

21회 한국로봇항공기경연대회

참가드론개발보고서

속	서울대학교
참가팀명	불나비
지도교수	이관중
대 표 자	손성일 외 9명

제21회 한국로봇항공기경연대회 참가드론개발보고서를 첨부와 같이 제출합니다.

2023년 7 월 24 일

대 표 자 명 손성일

지도교수 명 이관중

목 차

0. 참가팀 현황	1
1. 기술개발의 목표 및 내용	2
1-1. 개발 목표	2
1-2. 개발 내용	2
2. 기술개발 추진방법·전략	3
2-1. 기술개발 추진방향	3
2-2. 기술개발 전략	10
3. 자동비행 시스템 및 구현 기술	11
3-1. 유도, 제어, 항법시스템 설계	11
3-2. 구성품의 적정성	13
3-3. 지상 및 비행 시험 결과	16
4. 임무수행 구현기술	17
4-1. 임무수행 시나리오	17
4-2. 임무수행 방법	18
5. 안전장치에 대한 설명	19
5-1. 안전장치의 구성 및 성능	19
5-2. 안전장치의 기술적 적정성	21
5-3. 안전장치의 시뮬레이션 또는 실제 작동	21
6. 시스템 설계 및 제작상의 특징점	21
7. 개발 애로사항	
첨부. 비행시험 데이터 & 비행시험 동영상	

0. 참가팀 현황 (드론 조종사 - 비고란에 표시)

No	직 위	성 명	주민등록번호(앞7 자리)	학년	소속학과	E-mail	비고
1	팀 장	손성일	990518-1	4	기계항공공학부	alldridge99@snu.ac.kr	
2	팀 원	이종안	010721-3	3	기계항공공학부	johnny3357@snu.ac.kr	
3	팀 원	남궁호	990304-1	4	항공우주공학과	gur4944@snu.ac.kr	
4	팀 원	김석진	990619-1	4	기계항공공학부	ksjn1m@snu.ac.kr	
5	팀 원	곽승표	040520-3	1	기계공학부	ksp3269@snu.ac.kr	조종사
6	팀 원	김태중	030731-3	2	항공우주공학과	sae1809@snu.ac.kr	
7	팀 원	김태훈	020324-3	2	기계공학부	horse324@naver.com	
8	팀 원	허정화	000202-3	4	항공우주공학과	hjhcn22@gmail.com	
9	팀 원	김호준	010212-3	4	항공우주공학과	hojjunekim@snu.ac.kr	
10	팀 원	최진태	010516-3	3	항공우주공학과	chengu1000@naver.com	
	총	(10명)	대학원(명), 4학년(7명), 3학년(2명), 2학년(2명), 1학년(1명)				

1. 기술개발의 목표 및 내용

1-1. 개발 목표

본 팀에서는 도심 베란다 배송 임무를 수행하기 위해 센서 기반 장애물 탐지 기술, 장애물 회피 및 목표점 통과를 위한 경로 생성 기술, 베란다 화물 배송 임무 실현, 자동 착륙 기술을 개발하는 것이 목표이다.

1-2. 개발 내용

(1) 센서 기반 장애물 탐지

대회에서 주어진 장애물은 지상에 놓인 두 대의 사다리차이며, 두 장애물 사이가 주어진 경로점(WPT)으로 설정되어 있다. 이에 따라 본 팀은 전방 영상 센서를 이용하여 장애물은 탐지해 경로를 생성하고자 한다.

(2) 장애물 회피 및 목표점 통과를 위한 경로 생성

전방 영상 센서에서 생성한 Pointcloud의 2차원 군집화와 기체의 항법 좌표 등을 통해 장애물의 배치와 방향을 알아내어 목표점을 통과한다. 이때, 기존에 획득한 데이터와 새로 획득한 데이터를 모두 이용해 보다 정확한 장애물의 위치를 알 수 있도록 한다. 또한, 안정적인 비행을 위해 기체의 급격한 방향 전환 없이 속도를 유지하며 비행할 수 있는 경로를 생성해 목표점이 업데이트 되어도 안정적인 비행 상태를 유지하고자 한다.

(3) 베란다 화물 배송

기체는 임무 수행을 위하여 베란단에 접근해야 하는데, 임무장비는 이를 위해 안정적인 화물 배송과 비행 시 안정적인 기체 자세 유지를 위해 디자인 되어야 한다. 이 때문에 임무장비는 연장형으로 제작되었으며, 장비의 복잡도를 줄여 배선을 간단화하고, 고장 확률을 줄이기 위해 모터 하나의 동력을 이용하여 작동되도록 디자인한다. 또한, 임무 수행지 근처에서 객체 인식을 통해 베란단을 기준으로 정확한 자세를 유지할 수 있도록 한다.

(4) 자동 정밀 착륙

기체의 하방 카메라를 이용하여 착륙지에 놓인 Apriltag 마커를 인식해 최종 착륙 경로를 생성하여 신속하고도 정확한 착륙을 할 수 있도록 한다. 해당 마커와 기체의 항법 데이터를 이용하면 기체가 정확한 위치를 유지하면서 착륙할 수 있다.

2. 기술개발 추진방법·전략

2-1. 기술개발 추진방향

(1) 센서 기반 장애물 탐지

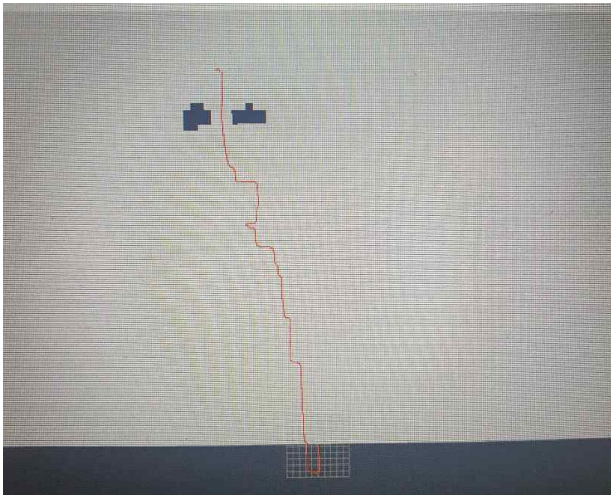
스테레오 카메라로부터 얻은 Pointcloud 데이터를 바탕으로 장애물을 탐지하여 목표점을 생성한다. 이렇게 생성된 목표점은 경로 생성 알고리즘에 전달되어 기체의 경로 계획에 사용된다. 본 기체에 사용된 ZED2i 스테레오 카메라는 ROS의 PointCloud2 메시지의 형태로 전방 물체에 대한 토픽을 전송한다.

Pointcloud는 두 단계를 거쳐 space frame으로 변환된다. 스테레오 카메라에 의해 생성된 pointcloud의 좌표는 이미지의 가로, 세로를 각각 x , y 축으로 이미지와 수직 방향을 z 축으로 설정된다. 첫 단계는 카메라 좌표계에서 드론의 body frame으로 변환하는 것이다. 두 번째 단계는 드론의 위치 정보와 자세 정보를 바탕으로 body frame에서 space frame으로 변환하는 것이다. Space frame은 이륙 위치를 원점으로하는 ENU 좌표계를 채택한다.

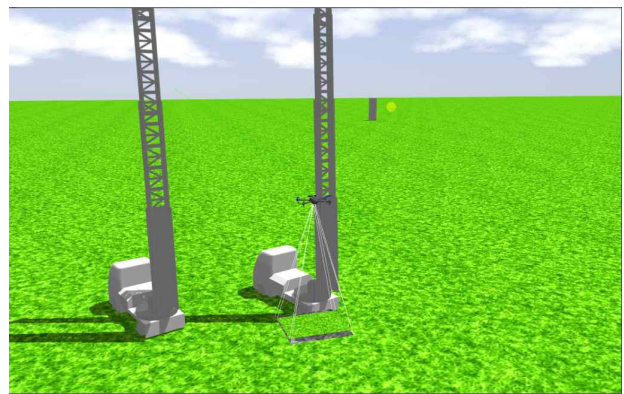
2D 점유격자지도 (2D Occupancy grid map)은 space frame 상에 위치한다. Space frame으로 변환된 pointcloud는 2D 점유격자지도에서 해당하는 격자의 위치에 맞추어 저장된다. 점유격자지도는 비행제한이 가로 200m, 세로 100m인 본 임무에 맞추어 0.1m의 해상도로 각각 4000개의 가로, 2000개의 세로 격자로 이루어진다. 일반적인 라이더의 경우 RANSAC 알고리즘을 통해 바닥면의 pointcloud를 처리한다. 그러나 이번 임무는 15m의 고도를 유지하며 비행을 하기에 이륙과 착륙을 제외하고는 z 방향 이동이 크지 않다. 따라서 space frame으로 변환할 때, pointcloud의 z 좌표가 3m 이상이고 드론의 현재 z 좌표와 2m 이하로 차이나는 pointcloud만 점유격자지도에 사용하는 방식으로 바닥면을 제외한 나머지 점들만 처리할 수 있도록 한다.

