Journal of Korea Robotics Society (2023) 18(3):316-322
https://doi.org/10.7746/jkros.2023.18.3.316 ISSN: 1975-6291 / eISSN: 2287-3961

쿼드콥터의 곡률 기반 3차원 경로 계획 알고리즘

Curvature-based 3D Path Planning Algorithm for Quadcopter

박 재 용[1] · 김 보 성[2] · 이 승 욱[2] · 마울라나 비시르 아즈하리[2] · 심 현 철†
Jaeyong Park[1], Boseong Kim[2], Seungwook Lee[2],
Maulana Bisyir Azhari[2], Hyunchul Shim†

Abstract: The increasing popularity of autonomous unmanned aerial vehicles (UAVs) can be attributed to their wide range of applications. 3D path planning is one of the crucial components enabling autonomous flight. In this paper, we present a novel 3D path planning algorithm that generates and utilizes curvature-based trajectories. Our approach leverages circular properties, offering notable advantages. First, circular trajectories make collision detection easier. Second, the planning procedure is streamlined by eliminating the need for the spline process to generate dynamically feasible trajectories. To validate our proposed algorithm, we conducted simulations in Gazebo Simulator. Within the simulation, we placed various obstacles such as pillars, nets, trees, and walls. The results demonstrate the efficacy and potential of our proposed algorithm in facilitating efficient and reliable 3D path planning for UAVs.

Keywords: UAV, 3D Path Planning, Obstacle Avoidance, Model Predictive Control

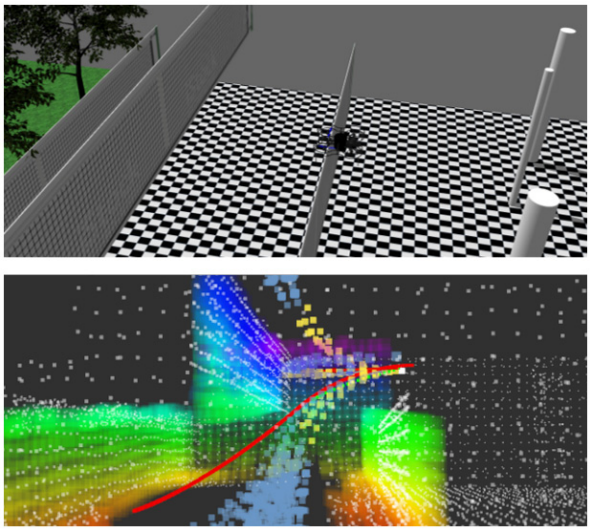
1. 서 론

UAV (Unmanned Aerial Vehicle, 무인 공중 이동체)는 배달, 사진 촬영, 미지 지역 탐사 등 민간 분야뿐만 아니라 정찰, 감시와 같은 군사적 용도로 활용 가능하다는 장점이 있어 점차 많은 관심을 받고 있다. 더불어 컴퓨터 기술의 발전으로 인해 점점 더 작고 경량화된 UAV가 개발되고 있으며, 이에 탑재된 컴퓨터를 통해 UAV가 사람의 개입 없이 자율적으로 비행하고 주어진 임무를 수행할 수 있는 능력을 활용하고자 하는 시도 또한 크게 주목받고 있다.

장애물이 많고 복잡한 환경에서 자율 비행 UAV를 운용하기 위해선 3차원 경로 계획 알고리즘이 요구된다. 이때 경로 계획 알고리즘은 낮은 계산량을 통한 실시간성과 부드러운 곡선 형태의 경로 생성이 요구된다. UAV는 자율주행 차량, 로봇 등에 비해 크기가 비교적 작은 특성으로 인해 탑재하는 컴퓨터의 성능이 제한적이며 그로 인해 계산량이 높은 알고리즘은 실시간성을 보장하기 어렵다. 또한 일반적으로 3차원 경로 계획 알고리즘에서 사용되는 A-star, RRT-star 같은 알고리즘은 결과 경로가 부드러운 곡선이 아닌 급격히 꺾인 형태를 가지는데, 이러한 경우 UAV가 추종하는 실제 비행 경로가 주어진 경로와 큰 차이를 보이거나, 또는 주어진 경로를 정확하게 비행하기 위해 크게 감속해야 하며 이는 비행 시 불안정성을 야기할 수 있다[1].

