



J. Korean Soc. Aeronaut. Space Sci. 50(9), 639-646(2022)

DOI: <https://doi.org/10.5139/JKSAS.2022.50.9.639>

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

미지 환경에서의 깊이지도를 이용한 쿼드로터 착륙방식 성능 비교

최종혁¹, 박종호², 임재성³

Performance Comparison of Depth Map Based Landing Methods for a Quadrotor in Unknown Environment

Jong-Hyuck Choi¹, Jongho Park² and Jaesung Lim³

Department of Military Digital Convergence, Ajou University, Suwon, Republic of Korea

ABSTRACT

Landing site searching algorithms are developed for a quadrotor using a depth map in unknown environment. Guidance and control system of Unmanned Aerial Vehicle (UAV) consists of a trajectory planner, a position and an attitude controller. Landing site is selected based on the information of the depth map which is acquired by a stereo vision sensor attached on the gimbal system pointing downwards. Flatness information is obtained by the maximum depth difference of a predefined depth map region, and the distance from the UAV is also considered. This study proposes three landing methods and compares their performance using various indices such as UAV travel distance, map accuracy, obstacle response time etc.

초 록

본 논문은 사전에 알려지지 않은 환경에서 깊이지도를 이용하여 정보를 획득하고, 이를 기반으로 쿼드로터의 착륙 선택 알고리즘 성능을 분석한다. 무인항공기의 유도 및 제어시스템은 궤적 계획 유도시스템과 위치 및 자세 제어기로 구성된다. 아래를 향하는 짐벌 시스템에 부착된 스테레오 비전 센서가 획득한 깊이 정보를 이용하여 착륙 지점을 선정한다. 평탄도 정보는 사전 정의된 깊이지도 영역의 최대 깊이 차이 및 UAV와의 거리를 고려하여 산출한다. 본 논문에서는 3가지 착륙 방법을 제안하며 다양한 성능지수들을 활용하여 성능을 비교한다. 성능지수로는 UAV 이동 거리, 지도 정확도, 장애물 대응 시간 등을 고려한다.

Key Words : Landing Site Search(착륙 지점 선택), Depth Map(깊이지도), Quadrotor(쿼드로터), Stereo Vision(스테레오 비전), Unmanned Aerial Vehicle(무인항공기)

1. 서 론

무인항공기(UAV)는 감시, 정찰, 탐사, 전달, 재난 지원 등 다양한 분야에서 활용되고 있다. UAV의 전체 시장 규모는 2021년 274억 달러로 추정되며, 2021년부터 2026년까지 연평균 16.4%의 성장률로 2026년에는 584억 달러를 이를 것으로 전망하고 있다[1]. UAV는 특히

국경 보안과 같은 다양한 임무를 위해 군에 의해 널리 사용되고 있고, 정부와 법 집행 기관들은 실험적으로 UAV를 사용하기도 한다. 민간 및 상업적 용도로는 Zipline과 같은 회사가 의료 용품을 운송하는 데 사용하고 있으며 항공 사진 촬영 용도로도 사용되고 있다. 또한 아마존과 같은 회사는 UAV를 비대면 배송에 이용하고 있다[2].

† Received : June 4, 2022 Revised : July 27, 2022 Accepted : August 1, 2022

¹ Undergraduate Student, ² Assistant Professor, ³ Professor

² Corresponding author, E-mail : parkjo05@ajou.ac.kr, ORCID 0000-0001-5406-0306

© 2022 The Korean Society for Aeronautical and Space Sciences

이러한 임무들을 수행하기 위해 GPS를 사용하며 임무 지역에 대한 정보를 알면 무인항공기가 안전하게 착륙할 수 있다. 그러나 임무 수행 중 UAV는 기계 고장, 연료 문제, 지상과제 시스템과의 통신 문제 등으로 급하게 착륙을 결정해야 하는 경우가 발생할 수 있다. 하지만, 착륙 지역에 대한 사전 정보가 없다면 착륙 가능한 평평한 지역을 확실히 알 수 없다. 그러므로 무인항공기는 상황에 따라 자동적으로 평평한 지역을 찾아 효율적으로 착륙할 수 있는 알고리즘을 보유해야 한다[3].

본 연구에서는 빠르고 효율적인 착륙지 검색 알고리즘을 제안한다. 평탄도 정보는 사전 정의된 깊이지도의 최대 깊이 차이로 얻으며, UAV와의 거리를 고려한다 [4-8]. 착륙에 필요한 정보들을 종합하여 후보 중 가장 좋은 착륙지를 결정한다. 3가지 착륙 방법인 Hovering Landing, Rotate and Landing, Spiral Landing을 제안하며, 다양한 성능지수들을 활용하여 성능을 비교한다. 각 착륙방식은 이동 거리, 지도 정확도, 지도 커버리지, 장애물 대응 시간 등의 성능지수를 이용하여 수치 시뮬레이션을 통해 성능을 비교 분석한다.

