|  |
| --- |
| **드론 GPS Spoofing 공격에 대응하기 위한 모듈 제작**  **(Anti GPS Spoofing Module 제작)**  **요 약**  최근 드론 기술이 발전함에 따라 여러 분야에 드론의 사용이 시도되고 있다. 정보의 중요성이 강조되는 분야에 있어 드론의 보안은 중요한 문제로 자리 잡는다. 하지만 드론의 무선 주파수 대역은 기존의 무선 통신의 취약점을 가지고 있으며, 일반적인 드론의 경우 보안을 고려하지 않은 채 설계된다. 본 연구는 PX4-Autopilot로 가동되는 UAV에서 실제 무선 통신을 이용한 해킹 기법인 GPS(Global Positioning System) Spoofing의 위협이 있는지 확인해보고, 이를 탐지하는 AGSM(Anti GPS Spoofing Module)를 제작하고자 한다. |

**1. 서론**

**1.1. 연구배경**

최근 드론은 기업뿐만 아니라 민간 분야에서도 사용되고 있는 추세이다. 한국항공우주산업진흥협회의 조사 결과에 따르면 2016년부터 2019년까지의 민수용 드론 시장이 연평균 약 3배 이상 성장하였다. 드론을 활용한 서비스 영역이 지속적으로 확장됨에 따라 지역적으로도 광범위하게 드론을 제어하는 무선통신 기술도 고도화가 되고 있다. 하지만, 단순 서비스에 초점을 둔 무선 통신 기술들이 보안성을 고려하지 않아 다양한 보안 취약점을 내포하고 있다. 현재 드론을 중심으로 수행되는 통신 체계는 MAVLink 프로토콜을 기반으로 지상 제어 시스템인 GCS(Ground Control System)와 통신하고 GPS 통신을 통한 드론의 자율 비행을 수행한다. 이 중, GPS 통신은 송신하는 신호가 저전력으로 매우 약하며, 데이터를 암호화하지 않고 수신한 데이터의 적절한 인증 체계을 갖고 있지 않아 다양한 보안 위협에 노출되어 있다. 실제 GPS 신호보다 더욱 강한 해킹 신호를 보내 드론과 GPS 신호를 탈취하는 GPS Spoofing 이 드론의 보안 취약점을 잘 드러내고 있다. 드론의 GPS Spoofing 공격 사례에는 2011년 12월 미국의 록히드마 틴과 이스라엘이 공동으로 제작한 무인 스텔스 RQ-170은 이란을 영내를 정찰하다가 포획 당한 사건이 있다.

따라서 본 연구에서는 PX4 Toolchain을 이용하여 드론의 비행 시뮬레이션 환경을 조성한 후, Fake GPS 신호를 통해 드론의 GPS Spoofing 과정을 시연하고자 한다. 또한 드론의 GPS Spoofing 공격을 탐지하기 위해, AGSM를 제작하고자 한다.

**1.2. 연구목표**

본 연구에서는 첫번째로, 오픈소스인 PX4 와 QGoundControl를 사용하여 드론의 비행을 직접 시연해보고자 한다. 이를 통해 현재 UAV의 작동 방식, UAV 와 GPS의 통신방법에 대해 연구할 것이다.

두번째로, 시뮬레이터를 활용하여 GPS-Spoofing 기반의 위협 시나리오를 전개하고 본 위협에 대한 UAV 보안 취약성을 검증한다. 공격 시나리오에서는 사용자가 직접 UAV를 조정하는 User Control 상황과 미리 설정된 mission을 수행하는 상황을 설정한다.

세번째로, 만약 실제 UAV가 GPS-Spoofing 공격에 의해 문제점이 발생하게 된다면, 이를 감지하는 AGSM (Anti GPS-Spoofing Module)을 제작하는 것이 본 연구의 목표이다.

**2. 관련연구**

**2.1. Spoofing**

Spoofing의 사전적 의미는 '속이다'이다. Spoofing 대상은 [MAC 주소](https://ko.wikipedia.org/wiki/MAC_%EC%A3%BC%EC%86%8C), [IP주소](https://ko.wikipedia.org/wiki/IP%EC%A3%BC%EC%86%8C), [포트](https://ko.wikipedia.org/wiki/%EC%BB%B4%ED%93%A8%ED%84%B0_%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4_%ED%8F%AC%ED%8A%B8) 등 네트워크 통신과 관련된 모든 것이 될 수 있고, Spoofing은 속임을 이용한 공격을 총칭한다. GPS Spoofing의 경우 GPS 수신기가 잘못된 위치에 있다고 믿게 만드는 지능형 형태의 간섭으로 근처에 있는 무선 송신기가 가짜 GPS 신호를 대상 수신기로 보내는 방식으로 행해진다.

