

고고도 장기체공 무인기의 유도 및 방향축 제어 알고리즘 설계

구소연¹ · 임승한^{1†}¹국방과학연구소

Design of Guidance Law and Lateral Controller for a High Altitude Long Endurance UAV

Soyeon Koo¹ and Seunghan Lim^{1†}¹Agency for Defense Development

Abstract

This paper elaborates on the directional axis guidance and control algorithm used in mission flight for high altitude long endurance UAV. First, the directional axis control algorithm is designed to modify the control variable such that a strong headwind prevents the UAV from moving forward. Similarly, the guidance algorithm is designed to operate the respective algorithms for Fly-over, Fly-by, and Hold for way-point flight. The design outcomes of each guidance and control algorithm were confirmed through nonlinear simulation of high altitude long endurance UAV. Finally, the penultimate purpose of this study was to perform an actual mission flight based on the design results. Consequently, flight tests were used to establish the flight controllability of the designed guidance and control algorithm.

초 록

본 논문에서는 고고도 장기체공 무인항공기의 임무 비행을 위한 방향축 유도, 제어 알고리즘에 대해 기술 하였다. 먼저 방향축 제어 알고리즘은 임무 기간 중 무인항공기가 전진비행을 할 수 없을 맞바람에 대해 제어 변수를 전환하는 알고리즘을 설계하였다. 유도법칙은 항로점 비행을 위해 Fly-over, Fly-by, Hold 속성에 대한 각각의 알고리즘을 적용하였다. 무인항공기의 비선형 시뮬레이션을 통해 각 유도, 제어 알고리즘의 설계 결과를 확인하였다. 본 연구는 설계 결과를 토대로 실제 임무 비행을 수행하는 것을 목적으로 한다. 따라서 본 연구 내용을 기반으로 비행 시험을 통해 설계한 유도 제어 알고리즘의 비행 운용성을 확인하였다.

Key Words : High Altitude Long Endurance UAV (고고도 장기체공 무인항공기), Autonomous Flight (자동 비행), Flight Test (비행 시험)

1. 서 론

태양광 장기체공 무인항공기는 내연기관을 사용하는 항공기와 달리 태양전지판을 통해 주간 비행동안 전력을 지속해서 충전하여 장기 체공 임무를 수행하는데 유리하다는 장점으로 국내외에서도 활발히 관련 연구

가 진행되고 있다[1-2]. 하지만 충분한 전력 생산을 위해서는 넓은 면적에 태양 전지를 설치해야 하므로, 기존의 무인항공기 형상에 비해 가로세로 비율이 높은 글라이더 형상으로 제작하고, 주익에 태양전지판을 부착한다. 또한, 고고도 임무를 수행하기 위해 고도 16km 이상의 성층권 영역에서 임무를 장기간 수행하게 되며, 조종의 편의와 통신 두절과 같은 비상 상황에 대기, 귀환을 수행하기 위해 자동 항로점 비행 기능이 요구된다. 자동 항로점 비행을 위한 무인항공기의 유도 기법은 LOS 유도, 트랙 유도, 벡터필드 유도

Received: Nov. 10, 2017 Revised: Apr. 09, 2019 Accepted: Apr. 10, 2019

† Corresponding Author

Tel: +82-42-821-2255, E-mail: shlim@add.re.kr

© The Society for Aerospace System Engineering

등 다양하게 연구 되고 있다[3-5].

장시간 체공을 목적으로 하는 저속 무인기는 임무 수행 중 바람의 영향에 취약하며, 성층권까지 도달하기 위해서 여름 평균 시속 65km의 제트기류의 영향을 받는다. 세계적으로 외란을 보상하여 무인기를 원하는 방향으로 유도 제어 할 수 있는 알고리즘이 연구되고 있으나[6-7], 제트기류와 같이 저속의 항공기가 전진할 수 없을 만큼 강한 바람을 맞게 되면 원하는 경로가 아닌 뒤로 밀리게 되어 자동비행을 수행 할 수 없다. 이에 대해 Byeon 및 Park[8]는 강한 맞바람을 받게 되었을 때 지나온 경로를 다시 추종하며 후진 비행하는 방법을 연구하였다. 이 연구에서는 바람이 다시 약해졌을 경우 단시간에 정상 임무에 복귀하기 위해 지나온 경로를 다시 추종하지만, 고고도 장기체공 무인기는 약 7~12km 까지 제트기류의 영향을 받기 때문에 단시간에 정상 전진 비행을 수행하기 어렵다. 따라서 고고도 장기체공 무인항공기의 유도 제어 알고리즘은 안정한 자세를 유지해주는 비행 제어능력이 요구된다.

본 논문에서는 고고도 장기체공 무인항공기의 임무 운용을 위한 방향측 자동비행 알고리즘을 설계하였다. 먼저, 전진비행을 할 수 없는 상황에서 안정적인 자세를 유지할 수 있는 경로각/기수방향 제어 전환 알고리즘을 구현하였다. 또한, 경로 비행이 가능한 상황에서 지정한 항로점에 대해 자동비행을 수행 할 수 있도록, 직선, 선회비행을 위한 LOS와 벡터필드 유도법칙을 적용하였다. 마지막으로 설계한 알고리즘을 무인항공기의 비선형 모델에 적용하여 시뮬레이션을 수행하고, 실제 비행시험을 통해 장기체공 무인항공기의 임무 운용에 적용 가능한 유도 제어 알고리즘의 성능을 검증하였다.