II. 본 론

2.1 착륙 선택 알고리즘

2.1.1 환경 모델 설계

UAV가 착륙 지점을 선택하기 위해서는 지형정보를 얻어야 한다. 3차원 지형정보는 단일 카메라의 SLAM 알고리즘, RGB-D 카메라 등을 통해 얻을 수 있다. 본 논문에서는 스테레오 카메라를 이용하여 깊이지도 생성한다. 스테레오 카메라는 이미지 획득과 동시에 거리정보를 얻을 수 있고, 비용 또한 RGB-D 카메라보다 저렴하다는 장점이 있다.

시뮬레이션 환경에서 본 스테레오 카메라는 UAV 동체 하부의 짐벌에 장착되어 UAV의 자세와 관계없이 지구 중심 방향(z축 방향)을 향한다.

카메라의 시계는 Fig. 1과 같이 네 개의 꼭짓점 v_1, v_2, v_3, v_4 를 정의하고 식 (1)-(4)를 통해 각각 구할 수 있다.

$$v_1 = [d_a \tan(V_{fv})/2 - d_a \tan(H_{fv}/2) \ d_a]^T \quad (1)$$

$$v_2 = [-d_a \tan(V_{fv})/2 - d_a \tan(H_{fv}/2) \ d_a]^T \quad (2)$$

$$v_3 = [-d_a \tan(V_{fv})/2 \ d_a \tan(H_{fv}/2) \ d_a]^T \quad (3)$$

$$v_4 = [d_a \tan(V_{fv})/2 \ d_a \tan(H_{fv}/2) \ d_a]^T \quad (4)$$

여기서 V_{fv} 과 H_{fv} 는 각각 카메라의 수직, 수평 방향 각도이다. d_a 는 UAV 동체와 지표면 사이의 거리를 나타내고, 좌표계인 NED_I 와 NED_B 의 중심점은 각각 기준점, UAV의 무게중심이며, 이들의 X, Y, Z 축은 북, 동, 지구 중심방향을 향한다.

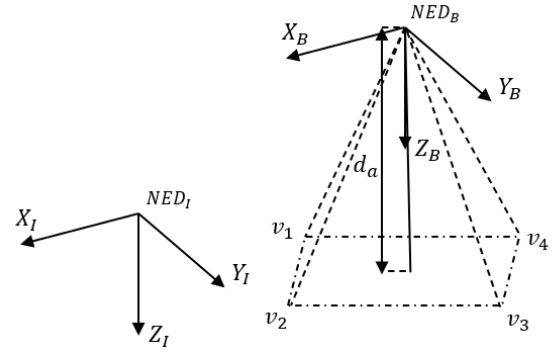


Fig. 1. Image plane model

UAV가 지형을 스캔하면서 획득하는 깊이정보 Z_{scan} 에 센서의 불확실성을 고려하기 위하여 실제 높이 값인 Z_t 를 평균으로 하는 정규분포의 임의의 값을 입력으로 설정한다. 또한 UAV의 높이(Z_{pos})가 높을수록 센서의 불확실성에 비례하도록 정규분포의 표준편차를 설정한다.

$$Z_{copy} = normrnd(Z_t, Z_{pos}) \quad (5)$$

여기서 $normrnd(a,b)$ 는 평균 a , 표준편차 b 를 가지는 정규분포에서 난수를 생성한다. 랜덤한 깊이정보의 신뢰성을 높이기 위해 각 time step마다 스캔한 높이의 정보를 축적하여 평균을 산출한다.

2.1.2 착륙 지점 선택 알고리즘

본 논문의 알고리즘은 불확실성을 가미한 데이터를 바탕으로 빠르게 유효한 착륙 지점을 찾아야 한다. 이를 위해 거리 데이터와 평탄도 데이터를 모두 고려한다.

거리 데이터는 UAV와 검색하는 지점과의 직선거리로 구성되며 식 (6)을 통해 구할 수 있다.

$$D_{data} = \sqrt{(X_I - X_B)^2 + (Y_I - Y_B)^2 + (Z_I - Z_B)^2} \quad (6)$$

평탄도 데이터(F_{data})는 Fig. 2과 같이 가로 및 세로 길이 t인 구역을 n개 고려하고 그 중 가운데 구역을 검증하려는 착륙 지점으로 지정한다. Fig. 2에 회색으로 표시한 영역(Z_{co})의 높이를 착륙 검증 구역과 함께 식 (7)을 이용하여 높이의 최대 최소 차이를 구한다. 이와 같은 방식으로 구한 평탄도 데이터는 기존의 Canny Edge

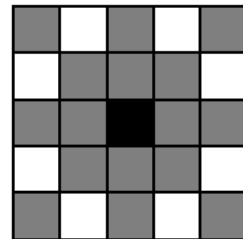


Fig. 2. Generating F_{data}

Table 1. Score of three candidate landing sites

	Candidate 1	Candidate 2	Candidate 3
D_{data}	10	5	8
F_{data}	6.1851	14.5487	9.8831

Detection Algorithm으로 가파른 경사면이나 절벽을 탐지하는 방법에 비해 실제로 높이를 획득한 후 구역의 높이 편차를 이용해 평탄도를 계산하므로 상대적으로 신뢰도가 높으며, 계산량 또한 적다[3].