**2.2 PX4**

PX4는 무인 차량 및 드론 등 소프트웨어적으로 조작 가능한 기기에 대한 원격조정 또는 자동화에 대한 오픈소스 소프트웨어인 ArduPilot의 하위 프로젝트이다. PX4는 무인 장치 중에서도 쿼드콥터같은 무인항공기의 원격조정 및 자율주행을 위해 설계된 NuttX 운영체제 기반의 소프트웨어로 이루어진 플라이트 컨트롤 보드에서 작동하는 완전한 플랫폼이다.

**2.3 QgroundControl**

QgroundControl는 MAVLink 프로토콜을 지원하는 드론의 GCS이다. QgroundControl은 GCS와 연결된 드론 실시간 위치를 보여주는 UI, PX4의 펌웨어 업로드. Mission planning 기능을 통한 예약 비행, 드론 기기 설정 및 캘리브레이션, MAVLink 통신을 통한 드론의 비행 Log 기록 등의 기능을 사용자에게 제공한다.

**2.4 jMAVSim**

jMAVSim은 PX4로 동작하는 쿼드콥터같은 무인항공기, 간단한 멀티 로터를 가상환경에서 조정할 수 있도록 해주는 Simulator이다. 실제 하드웨어가 없어도, 무인항공기의 설정을 변경하면서 차량의 이륙, 비행, 착륙 및 다양한 실패 조건 (예: GPS 실패)에 적절하게 대응할 수 있는지 테스트하는 데 사용할 수 있다는 장점이 있다.

**2.5 MAVLink**

MAVLink는 무인 기기를 위한 통신 protocol이다. 무인 기기 간에 교환되는 포괄적 인 메시지 집합을 지정하며, 주로 ArduPilot 과 PX4에서 무인 정찰기의 미션을 통제, 전달해주는 역할을 한다. MAVLink는 다양한 유형의 전송 레이어 및 매체를 지원한다는 특징을 지녔다. MAVLink는 WIFI, Ethernet 또는 sub-GHz 주파수를 통해 대규모 통신에 도달할 수 있으며, IP Network를 통해 GCS에게 MAVLink 메시지를 전송할 수 있다. 이때 UDP/TCP 통신을 모두 지원한다.

**3. Fake-GPS를 통한 해킹 시나리오 제안**

**3.1. 요구사항**

**3.1.1. PX4 기반 Simulator에 대한 요구사항**

PX4 기반 시뮬레이션을 사용하기 위해서는 PX4 Tool Chain을 사용하여 관련 프로그램들을 설치하여야 한다. 해당 과정은 linux ubuntu 18.04를 기준으로 한다.

1. Tool-chain 설치 스크립트 생성

$ sudo apt-get install git

$ git clone <https://github.com/PX4/Firmware.git> --recursive

$ cd Firmware

$ bash ./Tools/setup/ubuntu.sh

$ sudo reboot now

2. QGroundControl 설치

$ sudo apt-get remove modemanager -y

$ sudo apt install gstreamer1.0-plugins-bad gstreamer1.0-libavgstreamer1.0-gl -y

$Wget

<https://s3-us-west2.amazonaws.com/qgroundcontrol/latest/QGroundControl.AppImage>

3. QgroundControl & jmavsim 실행

- chmod +x ./QGroundControl.AppImage

- ./QGroundControl.AppImage

**3.1.2. jMAVSim 환경 구성**



[그림 1] jMAVSim(좌) QgroundControl(우) interface

[그림1] 은 앞서 3.1.1의 PX4 기반 Simulator에 대한 요구사항에 맞게 PX4 Project와 QgroundControl을 설치한 모습이다. 좌측의 경우 SITL 프로그램인 jMAVSim의 화면이며 우측은 QgroundControl의 화면이다. QgroundControl의 좌측 상단을 보면 톱니바퀴의 설정모양이 있다.