2. 자동비행 알고리즘 구조

자동비행 알고리즘은 유도법칙, Autopilot, SCAS로 구성되어있으며, 방향측 제어 알고리즘 구조는 Fig. 1과 같다. 방향측 제어를 위한 유도법칙은 항로점 비행을 위한 기수방향 명령을 생성하는 기능을 수행한다. 기수방향 제어기는 유도법칙에서 생성한 기수방향 명령을 추종할 수 있는 롤 명령을 생성한다. 생성한 롤

명령은 SCAS를 통해 최종 조종면 제어 명령을 생성한다.

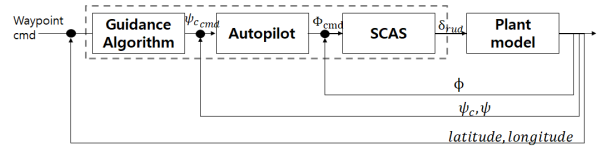


Fig. 1 Structure of Lateral controller

3. 방향측 제어 알고리즘 구조

3.1 방향측 제어 알고리즘 구조

방향측 제어 알고리즘은 경로각(ψ_c) 제어기와, 기수방향(ψ) 제어기로 구성되어있다. 경로각 제어기는 항공기가 원하는 궤적을 추종하기 위한 명령을 생성한다. 경로각 및 기수방향 제어기는 Fig. 2와 같은 구조로 설계하였다. 경로각 및 기수방향 명령에 대해 롤을 생성하며, 급격한 롤 명령을 방지하기 위해 경로각 오차 제한과 기울기 제한을 적용하였다.

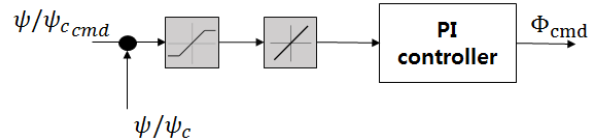


Fig. 2 Structure of Heading and Course controller

3.2 맞바람에 대한 경로각/기수방향 제어 전환 알고리즘

전진비행을 하면서 경로각 제어를 수행하면 항공기의 기수방향은 바람이 불어오는 방향을 지향하게 되어, 바람에 대해서 기체의 비행 안정성을 유지할 수 있다. 하지만 맞바람이 비행대기속도보다 빨라 전진비행이 불가능한 상황에서는 기체가 뒤로 밀리기 때문에 180도에 가까운 경로각 오차가 발생한다. 이에 따라 경로각 제어기는 계속해서 최대/최소로 진동하는 롤 명령을 생성하면서 후진하게 된다. Figure 3, 4는 강한 맞바람에 대한 경로각 제어의 시뮬레이션 결과다. 정북 방향을 향해 비행하는 상황에서 정북풍을 증가/감소시켰으며, 비행대기속도보다 바람속도가 빨라지는 60초 부근에서 후진을 시작하면서 경로각 명령을 추종하기 위해 좌우로 진동하는 것을 볼 수 있다.

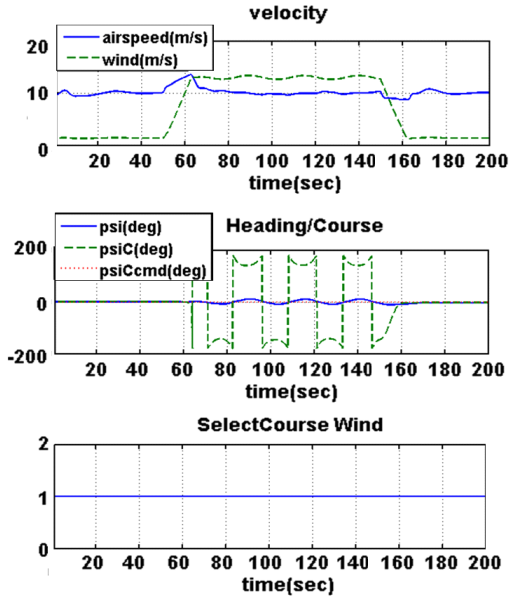


Fig. 3 Course Control Result under Wind (state)

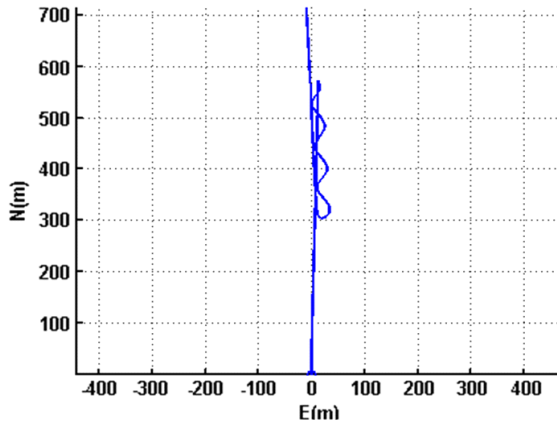


Fig. 4 Course Control result under Wind (trajectory)

맞바람이 비행대기속도보다 빠를 때에는 원하는 경로각으로 비행할 수 없는 상황임에도 이를 제어하기 위해 좌우로 진동하는 비효율적인 비행을 나타낸다. 따라서 본 논문에서는 경로각과 기수방향의 차이를 비교하여 일정 수치 이상의 차이가 발생하면 경로각 제어에서 기수방향 제어로 전환하는 알고리즘을 설계하였다. 먼저 경로각과 기수방향의 차이는 Eq. 1로 나타낸다.