$$F_{data} = \max(Z_{co}) - \min(Z_{co}) \quad (7)$$

이와 같이 획득한 거리 데이터와 평탄도 데이터는 식 (8)과 같이 가중치 R을 이용하여 성능지수를 산출한다.

$$J = R * D_{data} + (1 - R) * F_{data} \quad (8)$$

상황에 따라서 거리보다는 평탄도 또는 평탄도보다는 거리를 고려하도록 R을 설정한다. D_{data} 와 F_{data} 모두 수치가 낮을수록 거리가 가깝고, 평탄함을 의미하므로 성능지수(J)도 낮으면 착륙하기 좋은 장소로 판단한다. 본 연구에서는 $R < 0.5$ 로 설정한다. 일반적으로 착륙하는 데 거리보다는 평탄도의 중요도가 더 크다. 그러나 연료가 없는 상황에서는 평탄도보다는 가까운 거리의 착륙 지점을 찾는 것이 유리하다.

Table 1을 보면 3개의 착륙 후보지가 있다. 평탄도를 상대적으로 더 중요하게 생각하여 $R = 0.33$ 으로 정하면 후보 2와 3은 후보 1보다 가까움에도 불구하고 후보 1이 더 평탄하기에 최종착륙지점으로 선정된다. 만약 $R = 0.66$ 으로 정하면 후보 2가 최종착륙지점으로 선정된다.

2.2 착륙방식 비교

2.2.1 UAV 시스템 모델링과 착륙방식

본 연구에서 설계한 UAV 모델은 Fig. 3과 같다.

Trajectory Planner는 유도시스템으로 각 time step마다 착륙 지점 선택 알고리즘으로부터 산출된 위치명령

(x_d, y_d, z_d) 을 출력하여 Position Controller에게 전달한다. UAV는 x축, y축, z축 각각 V의 속력으로 기동하도록 설정한다.

Position Controller는 Trajectory Planner로부터 받은 위치정보로 UAV의 각도명령(ϕ_d, θ_d)을 다음과 같이 계산한다.

$$a_{x,cmd} = -2\zeta_h \omega_h \dot{x}^n + \omega_h^2 (x_d^n - x^n) \quad (9)$$

$$a_{y,cmd} = -2\zeta_h \omega_h \dot{y}^n + \omega_h^2 (y_d^n - y^n) \quad (10)$$

$$\begin{bmatrix} a_{u,cmd} \\ a_{v,cmd} \\ a_{w,cmd} \end{bmatrix} = C_{b/n} \begin{bmatrix} a_{x,cmd} \\ a_{y,cmd} \\ 0 \end{bmatrix} \quad (11)$$

여기서 ζ_h 와 ω_h 는 설계변수이고, $C_{b/n}$ 은 방향 코사인 행렬로 NED 좌표계에서 동체 좌표계로 변환해주는 역할을 한다. Roll과 pitch의 각도명령(ϕ_d, θ_d)은 다음과 같다.

$$\phi_d = \text{atan2}(a_{v,cmd}, g) \quad (12)$$

$$\theta_d = \text{atan2}(-a_{u,cmd}, g) \quad (13)$$

이를 이용하여 Attitude Controller에서는 쿼드콥터의 추력 및 모멘트를 계산한다. 이때 추력 T는 PD 제어기 형태로 설계하고, roll, pitch, yaw를 담당하는 M_1, M_2, M_3 는 PID 제어기 형태로 설계한다.

$$T_\Sigma = m \frac{2\zeta_v \omega_v \dot{z}^n + \omega_v^2 (z^n - z_d^n) + g}{\cos \phi \cos \theta} \quad (14)$$

$$M_1 = [K_{P,\phi}(\phi_d - \phi) + K_{I,\phi} \int (\phi_d - \phi) + K_{D,\phi}(p_d - p)] \quad (15)$$

$$M_2 = [K_{P,\theta}(\theta_d - \theta) + K_{I,\theta} \int (\theta_d - \theta) + K_{D,\theta}(q_d - q)] \quad (16)$$

$$M_3 = [K_{P,\psi}(\psi_d - \psi) + K_{I,\psi} \int (\psi_d - \psi) + K_{D,\psi}(r_d - r)] \quad (17)$$

