

멀티콥터 정밀 유도 제어를 위한 위치 추정기법에 대한 연구

A study of position estimation method for multi-copter accurate guided control

조동훈*, 송용규
한국항공대학교

초 록

본 논문에서는 실외 환경에서 GPS 센서 보다 빠른 주기의 위치 추정 기법을 소개한다. 빠른 제어주기가 필요한 멀티콥터 시스템에서 위치 제어에 GPS 센서를 사용하기 위해서는 상대적으로 느린 GPS 데이터의 출력주기를 보완하는 처리가 필요하다. 이 문제를 해결하기 위해 오픈소스 프로젝트를 분석하고 활용하여 가능성을 확인하는 선행 연구를 진행하였다. matlab에서 알고리즘을 검증하고 임베디드 장비에 직접 구동해봄으로써 성능을 확인하였다. 이 알고리즘에서는 멀티콥터의 위치 추정 계산을 위해 AHRS, GPS센서, barometer 센서를 사용한다.

1. 서론

현재 제자리 비행과 수직 이착륙이 가능한 장점을 가진 멀티콥터는 다양한 분야에서 사용되고 있다. 국내외 많은 업체에서 자동 비행이 가능한 상용 제품들을 출시하고 있는데 실외 비행이 가능한 대부분의 제품에는 GPS 모듈이 탑재되어 있다. 이를 통해 기체의 위치 제어에 GPS를 활용하는 방법이 널리 사용되고 있는 것을 알 수 있다. 그런데 개발과 구현의 관점에서 보면 모터의 회전에 의해 발생하는 양력에 전적으로 의존하는 멀티콥터의 특성상 제어기의 주기가 빠를수록 제어 성능이 향상되는데 상대적으로 느린 GPS의 업데이트 속도가 걸림돌이 될 수 있다. 대표적으로 DJI사의 'phantom', 3DR사의 'solo' 제품에서 사용하는 GPS 모듈은 uBlox사의 제품으로 최대 업데이트 주기가 10Hz[6]로 일반적인 멀티콥터 제어주기의 10분의 1 수준이다. 결론적으로 실시간적인 정밀 유도 제어를 구현하기 위해서 위치 정보에 대한 추정이 필요하다. 본 논문에서는 open source platform 중에서 국내외 많은 사람들에게 검증받은 Pixhawk에 탑재된 알고리즘을 토대로 한 분석과 실험 내용을 다룬다.

2. 본론

2.1 위치 추정 알고리즘 소개

알고리즘 소개에 앞서 Pixhawk[3] 프로젝트는 멀티콥터 오픈소스 프로젝트에서 가장 큰 규모를 형성하고 있다. 상용 제품 'solo' 를 출시한 이력이 있는 만큼 국내학회에도 이를 활용하여 어플리케이션을 시도한 사례들을 찾아볼 수 있다.

Pixhawk에는 두 가지 위치 추정 알고리즘이 존재한다. Extended Kalman Filter를 이용한 추정 알고리즘[1]과 Inertial Filter를 이용한 추정 알고리즘이 있는데 기본적으로 Inertial Filter를 이용한 알고리즘을 사용한다. 본 논문에서는 Inertial Filter를 이용한 알고리즘을 다룬다.

2.2 위치 추정 알고리즘

알고리즘을 정리하면 그림 Fig.1과 같다. GPS 센서, AHRS, Barometer 센서 데이터를 입력받아 GPS 데이터의 정확도를 지표로 각 센서들의 반영률을 조정하여 위치와 속도를 계산한다. 알고리즘의 핵심은 GPS 데이터 품질을 기준으로 가중치가 조정된다는 것이다. 이때 GPS 데이터 품질 지표로 Ublox사의 GPS 모듈에서 제공하는 horizon accuracy[7], vertical accuracy[7] 값을 이용한다. 위 알고리즘을 크게 네 부분으로 세

분화할 수 있고 이를 수식으로 정리하면 식 (1)~(6)과 같다. 여기서 GPS 위치 데이터를 local frame x,y,z로 변환하는 map projection[4]은 처리된 것으로 가정한다.

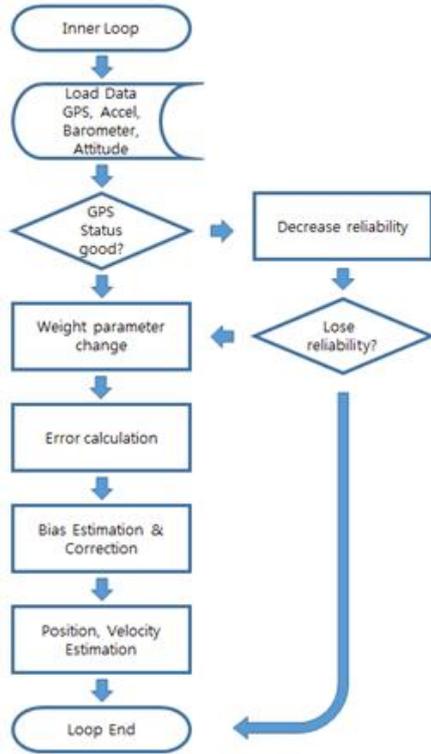


Fig. 1 위치 추정 알고리즘

첫 번째, weight parameter change 블록을 정리하면 식(1)과 같다.