$$\psi_{err} = |\psi - \psi_c| \quad (1)$$

후진 비행 시 경로각에서 기수방향으로 전환하는 조건은 다음 Eq. 2와 같다.

$$\psi_{err} > 60(\text{deg}) \quad (2)$$

여기서, 바람에 의해서 경로각 제어에서 기수방향 제어로 전환했을 때에는 경로각 제어 명령 및 상태변수를 기수방향으로 전환한다. 이후, Eq. 3 조건이 maxTime 동안 지속되면 바람이 잦아들어 전진비행이 가능하다는 것을 의미하므로 기수방향 제어에서 경로각 제어로 전환한다.

$$\psi_{err} < 20(\text{deg}), \text{ during maxTime} \quad (3)$$

여기서, maxTime이라는 판단 조건이 필요한 이유는 모드 전환의 채터링을 방지하기 위함이며, 2초로 설정하였다. 또한 20 및 60도에 해당하는 수치는 물리적으로 유도된 수치가 아니라 시뮬레이션 스타디를 통해서 선택된 값이다.

Figure 5, 6은 경로각/기수방향 전환 알고리즘을 적용하여 Fig. 3, 4와 동일한 비선형 시뮬레이션을 수행한 결과이다. 여기서, SelectCourse Wind 1은 경로각 제어이며 0은 기수방향 제어를 수행하는 상황이다. 시뮬레이션 결과 후진 비행 시에 기수방향 제어로 전환하였으며, 기수방향은 경로각 명령인 0 deg에 근접하기 때문에 0 deg의 기수방향 명령에 대한 기수방향 제어를 수행하면서 경로각 제어를 수행하기 위해 좌우로 진동하는 경향은 발생하지 않고 안정적으로 후진비행하는 것을 확인할 수 있다.

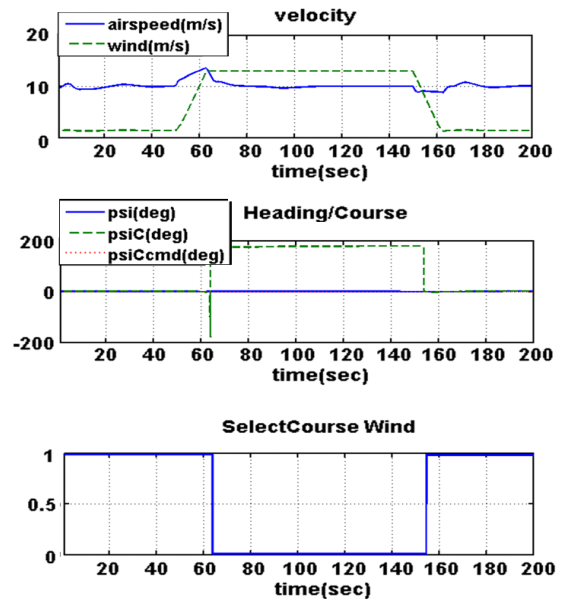


Fig. 5 Course Control Result under Wind with Controller Switch Algorithm (state)

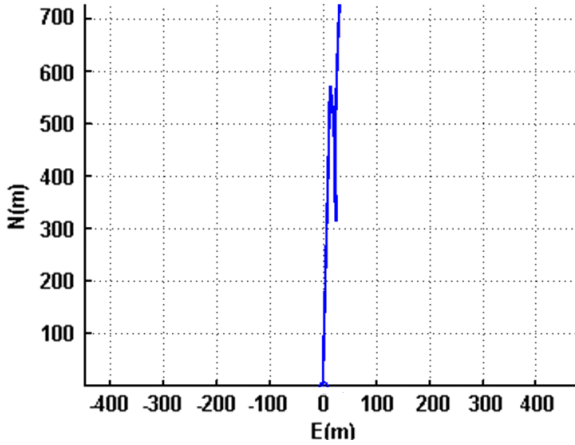


Fig. 6 Course Control Result under Wind with Controller Switch Algorithm (trajectory)

4. 유도법칙 설계

4.1 유도법칙 구조

고고도 장기체공 무인항공기를 위한 유도법칙은 3가지로 구성되어 있다. 먼저, Fly-over는 지정한 항로점을 직선으로 통과하는 유도법칙이다. 다음으로, Fly-by은 지정한 항로점에 도달 전 선회하여 다음 항로점으로 진입하는 알고리즘이다. 마지막으로, Hold는 지정한 항로점에서 일정한 반경을 가지는 원을 따라 선회비행을 하면서 지정한 시간동안 선회 비행을 유지하는 알고리즘이다.

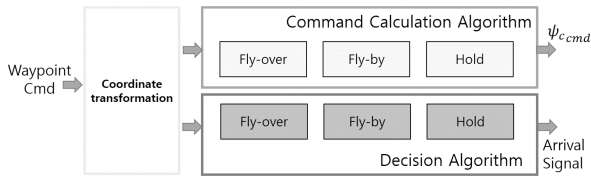


Fig. 7 Structure of Guidance Algorithm

유도법칙은 위도, 경도로 된 각각의 항로점 정보를 이용하여 항로점을 연결하는 가상의 기준 궤적을 생성한다. 먼저, 위도, 경도로 입력된 정보를 임의의 점을 기준으로 각 항로점 좌표를 거리로 환산한다. 따라서 WGS84 기준의 지구의 구면 좌표계의 위도, 경도를 NE(North-East) 좌표계로 변환하고, 유도법칙의 직관적인 연산을 위해 항로점 및 항공기의 위치정보를 유도 좌표계로 정의한다. 유도 좌표계는 Fig. 8 과 같이

자동비행 시작점 또는 과거의 항로점(Passed Way-point)을 기준으로 현재 항로점(Current Way-point)을 연결한 직선을 x 축 방향으로 설정하고, 시계방향으로 90도 회전한 방향을 y 축 방향으로 설정한다. 유도 좌표계는 항로점이 갱신될 때마다 과거의 항로점을 기준점으로 항로점간 상대적인 위치에 대한 좌표변환을 수행한다.