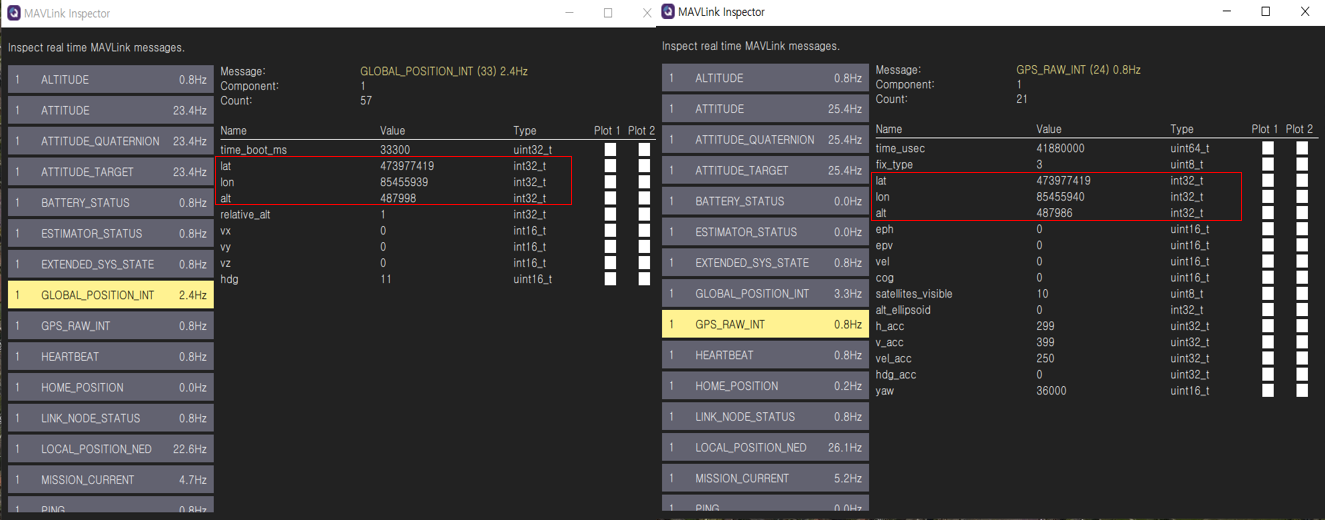


[그림 2] mission에 따라 비행을 수행하는 드론

이 설정을 통하여 [그림2] 와 같이 waypoint들을 만들어 드론이 수행할 mission을 지정해준다. 위와 같은mission 이 지정되어 있으면 그림1의 비행 버튼을 클릭하여 실제 jMAVSim 과 QgroundControl에서 해당 드론이 비행함을 볼 수 있다.

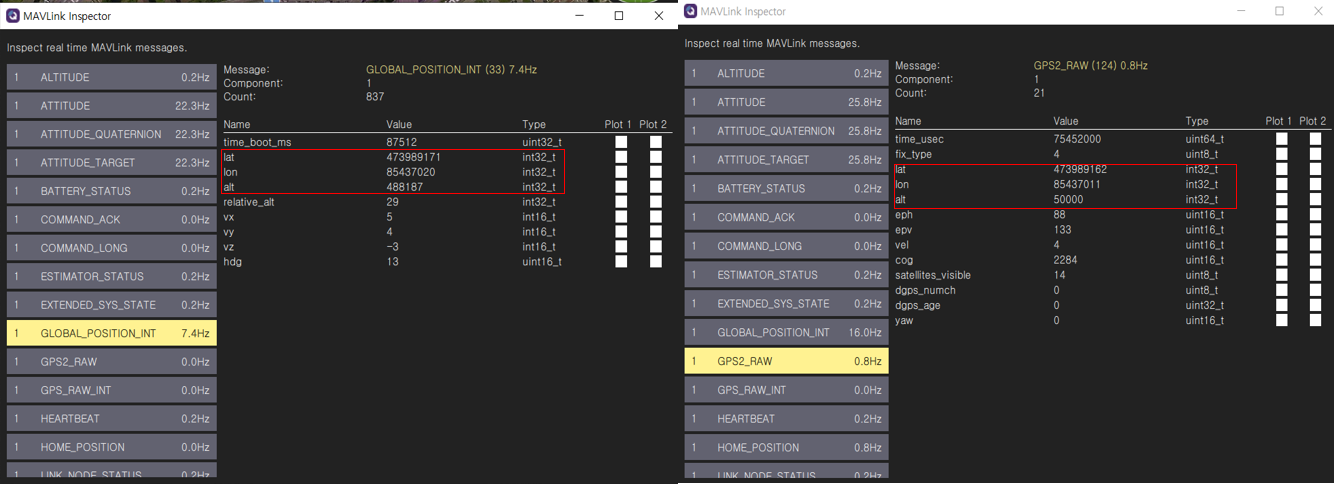
**3.2. Fake-GPS를 이용한 GPS-SPoofing 시연**

**3.2.1. GPS-Spoofing 상황을 위한 사전 작업**



[그림 3] GPS\_RAW\_INT를 기반으로 생성되는 GLOBAL\_POSITION\_INT 매개변수

본래 드론의 운행방법을 살펴보면, [그림 3] 과 같이 GPS에서 보내는 신호인 GPS\_RAW\_INT를 기준으로 드론 내부에서 속도, 값을 더해 자체 위치 추정치인 GLOBAL\_POSITION\_INT를 생성한다.