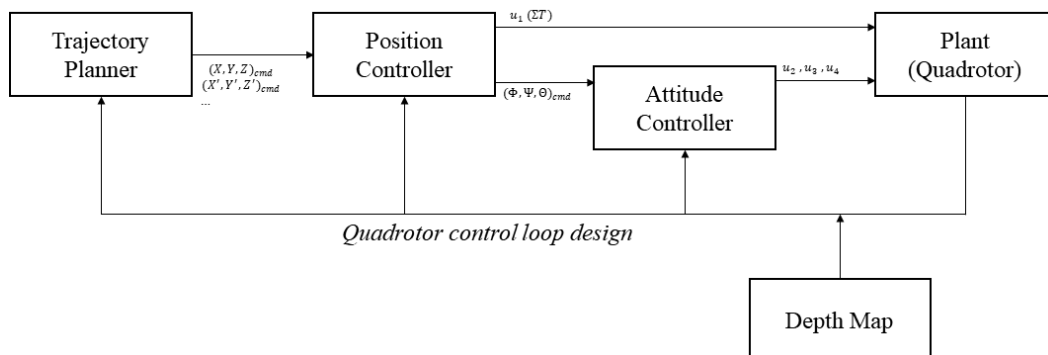


Fig. 3. Quadrotor control loop design

여기서 ζ_v , ω_v , $K_{P,\phi}$, $K_{P,\theta}$, $K_{P,\psi}$, $K_{I,\phi}$, $K_{I,\theta}$, $K_{I,\psi}$, $K_{D,\phi}$, $K_{D,\theta}$, $K_{D,\psi}$ 는 설계변수이다. 식 (9)-(17)을 이용하여 UAV의 제어시스템을 설계한다[4-8]. Fig. 3의 Plant는 UAV의 6자유도 동역학 모델을 나타낸다.

본 연구에서는 세 가지 착륙 방법을 제안한다. 첫 번째 방식은 Fig. 4에서 볼 수 있듯이 Hovering Landing이다. 회전익 무인항공기인 쿼드콥터가 호버링에 특화되어 있다는 점에서 기준으로 삼은 착륙방식이다. Hovering Landing은 UAV가 초기 위치에서 얻은 착륙 지점으로 수평 이동을 수행한 후 일정 고도를 감소하고 착륙 지점 선택 알고리즘을 수행하여 착륙 지점을 갱신한다. 이 과정을 착륙할 때까지 반복한다. Algorithm 1은 Hovering Landing 알고리즘의 의사 코드를 나타낸다.

Algorithm 1 Landing type 1 (Hovering Landing)

```

Initialization: flag=1; flag_f=0;
for entire simulation time
  if UAV is in position
    flag=1;
  end
  if flag==1 and flag_f==0
    run landing site search algorithm
    UAV next (x, y) position = next landing (x, y)
    position;
    flag_f=1;
    flag=0;
  end
  if flag_f==1 and flag==1
    if UAV z position < threshold height
      UAV z position +h; //Downward movement
      flag_f=0;
      flag=0;
    end
  end
end
end
end

```

두 번째 방식은 Fig. 5에서 볼 수 있듯이 Rotate and Landing 방식이다. UAV가 선회하면 깊이지도의 영역이 커지고 정확도 또한 향상된다. 이를 통해 상대적으로 더 많은 영역을 검색하므로 더 적절한 착륙지를 선정할 가능성이 커진다. Rotate and Landing 방식은 UAV가 초기 위치에서 얻은 착륙 지점을 기준으로 한 바퀴 선회하고 일정 고도를 감소한다. 다음 착륙 지점의 갱신을 위해 착륙지점 지점 선택 알고리즘을 수행하고, 이 과정을 착륙할 때까지 반복한다. Algorithm 2는 Rotate and Landing 알고리즘의 의사 코드를 나타낸다.

세 번째 방식은 Fig. 6에서 볼 수 있듯이 Spiral Landing 방식이다. Spiral Landing 방식은 Hovering Landing 방식과 Rotate and Landing 방식을 융합시킨

방식이다. Spiral Landing 방식은 UAV가 초기 위치에서 얻은 착륙 지점을 기준으로 한 바퀴 선회할 때마다 착륙 지점 선택 알고리즘을 수행하고 고도를 일정한 속도로 낮추면서 이 과정을 착륙할 때까지 반복한다. Algorithm 3은 Spiral Landing 알고리즘의 의사 코드를 나타낸다.