3차원 경로 계획 알고리즘의 실시간성 및 곡선 형태의 경로 생성을 달성하기 위한 많은 연구가 이루어졌다[2]. Motion-Primitive를 이용한 연구[3]에서는 많은 숫자의 경로를 미리 생성하고 그 경로를 기반으로 장애물과의 충돌 검사를 수행한 후, 확률 기반의 방법으로 최적의 경로를 선택하는 방법을 제안하였다. 또한 3D A-star및 B-Spline을 활용하여 쿼드콥터의 경로를 탐색하거나[4], Jump-Point-Search 알고리즘을 적용하여 3차원 공간에서 경로를 탐색한 후 3차 다항식을 활용하여 smoothing을 적용하였다[5]. 쿼드콥터 모델의 differential flatness를 활용한 연구[1] 에서는 시간에 대한 고차 다항식을 정의한 다음 L2 norm을 최소화하는 최적 문제를 제시하고 그 해를 계

Received : Jun. 7. 2023; Revised : Jul. 18. 2023; Accepted : Jul. 26. 2023
※ This project was financially supported by the Institute of Civil Military Technology Cooperation funded by the Defense Acquisition Program Administration and Ministry of Trade, Industry and Energy of Korean government under grant No.UM22206RD2
1. M. S. Student, Electrical Engineering, KAIST, Daejeon, Korea (yongee@kaist.ac.kr)
2. Ph. D Student, Electrical Engineering, KAIST, Daejeon, Korea (brain.kim, seungwook1024, mbazhari@kaist.ac.kr)
† Professor, Corresponding author: Electrical Engineering, KAIST, Daejeon, Korea (hcshim@kaist.ac.kr)



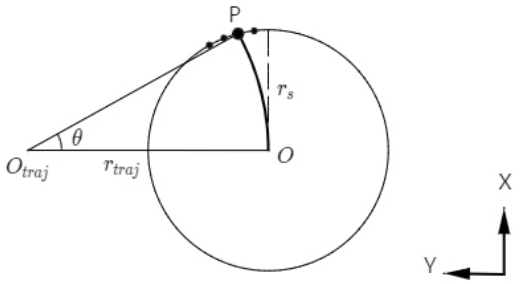
[Fig. 1] UAV flying through a complex environment in a Gazebo simulation, displaying the outcome of the path generation process

산함으로써 쿼드로터의 모델을 고려하며 주어진 목표 지점들을 지나는 경로를 생성하였다. 하지만 위에 언급된 알고리즘들은 탐색과 smoothing의 두 가지 과정이 요구되며 이때 두 과정 모두에서 결과 경로와 장애물 간 충돌이 없음을 보장하기 위해 충돌 검사를 중복으로 수행해야 하는데, 이는 계산량 측면에서 비효율적이다.

이러한 문제를 해결하기 위해 본 논문에서는 낮은 충돌 검사 계산량을 가지는 곡률 기반 3차원 경로 계획 알고리즘을 제안한다. 본 논문에서 제시하는 경로 계획 알고리즘은 다음과 같은 특징을 가진다.

- 결과 경로는 원호들로 구성되어 UAV의 원활한 경로 추종을 위한 추가적인 smoothing 과정을 필요로 하지 않는다.
- 모델 예측 제어기가 요구하는 reference point를 원호의 길이와 호의 각도 간 관계를 이용하여 생성된 경로에서 구한다.
- 경로와 장애물 간의 충돌 검사를 위해 경로 위의 모든 점과 장애물 포인트 클라우드 간의 거리를 검사하는 대신 경로를 그리는 원과 장애물 간 상관 관계를 바탕으로 충돌 검사의 계산량을 낮춘다

본 논문에서 제안하는 알고리즘을 검증하기 위해 [Fig. 1]과 같이 원기둥, 수평 벽 등 다양한 형태의 장애물이 설치된 Gazebo 시뮬레이션을 활용하였으며 제안하는 기법이 기존의 충돌검사 기법보다 낮은 계산량을 가지며 원호 형태의 경로를 생성함을 확인하였다.



[Fig. 2] Illustration of the generation of curvature-based trajectories. O denotes the starting point of the trajectory, r_s represents the search range, θ represents the angle between the starting point and the final point. Each trajectory is designed to form a segment of a circle that contacts the preceding direction, ensuring that the circle encompassing the trajectory is uniquely determined

2. 알고리즘

2.1 곡률 기반 경로 탐색

현재 진행 방향을 기준으로 하여 첫 번째 경로 후보들을 생성한다. 탐사 반경이 r_s 일 때 [Fig. 2]와 같이 중심이 탐색 시작 위치이고 반지름이 r_s인 원을 그린 다음 일정한 각도를 가지는 점들로 나눈다. 각 점들에 대해서 해당 점과 위에서 그린 원의 중심을 지나며, 원의 중심에서 진행 방향을 향하는 직선과 접하는 원은 유일하게 결정되며, 그 원의 호가 해당 점에 대응하는 경로 후보가 된다.