점유격자지도에 저장된 격자의 위치들을 DBSCAN (Density-based spatial clustering of applications with noise) 알고리즘을 통해 군집화한다. DBSCAN 알고리즘은 다른 군집 알고리즘들보다 계산 시간이 훨씬 적게 걸리며, k -means 알고리즘처럼 군집의 수를 hyperparameter로 지정해줄 필요가 없어 채택하였다. 또한, 3D pointcloud에서 직접적으로 DBSCAN 알고리즘을 적용하지 않고, 2D의 격자에 DBSCAN 알고리즘을 사용하여 외란에도 강건하고 시간복잡도 측면에서도 이점을 얻을 수 있다. DBSCAN의 경우 반경내에 위치해야 할 점의 수와 반경의 크기를 지정해주어야 한다. 반경의 크기는 격자의 해상도인 0.1m로 지정하였고, 점의 수는 noise에 영향을 받지 않을 수 있도록 20으로 설정하였다.

장애물들의 Pointcloud 정보가 수집된 이후, 군집화된 장애물들의 centroid와 그 크기를 바탕으로 장애물을 둘러싸는 원형의 envelope를 설정한다. 원형의 envelope의 반지름의 경우, 같은 클러스터에 위치한 격자들 중 centroid와 가장 먼 격자까지 거리에 안전거리로 1m를 더하여 설정하여 이를 침범하지 않도록 경로점을 설정하여 장애물을 회피한다. 사다리차 경로점의 경우, 특히 두 장애물 사이로 이동하여야 하기에, 두 군집 centroid의 중심과 법선벡터를 찾아 가장 안전한 경로로 이동할 수 있도록 한다.



[그림 2-1-(1)-1] 생성된 2차원 격자의 모습



[그림 2-1-(1)-2] SILS 상의 기체 모습

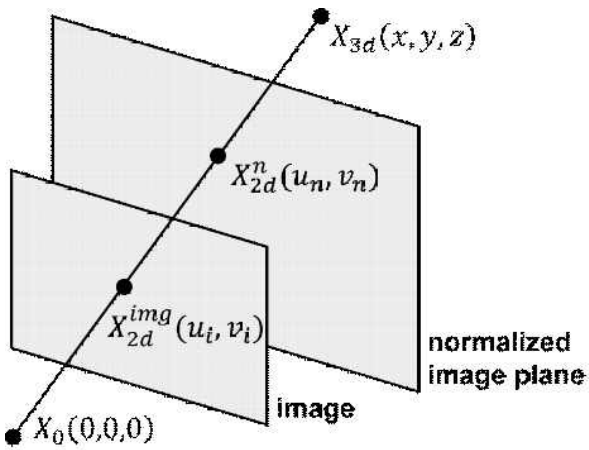
(2) 영상 기반 목표물 탐지

베란다에 화물을 정확히 배달하기 위해서는 베란다 3층 벽면에 위치한 십자가 모양의 마커를 탐지하여 드론의 방향을 정렬해야 한다. 이를 위해, 전방 카메라에서 얻은 영상 정보에서 직접 학습시킨 YOLOv5 모델을 이용해 십자가의 픽셀 정보를 얻은 후, 인식된 십자가의 픽셀들에 대한 깊이 정보를 이용해 십자가의 정확한 평면의 방정식을 얻어낸다. 이후, 해당 평면을 바라보도록 드론의 헤딩을 맞추어 평면을 바라보게 하면, 배송을 시작하기 전 항상 원하는 위치에 드론을 정렬시킬 수 있다.

대회에서 사용될 십자가의 형상과 크기가 정해지지 않았으므로, 해당 십자가를 강건하게 탐지하기 위한 자체 데이터셋을 만들어 벽면의 십자가를 탐지하고자 한다. 해당 모델은 여러 가지 십자가의 모양/크기가 있을 때 각각이 모두 십자가임을 탐지해야 하며, 온보드 컴퓨터의 연산량을 고려했을 때 부담이 크지 않아야 한다. 이에 따라, ROS2에서 얻은 topic을 처리할 수 있는 YOLOv5 계열 모델들이 고려되었으며, 가장 파라미터의 개수가 작은 YOLOv5-tiny 모델로도 여러 가지 십자가들을 정확하게 탐지하는 것을 관찰하여 해당 모델을 기반으로 custom model을 만들어 사용하기로 한다.

십자가를 탐지하여 해당 부분의 bounding box를 얻은 이후, 다시 전방 카메라에서 얻

은 깊이 데이터를 이용하여 해당 부분의 기울기를 얻을 수 있다. 이때, 해당 부분의 깊이 값을 알고 있기에 십자가의 크기와 기울기를 정확히 알 수 있으며, 그 방법은 [그림 2-1-(2)-1]에서 나온 것과 같다. 이를 통해, 기체는 전방에 놓인 벽면의 정렬 상태를 판단할 수 있으며, 십자가를 수직으로 바라보면서 정면에 위치할 수 있도록 기체의 위치를 조정한다.

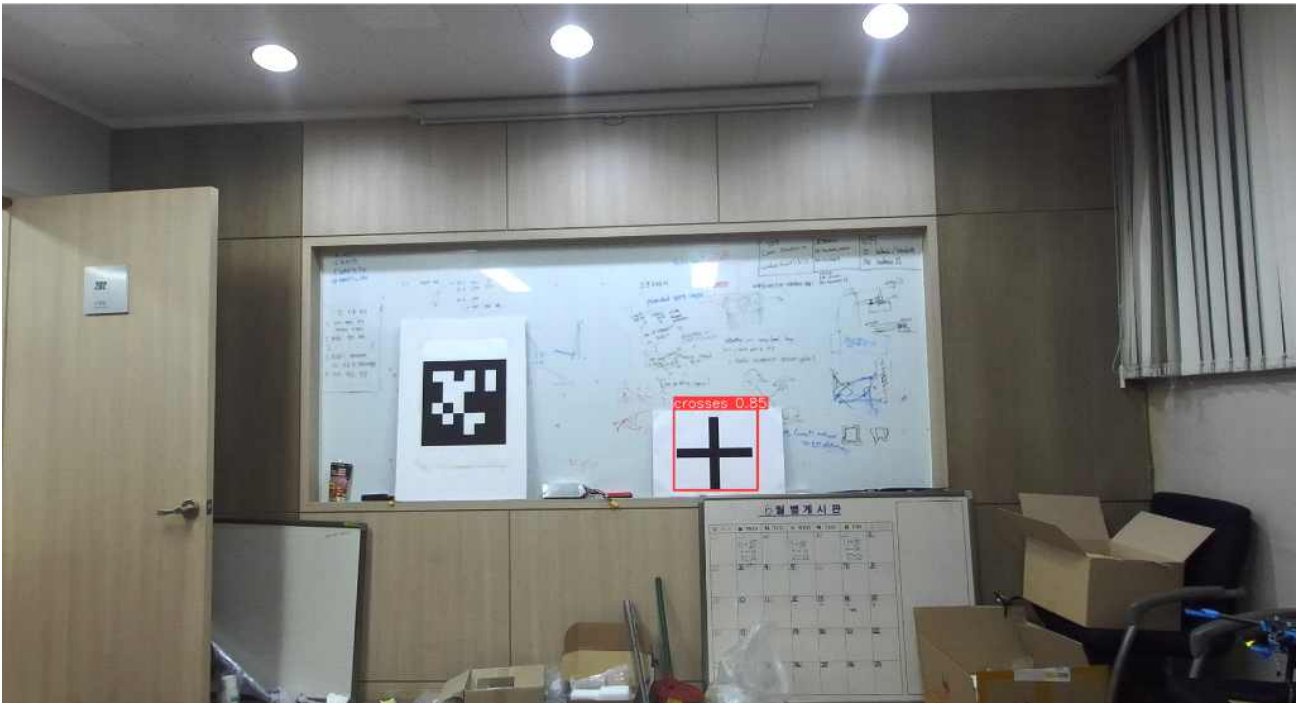


$$\begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = K \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \hat{z} \begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix}$$