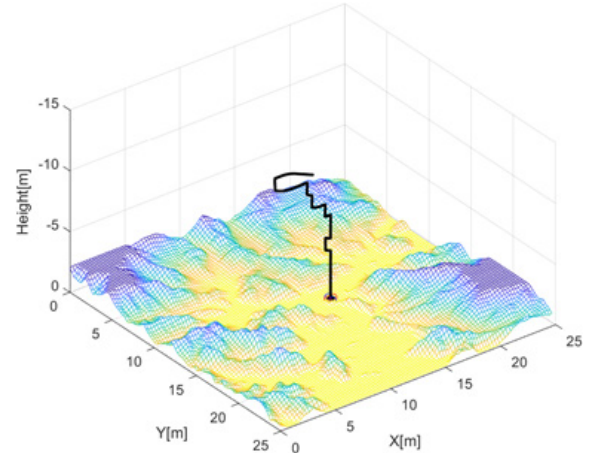


Fig. 4. Hovering Landing

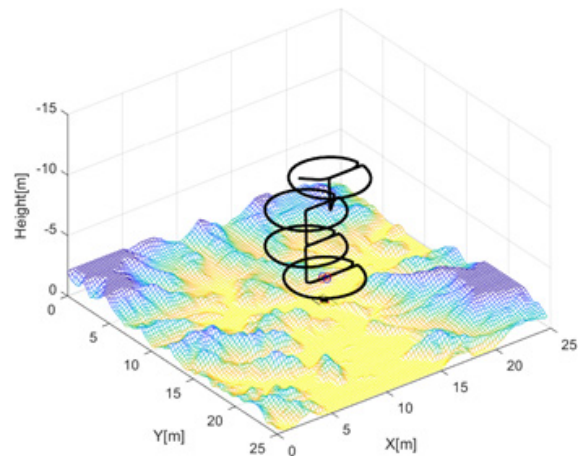


Fig. 5. Rotate and Landing

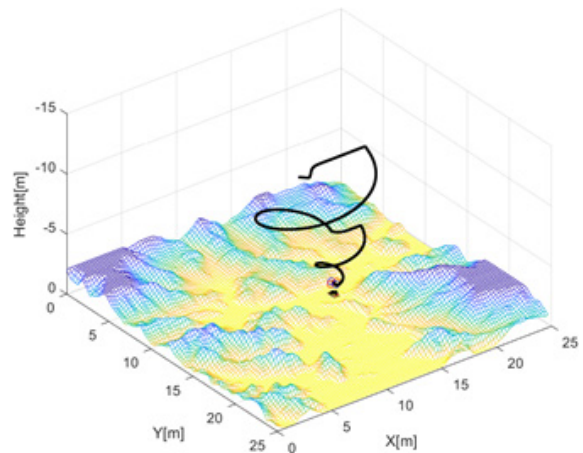


Fig. 6. Spiral Landing

Algorithm 2 Landing type 2 (Rotate and Landing)

```

Initialization: flag = 1; flag_circle = 0;
for entire simulation time
  if UAV is in position
    flag = 1;
  end
  if flag == 1 and flag_circle == 0
    run landing site search algorithm
    UAV next (x, y) position = next landing (x, y)
    position;
    flag_circle = 1;
    flag = 0;
  end
  if flag_circle == 1 and flag == 1
    x position = radius*cos(theta) + x target position
    y position = radius*sin(theta) + y target position
    if UAV goes around once
      target point = target position
      flag_circle = 2;
      flag = 0;
    end
  end
  if flag_circle == 2 and flag == 1
    if UAV z position < threshold height
      UAV z position + h; //Downward movement
      flag_circle = 0;
      flag = 0;
    end
  end
end
end

```

Algorithm 3 Landing type 3 (Spiral Landing)

```

Initialization: flag = 1; flag_circle = 0;
for entire simulation time
  if UAV is in position
    flag = 1;
  end
  if flag == 1 and flag_f == 0
    run landing site search algorithm
    UAV next (x, y) position = next landing (x, y)
    position;
    flag_circle = 1;
    flag = 0;
  end
  if flag_circle == 1 and flag == 1
    rhratio = UAV current height*radius_ratio;
    x position = rhratio*cos(theta) + x target position
    y position = rhratio*sin(theta) + y target position
    z position = UAV current height - landing ratio
    if UAV goes around once
      flag_circle = 0;
    end
  end
end
end

```

2.2.2 성능평가 항목

각 착륙방식의 성능을 수치화하기 위해 여러 가지 요소를 고려한다. 지도의 정확도는 한 구역 단위마다 실제 높이 값과 스캔한 높이 값의 차를 이용해 구한다.

$$Point_{map} = 100 + \frac{\sum |Z_t - Z_{scan}|}{\sum Z_t} * 100(\%) \quad (18)$$

여기서 Z_t 는 실제 높이 값을, Z_{scan} 은 UAV가 스캔한 영상에서 얻은 높이 값을 의미한다. 연료 소비 지수는 UAV가 최종 착륙 지점까지 이동한 거리에 비례하도록 다음과 같이 계산한다.

$$Point_{fuel} = d_b * D_t \quad (19)$$

여기서 d_b 는 UAV가 time step 동안 이동한 거리, D_t 는 UAV가 이동한 시간이다. 응급 상황에 대한 대처는 장애물을 탐지할 때 UAV가 얼마나 빨리 인식하고 최종 착륙 지점을 변경하는지를 Obstacle Recognition Time (ORT)을 통해 고려한다.

$$ORT = t_n - t_e \quad (20)$$

여기서 t_n 은 장애물을 인식한 시간, t_e 는 장애물이 생성된 시간이다. 마지막으로 스캔한 넓이가 착륙 지점 선택 및 성능에 영향을 주기 때문에 다음과 같이 고려한다.

$$Point_{scan} = S * R_{copy} \quad (21)$$

여기서 S는 한 단위 구역의 넓이, R_{copy} 는 UAV가 스캔한 전체 단위 구역의 개수이다. Rotate and Landing 방식과 Spiral Landing 방식의 경우, 회전 반지름에 따라서 스캔한 넓이와 지도의 정확도에 영향을 주므로 선회 반지름 또는 높이 대 선회 반지름 비율을 변화시켜가며 측정한다.

2.2.3 시뮬레이션 환경

UAV가 구동되는 환경은 Fig. 7과 같이 가로, 세로, 높이가 각각 25m, 25m, 15m이고 평지와 산맥이 골고루 분포되어 있다. 산맥 및 장애물이 있는 영역은 UAV가 착륙할 수 없다. UAV의 수평 초기 위치는 무작위로 설정되며, 높이는 11m로 고정한다. F_{data} 의 값을 가져오기 위해 가로 및 세로 길이 25cm로 설정하고($t = 25cm$) 25개 구역($n = 25$)을 고려한다.

