PX4 Autopilot 을 활용한 안티드론 개발

2017103989 박하늘
2019110627 김명주

요 약

최근 드론의 급격한 확산에 따라 부작용으로 드론 범죄 또한 지속적으로 증가하고 있는 추세이다. 따라서 미확인 드론을 탐지하고, 추적해 무력화하는 안티 드론 산업의 필요성이 증가하고 있다. 본 프로젝트에서는 PX4 Autopilot open source 와 SITL simulator 를 활용하여 드론의 탐지, 식별 뿐만 아니라 Soft Kill 과 Hard Kill 방식을 사용하여 무력화하는 안티 드론을 개발하고자 한다.

1. 서론

군사적 훈련을 목적으로 개발 되었던 드론이 이제는 다양한 민간분야에서 활용되고 있으며, 정부도 '무인이동체 기술혁신과 성장 10 개년 로드맵', '드론 활성화 지원 로드맵', '드론산업 발전 기본계획(`17~`26)'을 수립하여 추진하는 등 드론 산업 육성을 위하여 각종 지원을 아끼지 않고 있다. 단순한 취미나 레저 활동을 넘어서 방송, 공공행정 분야까지 여러 방면에서 활용이 되고 있으며, 아마존이나 보잉과 같은 글로벌 기업은 다양한 영역에 드론을 활용하기 위해 첨단 기술을 동원하여 드론 산업을 육성하고 있다.

그러나 최근 드론의 급격한 확산에 따라 무단비행, 사생활 침해 등 불법적 활용을 통한 드론 범죄가 지속해서 증가하고 있다. 드론의 확산에 따라 미확인 드론을 탐지하고, 무력화하는 안티드론(Anti Drone) 산업의 필요성이 증가하고 있다. 따라서 실제 드론을 탐지 및 식별하고 무력화시키는상황을 소프트웨어 환경에서 시뮬레이션해봄으로써 실제 상황에 대비할 수 있도록 한다. 딥러닝을 통해 드론 이미지를 학습하고 학습된 모델을 통하여 드론을 실제로 식별하는 것을 시연하고 다음으로 드론의 GPS 정보를 임의로 조작한 GPS 로 덮어씌워 드론을 원하는 위치에 착륙시킨다.

2. 관련연구

2.1 탐지 및 식별

미확인 드론을 탐지 및 식별하기 위해서는 다음과 같은 센서를 사용할 수 있다.

기술 분류	설명
음향 탐지 센서	드론이 동작할 때 프로펠러의 회전으로 인해 발생하는 특유의 소음을 탐지하는 기술로, 소음이 많은 환경에서는 탐지하기 어렵다는 단점이 있다. 그러나 가격이 싸다는 장점이 있다.
방향 탐지 센서	무인 비행체의 조종 신호는 Industrial Scientific and Medical(ISM) 대역인 2.4GHz 대역(제어 신호 송수신용)과 5.8GHz 대역(영상데이터 송수신용)을 주로 사용하고 있다. 이 대역의 RF(Radio Frequency) 신호의 방향과 위치를 방향 탐지 센서를 이용해 조종자 및 무인 비행체의 위치 추정이 가능하다. 단, WiFi 주파수와 같으므로 WiFi 가 많이 설치된 도심에서는 조종 신호와 구분하기 어려운 것이 단점이지만, 다른 센서와 달리 조종자의 위치까지도 추정할 수 있다는 장점이 있다.
영상 센서	가시광선 영역과 적외선 열화상 영역의 영상정보를 활용하여 움직이는 무인 비행체를 탐지하는 기술이다. 위협체의 형상을 운용자가 직접 확인할 수 있으므로, 접근 중인 위협체가 무인 비행체인지 아닌지 식별하기 위한 수단으로 활용되고 있다. 낮과 밤에 모두 활용하기 위하여 Electro Optic(EO) 장비와 Infrared(IR) 장비를 동시에 운용한다. 열화상 센서 관련 광학계의 제작비용이 매우 비싸다는 단점이 있다.
Lidar 센서	위에서 언급된 센서들과는 달리 스스로 에너지를 방사하는 센서(Active Sensor)로 특정 대역의 RF 신호를 송출하고 표적으로부터 반사되어 돌아오는 신호를 수신하여 표적을 탐지한다. 날씨, 온도, 낮/밤 등에 무관하게 안정적인 탐지 성능을 보장하는 전천후 센서로 최대 탐지 거리가 다른 센서와 비교하여 길다는 장점이 있다. 다른 센서에 비해 제작, 구매, 도입 비용이 매우 높은 편이고 스스로 신호를 송출하기 때문에 간섭 문제를 고려해야 한다.
Rader 센서	Rader 센서는 특정주파수의 전파를 송신한 후 해당 신호의 반사파를 감지하는 방식의 도플러효과를 적용하여 거리에 따른 주파수 변화를 검출하여 감지신호를 출력하는 것이다.. 따라서 주파수의 변화와 변화상태를 분석하여 움직이는 물체가 있는지, 물체가 어느 방향으로 움직이는지, 얼마의 속도로 움직이는지 등을 분석할 수 있다. 라디오 주파수를 사용하기 때문에 벽을 뚫고 송수신이 가능하며 넓은 지역으로 전파를 보낼 수도 있다. 하지만 적외선센서나 IR 센서 등에 비하여 비용이 많이 들고, PCB 회로를 따로 구현해야 하므로 온도습도등에 영향을 많이 받고 여유공간이 많이 필요하다는 단점이 있다.

2.2 무력화

2.2.1 Hard Kill

Hard Kill 이란 물리적으로 드론을 무력화하는 방식이다.