\hat{z} : estimated depth from stereo camera
 K : intrinsic matrix

[그림 2-1-(2)-1] 영상에서 얻은 bounding box와 추정된 depth 정보를 통한 3차원 정보 복원 예시

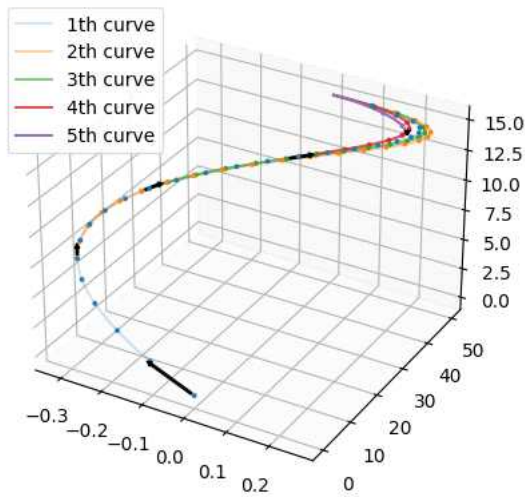


[그림 2-1-(2)-2] custom dataset을 이용해 학습시킨 모델의 십자가 검출 예시

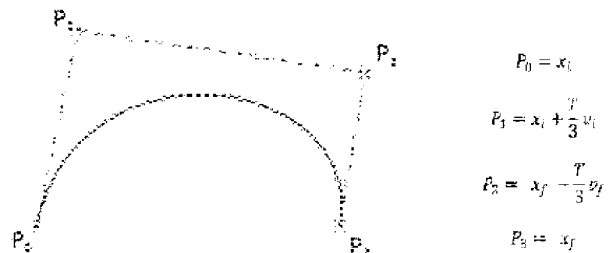
(3) 장애물 회피 및 목표점 통과를 위한 경로 생성

안정적인 기체 유도를 위한 경로 계획은 3차 Bezier curve를 지속적으로 업데이트하여 계획한다. 이때, 각 curve는 4개의 점을 이용한 Bernstein 다항식의 합으로 생성되며, 이 네 점은 시작 위치 x_i , 시작 속도 벡터 v_i , 최종 위치 x_f , 최종 속도 벡터 v_f 를 이용해 결정된다. 이때, 최종 위치 x_f 와 최종 속도 벡터 v_f 는 외부 노드에서 주어지며, 시작 위치와 속도는 기체가 현재 인식하고 있는 시작 위치와 속도를 사용한다. 이 결과로 얻은 경로의 점들을 FC에 publish하게 된다.

기체의 경로를 안정적으로 유도하기 위해, 새로 갱신된 경로의 처음 속도는 기존의 속도 v_i 와 같아야 하고, 마지막 속도는 v_f 와 같아야 한다. 그러나 FC의 속도 추정값은 실제로 부정확하고, 위치 추정값은 GPS의 정확도로 인해 비교적 정확한 값을 가지므로, 처음 속도 v_i 는 기존에 추종하던 경로의 현재 속도값을 사용하고 처음 위치 x_i 는 기체에서 추정하고 있는 위치값을 사용한다. 이에 따라 경로를 구성하는 네 개 위치점은 다음 [그림 2-1-(3)-2]와 같다. 기체의 비행 안정성을 위해 각각의 점들을 이은 convex hull 내에는 장애물이 존재하지 않아야 하며, 장애물이 존재할 경우 이를 고려한 새로운 경로점을 생성한다.



[그림 2-1-(3)-1] 5초마다 갱신되는 경로를 통해 $x_f = [0,50,10]$, $v_f = [0.01,0,0]$ 을 향해 가는 경로의 모습



[그림 2-1-(3)-2] 각 경로 생성에 사용된 Bezier curve와 그 설정점들의 setting

Pixhawk의 비행 제어 알고리즘은 P 제어를 사용하여 위치 정보를 추종한다. 따라서 각 시간에 본래 기체가 있어야 하는 위치를 publish 하면 기체는 해당 위치에 이미 있기

때문에 정보가 갱신될 때마다 움찔거리며 나아가게 된다. 반면 약 1초 이후의 위치 정보를 publish 했을 때 경로에서 의도하는 것과 동일한 경로를 따르는 것이 확인 되었다. 따라서 경로 생성 노드에서는 해당 시점의 목표점으로 약 1초 후의 위치 정보를 publish한다.

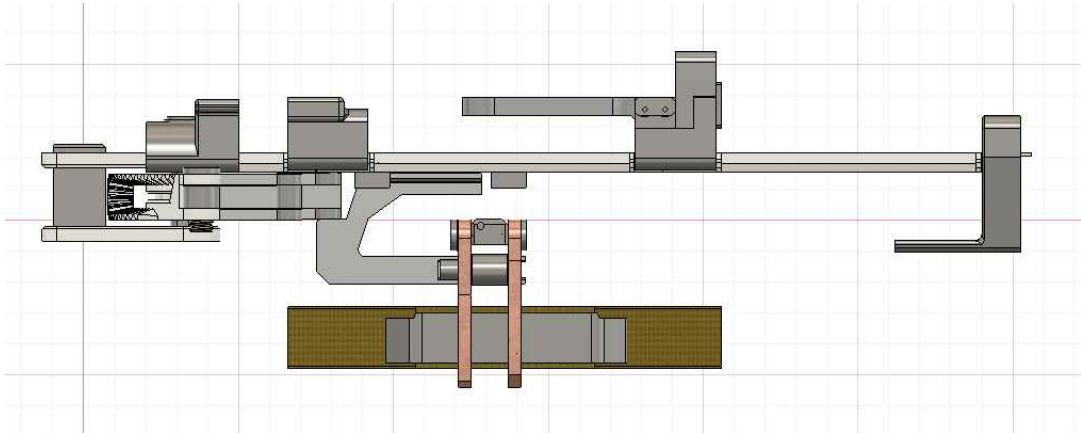
(4) 베란다 화물 배송

이번 대회의 임무는 베란다에 피자박스를 배달하는 것으로 배송하는 방법에 따라 임무장비의 설계가 달라진다. 임무 수행 방식으로 베란다 내부에 기체가 직접 들어가 화물을 놓고 오는 방식을 사용하는 경우 GPS 오차가 생길 가능성이 높기 때문에 베란다 밖에서 팔과 같은 형상을 연장하여 베란다 내부에 기체가 들어가지 않고 물체를 배송하는 방식을 채택한다. 배송 시에는 ground effect를 줄이기 위해 기체 중앙 부분에 피자박스를 수납해 비행하다가 마커를 인식하여 일정 거리 내에 도달하는 경우에 팔을 연장하여 피자박스를 배송한다.

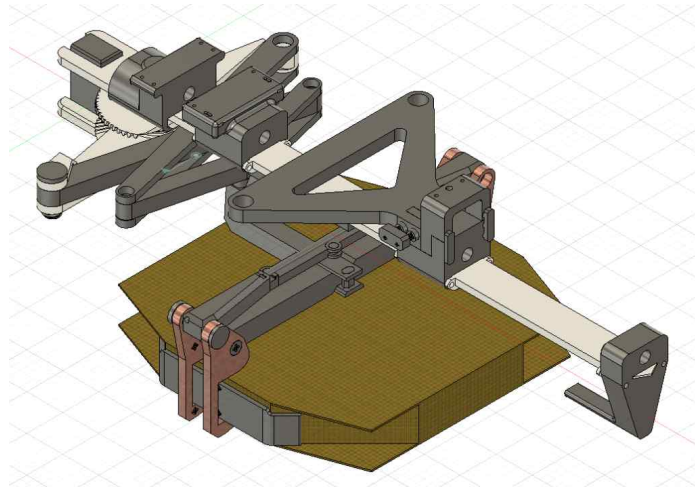
임무장비는 서보 모터를 하나만 사용하여 연결선과 모터의 failure와 같은 임무장비에 생길 수 있는 불확실성을 최대한 줄이고 무게를 줄여 payload를 낮출 수 있게 설계한다. 이를 위하여 팔을 연장하여 end effector가 레일의 끝부분에 도착했을 때 박스를 낙하하는 방법으로 구조 설계를 통한 기계적인 방식을 사용한다. 이 방식은 따로 고정 장치를 구동하는 모터 없이 고정 장치를 해제하여 박스를 낙하할 수 있기 때문에 배선 불량이나 모터의 불량 같은 failure의 가능성을 줄일 수 있다.

구조적 특징으로는 bevel gear를 이용하여 서보 모터의 토크를 scissor extending arm 형태의 팔로 전달하고 팔이 카본을 통해 보완된 레일을 따라 움직이도록 설계하여 진동과 임무장비 무게의 의해 생기는 파손 가능성을 낮춘다. End effector는 팔이 레일의 끝 부분에 도착해야만 bistable hinge가 구부러져 안정장치가 해제되어 상자가 낙하하도록 설계한다.