UAV는 X, Y, Z축으로 각각 2m/s로 구동하도록 설정한다 ($V = 2m/s$). Algorithm 1, 2에서 UAV의 고도는 각각 1m(height = -1m), 3m씩(height = -3m) 감소한다. Algorithm 1, 2, 3에서 Hovering Landing과 Rotate and Landing의 임계높이는 -1m로(threshold height = -1m) 설정하고, Spiral Landing은 -2m로 설정한다(threshold height = -2m).

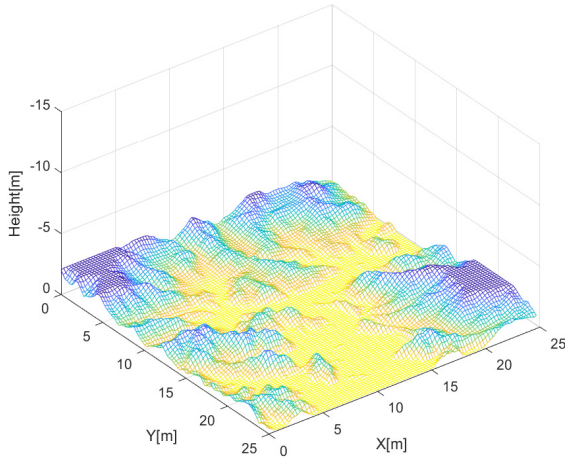


Fig. 7. Simulation environment

성능지수 중 식 (20)의 ORT를 측정하기 위하여 UAV가 5m 상공에 있을 때 선택한 착륙 지점에 높이가 평균 2m인 장애물을 생성한다. 각 착륙방식마다 시뮬레이션을 10번 수행하였다. Table 2는 수치 시뮬레이션에서 사용한 UAV 모델 및 제어기와 관련된 설계변수 값이다.

Table 2. UAV system's parameter

Parameter	Values
Mass	1.25 kg
Arm length	0.265 m
R	0.33
I_{xx}	0.0232 kg*m ²
I_{yy}	0.0232 kg*m ²
I_{zz}	0.0468 kg*m ²
H_{fv}	45°
V_{fv}	30°
$K_{P,\phi}$	10.0
$K_{I,\phi}$	0.8
$K_{D,\phi}$	0.1
$K_{P,\theta}$	10.0
$K_{I,\theta}$	0.8
$K_{D,\theta}$	0.1
$K_{P,\psi}$	8.0
$K_{I,\psi}$	1.5
$K_{D,\psi}$	0.1
ζ_v	0.8
ω_v	6.0
ζ_h	0.8
ω_h	6.0

2.2.4 시뮬레이션 결과

Figure 8은 산맥만 있는 지형에서 UAV를 동시에 출발시켰을 때 착륙방식에 따라 UAV의 움직임을 비교한 대표 결과이다. 그리고 대표 결과에 대한 위치와 오일러 각도의 시간 반응은 각각 Fig. 9, Fig. 10, Fig. 11과 같다. 여기서 검은 실선은 Hovering Landing(이하 Type 1)을 나타내고, 파란 실선은 Rotate and Landing(이하 Type 2)을 나타내고, 빨간 실선은 Spiral Landing(이하 Type 3)을 나타낸다.

Type 1이 Type 2와 Type 3에 비해 빠르게 착륙하는 것을 Fig. 9를 통해 확인할 수 있다. Type 1의 구동 알고리즘은 주어진 카메라의 시계 안에서 착륙 지점을 찾고 그 지점까지 직선 이동하기 때문에 전체 이동 거리가 짧고 전체 시간도 짧다. Fig. 9를 통해 Type 2의 x축 성분이 주기적으로 원운동을 하지 않는 구간이 있음을 확인할 수 있다. 이 구간은 UAV가 구역 스캔을 위해 한 바퀴 선회한 후 착륙지점 선택 알고리즘을 수행하여 착륙 지점을 갱신하고 해당 지점으로 이동하는 구간이다. 반면, Type 3의 위치 성분들을 통해 UAV의 회전 반경이 점차 작아지고 고도가 일정하게 낮아지는 것을 확인할 수 있고, Spiral 형태로 착륙함을 알 수 있다.