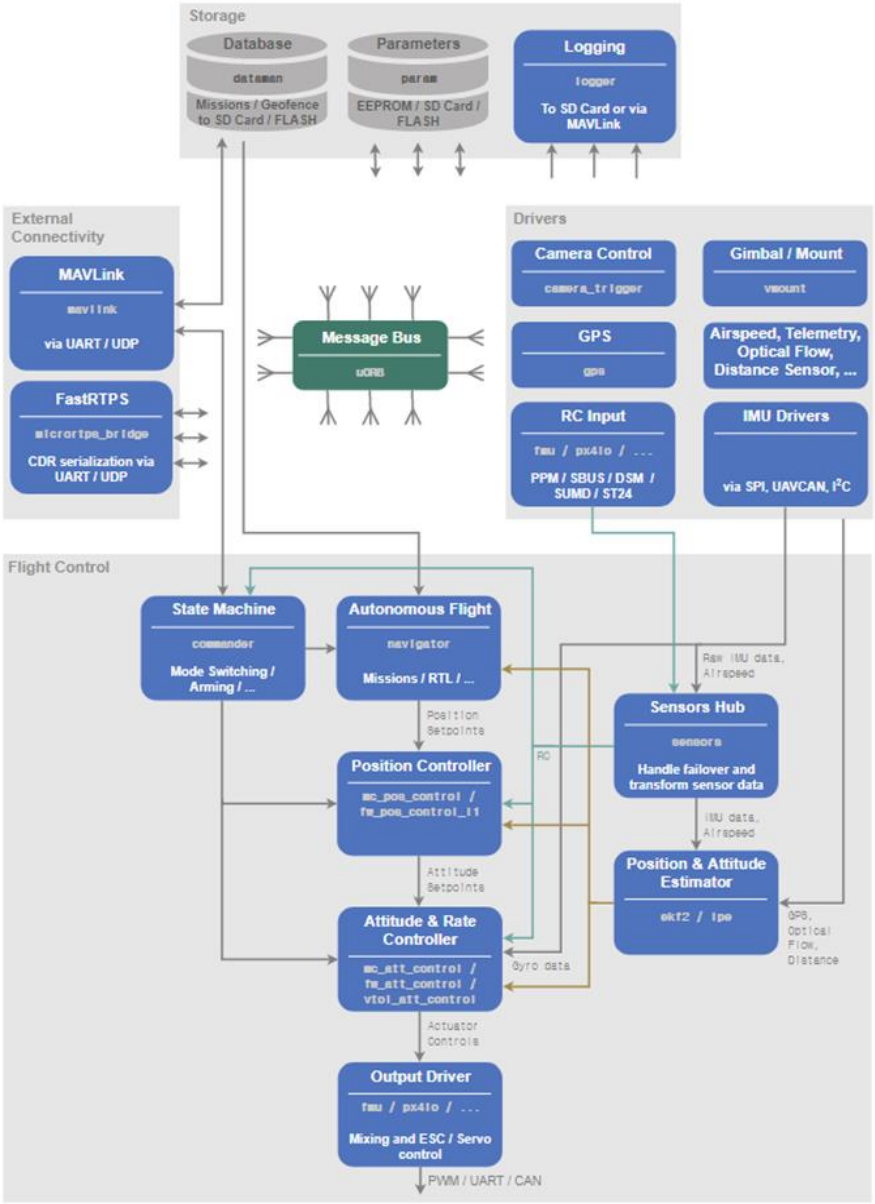
방법	설명
그물	무인 비행체를 탐지했다면 그물을 사용하여 포획하는 방법이다. 그물을 사용하여 드론은 프로펠러를 멈추어 비행이 불가능하도록 하는 방법으로 가장 간단한 방법이다.
레이저빔	레이저빔은 불법 드론을 감지하고 추적하는 데 효과적이다. 그런데 이 기술은 연구개발에서도 적잖은 비용이 들 뿐만 아니라 하늘의 점처럼 보이는 물체를 조준해서 맞추는 데 상당한 노하우가 필요하기에 여전히 진화를 거듭하는 중이다. 또한, 타격된 드론이 불타면서 수직 낙하할 경우 인력이나 시설에 2 차 피해를 발생시키고 불발될 경우에도 원치 않는 피해가 발생하기에 상당한 주의가 필요한 방법이다.

2.2.2 Soft Kill

Soft Kill 이란 전파, GPS 등을 활용하여 드론을 무력화하는 방식이다.

방법	설명
Jamming	jamming 은 전파 교란을 말하는데, 드론의 라디오 통신이나 GPS 항행에 혼선을 주는 것이다. 제조사나 제품의 사양에 따라 차이가 있지만, 일반적으로 드론은 조종자와의 통신이 끊기면 이륙한 곳으로 돌아가거나 통신을 회복할 때까지 제자리 비행 혹은 제자리 착륙을 하도록 프로그래밍 되어 있다. 따라서 전파 교란은 가장 저렴하고도 효율적인 안티드론 기술로 꼽힌다.
Geo-fencing	지리(Geography)와 울타리(Fence)를 결합한 단어로, 실제 위치에 기반하여 가상의 경계나 구역을 만드는 기술이다. 무인 비행체가 사용자가 지정한 울타리 내 출입을 불가능하게 만들거나 출입 현황을 알려줄 수 있다.
GPS Spoofing	공격자가 공격대상의 GPS 신호를 임의로 조작하여 원래 가고자 하는 목적지가 아닌, 공격자가 원하는 목적지로 가게 하는 것을 목표로 한다. 또한, 완전히 탈취하여 온전히 공격자의 것으로 만드는 Hijacking 방법으로 응용해서 사용할 수 있다.

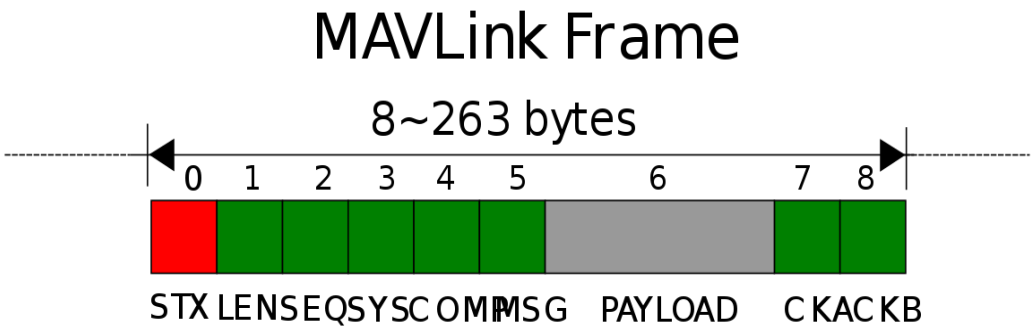
2.3 PX4



[그림 1] PX4 Building Block (Architecture)

PX4 는 무인항공기의 원격조정 및 자율주행을 위해 설계된 소프트웨어로 NuttX 운영체제와 ROS 를 기반으로 동작한다. NuttX 는 임베디드 시스템의 플랫폼 중 하나로 RTOS(Real Time Operating System)로 개발되었다. 또한, PX4 프로젝트는 QGroundControl 이라는 지상관제 프로그램과 드론키트라는 원격조정 어플리케이션의 오픈소스를 제공하고 있다.

2.4 MAVLink



필드 명(Field name)	고유 순서(Index) - 바이트(Bytes)	목적 및 기능
헤더1(Start-of-frame)	0	시작점을 가리키는 헤더(0xFE)
헤더2(Payload-length)	1	페이로드 길이값 (n)
헤더3(Packet sequence)	2	총 패킷에서 해당하는 순서값
헤더4(System ID)	3	발신자 시스템의 고유 ID
헤더5(Component ID)	4	해당 컴포넌트 고유 ID
헤더6(Message ID)	5	페이로드(payload) 정의 ID
데이타(Payload)	6 (n+6)	메세지 ID에서 참조되는 실질적인 데이타 값들 (없을수있다)
체크섬1 CRC	(n+7)	메세지 무결성 검사
체크섬2 CRC	(n+8)	네트워크 무결성 검사

[그림 2] MAVLink Frame 구성

MAVLink 는 Micro Air Vehicle Link 로 소형 무인 장치들 및 자체내의 서로 다른 내부 컴포넌트과 통신하기 위한 프로토콜로 data 교환을 제공하는 메시징이다. 소형 장치들과 통신하는 특성상 제한된 RAM 과 플래쉬 메모리의 리소스 제약이 있는 시스템에 맞춰 연산을 최소화하기 위해 암호화를 진행하지 않는다.