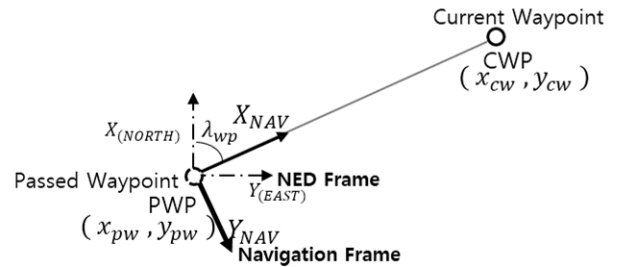


Fig. 8 Geometry between NED Frame and Navigation Frame

NED 좌표계에서 유도 좌표계로의 좌표 변환 행렬은 Eq. 4와 같다.

$$DCM_{NED}^{NAV} = \begin{bmatrix} \cos(\lambda_{wp}) & \sin(\lambda_{wp}) \\ -\sin(\lambda_{wp}) & \cos(\lambda_{wp}) \end{bmatrix} \quad (4)$$

여기서, λ_{wp} 는 항로점 간의 시선각이며 Eq. 5와 같다.

$$\lambda_{wp} = \tan^{-1}\left(\frac{y_{cw} - y_{pw}}{x_{cw} - x_{pw}}\right) \quad (5)$$

항로점에 대한 상대적인 좌표로 변환하고, 설정한 임무의 속성에 따라 유도법칙을 계산하여 경로각 명령을 생성한다. 이와 동시에 지정한 항로점에 대한 도달 상태를 판단하여 OFP(Operational Flight Program)에 도달 신호를 전송하며, OFP가 도달 신호를 수신하면 다음 항로점으로 갱신한다.

4.2 Fly - over 알고리즘

Fly-over 알고리즘은 직선 구간의 유도법칙으로 Fig. 9와 같이 경로각 명령을 생성한다. Fly-over 알고리즘의 경로각 명령은 항로점을 연결하는 직선에 대한 시선각과 기준 궤적으로 부터의 횡방향 이탈거리를 감소하기 위한 경로각 명령의 합으로 나타내며, 다음 Eq. 6과 같다[2].

$$\psi_{cmd} = \psi_{ref} + \Delta\psi_{cmd} \quad (6)$$

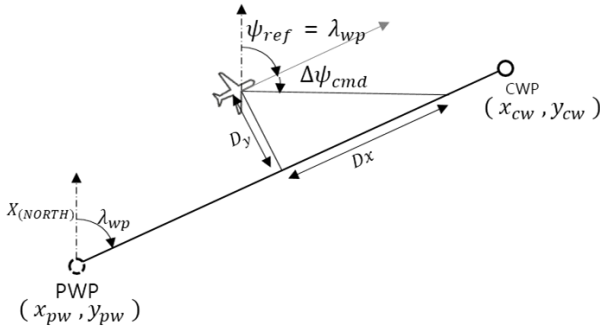


Fig. 9 Fly-over Guidance Algorithm

여기서, ψ_{ref} 는 항로점 간의 시선각 λ_{wp} 과 같은 값이며, $\Delta\psi_{cmd}$ 는 횡방향 이탈거리를 감소하기 위한 추가 경로각 명령이다. 추가 경로각 명령은 일정 시간 (T_{track}) 동안 비행체가 이동한 종방향 거리(D_x)와 기준 궤적과 항공기의 횡방향 이탈 거리(D_y)를 통해 도출된다. 추가 경로각 명령은 Eq. 7과 같다.

$$\Delta\psi_{cmd} = \tan^{-1}\left(\frac{D_y}{D_x}\right) \quad (7)$$

여기서, 종방향 거리(D_x)는 Eq. 8과 같다.

$$D_x = \sqrt{(V_N^2 + V_E^2)} \cdot T_{track} \quad (8)$$

여기서, V_N, V_E 는 항공기의 NE좌표에 대한 속도 성분이다.

Fly-over 알고리즘의 도달 판단조건은 Fig. 10 과 같다.