$$\begin{aligned}
 W_{xy} &= \frac{\min eph}{eph} \\
 W_z &= \frac{\min epv}{epv} \\
 W_{p_{xy}} &= K_{p_{xy}} \times W_{xy} \\
 W_{v_{xy}} &= K_{v_{xy}} \times W_{xy} \\
 W_{p_z} &= K_{p_z} \times W_z, \\
 W_{v_z} &= K_{v_z} \times W_z,
 \end{aligned} \quad (1)$$

위 식에서 eph는 horizontal accuracy, epv는 vertical accuracy를 의미하고 W_{xy} , W_z 는 GPS 상황에 따라 1 ~ 0.1의 값을 갖는다. 가중치에 곱해지는 K 값들은 상수로 필터의 반영 비율을 변화시킨다.

두 번째, error calculation 블록을 수식으로 표현하면 식(2)와 같다.

$$\begin{aligned}
 e_p &= P - \hat{P} \\
 e_v &= V - \hat{V}
 \end{aligned} \quad (2)$$

$$\begin{aligned}
 e_{w_b}(x) &= -e_p(x) \times W_{p_{xy}}^2 - e_v(x) \times W_{v_{xy}} \\
 e_{w_b}(y) &= -e_p(y) \times W_{p_{xy}}^2 - e_v(y) \times W_{v_{xy}} \\
 e_{w_b}(z) &= -e_p(z) \times W_{p_z}^2 - e_v(z) \times W_{v_z} \\
 w_{b1} &= C_n^b \times e_{w_b}(x) \\
 w_{b2} &= C_n^b \times e_{w_b}(y) \\
 w_{b3} &= C_n^b \times e_{w_b}(z) \\
 w_b &= \{w_{b1}, w_{b2}, w_{b3}\}
 \end{aligned} \quad (3)$$

a 는 가속도센서 입력 값, A 를 bias가 제거된 가속도, \bar{A} 는 A 를 기체 좌표계에서 관성 좌표계로 좌표 변환된 값, C_n^b 을 direction cosine matrix[1]라 하면 식(4)와 같다. DCM은 3-2-1 변환을 사용[5]하였다.

$$\begin{aligned}
 \bar{A} &= a - \omega_b \\
 \bar{A} &= C_n^b \times A
 \end{aligned} \quad (4)$$

마지막으로 position&velocity estimation 블록을 정리하면 다음과 같다. estimation part는 다시 prediction part와 correction part로 나눌 수 있다. 먼저 prediction part는 식(5)와 같다.

$$\begin{aligned}
 \dot{P} &= P + V \times dt + \frac{1}{2} \bar{A} \times dt^2 \\
 \dot{V} &= V + \bar{A} \times dt
 \end{aligned} \quad (5)$$

그리고 correction part는 식(6)과 같다.

$$\begin{aligned}
 \hat{P} &= \dot{P} + e_p \times W_p \times dt \\
 \hat{V} &= \dot{V} + e_v \times W_p \times dt(1 + W_p) \\
 \hat{V} &= \hat{V} + e_v \times W_v \times dt
 \end{aligned} \quad (6)$$

식(5), (6)을 보면 알 수 있듯이 bias가 제거된 가속도 값과 보정된 속도 값을 이용하여 다음 위치를 추정하는 것을 알 수 있다. 즉, 가속도와 속도 값을 적분하여 위치를 구하고 있으므로 inertial filter라는 명칭이 부여된 것으로 보인다.

위 수식을 matlab과 c코드로 구현하여 알고리즘 검증은 했고 결과는 다음 절에서 다룬다.

2.3 실험 결과

ARM Cortex-M4(168MHz)가 장착된 보드에 Ublox사의 NEO-M8N(GPS센서), Microstrain사의 3DM-GX45(AHRS), AMSYS사의 MS5611 (barometer 센서)을 연결하여 시스템을 구성하였고 100Hz 주기로 추정 알고리즘 연산을 수행하였다.