2.5 QGroundControl

QGroundControl 은 MAVLink 프로토콜을 지원하는 드론의 GCS(Ground Control System)이다. GCS 는 지상 제어 시스템으로 지상에서 사용자가 드론의 비행 상태를 확인하고 제어할 수 있는 환경을 제공하는 관제 시스템이다. 이외에도 QGroundControl 은 PX4 의 펌웨어 업로드, Planning, Flying Missions 기능을 통한 자동 비행, 드론에 장착된 카메라의 실시간 스트리밍 등의 기능을 사용자에게 제공한다.

2.6 ROS

ROS(Robot Operating System)는 오픈소스 기반 로봇 운영체제로 로봇 응용 프로그램을 개발할 때 필요한 하드웨어 추상화, 디바이스 제어, 프로세스 간의 메시지 패싱, 개발환경에 필요한 라이브러리와 다양한 개발 및 디버깅 도구를 제공한다.

2.7 Gazebo

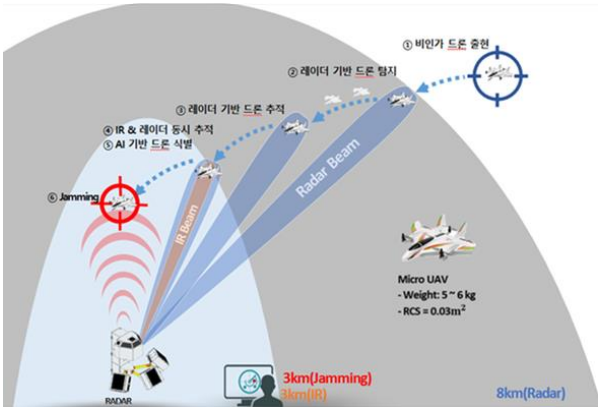
Gazebo 는 MAVLink 프로토콜을 지원하는 멀티로터 유형의 가상 드론 시뮬레이터이다. 사용자는 드론 기기를 사용하지 않고 작성한 소프트웨어를 안전하게 테스트 및 디버깅할 수 있다. 또한, 다양한 물리엔진과 센서, 높은 자유도를 제공하며 가상의 환경(World)을 직접 설계할 수 있다. Gazebo 에서 다양한 물건들을 배치하여 현실감 있는 드론 환경을 구축할 수 있다.

2.8 YOLO

객체 탐지란 미디어 내 대상이 되는 객체를 ROI(Region Of Interest)로 검출하고 올바른 클래스로분류하는 것을 말한다. 널리 알려진 객체 탐지 기법의 하나인 YOLO 는 one-shot detection 방식으로 정확도는 상대적으로 two-shot detection 에 비해 낮은 모습을 보이지만 실시간으로 객체 검출을 주 목적으로 하는 무인항공기 탐지에는 YOLO 가 적합하다는 것을 알 수 있다.

2.9 안티드론 활용 사례

2.9.1 국방



[그림 3] 국방 안티드론 솔루션

레이더 연동 안티드론 통합솔루션은 초소형 드론을 탐지, 식별, 전파교란 단계를 거쳐 작동 불능 상태로 만들어 미상의 드론이 군 중요시설에 접근하는 것을 차단할 수 있다.

2019 년 사우디 정유시설이 드론의 공격을 받은 것처럼 상용 드론을 군사용으로 개조해 무기로 활용하는 사례가 빈발하고 있으며, 최근 미군이 주둔하는 이라크 기지에 무인기 공격이 잇따르는 등

드론의 군사적 위협이 증가하고 있다. '레이더 연동 안티드론 통합솔루션'은 순수 국내 기술로 개발한 드론 대응체계이로 레이더는 레이더 반사면적이 0.01 ㎡ 크기의 초소형 드론을 8km 밖에서 탐지했다.

2.9.2 평창올림픽

[그림 4] 평창올림픽 '킬러 드론'

평창 올림픽 조직위는 올림픽의 새 위협으로 꼽히는 '드론 테러'를 차단할 3 중 대비책도 마련했다. 미확인 드론이 나타나면 전파 차단 기술로 무력화를 시도하고, 전문 요원이 드론에 산탄총을 쏴 격추를 노린다. 뿐만 아니라 안티 드론(킬러 드론)은 품고 있던 그물을 날려 수상한 드론을 포획하는 Hard Kill 방식을 채택하여 미확인 드론을 제압한다.

2.9.3 개트윅 공항 폐쇄 사태

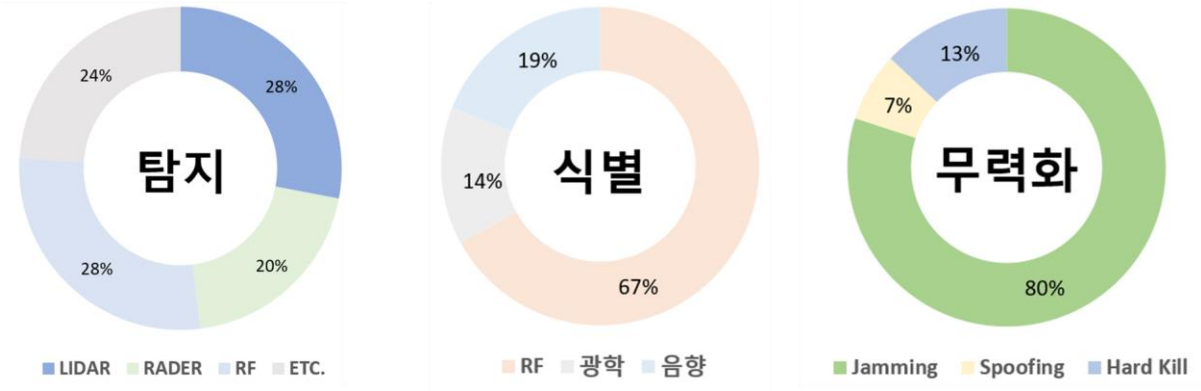
[그림 5] Gatwick Airport 폐쇄 사태

2018 년 12 월 런던 개트윅 공항(Gatwick Airport) 폐쇄 사태는 안티 드론 시스템의 필요성을 깨닫게 해주는 사례이다.. 미확인 소형 드론이 개트윅 공항으로 침입하면서 공항에 이착륙하는

비행기가 회항 또는 다른 공항으로 대피하는 소동이 벌어졌다. 미확인 드론이 활주로 인근에 출현하는 바람에 항공기 700 여 편이 36 시간 동안 운항에 차질을 맺었고, 12 만 명 승객의 발이 묶였다. 영국 국방부는 개트윅 공항 폐쇄 사태 이후 이스라엘 보안 기업 라파엘(Rafael Advanced Defense Systems)이 개발한 드론 방어 시스템 '드론 돔(Drone Dome)'을 공항 옥상에 배치했다. 드론 돔은 4 개의 레이더를 사용해 시설 주변 지역의 드론 비행을 감시한다. 레이더 탐지 거리는 16km 이며, 3.2km 떨어진 곳에서 최소 0.002 ㎡ 크기의 표적까지 탐지할 수 있다. 이뿐만 아니라 열화상 카메라, 드론과 조종사의 위치를 찾는 추적기, 방해 전파를 쏘아 드론을 격추할 수 있는 기능까지 갖췄다.