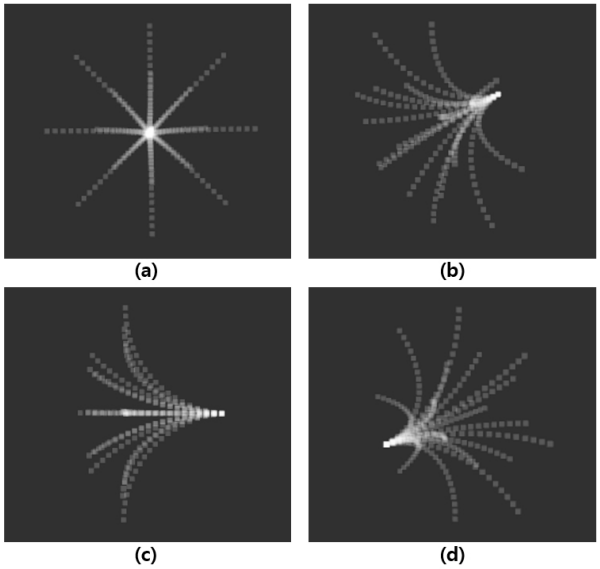
[Fig. 2]에서 O는 탐색 시작 지점, O_{traj}는 생성된 경로가 따르는 원의 중심, P는 생성된 경로와 탐색 반경을 반지름으로 하는 원이 만나는 점, r_s는 탐색 반경, r_{traj}는 경로가 따르는 원의 반지름이며, 그리고 θ는 경로가 따르는 원에서 유효한 경로와 대응하는 각도이고 식 (1)로 계산할 수 있다. 또한 경로를 구성하는 원의 반지름과 곡률은 각각 식 (2), 식 (3)을 통해 구할 수 있다. 식 (1)~(3)에서 p_x와 p_y는 점 P의 x,y좌표를 의미한다.

$$\theta_{valid} = atan2(p_x, p_y) \tag{1}$$

$$r_{traj} = \frac{p_x^2 + p_y^2}{2p_y} \tag{2}$$

$$\kappa_{traj} = \frac{2p_y}{p_x^2 + p_y^2} \tag{3}$$

앞서 언급한 과정을 3차원으로 적용하기 위해 [Fig. 3]와 같이 반평면을 일정한 각도로 회전하며 같은 과정을 진행한다. 평면의 회전 각도가 $\phi \in (o, \pi)$ 일 때 검사하는 평면의 수는 식 (4)로 주어진다.



[Fig. 3] Visualizations of the first set of trajectories. The plane rotation angle ϕ is 45°. (a) front view, (b) front-left view, (c) side view, (d) rear-right view. The generation of trajectories is repeated for each subsequent plane

$$number\ of\ planes = \left\lfloor \frac{2\pi}{\phi} \right\rfloor \tag{4}$$

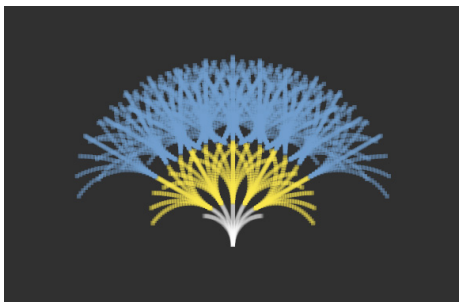
평면이 회전하는 각이 더 작을수록 탐색할 수 있는 경로 후보의 수는 많아지며 또한 연산 시간도 길어지게 된다. 특히 평면이 회전하는 각도가 180°일 경우 2차원 평면에서 경로를 찾는 특수한 경우가 된다.

2.2 충돌 검사 알고리즘

3차원 장애물 충돌 검사는 주로 라이다 기반 항법 알고리즘으로 생성한 정밀 3차원 pointcloud 지도 내에서 수행된다. 생성된 지도의 각 point들을 장애물로 고려할 때 아래 나열된 조건을 만족하면 충돌이라고 간주하였다.