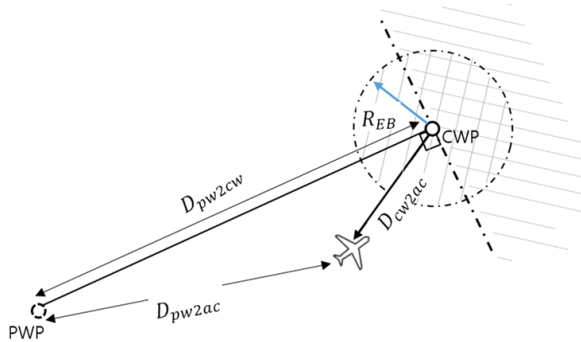


Fig. 10 Fly-over Arrival decision Algorithm

항로점 도달 판단은 항로점의 속성이 Fly-over일 때, Eq. 9와 같이 현재 항로점과 기체 사이의 거리(D_{cw2ac})가 설정한 반경(R_{EB})내에 도달했을 때 항로점 통과를 확인한다. 또는 바람이나 선회 반경의 제약으로 인해

반경 내에는 도달하지 못했지만, 과거 항로점과 현재 항로점 사이의 거리보다 현재 항로점과 기체 사이의 거리가 크면 다음 항로점으로 갱신한다.

$$R_{EB} > D_{cw2ac} \text{ or } D_{pw2cw} > D_{pw2ac} \quad (9)$$

4.3 Fly-by 알고리즘

Fly-by 알고리즘은 Fly-over 알고리즘을 적용하여 항로점 간의 기준 반경(R_{ref})에 대한 내접하는 점까지 직선 경로 추종을 유도한 뒤, 내접하는 지점에서 다음 항로점으로 갱신한다. 따라서 항로점이 갱신된 다음 항공기의 선회 반경에 의해 항로점으로 도달하기 위한 선회 비행을 수행한 후 다시 목표 항로점으로 직선 경로를 추종하게 된다. 먼저, 항로점 간의 기준 반경은 Eq. 10과 같이 나타낸다.

$$R_{ref} = \frac{VT_{ref}^2}{g \cdot \tan(\phi_{ref})} \quad (10)$$

여기서, VT_{ref} 는 항로점의 TAS 명령이며, g 는 중력가속도 이고, ϕ_{ref} 는 선회반경의 기준 롤 명령이다. 성층권까지 임무를 수행하기 때문에 고도 별 밀도 변화를 고려해야 하며, 항로점의 TAS(True Airspeed)를 이용하여 비행 고도에 따른 기준 선회 반경을 계산한다.

이에 따라 Fly-by 알고리즘의 다음 항로점으로 갱신하는 위치인, 기준 반경이 내접하는 위치(Dx_{FB})는 Fig. 11과 같으며, 이 위치에 도달하면 다음 항로점으로 갱신하게 된다. 만약 세 항로점이 이루는 각이 예각일 경우에는 Dx_{FB} 가 두 항로점 사이의 거리보다 커질 수 있으므로, 이런 경우에 대비하여 Dx_{FB} 에 최대값을 설정하여야 한다.

$$Dx_{FB} < Dx_{ac2cw} \quad (11)$$

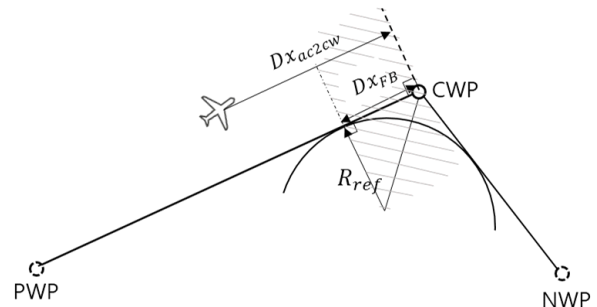


Fig. 11 Fly-by Arrival decision Algorithm

4.4 Hold 알고리즘

Hold 알고리즘은 항로점에서 원하는 반경을 따라서 선회하는 비행을 유도한다. 먼저, 현재 항로점까지 Fly-over비행을 하고, 항로점 도달 후부터 선회 비행을 수행한다. 선회 비행 알고리즘은 원형 경로를 안정적으로 추종할 수 있는 벡터 필드 기반의 유도법칙을 적용하였다[6, 7]. Figure 12는 원형 경로와 항공기 사이의 기하를 나타낸다.

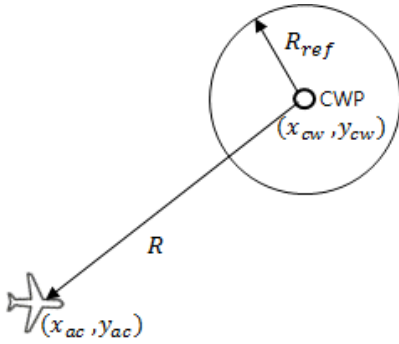


Fig. 12 Hold (Stationary Loiter Circles) Algorithm

원형 경로를 유도하는 벡터필드는 [6, 7]에서 사용한 Eq. 12를 적용하였다.

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \alpha \left(\frac{-\nu_0}{R} \right) \cdot \begin{bmatrix} x_r \cdot \frac{R^2 - R_{ref}^2}{R^2 + R_{ref}^2} + y_r \cdot \frac{2RR_{ref}}{R^2 + R_{ref}^2} \\ y_r \cdot \frac{R^2 - R_{ref}^2}{R^2 + R_{ref}^2} - x_r \cdot \frac{2RR_{ref}}{R^2 + R_{ref}^2} \end{bmatrix} \quad (12)$$

여기서, $[\dot{x}_d, \dot{y}_d]^T$ 는 원하는 관성 속도이고, α, ν_0 는 각각 양의 상수이다. $[x_r, y_r]^T = [x_{ac} - x_{cw}, y_{ac} - y_{cw}]^T$ 이고, R 은 항공기와 현재 항로점 사이의 거리이며, R_{ref} 은 원하는 원형 경로의 반지름으로 Eq. 10와 같다. Figure 13은 Eq. 12에 대해 원점을 기준으로 하는 반지름 100m인 원을 따라 선회하는 궤적을 벡터 필드로 나타내었으며, 현재 항로점을 지나 원형 경로를 추종하는 경로각 명령은 Eq. 13과 같이 나타난다 [7].