2.10 안티드론 기술 사례

사례	탐지 기술	무력화 기술	특징
HOGREEN AIR ANTI DRONE	LIDAR 360 RADER 소음 감지 센서 SDR 수신기	JAMMING HARD KILL	AI, 비전 사용, RF JAMMING
ADRIAN - ANTI DRONE	LIDAR 음향 센서 ESM (Electronic Support Measures)	GNSS SPOOFING JAMMING	GNSS(Global Navigation Satellite System)Spoofing 은 드론을 안전한 지역에 착륙시키거나 출발점으로 되돌리도록 할 수 있다.
NQDEFENSE ANTI DRONE	RADER	JAMMING	1. 6 frequency bands of jamming signals 2. 3D active phased-array rader 3. (ND-Bu002 는 9 frequency bands of jamming signals)
NQDEFENSE2 ANTI DRONE	RF 센서	JAMMING	6 frequency bands of jamming signals, directional jammer
Mistral AI ANTI DRONE	LIDAR 3D RADER RF 센서 음향 센서	GPS SPOOFING	AI 알고리즘을 통해 새, 구름, 기타 비행물체와 구분할 뿐만 아니라 드론의 제조사와 모델을 분류



[그림 6] 기술 사례 조사 결과

위의 표 외에도 30 개의 기업을 조사했고 35 개의 기업의 통계 결과이다. 탐지는 라이더와 RF 센서를 가장 많이 사용하고 있고 RF 와 광학 음향 순으로 많이 식별하는 것을 확인할 수 있었다. 무력화의 경우 구축하기 용이한 Jamming 방식이 가장 많았고 그 다음으로 Hard Kill, Spoofing 을 사용하는 것을 확인할 수 있다.

3. 시나리오

3.2 요구사항

3.2.1 탐지

- 방대한 탐지 범위를 가지고 있다면 너무 많은 탐지가 이루어지므로 탐지 범위를 적절하게 조절해야 한다.

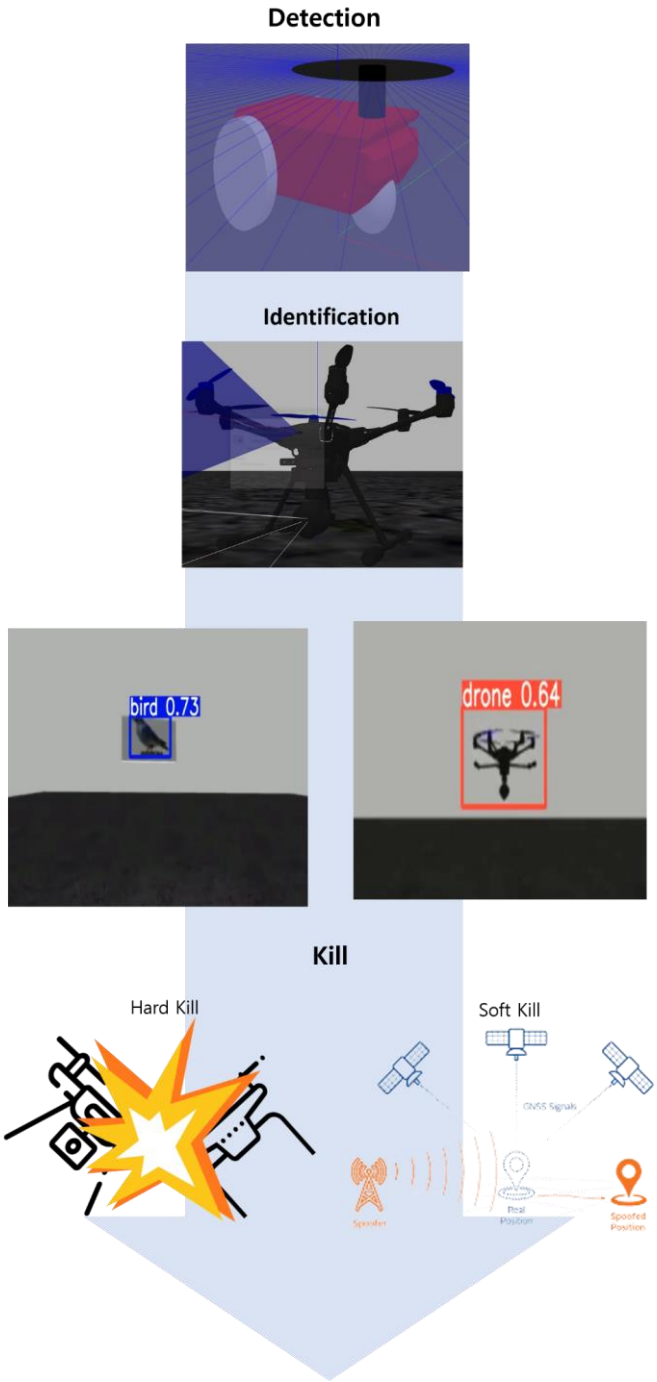
3.2.2 식별

- 무인비행체라고 판단되면 바로 무력화를 진행하기 때문에 정확한 드론 식별이 요구된다.
- 어두운 곳에서도 드론을 잘 식별할 수 있어야 한다.
- 식별을 통해 장애물을 회피하면서 이동해야 한다.

3.2.3 무력화

- 바람의 세기, 강수 등을 고려하여 Hard Kill 을 수행해야 한다. (정확한 무력화를 위해서)
- Soft Kill 은 우리 진영으로 미확인 드론를 hijacking 하는 시나리오, 바위 등 장애물에 추락시켜 파괴하는 시나리오 등을 다양하게 구상해본다.

3.2 시나리오



[그림 7] 안티 드론 flow

[그림 7]은 우리가 생각한 안티 드론의 시나리오이다. 가장 먼저 LIDAR 센서를 활용하여 드론을 탐지한다. 탐지에서 라이더 센서를 채택한 이유는 넓은 탐지범위와 빠른 탐지 속도를 가지고 있고 SITL 환경에서 직관적으로 표현할 수 있기 때문이다.

탐지 후 관제시스템에서 해당위치로 드론을 보낸다. 그리고 deep learning 을 활용하여 물체를 식별한다. 이후 미확인 드론을 판단하면 무력화를 진행한다. 무력화는 크게 Hard Kill 과 Soft Kill 이

있는데, Hard Kill 은 충돌에 의한 무력화 방법을 선택했고, Soft Kill 은 GPS Spoofing 방법을 선택했다.