- 장애물과 원의 중심 간의 거리와 원의 반지름의 차이가 safety distance 이내
- 장애물과 현재 평면 사이의 거리가 safety distance 이내
- 장애물과 탐색 시작 위치 간 거리가 탐색 반경 이내

위에 주어진 조건을 모두 만족한다면 현재 검사하는 경로와 장애물 간 충돌이 발생한 것으로 고려한다. 충돌이 발생한 경우 해당 경로를 충돌 발생 부분으로부터 safety distance만큼 뺀 경로를 결과 경로로 가진다.



[Fig. 4] Trajectories obtained after the 3rd search. The white line represents the first trajectory, the yellow line represents second trajectory, and the blue represents result of the third trajectory. This figure only depicts the 2-dimensional aspect

2.3 경로 추가 탐색

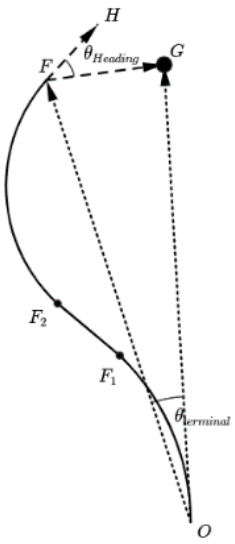
앞서 경로 후보들을 생성하고 해당 경로 후보들에 대해 충돌 검사를 수행한 후, 각 경로가 끝나는 부분에서 경로 생성 및 충돌 검사 과정을 다시 수행하여 추가로경로를 탐색한다. 매 탐색 시 이전 탐색에서 생성된 경로의 끝 부분에서 시작하며, 이때 이전 탐색 경로의 진행 방향은 다음 탐색에서의 중심 방향이 된다. 탐색 횟수가 늘면 더 먼 경로를 찾을 수 있으나 탐색 시간이 길어진다는 단점이 있다. 본 논문에서는 총 3번 탐색을 진행하였다. [Fig. 4]는 장애물이 없는 공간에서 경로 탐색을 총 3번 진행한 결과를 평면으로 잘랐을 때를 보인다. 흰 선은 첫 번째, 노란색 선은 두 번째, 그리고 파란색 선은 세 번째 탐색 결과를 의미한다. 이전 탐색에서 생성된 경로의 끝 부분에서 해당 경로의 진행 방향을 고려하여 다음 탐색의 경로가 생성되었고, 따라서 두 경로가 서로 이어짐을 확인할 수 있다.

2.4 경로 선택

2.1에서 제안하는 기법으로 탐색하고 2.2에서 제시한 방법으로 충돌 검사를 진행한 경로들 중 최적의 경로를 찾기 위해 각 경로의 score를 검사하여 최고의 score를 가지는 경로를 최종 경로로 택한다. 각 경로의 총 score는 식 (5)와 같다.

$$\theta J_{traj} = w_1 J_{terminal} + w_2 J_{heading} + w_3 J_{distance} - w_4 J_{curvature} \tag{5}$$

이때 $J_{terminal}$은 경로 시작 위치에서 경로의 끝을 가리키는 단위 방향 벡터와, 경로 시작 위치에서 목적지를 가리키는 단위 방향 벡터의 내적으로 주어진다. $J_{heading}$은 경로의 끝에서 쿼드로터의 진행 방향을 가리키는 단위 방향 벡터와 경로 끝에서 목적지를 가리키는 단위 방향 벡터의 내적이다. $J_{distance}$는 경로의 끝과 목적지 사이 거리를 고려한 score으로 다른



[Fig. 5] Illustrations of the angles between the goal point, the end of the trajectory, and the heading vector at the end of the trajectory. The vectors $\overrightarrow{OG}$, $\overrightarrow{OF}$, symbolize the directions towards the goal point and the end of the trajectory from the starting point. The vectors $\overrightarrow{FG}$, $\overrightarrow{FH}$ represents the directions towards the goal point, and heading vector at the end of the trajectory from the end of the trajectory

score 항과 비슷한 크기를 가지기 위해 지수 함수를 사용하였다. $J_{curvature}$은 각 경로를 구성하는 호의 곡률을 고려하여 결정된다. w_1, w_2, w_3, 그리고 w_4는 각 항의 비중을 결정하기 위한 계수이다. 식 (6)~(9)는 식 (5)의 각 항을 나타낸다.

$$J_{terminal} = \cos(\theta_{terminal}) \tag{6}$$

$$J_{heading} = \cos(\theta_{Heading}) \tag{7}$$

$$J_{distance} = e^{-d} \tag{8}$$

$$J_{curvature} = \frac{\kappa_1}{r_1} + \frac{\kappa_2}{r_2} + \frac{\kappa_3}{r_3} \tag{9}$$