[그림 4] GPS2\_RAW를 기반으로 생성되는 GLOBAL\_POSITION\_INT 매개변수

3.1 환경에서 PX4 자체 Module인 Fake\_gps를 $fake\_gps start 명령어를 통해 [그림 4]와 같이 기존 드론이 참고하고 있던 GPS\_RAW\_INT 이외의 GPS\_RAW2 라는 GPS를 생성하였다. GPS\_RAW2라는 새로운 GPS가 생성됨에도 불구하고, 드론은 기존의 GPS\_RAW\_INT를 참고하여 GLOBAL\_POSITION\_INT를 생성한다. 따라서, 기존의 GPS인 GPS\_RAW\_INT 매개변수 전송을 종료하기 위하여 PX4에서 제공하는 GPS off 상황을 만들어 주는 $failure gps off 명령어를 사용하였다. 이에, 기존의 GPS\_RAW\_INT 매개변수 전송이 중단되므로, [그림 4]와 같이 Fake-gps가 제공하는 gps\_raw2를 기반으로 GLOBAL\_POSITION\_INT 값이 생성됨을 확인할 수 있다.

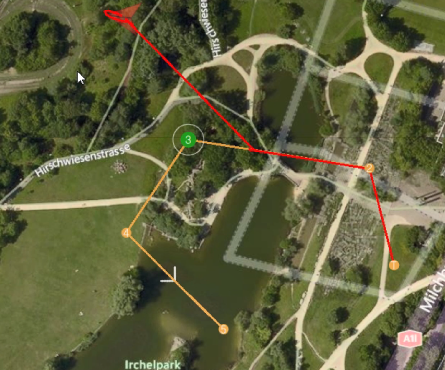


[그림 5] Fake\_GPS 동작을 위한 PX4 Project 명령어

[그림 5]는 PX4의 내부 함수로 Spoofing 동작을 구현하기 위한 명령어들이다. 앞서 설명한 바와 같이 GPS신호가 두 개 이상일 경우 드론은 지속적으로 GPS\_RAW\_INT 값만을 참고하기 때문에 해당 신호를 멈추기 위해 $failure gps off를 사용해야한다. 이를 위해서 우선적으로 $ param set SYS\_FAULURE\_EN 1으로 failure command를 사용할 수 있도록 설정해주어야 한다. GPS 신호가 사라진 후 바로 GPS2\_RAW 매개변수를 참조할 수 있도록 $failure gps off 가 입력되기전 fake\_gps 함수를 사용하였다.

**3.2.2. GPS-Spoofing 실행 결과**

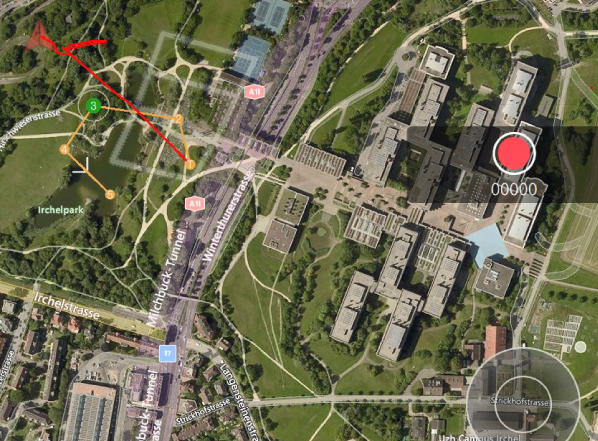
**3.2.2.1. Mission 과정에서의 GPS-Spoofing 공격 결과**



[그림 6] Fake-GPS 공격으로 경로가 바뀐 상황 (Mission 과정)

[그림 6]을 살펴보면 드론이 3번 waypoint로 비행을 하는 도중 경로를 이탈하여 특정 위치로 비행하고 있음을 살펴볼 수 있다. 실제 GPS가 종료된 후, 드론은 GPS가 종료된 곳에 머무르다 Fake-GPS의 신호를 받기 시작하면 [그림 6] 와 같이 사전에 사용자가 정의한 Fake-GPS의 값으로 자신의 위치를 인식하게 된다. 드론은 Fake-GPS에가 전송하는 위치에 도달하기 위해 빠른 속도로 해당 방향으로 움직이다 해당 위치에 도달하게 되면 근방을 배회하다 해당 위치에서 저속으로 비행하게 된다.