임무장비의 서보모터는 Jetson Xavier NX에서 Jetson의 GPIO핀을 이용해 PWM 신호를 출력하여 구동한다. 이러한 방법을 사용하면 ROS 노드에서 전달받은 명령을 PWM 입력으로 바꾸어 모터를 조절할 수 있으며 따라서 비행 실험 시에 Jetson Xavier NX와 ground station 사이의 SSH 통신을 통해 ground station에서 임무장비를 제어할 수 있다.



[그림 2-1-(4)-1] 임무장비의 측면 설계도



[그림 2-1-(4)-2] 임무장비의 정면 설계도

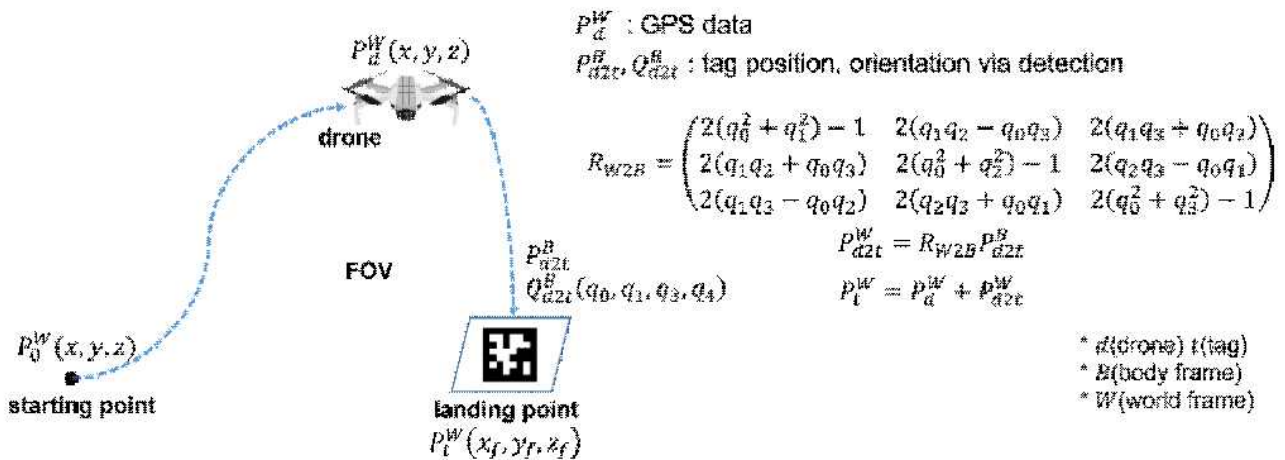
(5) 자동 정밀 착륙

Pixhawk의 자동 정밀 착륙 알고리즘은 IR 등의 신호기를 이용하여 해당 신호기 직상 방으로 이동한 뒤 수직 하강하는 방식으로, 안전하지만 시간이 오래 걸리는 단점이 있다. 본 팀은 이를 Apriltag를 이용한 경로생성 방식으로 대체하여 유동적이지만 안정적인 착륙 경로를 생성하여 착륙하고자 한다.

사전에 종류와 크기를 미리 알고 있는 Apriltag를 이용하여 하방 카메라를 통해 탐지하면, 기체의 body frame에 대한 마커의 position과 orientation을 얻을 수 있다. Apriltag 마커는 카메라를 통해 얻은 이미지의 gradient를 통해 harris corner extraction 등의 방법으로 edge를 찾아 사전에 정의된 모양의 tag를 찾는 방식으로 탐지한다. 이에 따라 카메라의 intrinsic/extrinsic parameter를 이용하여 raw image를 보정하지 않으면, 위치 정보가 큰 noise를 갖게 되어 탐지하기 전의 이미지에 preprocessing이 필요하다.

이를 통해 얻은 tag의 위치 정보 데이터는 14m까지 약 5cm 내외의 noise를 가지며 높은 position 정확도를 보인다. 또한, 착륙을 위한 마커는 지면에 고정되어 있으므로, world frame으로 변환된 마커의 위치는 stationary 해야 한다는 점, 그리고 카메라가 기체의 하방을 바라보고 있어 기체의 orientation을 따르기 때문에 Kalman filter를 이용해 FC에서 구해진 orientation을 같이 이용할 수 있다는 점을 이용해 오차를 보정하여 착륙지의 위치를 알 수 있다.

해당 방법으로 얻은 드론의 pose 데이터를 이용하면 기체에 대한 상대적 tag position을 알 수 있고, 이를 드론의 world frame 좌표를 이용하여 해당 좌표계에서의 tag 좌표를 알 수 있다. 따라서 착륙지점의 좌표는 noise가 Gaussian임을 가정하여 쉽게 구할 수 있어 기체의 경로 계획 목표점 와 그때의 방향 v_f 를 구할 수 있다.

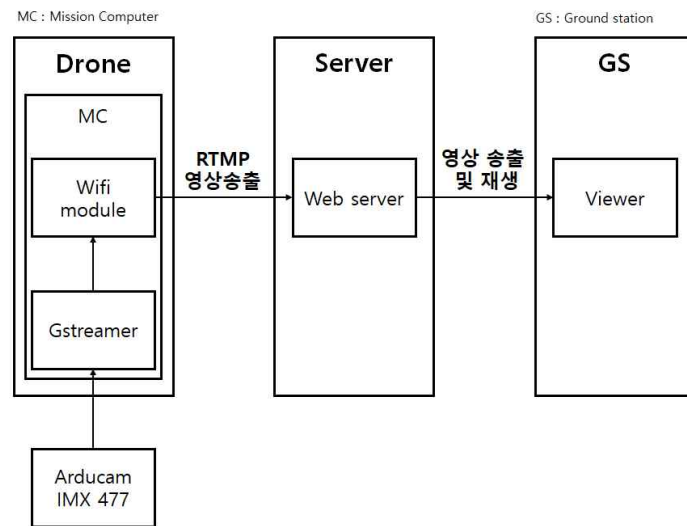


[그림 2-1-(5)] Apriltag를 이용한 정밀착륙 개요 및 수식

(6) 실시간 비행 영상 송수신

기체에서 촬영한 영상을 ground station에 실시간으로 스트리밍하기 위해 RTMP(Real-Time Messaging Protocol) 프로토콜을 사용한다. RTMP 프로토콜은 지속적인 연결을 통해 미디어 스트림을 스트리밍 서버로 전송하여 low-latency streaming을 제공하고, 적응형 비트 전송률 스트리밍을 통해 버퍼링을 방지할 수 있다. 고성능 웹 서버 소프트웨어인 Nginx를 이용하여 웹 스트리밍 서버를 구축한다. Nginx는 트래픽에 따른 속도지연이 적어 일관된 latency를 유지하는 데 유리하다. 웹캠 미디어 데이터를 캡처, 인코딩하여 RTMP 프로토콜을 통해 서버로 보내는 데에는 모듈식 멀티미디어 프레임워크가 광범위한 코덱을 지원하여 맞춤형 시스템을 제작하기에 유리하다고 판단하여 gstreamer 프레임워크를 사용한다. 또한 ground station에서 서버의 영상을 받아 디코딩하고, 재생하는 과정에도 gstreamer를 사용한다.

Latency는 다양한 요소에 의해 결정되게 되는데, 기체 쪽 인터넷을 1Mbps의 휴대폰 핫스팟에서 150Mbps 이상의 와이파이로 변경한 결과 대략 1초 정도의 latency 감소를 확인하였다. 동영상을 디코딩하는 과정에서 특히 latency가 많이 발생한다고 판단하였고, Gstreamer 파이프라인 최적화를 통해 6초 가량의 latency를 감소시켜 최종적으로 약 4초의 latency로 영상 스트리밍 시스템을 구축하였다.



[그림 2-1-(6)] 기체 영상 전송 개요도

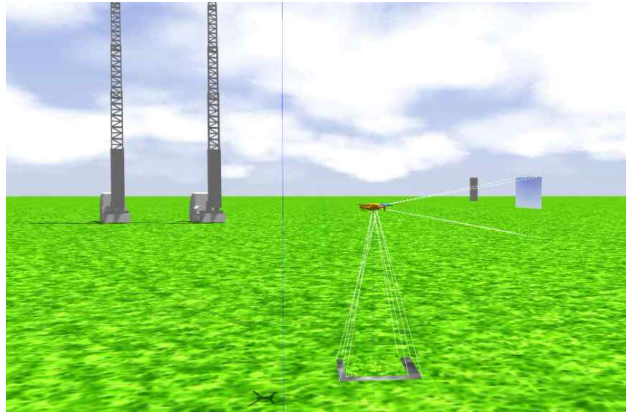
2-2. 기술개발 전략

(1) 시뮬레이션 환경 구축

도심 베란다 배송 임무 환경을 직접 구축하기는 많은 시간과 비용이 필요하여 현실적으로 불가능하다. 또한, 검증되지 않은 로직을 실제 드론에 적용하는 것은 드론 및 기타 임무 장비의 파손 위험이 크다. 따라서 시뮬레이션 환경을 구축하여 임무 수행을 위한 알고리즘이 강건하게 수행됨을 확인한 후, 실제 드론에 적용하도록 한다.