$$\begin{aligned} \psi_{cmd} &= \arctan\left(\frac{\dot{x}_d}{\dot{y}_d}\right) \\ &= \arctan\left(\frac{y_r \cdot (R^2 - R_{ref}^2) - x_r \cdot 2RR_{ref}}{x_r \cdot (R^2 - R_{ref}^2) + y_r \cdot 2RR_{ref}}\right) \end{aligned} \quad (13)$$

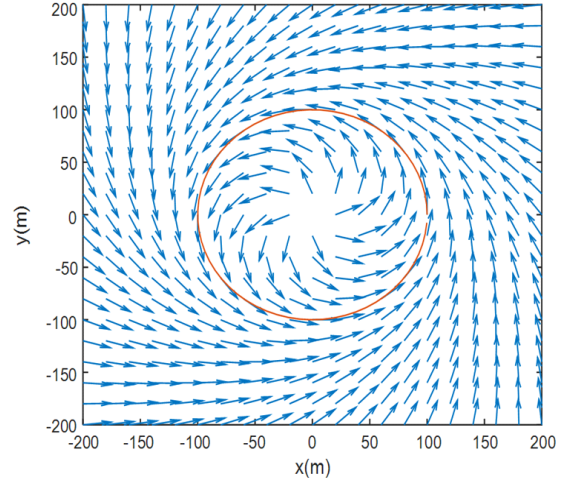


Fig. 13 Vector Field for Stationary Loiter Circles

Hold 알고리즘의 도달 판단은 현재 항로점까지 직진 비행 할 때 Fly-over와 같은 조건으로 판단한다. Hold 알고리즘은 판단 시기부터 지정한 시간동안 원형 경로를 따라 선회하는 기동을 유지하며, 지정한 시간 및 다음 항로점 갱신에 대한 판단은 OFP에서 수행한다.

4.5 유도법칙 비선형 시뮬레이션

설계한 유도법칙의 성능을 확인하기 위해 비선형 시뮬레이션을 수행하였다. 고고도 장기체공 무인항공기의 풍동 데이터를 이용한 비선형 모델을 구성하고, MATLAB/Simulink를 기반으로 자동비행 시뮬레이션을 수행하였다. 시뮬레이션 수행 조건은 다음과 같다.

- 고도 명령: 500 m (모든 항로점에서 동일)
- 속도 명령: 10 m/s (모든 항로점에서 동일)
- 바람 조건: 0m/s

시뮬레이션에 적용한 항로점의 속성은 다음과 같다.

- WP.No = [1 2 3 4 5 6 7 8 9]
- WP.Type = [0 1 1 1 1 1 1 1 2]

여기서, 항로점 속성 0은 Fly-over, 1은 Fly-by, 2는 Hold이다.

Figure 14, 15는 자동비행 시뮬레이션 결과를 나타낸다. 먼저, Fig. 14에서 Fly-over 속성으로 1번 항로점으로부터 설정한 반경 내에 도달하여서 2번 항로점으로 변경되었다. 2번 항로점부터 8번 항로점까지는 Fly-by 속성으로 항로점에 도달하기 전, 일정한 거리

에 도달하면 다음 항로점으로 갱신하는 것을 확인 할 수 있다. 마지막으로 9번 항로점에서는 Hold 속성으로 항로점을 지나간 뒤부터 원형 경로를 따라 선회한다. Figure 15는 유도법칙에서 생성한 경로각 명령을 따라, 경로각 제어기, SCAS 제어기의 제어 성능을 확인 할 수 있으며, 과도한 제어 명령 없이 안정적으로 자동비행에 대한 시뮬레이션 결과를 확인 할 수 있다.

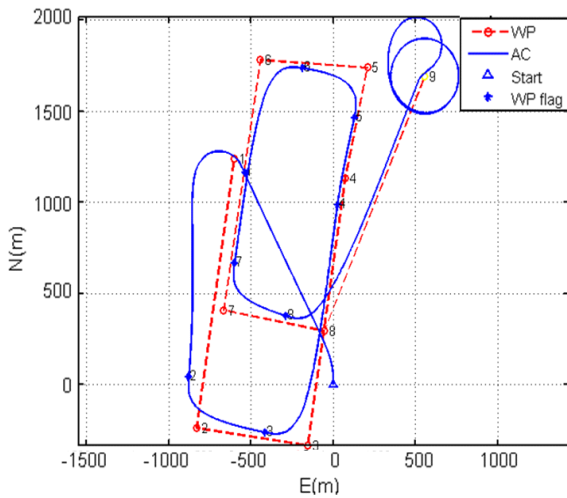


Fig. 14 Guidance Simulation Result(Trajectory)

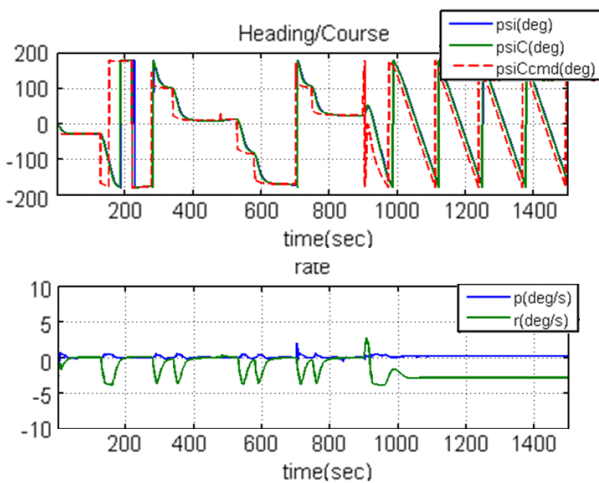


Fig. 15 Guidance Simulation Result (Course Control)