**3.2.2.2. User Control 과정에서의 GPS-Spoofing 공격 결과**

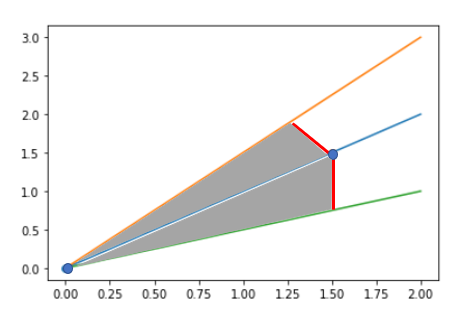


[그림 7] Fake-GPS 공격으로 경로가 바뀐 상황 (User Control 과정)

[그림 7]의 상황은 mission이 아닌 사용자의 컨트롤러 입력으로 드론이 비행하는 도중, GPS-Spoofing공격을 받은 상황이다. 사용자의 조정으로 출발지에서 왼쪽 대각선 방향으로 이동하던 드론이 Fake-GPS 값을 받게 되어 자신의 위치를 혼동하게 된다. 시간이 더 흐르게 되면, Fake\_GPS 때문에 드론의 방향 정보에 혼동이 생긴다. 이에 [그림 7]과 같이 컨트롤러를 북동쪽으로 조정하여도, QgroundControl 상에서는 드론이 북서쪽으로 이동하는 현상도 발생하게 된다.

**4. AGSM 설계**

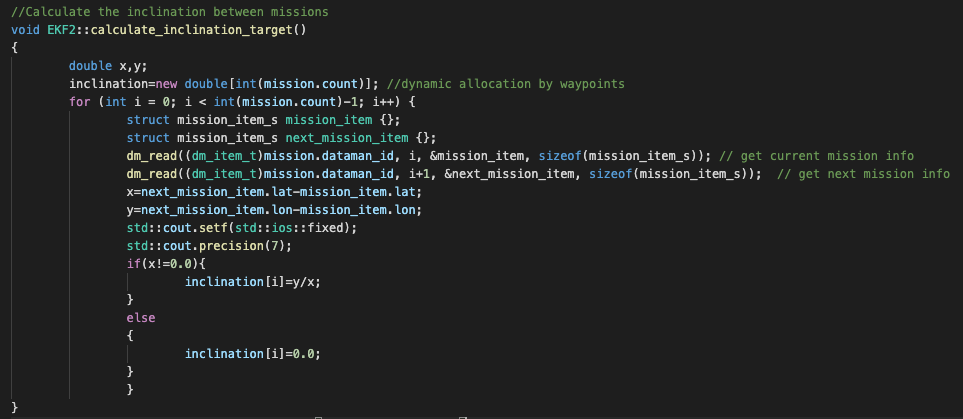
**4-1 기울기를 통한 드론 정상 운행 판단 방법**



[그림 8] 기울기를 통한 드론 정상 운행 판단 방법

파란색의 두 점을 각각 waypoint1, waypoint2 라고 하였을 때, 두 waypoint를 통하여 기울기를 계산할 수 있다. 계산된 기울기를 통하여 [그림 8]과 같이 오차범위에 해당되는 기울기 2개를 추가로 생성한다. 3개의 기울기와 waypoints를 이용하여 회색 영역과 같은 드론의 유효 범위를 지정한다. 만약 드론이 유효 범위 밖에서 10초 이상 머물러 있다면 가장 가까운 waypoint로 복귀 명령을 진행한다.

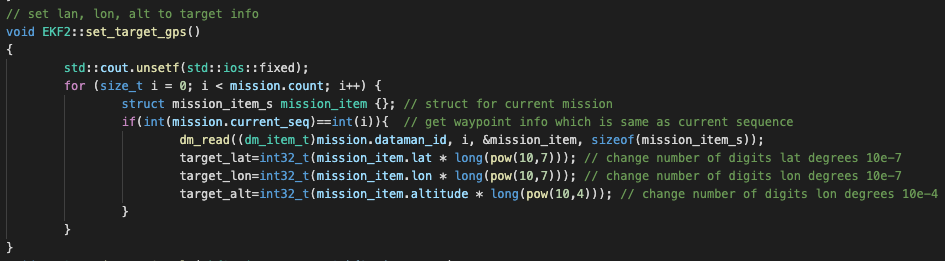
**4-2 두 waypoint의 위치 정보를 통한 기울기 생성 구현**



[그림 9] 두 waypoint의 위치 정보를 통한 기울기 생성 코드

우선, 현재 mission의 위치와 다음 mission의 위치 정보를 담기 위한 mission\_item 과 next\_mission\_item이라는 두 개의 구조체를 선언한다. dm\_read함수를 이용하여 드론 내의 자체 storage인 dataman에 접근하여 미션정보를 가져온다. 이 때, 으\_read는 전체 미션의 개수(mission. count) 만큼 반복된다. mission. dataman\_id를 통해 미션 데이터에 접근하여 현재 mission의 정보가 있는 i번째의 item을 읽어와 mission\_item에 값을 할당한다. 같은 방식으로 i+1번째에 위치하여 있는 다음 목적지의 정보를 next\_mission\_item에 할당한다. 이렇게 저장한 두 값의 경도와 위도를 이용하여 x의 증가량과 y의 증가량을 구한 후, inclination에 값을 할당한다. 마지막으로 현재위치와 출발지의 기울기 함수(4-4 참고)에서 나오는 결과값과 자릿수를 맞춰 주기 위하여 precision(7)를 사용하였다.