시뮬레이션 환경은 Ubuntu 20.04, ROS2 foxy, Gazebo11, PX4 Autopilot를 이용하여 구축한다. 시뮬레이션이 작동하는 모습은 [그림 2-2-(1)]와 같다. ROS2의 Depth camera 모듈로부터 PointCloud와 이미지를 받아 장애물 인식 및 회피, 마커 인식 알고리즘을 검증하였고, Downwards camera 모듈로부터 이미지를 받아 Apriltag를 이용한 정밀 착륙 알고리즘을 검증하였다. 또한, Gazebo 환경에서의 Offboard control을 통해 Bezier curve를 이용한 경로점 생성 알고리즘도 검증하였다.

우선 Ryzen3700U, RX Vega10 PC 환경에서 SILS를 진행하였고, 그 후에 onboard computer인 Javier Xavier NX를 이용하여 HILS를 진행하였다.



[그림 2-2-(1)] SILS 시뮬레이션 환경의 모습

(2) ROS2 기반 기술개발

Companion Computer의 기술개발은 ROS2를 이용하여 진행되었다. 이는 ROS2가 roscore 모듈이 없고 더 향상된 Middleware를 사용하여 기존의 ROS보다 더 빠른 내부 통신을 지원하기 때문이다. 또한, ROS는 Ubuntu 20.04를 마지막으로 지원하며, 2024년을 마지막으로 사라질 것이기 때문에 지속가능한 프로그램 개발과 유지를 위하여 ROS2를 이용하여 개발을 진행하였다.

(3) Bottom-up 방식

전체적인 임무 수행은 Localization, Perception, Planning, Control, Network system 등 여러 모듈이 합쳐져 이루어진다. 각 팀별로 부분의 임무를 나누어 개발한 다음, 후에 합치는 방식으로 진행한다. 하드웨어팀은 통신팀, 제어팀, 임무장비팀으로 나누어지고, 소프트웨어팀은 장애물 회피팀, 마커 인식팀, 경로 생성팀, 정밀 착륙팀으로 나누어져 모듈 단위의 임무를 개발하여 효율성을 높인다.

3. 자동비행 시스템 및 구현 기술

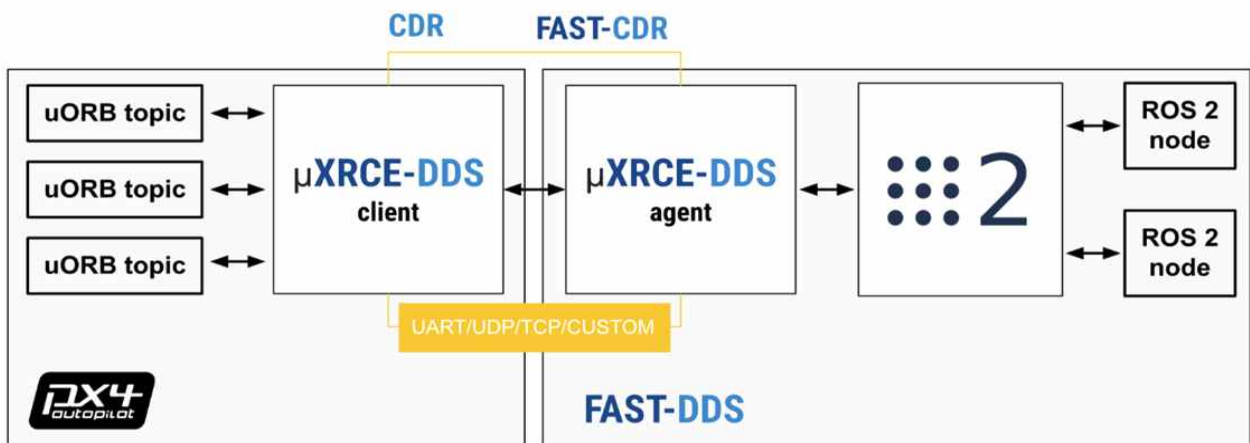
3-1. 유도, 제어, 항법시스템 설계

(1) 유도 시스템

기체 유도를 위한 경로 생성 시스템은 ROS2를 이용해 Bezier curve를 기반으로 제작한다. 각각의 경로는 2-1-(3)에서와 같이 생성되고, 이 경로는 하나의 배열로 CC에 저장된다. 또한, CC는 50Hz 간격으로 생성된 경로점들을 FC에 publish하여 위치 기반 제어 명령을 생성한다.

드론 하드웨어를 직접 제어하는 FC와 유도 임무 소프트웨어를 실행하는 CC 사이의

통신에는 XRCE-DDS bridge를 사용한다. 이는 XRCE-DDS 미들웨어를 이용하여 PX4와 ROS2 노드 사이에서 통신을 하는 소프트웨어인데, FC에서 실행되는 client와 CC에서 실행하는 agent로 이루어진다. Client는 정해진 uORB topic을 DDS data space에 송수신한다. Agent는 DDS data space와 ROS2 노드 사이에서 proxy의 역할을 하며 두 공간을 이어준다. XRCE-DDS bridge는 통신을 위해 UDP 혹은 UART 연결을 사용하므로, 우리 기체에서는 Pixhawk 6X의 TELEM2 port와 Jetson Xavier NX의 GPIO 포트를 연결하여 UART로 통신한다.



[그림 3-1-(1)] PX4와 ROS2 간 통신의 개요

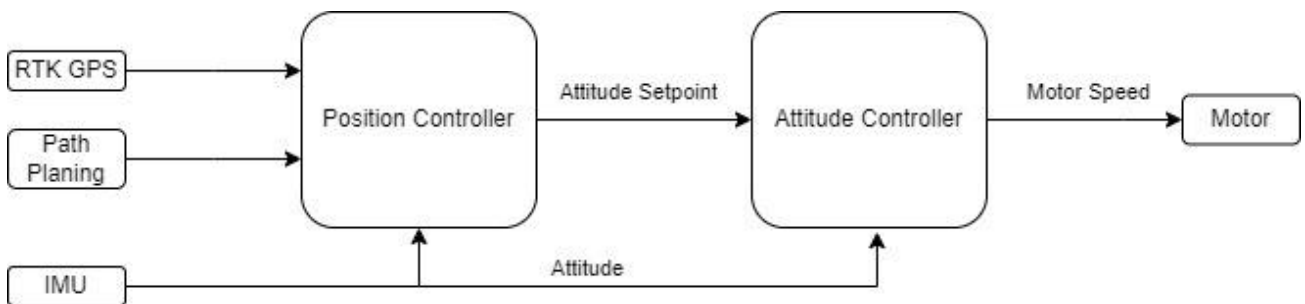
(2) 항법 시스템

기체의 위치 추정을 위해 CDGPS의 일종인 RTK GPS를 사용한다. 임무 수행을 위해서는 정확도가 높고 redundancy를 가진 항법 시스템이 필요했기 때문에 미터 단위의 정밀도를 가진 일반적인 GPS보다 더 높은 정밀도를 가진 RTK GPS를 선택하였다. Base station과의 비교를 통해 전리층이나 대류층에 의한 시간지연이나 다중 경로에 의해 생기는 bias error를 없앨 수 있는 DGPS방식과 반송파의 위상차를 통해 random error를 보정하는 carrier-phase tracking 방식을 융합한 CDGPS 방식인 RTK GPS는 stand-alone방식의 GPS보다 정확성이 뛰어나다. Base GPS는 노트북과 연결하여 QGroundControl을 통해 ground station 역할을 수행하고 rover GPS는 기체에 장착하여 위치를 특정할 수 있도록 모듈을 구성한다.

(3) 제어 시스템

기체는 위의 2-1-(3)에서 서술하였듯이, 유도 시스템에서 생성된 연속적이고 부드러운 경로를 생성할 수 있는 3차 Bezier curve 경로를 따라 비행해야 한다.