식 (8)에서 d는 경로의 끝과 목적지 사이의 Euclidian distance로 주어지며, $\theta_{terminal}$은 시작점을 기준으로 하여 목적지를 향하는 단위벡터와 경로의 끝을 향하는 단위벡터의 내적이고, $\theta_{Heading}$은 경로의 끝점을 기준으로 하여 목적지를 향하는 단위벡터와 경로의 끝에서의 진행 방향을 향하는 단위벡터의 내적이다. 임의의 경로에서 score을 구하기 위해 필요한 각각의 변수들을 [Fig. 5]에 표시하였다. 식 (9)에서 κ_i는 i번째 경로의 곡률이며, r_i는 i번째 경로의 반지름을 의미한다. 원의 경우 반지름이 클수록 곡률이 작아지므로, 탐색 반경에 의한 영향을 최소화하기 위해 반지름을 나누었다. 경로의 전체 score을 계산할 때 $J_{curvature}$을 뺌으로써 곡률이 작을수록 더 큰 score을 가지도록 하여 곡률이 작은 경로가 더 높은 score을 가질 수 있도록 하였다.

2.5 모델 예측 제어기

생성된 경로를 추적하기 위해 [6]에서와 같이 쿼드콥터의 내부 자세 제어기 의해 자세 안정화가 된 모델을 기반으로 한 비선형 모델 예측 제어기를 구현하여 적용하였다. 아래에 제시한 쿼드콥터 모델은 [7]을 바탕으로 한, 내부 자세 제어기를 가정하여 식 (10)~식 (13) 으로 나타내었다.

$$\dot{\boldsymbol{p}} = \boldsymbol{v} \tag{10}$$

$$\dot{\boldsymbol{v}} = \frac{1}{m}(\boldsymbol{R}_{IB}\sum_{i=0}^{N_r}\boldsymbol{F}_{T,i} - \boldsymbol{R}_{IB}\sum_{i=0}^{N_r}\boldsymbol{F}_{aero,i}) \tag{11}$$

$$\dot{\boldsymbol{R}}_{IB} = \boldsymbol{\Omega}\boldsymbol{R}_{IB} \tag{12}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega}\times\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{A}\left[n_1^2, \cdots, n_{N_r}^2\right]^T \tag{13}$$

이때 $\boldsymbol{p}$, $\boldsymbol{v}$는 각각 쿼드콥터의 3차원 위치 벡터 및 속도 벡터이다. $\boldsymbol{R}_{IB}$는 현재 쿼드콥터의 회전 행렬이고, $\boldsymbol{\omega}$는 각속도 벡터이며 $\boldsymbol{\Omega}$는 $\boldsymbol{\omega}$에 대응하는 교대 행렬(skew-symmetric matrix)이다. $\boldsymbol{F}_{T,i}$는 i번째 모터에서 발생하는 추력, $\boldsymbol{I}$는 관성 텐서이다. n_i는 i번째 모터의 회전 속도를 의미하며, $\boldsymbol{A}$는 각 모터의 회전수를 돌림힘 벡터로 표현하기 위한 행렬이다. 내부 자세 제어기가 포함된 쿼드콥터 모델의 경우 롤, 피치, 요 각도는 식 (14), (15), (16)과 같이 1차 미분 방정식으로 근사할 수 있다.

$$\dot{\varphi} = \frac{1}{\tau_\varphi}(k_\varphi\varphi_{cmd} - \varphi) \tag{14}$$

$$\dot{\theta} = \frac{1}{\tau_\theta}(k_\theta\theta_{cmd} - \theta) \tag{15}$$

$$\dot{\psi} = \psi_{cmd} \tag{16}$$

식 (14)~식 (16)에서 φ, θ, ψ는 각각 쿼드콥터의 롤, 피치, 그리고 요 각도이다. k_φ와 k_θ는 롤, 피치 게인이며, τ_φ, τ_θ는 각각 롤과 피치 각도의 시간 상수를 의미한다. 식 (10)~(16)을 기반으로 한 모델 예측 제어기의 비용함수는 식 (17)으로 주어진다.

$$\min_U \int_{t=0}^{T} \|\boldsymbol{x}(t) - \boldsymbol{x}_{ref}(t)\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}(t)\|_{\boldsymbol{R}}^2 dt + \|\boldsymbol{x}(T) - \boldsymbol{x}_{ref}(T)\|_P^2 \tag{17}$$

subject to $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$;
$\boldsymbol{u}(t) \in U$
$\boldsymbol{x}(0) = \boldsymbol{x}(t_0)$

식 (17)에서 $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$는 식 (10)~(17)을 통하여 구성할 수 있다. 비선형 모델 예측 제어기의 구현 및 검증에 관해서는 [7]를 참고하였으며 최적 제어 문제의 해를 찾기 위해 자동 제어와 동적 최적화를 위한 오픈소스 라이브러리인 ACADO Toolkit[8]을 활용하였다.