5. 비행 시험 결과

5.1 비행시험 1 : 맞바람에 대한 경로각 제어

고고도 장기체공 무인항공기의 임무는 고도를 상승했다가 하강하는 비행을 수행한다. 고도 10km 근처에서 제트기류의 영향으로 Fig. 16과 같이 비행대기속도보다 바람속도가 더 큰 영역에서도 비행하였으며, 이 경우에 무인항공기는 후진하였다. 이때 3.2절에서 제안된 경로각/기수방향 제어기 전환 알고리즘에 의해서 Fig. 16, 17과 같이 비행 중에 진동 없이 안정적인 자세를 유지하는 것을 실제 비행을 통해 확인하였다. 다만, 2.5×10^4 초에서 일시적으로 경로각 제어로 전환되는데 이는 경로각과 기수방향 차이가 20도 내로 들어왔기 때문이다. 이런 현상을 완전히 방지하여 더 안정적으로 비행하기 위해서는 Eq. (3)의 maxTime을 보다 크게 설정하면 된다. 상황에 따라 그 수치는 달라질 수 있으므로 실시간으로 적절한 값을 찾을 수 있는 방법에 대한 연구가 더 필요하다고 보인다.

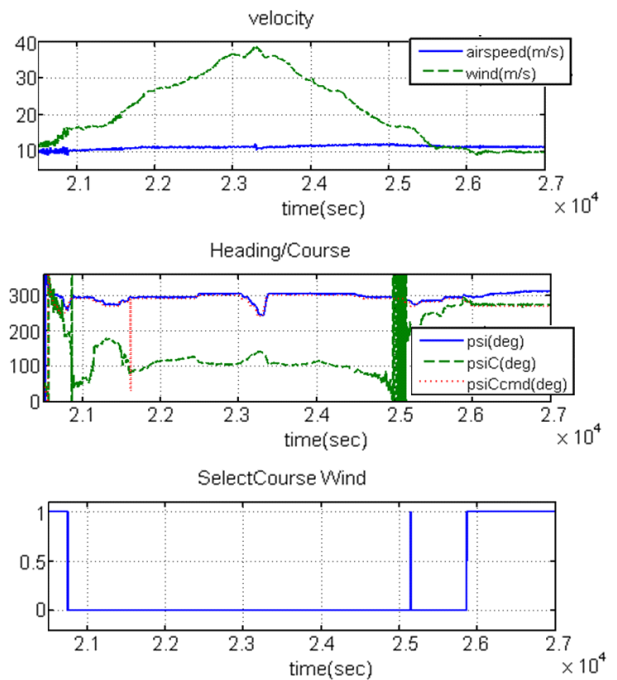


Fig. 16 Flight Test Result under Wind with Controller Switch Algorithm (State)

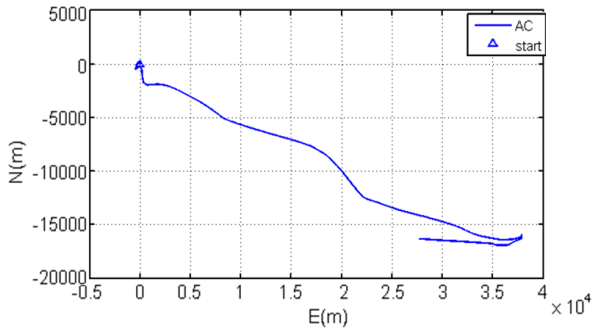


Fig. 17 Flight Test Result under Wind with Controller Switch Algorithm (Trajectory)

5.2 비행시험 2 : 자동 항로점 비행 시험

설계한 자동비행 알고리즘을 적용하여 실제 비행시험을 통해 검증하였다. 비선형 시뮬레이션과 동일한 항로점과 속성에 대해서 비행시험을 수행 하였으며, 1번 항로점부터 3번 항로점까지 Fly-over, Fly-by 비행을 수행한 뒤 3번 항로점과 4번 항로점 사이에서는 AVHC(Altitude/Velocity/Heading/Course) 모드로 전환하였다. 또한, Hold 속성에 대해서는 임의의 반자동 항로점에 대해 Fly-over를 수행한 뒤에 원형 경로를 따라 선회 비행함을 확인하였다. Figure 18, 19는 Fly-over, Fly- By 과 Hold 비행을 수행하는 동안의 경로각 제어 결과를 보여주며, 유도 명령에서 생성된 경로각 명령에 대해 안정적으로 비행함을 확인하였다.