4. 설계 및 구현

4.1 탐지(detection)

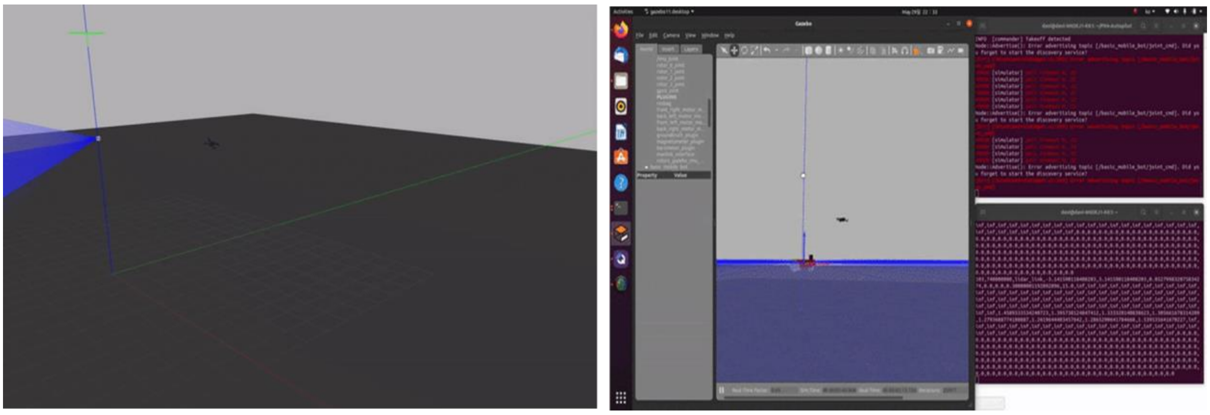
LIDAR 탐지는 직진성을 가진 광자로 물체를 탐지하는 기술로 정밀성이 매우 뛰어나다. LIDAR 센서는 물체의 위치를 탐지하 는데 오차범위가 mm 으로 작은 물체를 정확히 탐지할 수 있는 장점이 있다. 그러므로 비교적 크기가 작은 드론을 정확하게 탐지하는데 적합한 센서이다. RF 센서나 다른 센서들보다 가격 이 비싸고 발열을 잡기 힘든 단점이 있지만, Simulator 상에서는 해당 사항이 없다.

```
47    <sensor type="ray" name="sensor">
48
49      <!-- Position the ray sensor based on the specification. Also rotate
50           it by 90 degrees around the X-axis so that the <horizontal> rays
51           become vertical -->
52      <pose>0 0 -0.004645 1.5707 0 0</pose>
53
54      <!-- Enable visualization to see the rays in the GUI -->
55      <visualize>true</visualize>
56
57      <!-- Set the update rate of the sensor -->
58      <update_rate>30</update_rate>
59      <ray>
60        <noise>
61          <!-- Use gaussian noise -->
62          <type>gaussian</type>
63          <mean>0.0</mean>
64          <stddev>0.02</stddev>
65        </noise>
66        <!-- The scan element contains the horizontal and vertical beams.
67             We are leaving out the vertical beams for this tutorial. -->
68        <scan>
69
70          <!-- The horizontal beams -->
71          <horizontal>
72            <!-- The velodyne has 32 beams(samples) -->
73            <samples>32</samples>
74
75            <!-- Resolution is multiplied by samples to determine number of
76                 simulated beams vs interpolated beams. See:
77                 http://sdformat.org/spec?ver=1.6&elem=sensor#horizontal_resoluti
78                 -->
79            <resolution>1</resolution>
80
81            <!-- Minimum angle in radians -->
82            <min_angle>-0.53529248</min_angle>
83
84            <!-- Maximum angle in radians -->
85            <max_angle>0.18622663</max_angle>
86          </horizontal>
87        </scan>
88
89        <!-- Range defines characteristics of an individual beam -->
90        <range>
91
92          <!-- Minimum distance of the beam -->
93          <min>0.05</min>
94
95          <!-- Maximum distance of the beam -->
96          <max>70</max>
97
98          <!-- Linear resolution of the beam -->
99          <resolution>0.02</resolution>
100       </range>
101   </ray>
```

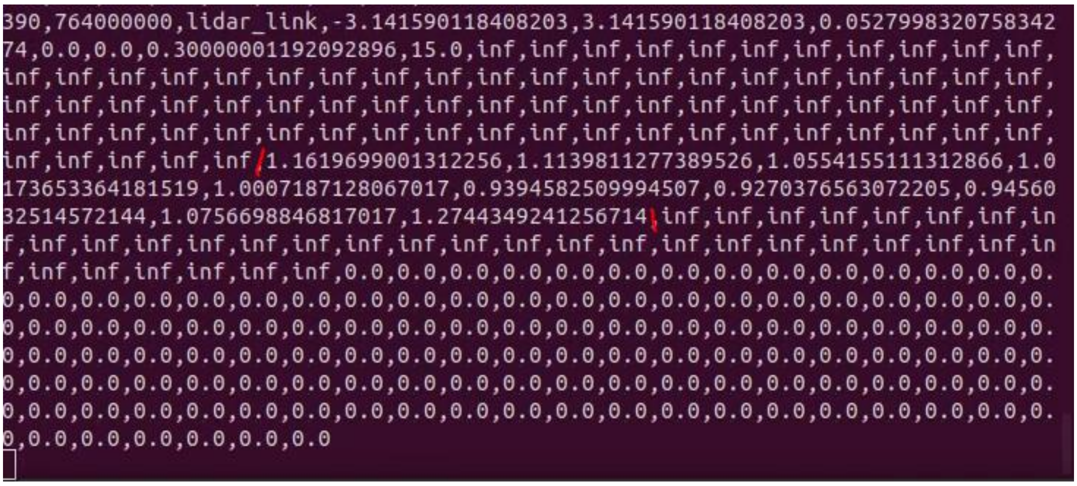
[그림 8] LIDRA.sdf

[그림 8]은 직접 제작한 LIDAR 센서 모델의 코드 일부이다. Gazebo 에서 제공하는 ray 모듈을 사용하여 ray 센서를 구축하 고 해당 센서에 ROS libgazebo_ros_ray_sensor 플러그인을 추가 하여 탐지된 물체의 정보를 전달받을 수 있도록 구현하였다. 해당 모델은 실제 LIDAR 를 제작하는 회사인 'Velodyne Lidar' 사의 모델을 모방하여 제작하였다. [그림 9]은 제작한 모델을 사용하여 Simulator

상에서 드론을 탐지하는 사진으로 왼쪽은 360 도를 회전하면서 탐지하는 LIDAR 모델, 오른쪽은 수직으로 이동하며 360 도 탐색하는 LIDAR 모델이다.



[그림 9] 미확인 물체 탐지



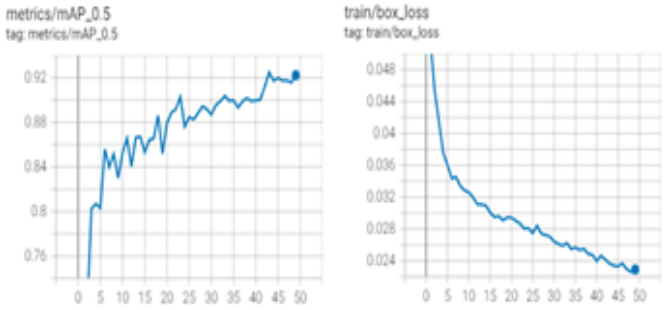
390,764000000,lidar_link,-3.141590118408203,3.141590118408203,0.0527998320758342
74,0.0,0.0,0.30000001192092896,15.0,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,
inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,
inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,
inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,
inf,inf,inf,inf,inf,1.1619699001312256,1.1139811277389526,1.0554155111312866,1.0
173653364181519,1.0007187128067017,0.9394582509994507,0.9270376563072205,0.94560
32514572144,1.0756698846817017,1.2744349241256714,inf,inf,inf,inf,inf,inf,inf,in
f,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,inf,in
f,inf,inf,inf,inf,inf,inf,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0

[그림 10] 탐지 후 결과

[그림 10]은 [그림 9]에서 탐지한 후의 terminal 출력을 나타낸 것으로 아무 물체도 탐지되지 않았다면 infinity 의 약자인 inf 를 나타낸다. 그 후 물체를 탐지했다면 빨간 괄호에 나와 있듯이 실수로 값을 반환한다. 이 실수는 탐지 모델로부터 거리를 의미한다. 해당 실수 값과 LIDAR 모델의 각도 값을 사용하여 물체의 위치를 알아낼 수 있다.