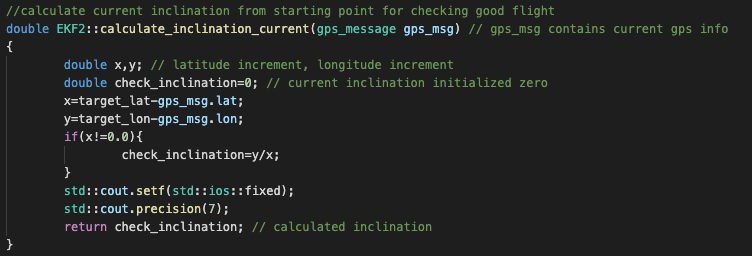
**4-3 드론의 mission waypoint(출발지) 위치 저장 코드**



[그림 10] 드론의 mission waypoint(출발지) 위치 저장 코드

print\_target\_gps 함수는 드론에 저장 되어있는 미션을 읽어 현재 진행중인 mission의 출발지 위치 정보를 전역 변수인 target\_lat, target\_lon, tartget\_alt에 할당한다. dm\_read 함수를 통해 현재 미션만을 가져온다. 현재 미션만을 가져오기 위하여 mission. current\_seq을 확인하는 조건을 부여하였다.

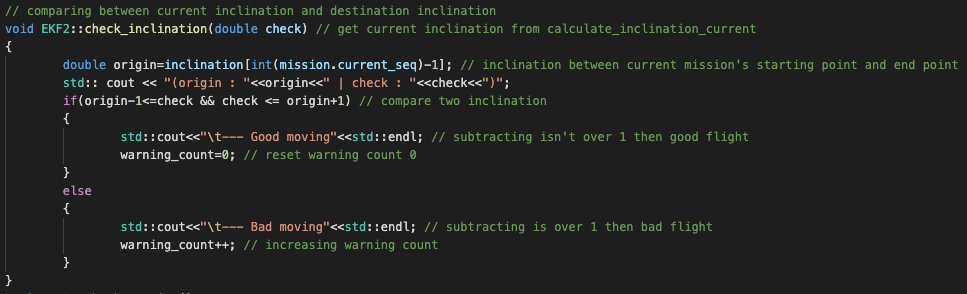
**4-4 현재 드론의 위치와 mission waypoint 사이의 기울기 생성 코드**



[그림 11] 드론의 현재 위치와 목적지 사이의 기울기 생성 코드

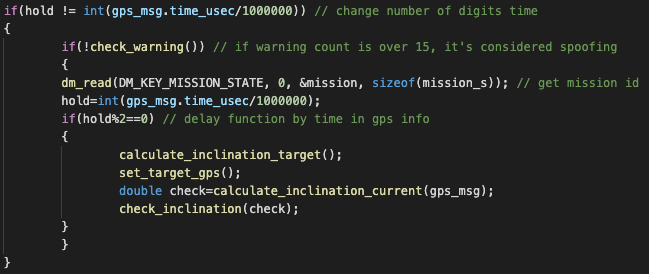
calculate\_inclination\_current의 함수의 경우 출발지의 위치 정보에서 현재 드론의 위치 정보를 통하여 기울기를 생성한다. 매개변수로 PX4 내장 함수인 collect\_gps를 통해 수집한 GPS 데이터 정보가 들어있는 gps\_msg를 사용한다. Collect\_gps는 실시간으로 드론의 현재 위치를 담고 있다. 따라서, 4-3 과정에서 저장된 mission 출발지의 경도 위도를 이용하여 실시간 드론 위치에 맞는 기울기를 생성이 가능하다.

**4-5 목적지간 기울기와 현재 비행 중 기울기 비교**



[그림 12] 목적지간 기울기와 현재 비행 중 기울기 비교 코드

Origin의 경우 출발점의 위치정보와 목적지 정보를 통해 생성된 기울기를 저장하는 변수이다. check는 출발점의 위치 정보와 현재 드론의 위치를 통해 생성된 기울기를 저장하는 변수이다. origin과 check를 비교하여 만약 기울기 차이가 1이상 차이가 나면 잘못된 비행으로 판단하여 사용자에게 “Bad moving” 이라는 경고를 보낸다. Bad moving이 발생하면 warning count가 누적되며 연속 15회 이상 누적 시 “abnormal moving detected를 출력하게 된다. Warning count는 Good moving시 발생시 Reset을 시켜 단순한 자연 환경 변화로 발생하는 이상 비행에 대한 오탐을 방지한다.