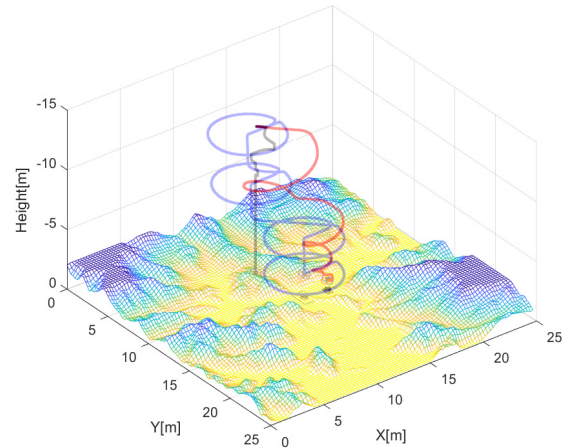


Fig. 8. Overall simulation

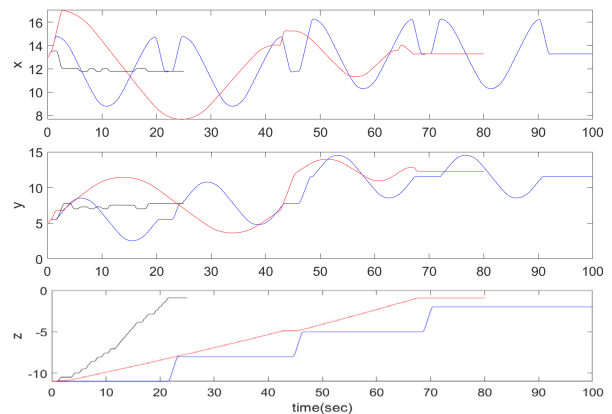


Fig. 9. Time histories of UAV's x, y, z position components in Fig. 8

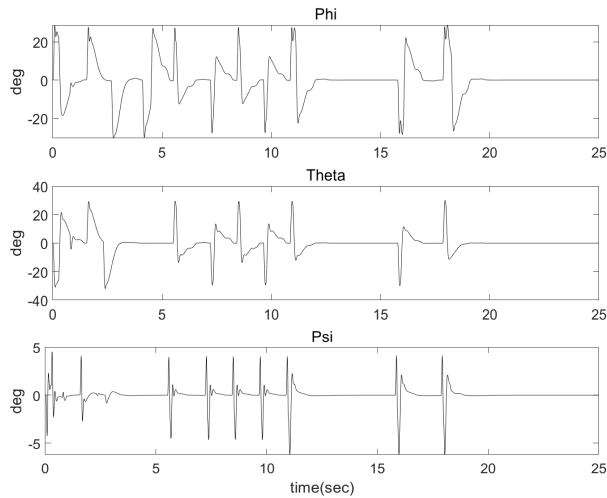


Fig. 10. Time histories of Type 1 UAV's roll, pitch, and yaw angles in Fig. 8

Figure 8과 Fig. 9를 통해 Type 1(검정 실선)은 UAV가 산맥 지형을 벗어나지 못하고 산맥 안에서 가장 평평한 지형에 착륙한 것을 확인할 수 있다. 반면 Type 2(파란 실선)와 Type 3(빨간 실선) 알고리즘의 경우, UAV가 산맥 지형을 벗어나서 가까운 평지에 착륙한 것을 알 수 있다. 결론적으로 Type 1은 UAV가 빠르게 착륙하지만 다른 지역을 스캔하지 못하고 더 좋은 착륙 장소를 고려하지 못한 채로 착륙해야 하는 단점이 있다. 그러나 Type 2와 Type 3은 더 넓은 곳을 스캔하는 특성상 더 좋은 착륙 장소를 발견할 가능성이 커진다. 착륙 알고리즘의 종류와 상관없이 롤, 피치, 요 각이 일정 각도 이상 발산하지 않고 최종착륙 위치까지 이동하는 것 Fig. 10과 Fig. 11을 통해 확인할 수 있다. Type 2와 Type 3의 롤, 피치 각이 진동하는 것은 매 time step마다 위치 변화가 많은 원운동을 하도록 위치명령을 갱신하고 이에 따라 UAV의 제어시스템에 의해 UAV의 각도가 움직이기 때문이다. Fig. 9와 마찬가지로 Fig. 11의 Type 2와 Type 3의 롤, 피치 각이 0으로 수렴하는 구

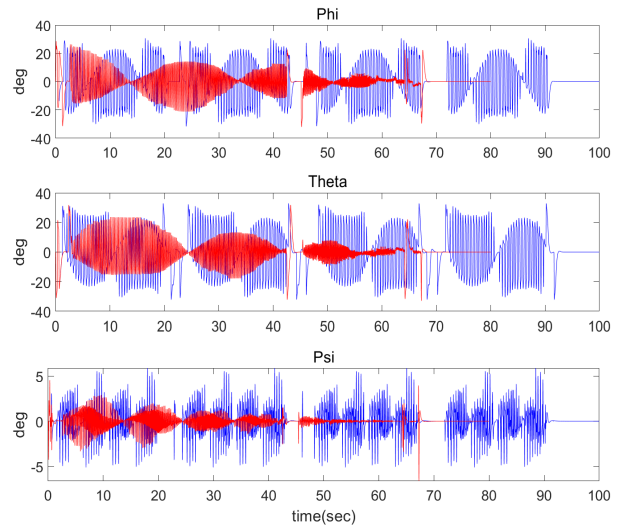


Fig. 11. Time histories of Type 2 and Type 3 UAV's roll, pitch, and yaw angles in Fig. 8

간이 있다. 이 구간은 마찬가지로 착륙 지점 선택 알고리즘을 수행하여 착륙 지점을 갱신하는 구간이다. 최종 착륙 지점에 도달하거나 착륙 알고리즘으로 최종 착륙 지점이 결정되면 롤, 피치각이 0으로 수렴하면서 UAV가 안정적인 자세를 구현하는 것을 확인할 수 있다.