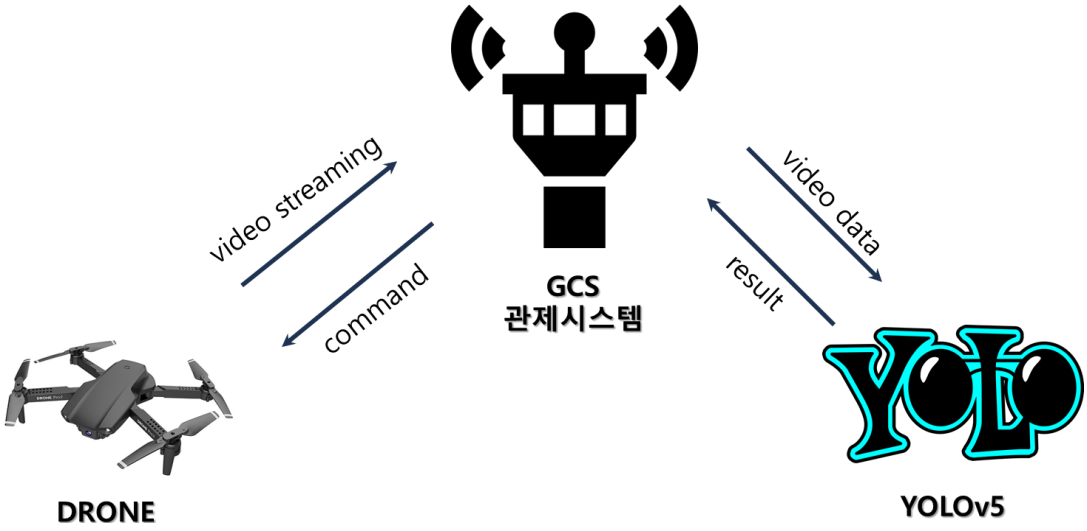
4.2 식별(identification)

[그림 11] 이미지 보정



[그림 12] 학습 결과

다양한 비행 물체 중 드론을 정확하게 식별하기 위해 인공지능을 사용하고자 한다. 'kaggle'을 통해 총 3,600 개의 드론 이미지를 수집했고 [그림 11]과 같이 어두운 이미지의 경우 밝기 조절과 외곽선을 강조하는 보정 작업을 수 행했다. YOLO v5 로 학습을 진행하기 위해 수집한 이미지의 90%는 Train Set 으로 10%는 Validation Set 으로 구분했다. 그리고 설정 가능한 image size, batch size, epochs 각각을 416, 16, 50 으로 지정하여 이미지 학습을 진행하였으며, 그 결과는 [그림 12]와 같다



[그림 13] 식별 flow

[그림 13]은 식별 과정으로 드론의 비디오 스트리밍 데이터를 실시간으로 관제시스템으로 전달한다. 이후 GCS 에서 YOLOv5 를 통해 식별을 진행한다. 그리고 식별 결과를 역순으로 제공한다.

[그림 14] 식별 결과

[그림 14]는 탐지 후 식별까지의 결과로 물체를 탐지하고 관제시스템에서 해당 위치로 드론을 보낸다. [그림 14][1]는 식별 결과 드론이 아닌 새라는 것을 성공적으로 파악했다.

[그림 15] 경계 구역 지정

이미지 학습이 끝났으므로 미확인 드론이 식별 가능하다. 그래서 [그림 15]와 같은 새로운 시나리오를 구상해볼 수 있다. Gazebo 환경에서 카메라가 장착된 안티드론(typhoon h480)을 생성했고 경계 구역을 설정했다. OGroundControl에서 안티드론은 경계 구역을 따라 순찰 하도록 경로를 지정했고, 무인 비행체는 경계 구역의 근처로 비행하도록 경로를 설정했다.

---

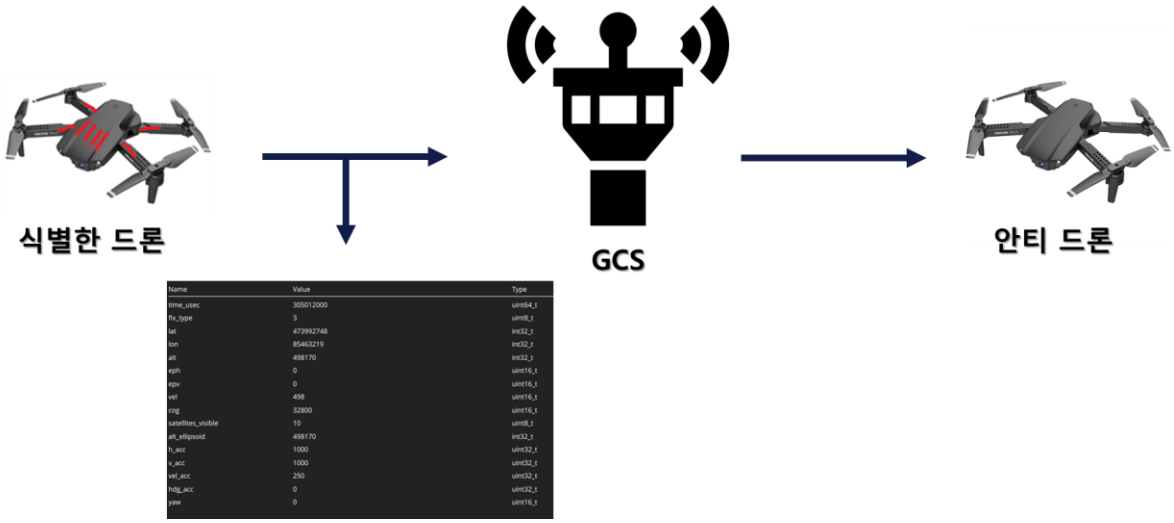
[1] https://youtu.be/wt09c1zPeDw

[그림 16] 실험결과

[그림 15][2]의 결과를 바탕으로 경계 구역 내에서 무인 비행체를 성공적으로 식별하는 것을 확인할 수 있다.