[그림 13] 구현된 함수의 실제 사용 모습

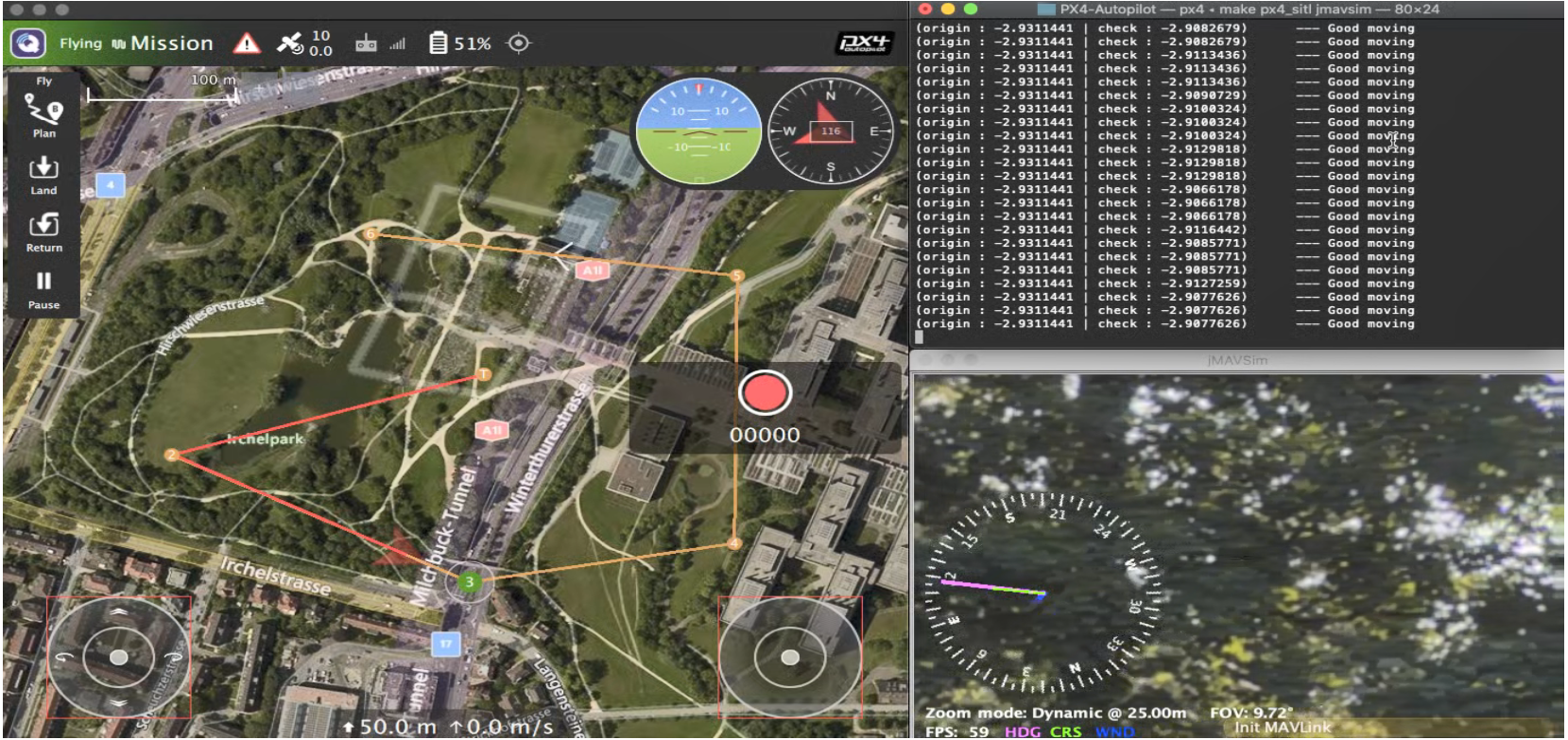
[그림 13]은 ekf2.cpp file의 UpdateGpsSample 함수 내부의 모습이다. 우선적으로 조건문을 통해 spoofing 상황인지를 판단하고 spoofing 상황이 아니라면 함수를 계속해서 진행한다. dm\_read 함수를 통해 mission\_id를 획득하고, 2초 마다 앞서 정의한 함수를 차례로 실행한다.