Table 3, Table 4, Table 5는 Type 1, Type 2, Type 3에 대한 통계 결과를 나타낸다. Table 3, Table 4, Table 5를 통하여 지도의 정확도는 Type 1을 기준으로 Type 2는 15~60%, Type 3은 65~75% 향상된 것을 확인할 수 있다. 표준편차 결과에 의해 Type 1보다 Type 2와 Type 3의 지도 정확도 일관성이 크다고 판단할 수 있다. 스캔한 영역은 Type 1을 기준으로 Type 2는 1.2~2.2배, Type 3은 1.8~3.5배 늘어난 것을 확인할 수 있다. 표준편차 결과에 의해 Type 1보다 Type 2와 Type 3의 일관성이 크다고 판단할 수 있다. 그러나 UAV가 이동한 거리는 Type 1 대비 Type 2는 2~4.5배, Type 3은 3.8~4.5배 증가한다. 결론적으로 스캔한

Table 3. Hovering Landing simulation result

Type 1	Region (m^2)	Accuracy (%)	Distance (m)	ORT (s)
average	83	118	21.99	2.33
stdev	9.85	4.4	1.98	0.43

Table 4. Rotate and Landing simulation result

Type 2	Region (m^2)	Accuracy (%)	Distance (m)	ORT (s)	Radius (m)
average	168	106	70.16	23.46	3
stdev	1.5	1.4	0.46	0.15	
average	136	109	64.92	18.21	2
stdev	7.3	2.1	0.58	0.13	
average	108	115	43.23	11.87	1
stdev	5.0	1.7	0.52	0.12	

Table 5. Spiral Landing simulation result

Type 3	Region (m^2)	Accuracy (%)	Distance (m)	ORT (s)	Radius-ratio
average	283	102	73.53	5.65	0.7
stdev	4.2	1.2	0.86	0	
average	208	103	64.33	4.2	0.5
stdev	1.0	0.6	0.6	0	
average	141	104	60.58	10.65	0.3
stdev	7.9	1.8	0.2	0.01	

넓이가 커지면 정확도는 늘어나지만 UAV가 이동해야 할 거리가 증가하므로 명확한 trade-off가 존재한다. Type 2와 Type 3의 결과를 비교하면, 선회 반지름 또는 높이 대 선회 반지름 비가 증가하면 정확도는 향상되지만 이동거리가 증가한다. 그러나 같은 넓이 영역을 탐색하면 이동거리는 유사하지만 Type 2보다 Type 3의 정확도가 더 높음을 확인할 수 있다.

ORT의 경우 Type 1이 가장 빠르고 Type 2는 UAV가 한 바퀴 선회하는 시간과 같다. 반면 Type 3은 UAV가 한 바퀴 선회하기 전 장애물을 인식함을 확인할 수 있다.

III. 결 론

본 연구에서는 깊이지도 이용한 회전익 무인항공기의 착륙 선택 알고리즘들을 비교 분석하였다. Hovering Landing 방식은 최단 시간에 최소거리로 UAV가 착륙하지만, 착륙 지점의 신뢰도는 낮다. 비상 상황에서 사용할 수 있는 착륙방식이지만, 환경에 따라서 착륙 성능의 일관성이 떨어지는 경향을 보인다. Rotate and Landing 방식과 Spiral Landing 방식은 이동거리에 대한 제약이 없고 착륙 성능의 신뢰성이 요구될 때 사용할 수 있는 착륙방식이다. 다만 Rotate and Landing 방식보다는 Spiral Landing 방식이 정확도 및 장애물 인식 시간 관련하여 이점을 보이기 때문에 후자의 방식이 더 효율적이라고 판단할 수 있다.

후 기

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1G1A1003429).

References

- 1) Markets and Markets, "Unmanned Aerial Vehicle (UAV) Market," Markets and Markets, June 2021, accessed 11 January 2022.
- 2) Jung, S. H. and Kim, H. S., "Analysis of Amazon Prime Air UAV Delivery Service," *Journal of Knowledge Information Technology and Systems*, Vol. 12, No. 2, 2017, pp. 253~266.
- 3) Park, J. H., Kim, Y. D. and Kim, S. W., "Landing Site Searching and Selection Algorithm Development Using Vision System and Its Application to Quadrotor," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 2, March 2015, pp. 488~503.
- 4) Park, J. H. and Kim, Y. D., "Collision Avoidance for Quadrotor Using Stereo Vision Depth Maps," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 51, No. 4, October 2015, pp. 3226~3241.
- 5) Park, J. H. and Cho, N. H., "Collision Avoidance of Hexacopter UAV Based on LiDAR Data in Dynamic Environment," *Remote Sensing*, Vol. 12, No. 6, March 2020, p. 975.
- 6) Paul Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Second Edition, Artech, 2013.
- 7) Kim, B. S., Kim, Y. D., Bang, H. C. and Hong, S. G., *Flight Dynamics and Control*, Kyongmoon, 2004.
- 8) Leishman, J. G., *Principles of Helicopter Aerodynamics*, Second Edition, Cambridge University Press, 2016.