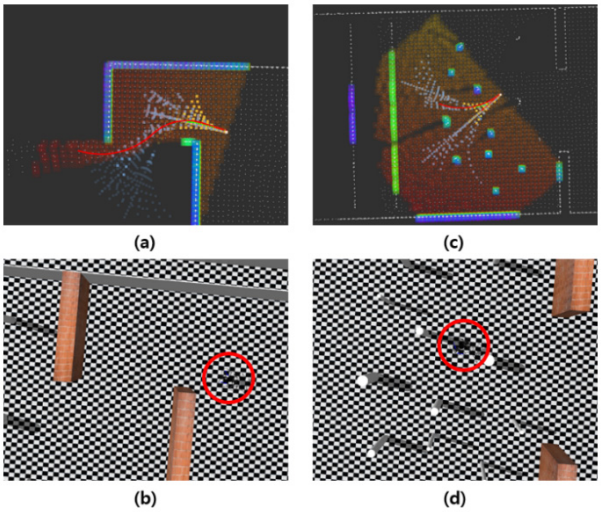
모델 예측 제어기의 비용함수 식 (17)의 reference point x_{ref}는 생성한 경로에서 원의 각속도와 선속도 간의 관계를 이용하여 계산할 수 있다. 모델 예측 제어기의 sampling time을 고려하면 reference point 간 각도 θ_{sample}은 식 (18)을 통해 구할 수 있다. 이때 l_{sample}은 time interval과 속도를 고려한 reference point 사이 거리를 의미한다.

$$\theta_{sample} = \frac{l_{sample}}{r} \tag{18}$$

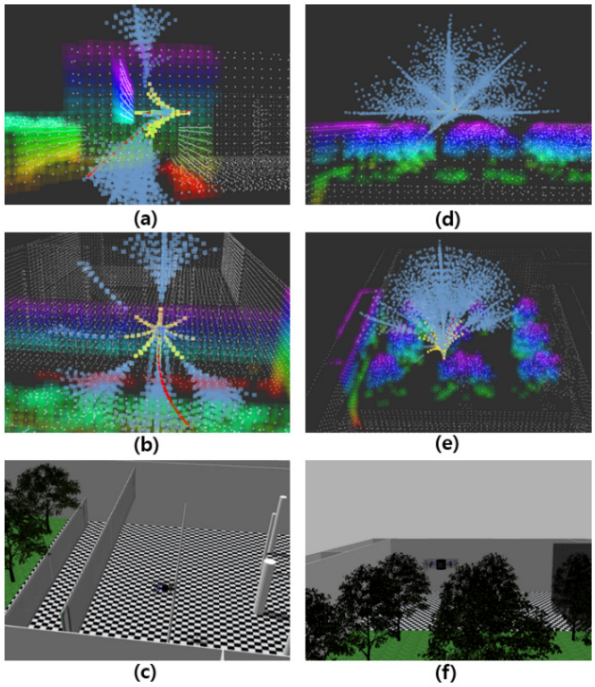
3. 시뮬레이션 검증

3.1 경로 생성 결과

제안한 3차원 경로 계획 알고리즘을 검증하기 위해 Gazebo 시뮬레이션을 활용하였다. Gazebo 시뮬레이터는 ROS와 연동이 가능하고 다양한 형태의 장애물을 설치하여 테스트할 수 있다는 장점이 있어 사용하였다. 시뮬레이션 환경에서 수평 방향 복잡한 기동과 수직 방향 기동이 필수적인 복잡한 환경을 구성하여 알고리즘을 검증하였다.