제어 시스템에서는 항법 시스템에서 얻은 RTK Position, FC 내에서 얻은 가속도/각가속도 정보와 합쳐 Path Planning을 통해 생성된 비행 명령을 따라간다. 이 세 가지를 통해 [그림]에 나타난 것과 같이 Motor Speed를 제어할 수 있다. PX4 Autopilot 비행제어 시스템은 위치와 각도에서 상/하한이 제한된 P 제어기를, 속도와 각속도에서 PID 제어기를 사용해 Motor speed를 조절하며, 각각의 이득 값을 기체 비행을 통해 결정해주어야 안정적인 비행이 가능하다. 이때, 각각의 센서에서 생성되는 high frequency noise를 제거하기 위해 low pass filter를 각속도 제어기 미분항에 붙여 제어한다.



[그림 3-1-(3)] FC Block Diagram

3-2. 구성품의 적정성

(1) 기체



WJD 1100HZ	
중량	4.8kg(empty) 10kg (full)
크기	800x800x400mm
배터리	6S 22.2V 16000mAh 30C
프로펠러	D: 22inch, pitch: 7inch
모터	T-Motor P60

대회에 참가할 기체로는 WJD사의 1100HZ를 선정하였다. 먼저, 배달해야 할 물체의 크기가 가로/세로 35cm의 피자박스 형태이기에 이를 무리 없이 하부에 수납할 수 있는 사이즈의 기체를 선정했다. 또한, 모래 환경에서 운용되는 드론이므로 모터와 중심 부분이 외부 이물질로부터 보호될 수 있는 형태의 드론을 선정하려 하였다. 마지막으로, 하부에 여유 공간이 있어 배터리와 다른 기기들을 부착할 수 있는 견고한 판을 부착할 수 있는 드론을 선정하였다. 프레임과 모터, 프로펠러, ESC, 임무장비와 배터리, 피자박

스를 포함한 무게는 약 10kg 내외이기 때문에 모터는 T-Motor사의 P60(340KV)를 선정하고, 프로펠러로는 Dualsky사의 2270 folding propeller를 선정한다. 이는 상기 모터와 프로펠러를 결합하였을 때 50% 출력에서 개당 2.7kg 내외의 추력을 낼 수 있을 뿐만 아니라, 프로펠러에 날개 끝 실속을 방지하기 위해 swept wing 형상의 설계가 적용되어 있으며, 접는 형식으로 제작되어 실험 및 대회 참가를 위한 이동 시에 용이하게 사용할 수 있기 때문이다.

(2) 스테레오 카메라



ZED 2i	
중량	230g
크기	175.25 x 30.25 x 43.10mm
Depth range	0.2 – 20m

전방 카메라로는 스테레오 카메라인 ZED 2i를 사용한다. 야외 환경에서 대회가 진행되어, 실내에서 주로 사용되는 RGBD 카메라의 신뢰성이 떨어질 수 있기 때문에 편광 필터가 장착 가능하고, RGB 카메라의 선예도가 더 높은 스테레오 카메라 모델인 ZED 2i를 선택했다.

(3) LiDAR



LIDAR-Lite V3	
중량	22g
크기	40 x 48 x 20mm
Range	0.05 – 40m
정확도	2.5cm

하방 거리 센서는 대회 평가 요소인 일정 고도의 지속적인 유지를 위해 사용하였다. 먼 거리에서도 정확성을 보장하고, 최대 측정 거리가 15m보다 월등히 길며 현재 FC와 호환되어 쉽게 운용할 수 있는 LIDAR-Lite v3로 결정했다.

(4) 하방 카메라



Arducam IMX 477	
크기	38 x 38 x 31mm
초점거리	3.9mm
FOV	75°
화상도	12.3 MP

하방 카메라는 arducam IMX477 카메라를 사용한다. 75도의 화각을 제공하며, 작은 크기에도 불구하고 높은 화질의 영상을 얻을 수 있기 때문이다. 또한, CSI 커넥터를 이용하기 때문에 Jetson 보드의 입출력 포트에 가하는 부담을 줄이면서 좋은 화질을 유지할 수 있다는 장점이 있다. 이에 따라, 먼 거리와 가까운 거리 모두에서 Apriltag를 추종하여 착륙할 수 있을 것으로 기대한다.

(5) Companion computer



NVIDIA Jetson Xavier NX	
전력	12V, 20W
크기	103 x 90.5 x 34.66mm
중량	172g
GPU	384-core NVIDIA Volta™ GPU with 48 Tensor Cores
CPU	6-core NVIDIA Carmel ARM® 64-bit CPU
모터	8 GB 128-bit LPDDR4x

미션을 진행하는 동안, CC는 영상 Feature들을 통해 ZED 2i에서 Depth를 구하고, 이 정보를 이용해 경로점을 만들어 내서 FC에 비행 경로를 전달하는 동시에 GCS로 처리한 영상을 송신한다. 또한, 하방 카메라를 이용해 마커와의 상대적인 pose를 계산해서 정밀 착륙을 수행한다. 특히 영상의 처리는 많은 연산량을 필요로 하기에, 온보드 GPU를 장착하여 하드웨어 가속을 사용할 수 있으며 파워 소모량이 적은 Jetson Xavier NX를 사용하기로 결정하였다.

(6) GPS



Holybro H-RTK F9P GNSS	
중량	106g
크기	Φ76mm x h 20mm
정확도	1-10cm

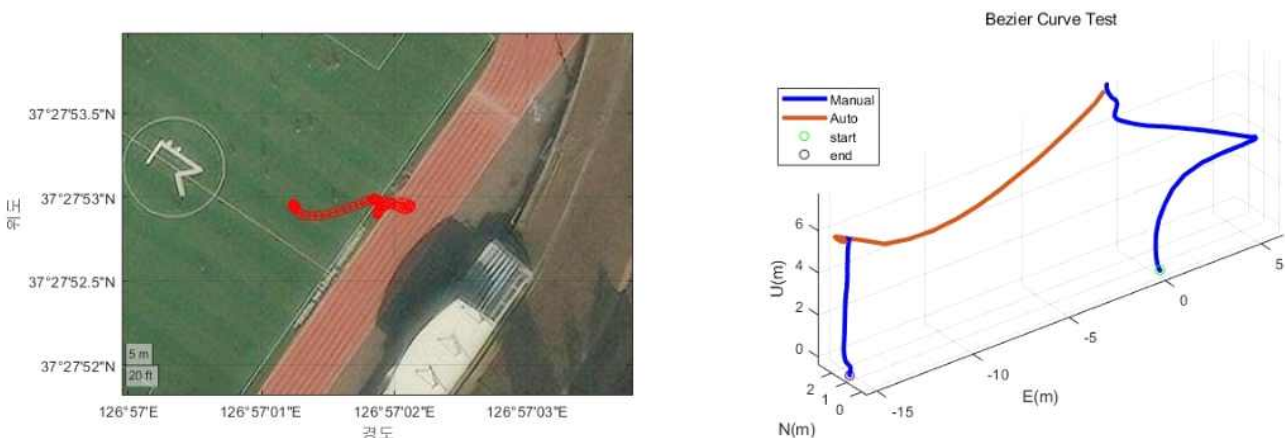
두 개 이상의 gps를 사용하면 redundancy가 증가하고 두개를 융합하여 yaw를 측정할 수 있는 장점이 있지만 IMU와 자기계를 같이 융합하면 하나의 정밀한 GPS만으로 yaw를 측정하여 기체의 flat output을 모두 측정할 수 있기에 FC와 연결이 용이하고 정확도가 높은 Holybro사의 H-RTK F9P GNSS를 선택했다.

3-3. 지상 및 비행 시험 결과

(1) 비행 시험 데이터 분석

비행 데이터는 FC인 Pixhawk 6x에 저장된 비행 log를 GCS 소프트웨어인 QGroundControl에서 ulog 파일로 저장한다. 이 중 vehicle_control_mode와 vehicle_gps_position 두 가지 message type에서 자동, 수동 Flag와 GPS Time, LLA 좌표를 얻는다. 이를 이용하여 Bezier curve로 경로를 생성하여 West 방향으로 15m 이동하는 비행 실험의 데이터를 분석하였다.

실제 RTK GPS를 통한 센서 데이터를 이용하여 Bezier curve를 통해 생성된 경로를 따라 초기 위치를 기준으로 West 방향으로 15m를 성공적으로 이동하는 것을 확인할 수 있다. 이를 통해 항법 시스템의 정밀함과 제어 시스템을 통해 생성된 경로를 드론이 잘 추종함을 확인할 수 있다.



