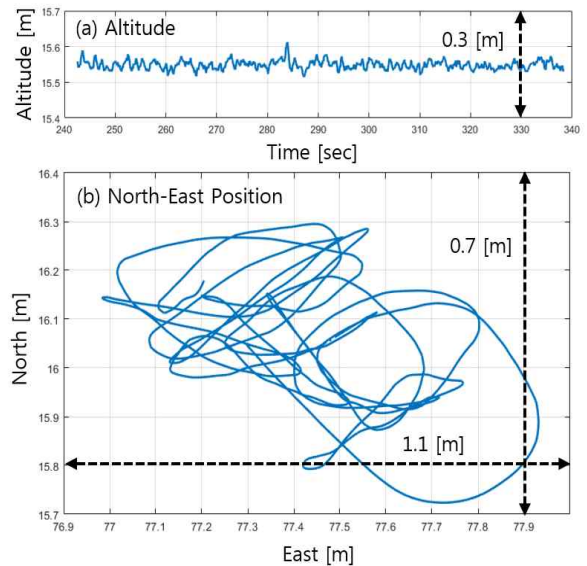


Fig. 16. Hovering flight results

였고, 비행궤적을 Fig. 16에 나타내었다. 총 100초 동안 제자리 비행을 수행하였고 North 0.7, East 1.1 m, Altitude 0.3 m 안에서 매우 정밀하게 제자리 비행이 되었음을 확인하였다.

2.4.3 경로점 비행 실험

경로점 비행실험은 총 4개의 사각형 모양의 경로점을 차례대로 지나가도록 하여 비행실험을 수행하였다. 경로점 비행 알고리즘은 Linux 컴퓨터에서 실행되며, 실행 결과로서 기체의 각 축에 대한 속도추종 명령을 생성하고 UART 통신을 통하여 마이크로컨트롤러에 전달되어 최종적인 비행제어가 이루어지게 된다.

Figure 17에서 옥토로터가 각각의 경로점을 직선의 형태로 정밀하게 비행하였음을 확인할 수 있으며, Fig. 18에서는 각 축에 대한 속도제어 결과를 보여준다. 이는 본 연구에서 설계한 속도추종 알고리즘을 활용한 결과이며, 이로써 실제 환경에 성공적으로 적용됨을 확인하였다.

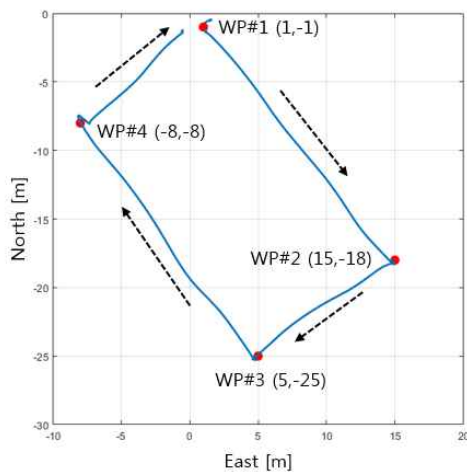


Fig. 17. Way-point flight results

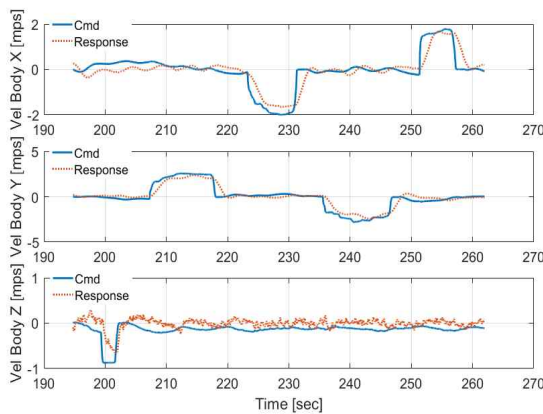


Fig. 18. Velocity control result

III. 결 론

본 논문에서는 소형 멀티로터에 적합한 저가 하드웨어 기반의 비행제어 컴퓨터를 개발하고, 항법 및 제어 알고리즘을 구현하여 실제 비행실험을 통한 검증과정에 대하여 서술하였다. 개발된 비행제어 컴퓨터는 저가 하드웨어 기반으로 기존 값비싼 비행제어 컴퓨터를 대체하면서도 비행성능은 그대로 유지하였으며, 실제 비행실험을 통하여 그 성능을 확인하였다. 개발된 비행제어 컴퓨터는 멀티로터의 다양한 비행임무에 기초가 되는 시스템으로 활용할 예정이며, 지속적으로 항법 및 제어 알고리즘을 개선해 나갈 예정이다.

후 기

본 연구는 산업통상자원부 산업현장핵심기술

수시개발사업의 “소형 무인 멀티콥터용 비행제어시스템 핵심 SW 기술 개발”과제 지원으로 수행되었습니다(과제번호 10060382).

References

- 1) H. Lim, J. Park, D. Lee, and H. Kim, "Build Your Own Quadrotor: Open-Source Projects on Unmanned Aerial Vehicles", *IEEE Robotics & Automation Magazine*, Vol. 19, Issue 3, Sep. 2012, pp. 33-45.
- 2) L. Meier, P. Transkanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision", *2011 IEEE International Conference on Robotics and Automation*, May 9-13, 2011.
- 3) Sung-Han Park, Jae-Yong Kim, In-Je Cho, Byung-Moon Hwang. (2010.2). Redundancy Management Design for Triplex Flight Control System. *Journal of The Korean Society for Aeronautical and Space Sciences*, 38(2), 169-179.
- 4) Hye-jeong Cha, Jae-yong Kim, In-je Cho, Sang-yool Lee, Wun-suk Jeong. (2012.9). Development of UAV Integrated Flight Control Computer Using PowerPC 8349E. *Communications of the Korean Institute of Information Scientists and Engineers*, 30(9), 71-78.
- 5) W. T. Higgins, "A Comparison of Complementary and Kalman Filtering", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-11, Issue 3, February 2007, pp. 321-325.
- 6) J. F. Vasconcelos, J. Calvario, P. Oliveira, and C. Silvestre, "GPS Aided IMU for Unmanned Air Vehicles", *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, July 5-7, 2004.
- 7) Seongpil Kim, Chang-sun Yoo, Oleg S. Salychev, Eung Tai Kim, "Development of Altitude Determination System by Using GPS/INS/Baroaltimeter", *Journal of The Korean Society for Aeronautical and Space Sciences*, 33(6), 51-56.
- 8) D. G. Luenberger, "Observers for Multivariable Systems", *IEEE Transactions on Automatic Control*, Vol. AC-11, No. 2.