[Fig. 6] Result of path generation in an environment that requires the generation of a 2D trajectory to avoid collisions. (a) Path generation result in front of two big walls, (b) Gazebo simulation environment with two big walls in front of the UAV. (c) Path generation result between cylinders. (d) Gazebo simulation environment with many cylinders



[Fig. 7] Result of path generation in the environment that necessitates a 3-dimensional trajectory. (a), (b) Path generation result in front of the net, (c) Gazebo simulation environment with a net in front of the UAV, (d), (e) Path generation result above trees, (f) Gazebo simulation environment with trees below the UAV

[Fig. 6]은 2차원 회피 경로를 필요로 하는 환경에서의 알고리즘 검증 결과를 나타낸다. 경로를 생성할 땐 3차원의 경로를 모두 고려하였으나 확인의 편의를 위하여 평면 위의 경로 후보들만 표시하였다. [Fig. 6(b)]는 경로 생성 결과가 충분히 smooth한지 확인하기 위해 전방에 벽이 2개 놓여있는 환경을 구성한 Gazebo 시뮬레이션이며, [Fig. 6(a)]는 이때 경로 계획 알고리즘의 경로 생성 결과를 나타낸다. 빨간색으로 표시된 경로는 score가 가장 높아 최종적으로 선택한 경로이다. 생성된 경로 후보들과 최종 선택 경로들 모두 추가적인 smoothing 과정 없이도 급격하게 꺾이는 부분 없이 부드러운 직선 및 곡선으로 경로가 구성되어 있음을 보인다. [Fig. 6(d)] 는 장애물 검사 결과를 검증하기 위해 Gazebo 시뮬레이션 환경에서 원기둥 형태의 장애물이 여러 개 놓은 모습이며, [Fig. 6(c)]는 이때 경로 계획 알고리즘의 경로 생성 결과를 보여준다. 각 원기둥 근처에는 경로 후보들이 생성되지 않으며 충돌이 없는 경로 후보들 중 가장 score가 높은 경로가 선택되었다.

[Fig. 7]은 2차원으로 이동하는 경로가 존재하지 않을 때의 경로 생성 결과를 보여준다. [Fig. 7(c)]와 같이 UAV 전방에 수평 방향 장애물인 그물이 있는 환경을 Gazebo 시뮬레이션 환경에서 구성하였을 때, [Fig. 7(a)]와 [Fig. 7(b)]는 3차원 공간에

서 생성된 경로 후보들을 바탕으로 다음 경로 탐색이 이루어져 벽 위와 아래로 진행하는 경로들이 생성되는 모습을 보이며, 최종적으로 벽 아래로 진행하는 경로가 선택되었다. [Fig. 7(f)]는 UAV 주위에 여러 나무가 있는 상황을 Gazebo 시뮬레이션 환경을 구성한 모습이다. 이때 [Fig. 7(d)], [Fig. 7(e)]에서 확인할 수 있듯이 나무로 인해 비행할 수 없는 경로들은 생성되지 않았고 목적지에 도달하기 위해 직선으로 쭉 뻗는 경로가 선택되어 UAV가 이를 따라 비행하였다.

3.2 충돌 검사 시간 검증

제안한 충돌 검사 방법이 경로 위 모든 점과 장애물 간 거리를 검사하는 방법보다 연산량이 적음을 보이기 위해 두 방법의 충돌 검사 연산 시간을 비교하였다. [Table 1]은 0.1 m 간격으로 경로 위 모든 점을 검사한 경우의 연산량 비교를 보인다. 모든 경우에서 제안한 알고리즘이 경로 위 모든 점과 장애물 간 거리를 검사하는 방법보다 연산 시간이 적게 나타났다. [Table 2]는 0.2 m 간격으로 경로 위 모든 점을 검사한 경우의 연산량 비교를 보인다. Pointcloud가 많은 경우 첫 번째 탐색 경로를 검사할 때는 기존 방법이 연산 시간이 적지만 세 번째 탐색 경로까지 검사를 마친 후에는 제안한 검사 방법이 연산 시간이 더 작음을 보인다. [Table 1]과 [Table 2]는 제안한 장애물 검사 방법이 경로 위 모든 점을 검사하는 방법에 비해 연산 시간이 짧을 뿐만 아니라 pointcloud의 개수가 늘어났을 때 연산 시간의 증가량도 작음을 보인다.