[그림 3-3-(1)] Offboard 비행시험 데이터

(2) Offboard control 시험

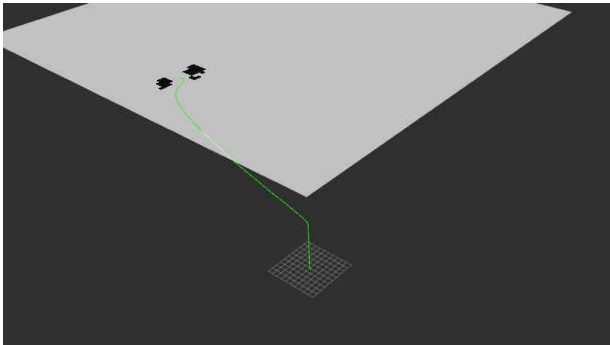
본 미션에서 경로 제어는 ROS2내 px4_msgs의 trajectory_setpoint topic을 이용하여 드론을 제어한다. 이 메시지는 기체의 위치/속도/가속도를 명령하는 데에 사용된다. 이 제어 방법을 시험하기 위해서 키보드를 이용하여 기체의 속도를 제어하는 노드를 제작했다. 실제 기체에 시험하여 기체를 GCS에서 원격으로 조종해본 결과, 속도 명령에 실시간으로 반응하는 것을 확인할 수 있었다.

Bezier control 노드 또한 실제 기체 실험을 진행하였다. 해당 실험에서는 고도 6m에서 서쪽으로 15m 이동하되, 속도 벡터가 북쪽을 가리키는 상태에서 도착하도록 하였다. 목표지 명령은 간단한 ROS2 publisher를 이용해 $\mathbf{p} = [0, -15, 6]$, $\mathbf{v}_f = [0.5, 0, 0]$ 의 목표점을 1Hz의 주기로 전달되었으며, bezier_control 노드가 앞에서 서술했듯이 이를 기

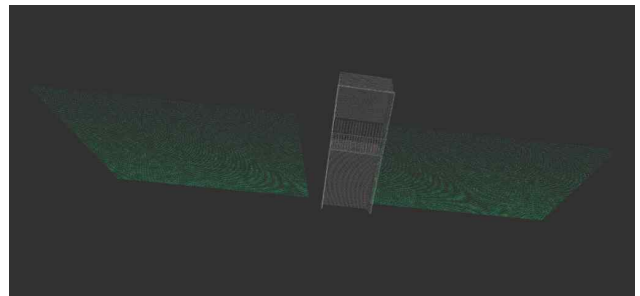
반으로 bezier curve를 생성해 50Hz로 위치 명령을 내렸다. 시험 비행에서의 로그는 [그림 3-3-(1)]과 같다. 이 실험을 통해 Bezier curve가 잘 생성되며 기체가 원활하게 이 경로를 따라갈 수 있었다는 것을 확인할 수 있다.

(3) 장애물 인식 및 회피 시뮬레이션

Pointcloud를 이용한 장애물 인식 및 회피 알고리즘을 검증하기 위해 시뮬레이션을 진행하였다. ZED2i 스테레오 카메라와 같이 최소 depth 0.2m, 최대 depth 20m로 설정한 depth camera 플러그인을 이용하여 스테레오 카메라를 구현한다. SILS 상에서 8m의 거리를 두고 두 사다리차를 위치하고 그 중점을 초기 경로점(WPT)으로 지정한다. ROS2 Nav_msgs의 OccupancyGrid 메시지를 이용하여 그 결과를 Rviz2에 시각화한다. 시뮬



[그림 3-3-(3)-1] 2D 점유 격자 지도와 Bezier curve를 이용하여 생성된 경로



[그림 3-3-(3)-2] Pointcloud를 이용하여 인식된 배란다의 모습

레이션에서는 Grid의 해상도가 1m이고, 가로, 세로 100m인 격자를 이용하였다. 최종적으로 Bezier curve를 이용한 경로 생성 알고리즘과 장애물 인식 알고리즘을 결합하여 경로점(WPT)까지 경로를 생성하였다.

4. 임무수행 구현기술

4-1. 임무수행 시나리오

임무 수행은 다음 6가지 단계로 이루어진다. 더불어 이륙부터 착륙까지 GCS와 통신을 통해 실시간 영상 송수신을 같이 진행한다.

- (1) 목표 고도 15m까지 자동 이륙
- (2) 장애물과 충돌하지 않고, 사다리차 사이에 위치한 경로점(WPT) 통과
- (3) 배송 지점으로 이동하여 마커 인식 및 배송 준비
- (4) 피자 배달 임무 수행

- (5) 사다리차 사이에 위치한 경로점(WPT) 통과하여 이륙 위치로 복귀
- (6) AprilTag를 이용한 정밀 착륙

4-2. 임무수행 방법

(1) 자동 이륙 및 고도 유지

먼저, 기체를 arm 한 뒤 offboard 모드로 전환한다. Offboard 모드에서는 bezier_control ROS2 노드를 이용해 정해진 위치로 이동하고, 자동 이륙의 경우에는 현재 위치에서 수직 위로 이동한다. 주어진 명령에서 고도를 변경하지 않으면 고도는 15m로 자동으로 유지한다.

(2) Bezier curve를 이용한 경로 생성

각 임무를 실행하는 ROS2 노드는 목적지의 위치 및 도착 속도 벡터를 NED로 계산하여 publish한다. Subscribe 받은 목적지의 위치와 도착 속도 벡터를 현재 기체의 위치와 속도를 이용해 4-point bezier curve를 생성한다. 생성된 bezier curve를 적절한 최대 속도, 가속도를 이용해 위치 정보 배열을 생성해 1초 뒤 기체의 desired position을 publish한다.

(3) 스테레오 카메라를 이용한 장애물 인식 및 회피

Pointcloud로 사다리차를 인식하고, 인식된 두 군집의 centroid를 바탕으로 드론이 사다리차 사이를 통과하기 위해 나아가야 할 경로점과 방향을 지정한다. 경로점은 두 centroid의 중점으로, 방향은 두 centroid를 이은 직선에 수직인 방향으로 설정한다. 경로점과 방향은 v_f 의 topic으로 Bezier curve 경로 생성 알고리즘에 전달되어 사다리차와 충돌하지 않도록 한다.

(4) 마커 탐색을 위한 기동

경로점(WPT)의 근방에 도달한 기체는 배송 지점으로 이동한다. 사다리차 사이를 빠져나올 때는 들어올 때와 마찬가지로 두 centroid를 이은 직선과 수직인 방향으로 이동하도록 한다. 배송 지점에 위치한 기체는 pointcloud로 베란다의 군집을 인식하여 centroid를 찾는다. 베란다의 중심 지점 주변을 베란다와 충돌하지 않을 충분히 안전한 거리에서 원형으로 이동하며 십자가 모양의 마커를 탐지할 수 있도록 한다.

(5) 마커 인식 및 정렬

기체는 3층 베란다 벽면에 설치되어 있는 십자가 모양의 마커를 인식하면 해당 마커를 중앙에 놓으려 움직인다. 이때, 기체는 탐지된 십자가 마커의 bounding box 양쪽의 depth 차이를 이용해 벽면이 기체에 대해 기울어져 있는지 감지하고, 벽면에 수직하도록 기체의 위치를 정렬한다.

(6) 피자 배달 임무 수행

기체와 벽면과의 거리를 연장장치를 고려한 길이인 2.3m로 유지한다. 충분히 기체가 안정되면, 임무장비를 연장하여 피자박스를 기체 앞 쪽으로 이동한다. 피자박스가 레일의 끝에 도착하면 bistable hinge가 구부러져 고정장치가 해제되어 피자박스가 베란다 위로 배송된다.

(7) AprilTag를 이용한 정밀 착륙

기체는 사전에 저장된 자신의 이륙 위치로 돌아가면서 하방 카메라로 Apriltag을 지속적으로 탐지한다. 해당 마커가 탐지되면 마커를 기준으로 최종 위치와 속도를 확인하여 안전하고 신속하게 착륙할 수 있도록 경로를 생성한다.

5. 안전장치

5-1. 안전장치의 구성 및 성능

(1) Flight Computer - Companion computer 통신

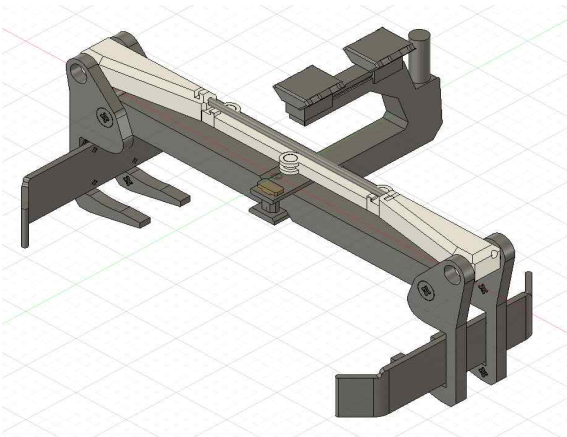
기체는 Offboard 모드를 실행 중 CC로부터 OffboardControlMode 메시지를 2Hz 이상의 주기로 수신해야만 해당 모드를 유지한다. 이 때문에 CC쪽에서 제어 노드가 실행되지 않았거나, FC와 CC 사이의 serial connection에 문제가 발생하여 통신이 두절된다면 기체는 Offboard control이 멈췄다는 것을 탐지하고 자동으로 hold 모드로 전환하여 GCS와 RC 명령을 기다린다.