4.3 무력화(kill)

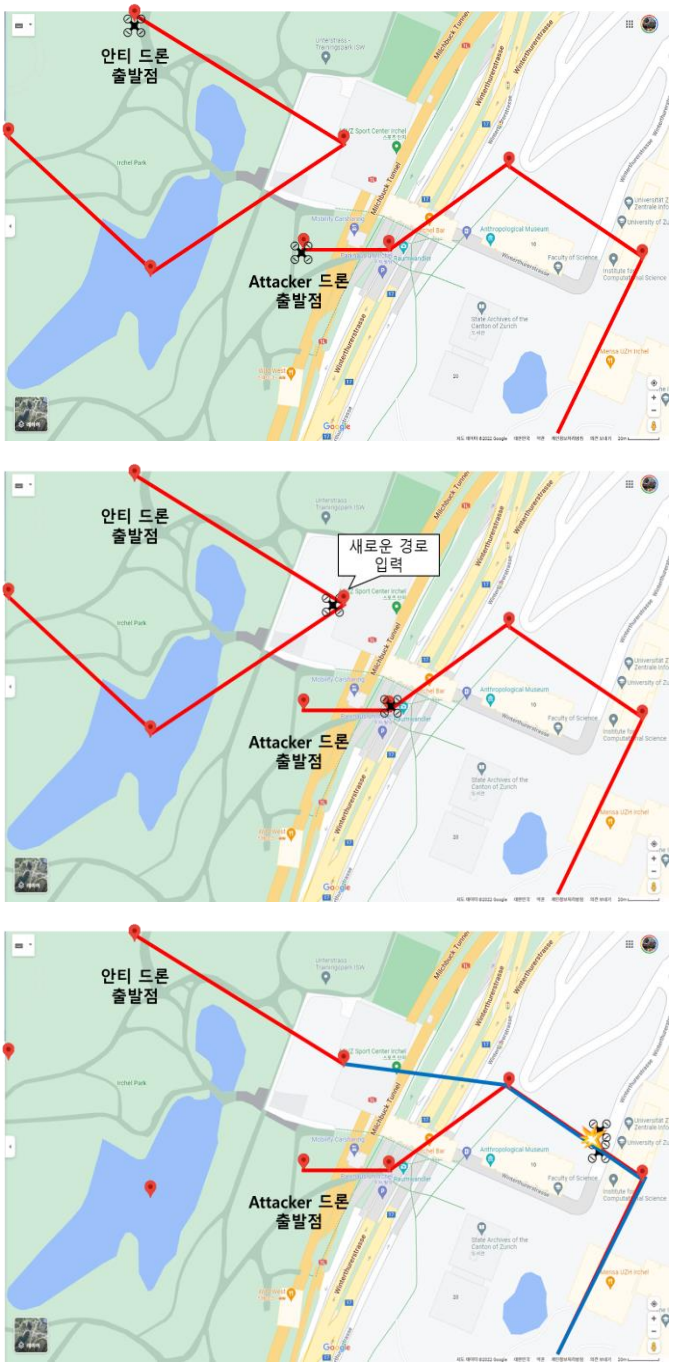
4.3.1 Hard Kill



[그림 17] Hard Kill flow

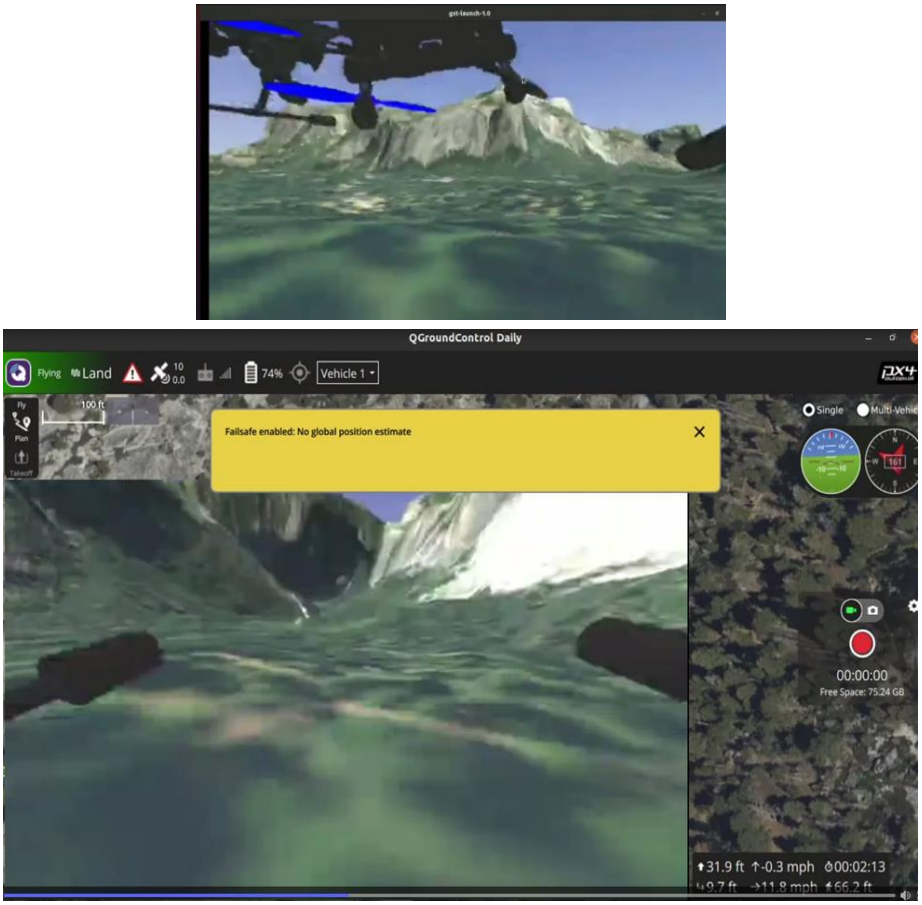
[그림 17[은 hard kill flow 로 충돌 방식을 채택한 것을 파악할 수 있다. 즉, 안티 드론이 미확인 드론을 식별했다면 관제시스템(GCS)에서 안티 드론에게 식별한 드론의 위치를 수동으로 지정해준다. 이 정보를 활용하여 안티 드론의 속도를 더 빠르게 지정하여 미확인 드론과 충돌하는 것이다.

[2] https://youtu.be/yxvvwENwbM4



[그림 18] 스토리보드

[그림 18]은 충돌 알고리즘을 시각화 한것으로 첫 번째 사진에서 미확인 드론과 안티 드론이 각각의 경로로 순찰을 진행 중이다. 두 번째 사진에서 안티 드론이 미확인 드론을 식별하고 새로운 경로를 관제시스템으로부터 입력 받는다. 마지막 사진에서 안티 드론의 속도를 증가시켜 충돌시키는 장면을 볼 수 있다.



[그림 19] 실험 결과

[그림 19][3]는 실험 결과로 충돌 후 미확인 드론이 관제 시스템과 연결 종료되는 것을 확인할 수 있다.

4.3.2 Soft Kill

GPS spoofing 을 활용하여 soft kill 을 구현했다. GPS spoofing 은 가짜 GPS 를 주입하여 강제로 이동, 착륙하도록 만드는 방법이다. 대표적으로 위성의 위치를 옮기거나 시간을 이동시켜 GPS 수신기가 위성으로 받은 신호 대신 GPS spoofing 신호를 선택하도록 위치와 시간을 조작할 수 있다. 그러나 이런 환경은 SITL 환경에서 제공하지 않으므로 미확인 드론에게 직접 설계한 fake_gps 모듈을 삽입하는 방향으로 실험을 진행했다.