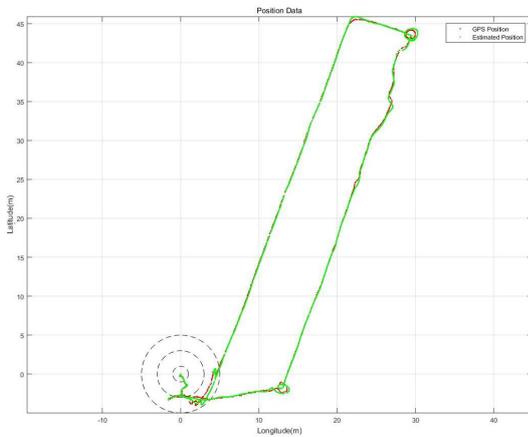


Fig. 2 GPS 위경도와 추정된 위경도 비교

위 그래프는 실험 장비를 들고 운동장 궤적을 따라 걸으며 수집한 데이터를 시각화한 것이다. GPS 센서로 부터 수집한 데이터와 추정 알고리즘을 통해 계산된 데이터를 비교하였다. 위치 데이터는 가속이 발생하는 운동을 제외하면 오차가 1m 이내로 GPS 센서의 업데이트 주기를 보완하기에 충분한 것으로 판단했다. 단, 가속이 있는 운동에서 반경이 커지는 현상이 있는데 실제 멀티콥터로 비행을 해보면 관성이 작용하기 때문에 이러한 특성이 제어에는 적합할 것으로 보인다. 한편, 100Hz 주기로 계산을 수행했을 때 소모되는 시간이 0.7ms ~ 1ms로 측정되었고 이는 제한 조건 10ms 이내를 충족함을 알 수 있다.

3. 결론

본 논문에서는 open source project를 활용하여 위치 추정 기법을 분석하였고 성능을 확인하

였다. 전 세계적으로 오픈소스 프로젝트가 많아지고 있는 추세에서 이를 잘 활용하면 목적에 따라 응용이 가능한 부분이 많을 것으로 생각한다.

실험 결과와 같이 저렴한 GPS 모듈을 사용하여 더 빠른 주기의 위치 데이터를 얻어내려는 목적을 달성하였고 추후에는 실제 멀티콥터 시스템에 탑재하여 비행 성능을 확인할 것이다. 또한, EKF를 활용한 추정 기법과 같이 다른 추정기법을 추가하여 본 결과와 비교를 수행할 것이다.

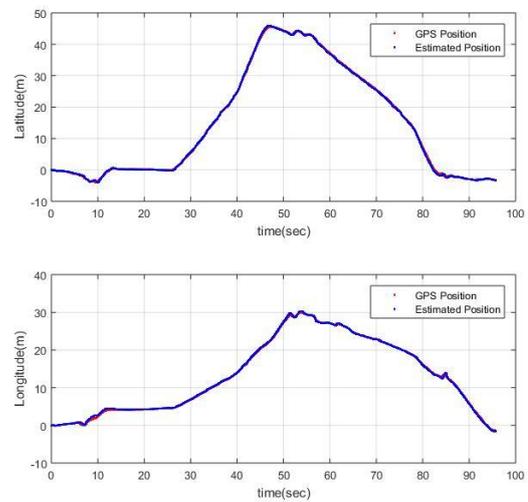


Fig. 3 원점 기준 위경도 비교

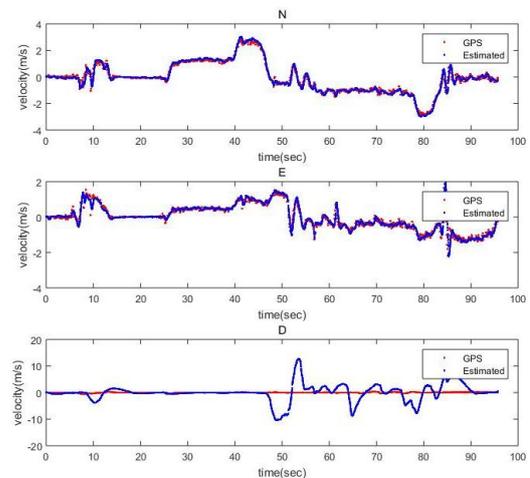


Fig. 4 GPS 속도와 추정된 속도 비교

후 기

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 방송중계차량 탑재형 멀티콥터캠 시스템 기술 개발 사업의 연구결과로 수행되었음.

참고문헌

- [1] Vikas Kumars N, July, 2004, "Integration of Inertial Navigation System and Global Positioning System Using Kalman Filtering"
- [2] Jan Wendel, Oliver Meister, Christian Schlaile and Gert F. Trommer, 2006, "An Integrated GPS/MEMS-IMU navigation system for an autonomous helicopter", volum 10, issue 6, German Aerospace Congress, pp. 527-533
- [3] Software Choice for Pixhawk, PX4 Flight Stack, <https://pixhawk.org/choice>(2016.05.04)
- [4] Melita Kennedy, 1994, "Understanding Map Projections", Environmental Systems Research Institute, Inc, New York.
- [5] Hanspeter Schaub and John L. Junkins, 2003, "Analytical Mechanics of Space Systems", AIAA, USA
- [6] NEO-M8Q/M8M Product Summary, [https://www.u-blox.com/ko/product-resources?f\[0\]=property_file_product_filter%3A2688](https://www.u-blox.com/ko/product-resources?f[0]=property_file_product_filter%3A2688), (2016.01.25.)
- [7] u-blox 8 / u-blox M8 Receiver Description Protocol Specification, [https://www.u-blox.com/ko/product-resources?f\[0\]=property_file_product_filter%3A2688](https://www.u-blox.com/ko/product-resources?f[0]=property_file_product_filter%3A2688), (2016.03.03)