(2) RC 통신 두절

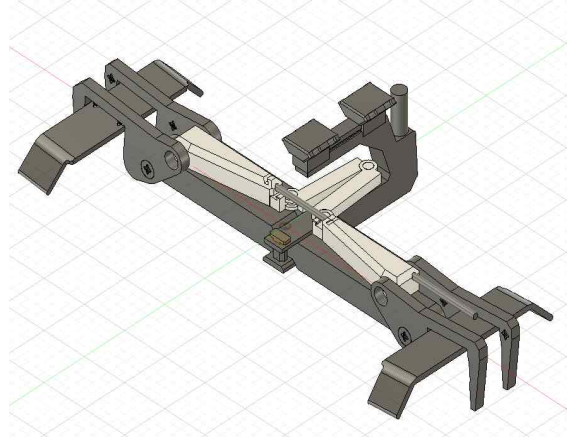
기체는 Offboard 비행 중 발생할 수 있는 불의의 상황에 대응할 수 있도록 RC제어기의 신호를 항상 수신하고 있어야 한다. 따라서 RC 통신이 두절되면 기체는 즉각 임무를 멈추고 hold 모드로 전환하여 해당 위치에 대기하며 RC 신호가 다시 연결되기를 기다린다.

(3) 임무 장비의 안전성

피자 박스를 운반하는 임무장비는 비행 시에 피자박스를 안정적으로 고정할 수 있어야 한다. 피자 박스가 안정적으로 고정되지 않으면 배송 시 진동으로 인해 떨어질 수 있기 때문에 bistable hinge를 이용하여 안정적인 end effector를 설계한다. 임무장치의 팔을 뻗어 드론의 앞부분에서 피자박스를 배송한다는 점을 이용하여 배송장치의 연장이 끝났을 시에만 bistable hinge가 접혀 화물의 고정이 풀리며 배송이 이루어질 수 있도록 한다. 따라서 팔이 끝까지 펴지지 않을 시에는 다른 동력원 없이 물리적으로 안정적인 고정이 가능하여 배송 중에 피자 박스가 떨어지지 않도록 설계하였다.



[그림 5-1-(3)-1] 임무 장비 닫힌 상태



[그림 5-1-(3)-2] 임무 장비 열린 상태



[그림 5-1-(3)-3] 실제 장비 닫힌 상태



[그림 5-1-(3)-4] 실제 장비 열린 상태

5-2. 안전장치의 기술적 적정성

(1) Flight Computer - Companion Computer 통신

Companion computer와 통신이 두절된다면 더 이상의 임무 수행이 불가능하다. 비행하는 장소에는 사다리차등 여러가지 장애물이 있기 때문에 companion computer의 제어가 없을 시에는 기체가 스스로 움직이는 것은 위험하다. 따라서 hold 모드로 전환을 하는 것으로 안전 장치를 구현한다.

(2) RC 통신 투절

직접 임무 수행에 사용되지는 않지만, 안전한 비행을 위해서는 끊임없이 숙련된 조종사가 모니터링할 필요가 있다. 이때 필요한 통신이 기체와 RC 컨트롤러 사이의 연결과 기체와 ground station 사이의 연결이다. (1)에서와 마찬가지로 비행 장소에 여러 장애물이 존재하기 때문에 수동으로 조종할 수 없는 상태에서 함부로 움직이는 것은 위험하다. 따라서 마찬가지로 hold 모드로 전환하는 것으로 안전 장치를 구현한다.

(3) 임무 장비의 안전성

배송지를 향해 비행하는 도중 피자 박스가 임무 장비에 안정적으로 고정되지 않아 피자 박스가 떨어지면 더이상 임무를 수행하는 의미가 없으므로 임무 장비의 안정적인 고정은 필수적이다. 배송지에 도착한 경우에만 팔을 뻗는다는 사실을 이용하여 팔이 접혀 있는 경우에 물리적으로 고정이 풀리지 않는 안전 장치를 구현한다.

5-3. 안전장치의 시뮬레이션 또는 실제 작동

발표 시 다른 영상 자료들과 함께 시연 예정

6. 시스템 설계 및 제작상의 특징점

(1) CC-GCS 통신

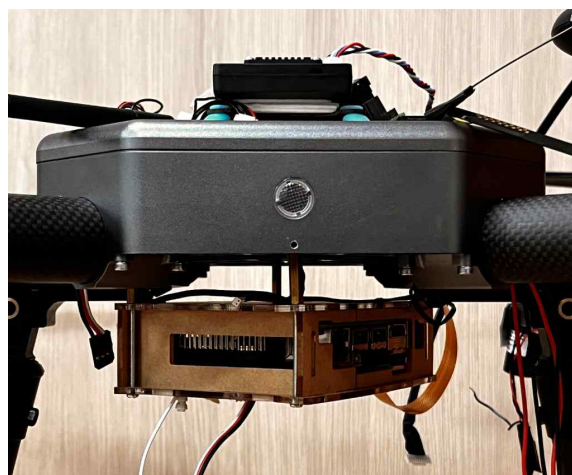
드론에 장착된 상태에서는 CC인 Jetson Xavier NX를 제어할 수 없으므로, GCS에서 원격 제어를 통해 CC를 제어해야 한다. 따라서 Ubuntu에서 기본으로 지원하는 원격 프로그램인 SSH를 사용하여 GCS와 CC 간의 제어 명령을 전송한다. 또한, CC는 드론과 함께 멀리 날아가기 때문에 GCS와 연결할 때 거리의 한계를 극복할 수 있는 네트워크

체계를 고려해야 한다. 따라서, 원거리에서도 상용망을 이용하여 CC에 안정적인 네트워크를 공급할 수 있는 단말기를 부착해 상용망이 있는 곳에서는 어디서든 통신이 이루어질 수 있도록 하였다.

GCS에서 CC의 SSH에 접속하기 위해서는 CC와 GCS가 같은 네트워크에 있어 내부 IP를 통해 통신할 수 있어야 한다. 이때, CC가 GCS에서 물리적으로 멀리 떨어지면 두 개체는 같은 Wi-Fi 네트워크에 연결할 수 없는 상태가 된다. 이 때문에 CC는 상용망 단말기를 통해 네트워크에 연결되는데, 이 때에도 내부망을 통해 통신하는 것과 같은 효과를 내기 위해 VPN을 이용하여 두 개체를 하나의 네트워크로 묶어 제어 명령과 영상을 전송한다.

(2) 기체의 적층적 구조

자동비행을 위한 기체 전자부의 구성으로는 크게 드론의 모터, ESC 및 PDB, Pixhawk 6x(FC), Jetson Xavier NX가 있다. PDB는 외부로 노출되지 않는 것이 좋기 때문에 메인 프레임 안에 위치하고 Pixhawk 6x는 비행 실험시 배선의 교체가 용이하고 진동을 제거하기 위한 damper를 설치할 수 있어야 하기 때문에 메인 프레임의 상단에 위치한다. Jetson Xavier NX는 임무장비의 구동과 stereo camera의 통신을 할 수 있어야 하기 때문에 메인 프레임의 하단에 위치한다. 이 구조는 배선을 최대한 깔끔하게 할 수 있고 구성품 중 오염과 충격에 가장 취약하며, 수리 부품을 구하기 어려운 Jetson Xavier NX를 기체의 중심부에 위치시켜 기체에 파손이 생기더라도 Jetson Xavier NX에 데미지가 생기지 않도록 할 수 있다.



[그림 6-(2)] 기체 적층적 구조

(3) 임무장비의 신뢰성 확보

이번 미션에 사용되는 임무장비는 구조를 간단화하여 모터 하나로 이루어진 형태를 취하고 있다. 이런 간단한 구조는 여러 개의 서보모터를 사용하는 것보다 배선을 하기에 간편하고, 신뢰성 있는 모터를 사용했을 시 고장의 염려가 작다는 장점이 있다. 또한, 작동 시를 제외한 모든 상황에서 기체 중심부에 위치해 비행중 CG가 기체 내부에 위치하여 안정적인 제어를 달성할 수 있다.

7. 개발 애로사항

첨부. 비행시험 데이터 & 비행시험 동영상