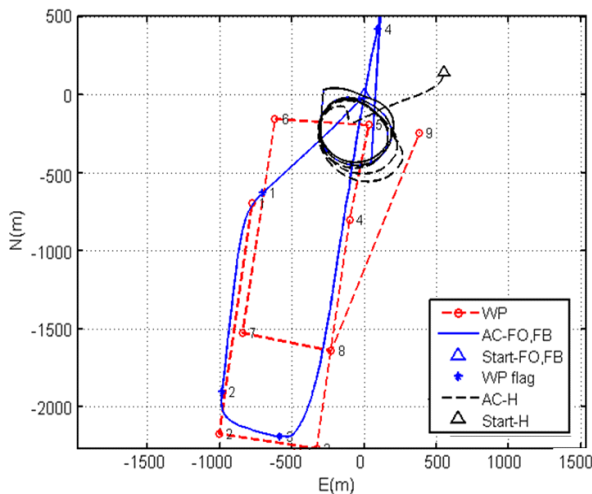


Fig. 18 Autonomous Flight Test Result (Trajectory)

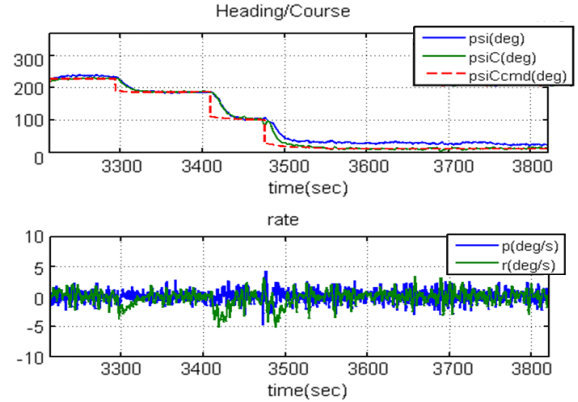


Fig. 19 Autonomous Flight Test Result during Fly-over, Fly-by (Course Control)

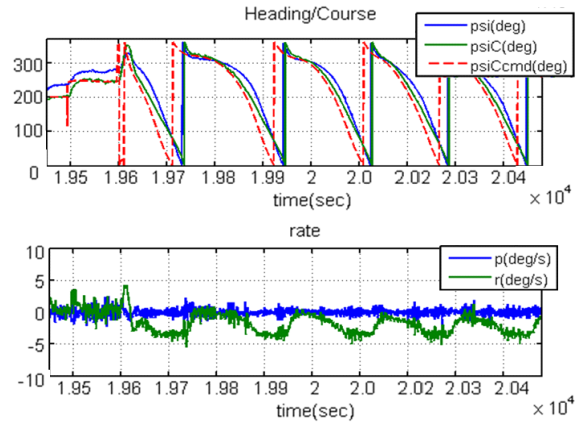


Fig. 20 Autonomous Flight Test Result during Hold (Course Control)

6. 결 론

본 논문에서는 고고도 장기체공 무인항공기의 임무 운용을 위한 방향축 자동비행 알고리즘을 설계하였다. 고고도 장기체공 무인항공기는 일반 무인기와는 다르게 비행대기속도가 매우 느려 바람속도보다도 작은 상황에서 자동비행해야 하는 경우가 발생한다. 이런 경우는 제트기류가 있는 고도에서 주로 일어나며, 이 상황에서도 안정적으로 비행할 수 있는 유도 및 방향축 제어 알고리즘을 설계하고 실제 비행을 통해 운용성을 검증하였다.

경로각 제어 중에 강한 맞바람을 맞아 후진 비행하게 되는 상황에서는 경로각 제어에서 기수방향 제어로

전환하는 알고리즘을 적용하여 안정적인 비행이 가능하도록 하였다. 또한, 유도법칙의 Fly-over, Fly-by, Hold의 속성에 대해 경로각 명령을 생성하고, 항로점 도달 판단에 대한 조건을 수립하였다. 이와 같은 제어 알고리즘과 유도법칙에 대해서 비선형 시뮬레이션과 실제 비행을 통해서 강한 바람에서도 안정적인 자동비행이 가능함을 검증하였다.

References

- [1] S. Hwang, S. Kim, Y. Lee, "Developing High Altitude Long Endurance (HALE) Solar-powered Unmanned Aerial Vehicle (UAV)," *Journal of Aerospace System Engineering*, Vol.10, No.1, pp.59-65, March, 2016.
- [2] H. Ryu, H. Byun and S. Park, "Efficient Path Planning for Long Term Solar UAV Flight," *Journal of Aerospace System Engineering*, Vol.8, No.4, pp.32-38, December, 2014.
- [3] K. Jung, J. Sung, B. Kim, J. Je and S. Lee, "Design and flight test of path following system for an unmanned airship," *Journal of Institute of Control, Robotics and Systems*, Vol. 16 no. 5, pp. 498-509, 2010.
- [4] D. Lee, S. Kim and J. Suk, "Design of a track guidance algorithm for formation flight of UAVs." *AIAA Guidance, Navigation, and Control Conference* p. 1315, 2015.
- [5] S. Lim, "Standoff target tracking using a vector field for multiple unmanned aircrafts," *Journal of Intelligent & Robotic Systems*, 69(1-4), pp. 347-360, 2013
- [6] S. Park, "Autonomous aerobatics on commanded path," *Aerospace Science and Technology*, vol. 22, no. 1 pp.64-74, 2012.
- [7] J. Osborn and R. Rysdyk, "Waypoint guidance for small UAVs in wind," *AIAA Infotech@ Aerospace*, vol. 193, no.1-4, pp.1-12, 2005.
- [8] G. Byeon and S. Park, "Backward Path Following Under a Strong Headwind for UAV," *J. of The Korean Society for Aeronautical and Space Sciences*, vol. 42, no. 5, pp. 376-382, 2014.