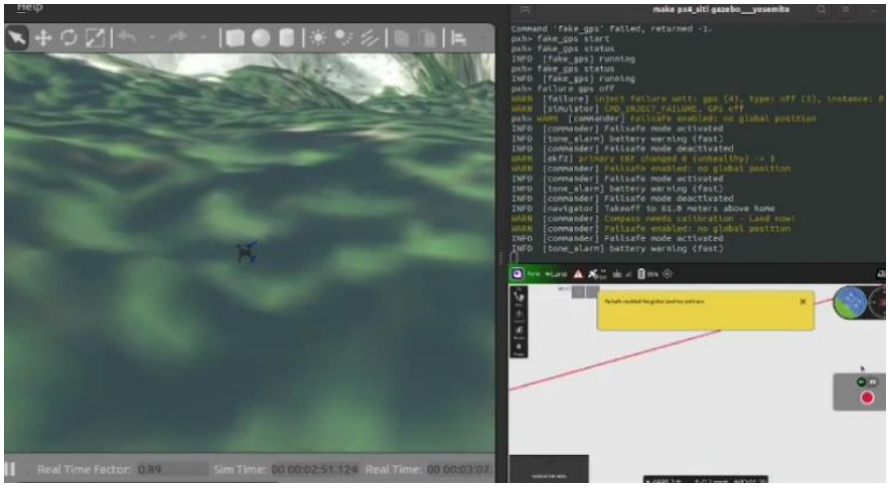
[3] https://youtu.be/di0vd0UJlf8

```
private:
        static constexpr uint32_t SENSOR_INTERVAL_US{1000000 / 50}; // 5 Hz

        void Run() override;

        uORB::PublicationMulti<sensor_gps_s> _sensor_gps_pub{ORB_ID(sensor_gps)};

        int32_t _latitude{473977507};    // Latitude in 1e-7 degrees
        int32_t _longitude{85456073}; // Longitude in 1e-7 degrees
        int32_t _altitude{0};        // Altitude in 1e-3 meters above MSL, (millimetres)
```

```
void FakeGps::Run()
{
        if (should_exit()) {
                ScheduleClear();
                exit_and_cleanup();
                return;
        }

        sensor_gps_s sensor_gps{};
        sensor_gps.time_utc_usec = hrt_abs
        sensor_gps.lat = _latitude;
        sensor_gps.lon = _longitude;
        sensor_gps.alt = _altitude;
        sensor_gps.alt_ellipsoid = _altitu
        sensor_gps.s_variance_m_s = 0.3740f;
        sensor_gps.c_variance_rad = 0.6737f;
        sensor_gps.s_variance_m_s = 0.3740f;
        sensor_gps.c_variance_rad = 0.6737f;
        sensor_gps.eph = 0.0f;
        sensor_gps.epv = 0.0f;
        sensor_gps.hdop = 0.0f;
        sensor_gps.vdop = 0.0;
        sensor_gps.noise_per_ms = 100;
        sensor_gps.jamming_indicator = 0;
        sensor_gps.vel_m_s = 0.0420f;
        sensor_gps.vel_n_m_s = 0.0370f;
        sensor_gps.vel_e_m_s = 0.0200f;
        sensor_gps.vel_d_m_s = -0.0570f;
        sensor_gps.cog_rad = 0.3988f;
        sensor_gps.timestamp_time_relative = 0;
        sensor_gps.heading = NAN;
        sensor_gps.heading_offset = 0.0000;
        sensor_gps.fix_type = 4;
        sensor_gps.jamming_state = 0;
        sensor_gps.vel_ned_valid = true;
        sensor_gps.satellites_used = 14;
        sensor_gps.timestamp = hrt_absolute_time();
        _sensor_gps_pub.publish(sensor_gps);
}
```

[그림 20] fake_gps.hpp, fake_gps.cpp

[그림 20]은 직접 설계한 fake_gps 모듈로 lat 은 위도, lon 은 경도, alt 는 고도, s_variance_m_s 는 속도, s_variance_rad 는 각도를 의미한다. 즉, 해당 코드만 보았을 때 일정 속도와 각도를 유지하며 높이가 0 인 지점으로 드론이 간다는 것을 확인할 수 있다. 이는 단순이 드론을 착륙시키는 목적이 아닌 추락하여 드론을 파괴하는데 목적이 있다.



[그림 21] 실험 결과

[그림 21][4]은 실험 결과로 성공적으로 드론이 바닥으로 추락 후 파괴되었고 관제시스템과 통신이 종료되는 것을 확인할 수 있었다.

5. 결론

5.1 기대효과

인공지능 기반의 식별은 기존의 수동으로 식별하는 방법과 드론의 프로펠러 소리 등으로 해결하는 방법의 한계를 극복할 수 있다. 이를 통해 다른 물체를 드론이라고 식별하는 등의 위험 요소를 줄일 수 있다. 또한, GPS spoofing 을 통해 미확인 드론을 우리 진영으로 hijacking 하여 자원적, 정보적 우위에 설 수 있다.

5.2 향후연구

이번 프로젝트에서는 드론 탐지 및 식별 그리고 무력화 시나리오를 제시하고 각 시나리오를 SITL Simulator 를 통해 시각화함으로써 실제 상황에서 어떻게 안티드론 시스템이 작동하는지 설명하였다. 또한, 드론의 잠재적 취약성을 시사하고 이에 대응하기 위한 보안 연구의 필요성을 제시할 수 있다. 드론을 생산하는 기업이 다양하고 해마다 드론이 급변하고 있어 드론 외관을 확정하는데 어려움이 있고 컴퓨터 성능상 많은 데이터를 학습시키는 데 어려움이 있었다. 현재 학습모델은 천천히 움직이는 드론 인식률이 34 프레임 중 31 프레임 정도로 비교적 우수했지만 빠르게 움직이는 드론은 60%에 머물렀다. 더 다양한 기업의 드론 Dataset 을 구해 드론 식별률을 향상할 예정이다.

그리고 PX4 드론이 보안에 취약한 Mavlink protocol 을 사용한다는 점을 활용하여 GPS spoofing 을 구현해보았는데 향후 연구는 Mavlink 의 보안 취약점에 대해 연구하고 개선 방안을 도출하고 자 한다.

---

[4] https://youtu.be/33fICHTaGWE

6. 참고문헌

[1] 학술연구정보서비스, 'Development of ROS Based Indoor Autonomous Robot', academic journal, 2017

[2] Mordor Intelligence, 'ANTI-DRONE MARKET-GROWTH, TRENDS, COVID-19 IMPACT, AND FORECASTS(2022-2027)', Industry Reports, 2021.

[3] Dardoize Tristan, 'Implementation of Ground Control System for Autonomous Multi-agents using QGroundControl', Research, 2019

[4] 한국통신학회, 'Implementation of Mobility Detection Model in Media using YOLO v3', academic journal, 2022.

[5] 한국위성정보통신학회, 'Efficient Drone Detection method using a Radio-Frequency', K12 4-4, 2017

[6] 한국차세대컴퓨팅학회, 'Patent Trend Analysis of Anti-Drone : Focusing on the Neutralization Means and Methods', vol.16, no2, 2020

[7] 한국통신학회, 'NRP-Sys: Nonlinear Regression Prediction System forYOLO-based Separation Detection of Identical Object', 9 - 17, 2022.