[Table 1] The performance comparison results between the proposed collision check method and the method of checking every point in the trajectory with an interval of 0.1 m

# of pointcloud	Collision check method	Processing Time [ms]			
		Total	1st	2nd	3rd
2993	Proposed	160	12	62	86
	Interval-0.1 m	636	47	104	485
3139	Proposed	123	12	50	61
	Interval-0.1 m	641	49	76	516
5257	Proposed	233	51	87	95
	Interval-0.1 m	1553	82	122	1349
5258	Proposed	238	20	103	115
	Interval-0.1 m	1715	81	174	1460

[Table 2] The performance comparison results between the proposed collision check method and the method of checking every point in the trajectory with an interval of 0.2 m

# of pointcloud	Collision check method	Processing Time [ms]			
		Total	1st	2nd	3rd
2956	Proposed	117	11	42	64
	Interval-0.2 m	322	23	37	262
2984	Proposed	115	12	50	53
	Interval-0.2 m	248	22	33	193
5215	Proposed	246	49	111	86
	Interval-0.2 m	877	39	131	707
5262	Proposed	293	50	116	127
	Interval-0.2 m	883	36	131	716

4. 결 론

본 논문에서는 곡률을 기반으로 한 3차원 경로 계획 알고리즘을 제안하고 이를 Gazebo 시뮬레이션 환경에서 검증하였다. 곡률 기반으로 경로 후보들을 생성한 다음 장애물 검사를 진행하는 과정을 3번에 걸쳐 반복하여 경로 후보들을 생성하였고, 생성된 경로 후보들의 score를 계산하여 최종 경로를 선택하였다. 곡률을 기반으로 3차원 경로를 탐색할 경우 추가적인 smoothing 과정을 생략할 수 있으며 또한 장애물 충돌 검사 시 연산량을 줄일 수 있음을 보였다. 제안한 알고리즘은 Gazebo 시뮬레이터를 기반으로 복잡한 움직임을 요구하는 다양한 환경에서 검증되었다. 시뮬레이션 결과 3차원 경로를 생성하여 복잡한 장애물을 피해갈 수 있음과 연산 시간이 감소함을 확인하였다.

References

[1] D. Mellinger and V. Kumar, “Minimum Snap Trajectory Generation and Control for Quadrotors,” *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, pp. 2520-2525, 2011, DOI: 10.1109/ICRA.2011.5980409.

[2] A. A. Saadi, A. Soukane, Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, “UAV Path Planning Using Optimization Approaches: A Survey,” *Archives of Computational Methods in Engineering*, vol. 29, pp. 4233-4284, Apr., 2022, DOI: 10.1007/s11831-022-09742-7.

[3] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, “Falco: Fast Likelihood-based Collision Avoidance with Extension to Human-guided Navigation,” *Journal of Field Robotics*, vol. 37, no. 8, pp. 1300-1313, Dec., 2020, DOI: 10.1002/rob.21952.

[4] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, “Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529-3536, Oct., 2019, DOI: 10.1109/LRA.2019.2927938.

[5] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, “Faster: Fast and safe trajectory planner for navigation in unknown environments,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922-938, Apr., 2022, DOI: 10.1109/TRO.2021.3100142.

[6] M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010, DOI: 10.1109/ROBOT.2010.5509920.

[7] M. Kamel, M. Burri, and R. Siegwart, "Linear vs nonlinear MPC for trajectory tracking applied to rotary wing micro aerial vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3463-3469, Jul., 2017, DOI: 10.1016/j.ifacol.2017.08.849.

[8] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO Toolkit–An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298-312, May, 2011, DOI: 10.1002/oca.939.

박 재 용

2022 경북대학교 전자공학부(학사)

2022~현재 KAIST 전기 및 전자공학부 석사과정

관심분야: UAV, 경로 계획, 항법, 모델 예측 제어

Maulana Bisyir Azhari

2020 University of Indonesia, Electrical Engineering (B.S.)

2022~now KAIST, Electrical Engineering (Ph. D. Student)

Field of Intrest: Drone, Robotics, Autonomous Robot

김 보 성

2017 인하대학교 전자공학(학사)

2019 KAIST CCS 모빌리티 대학원(석사)

2019~현재 KAIST 전기 및 전자공학부 박사과정

관심분야: 로보틱스, 항법, UAV, 탐사, 경로 계획

심 현 철

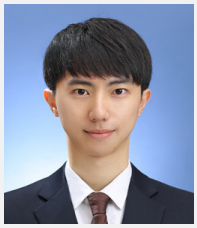
1991 서울대학교 기계공학(학사)

1993 서울대학교 기계공학(석사)

2000 University of California, Berkeley Mechanical Engineering(박사)

2007~현재 KAIST 교수

관심분야: 로보틱스, 드론, 자율로봇, 자율주행차량



이 승 욱

2018 한양대학교 기계공학(학사)

2020 KAIST 로봇공학학제전공(석사)

2020~현재 KAIST 전기 및 전자공학부 박사과정

관심분야: 로보틱스, 안티 드론, UAV, 제어