**5. 프로젝트 결과**

**5.1. 연구 결과**

본 프로젝트의 주안점은 실시간으로 드론의 위치를 판단하여 해당 드론의 비행이 유효한지 여부를 실시간으로 사용자에게 알려주는 것이다.

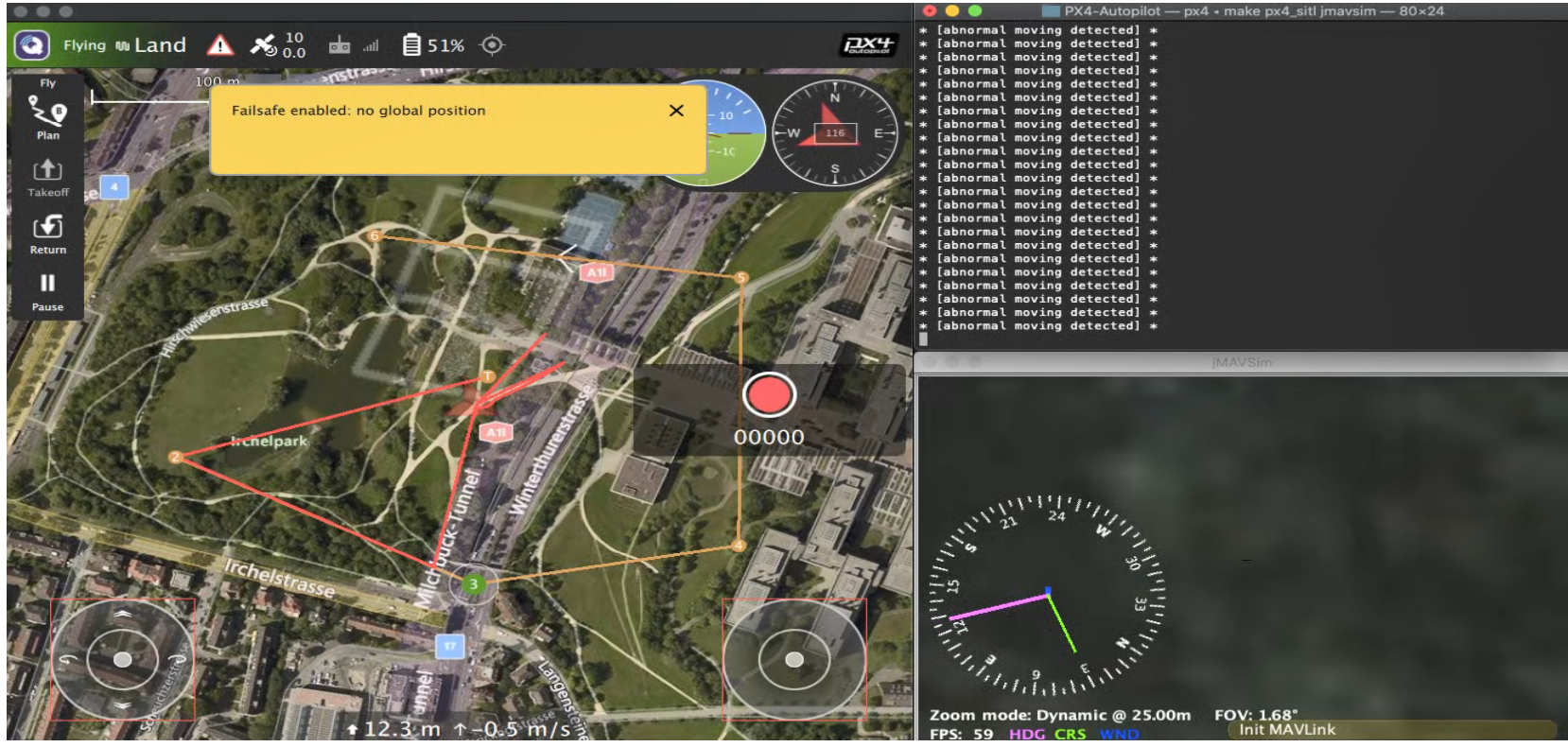
**5.1.1. 정상 운행 중일때의 AGSM 상황**



**[그림 14] AGSM을 통한 드론 비행 모니터링 (정상 운행)**

[그림 14]의 경우 드론의 비행 중 구현된 AGSM를 실행하여 실시간으로 드론의 운행을 모니터링 하고 있는 상황이다. 4-5에서 언급한 방식대로 기울기의 차이로 정확한 경로를 판단하고 있으며, 기울기의 차이가 크지 않기 때문에 “Good moving”을 출력하고 있는 상황이다.

**5.1.2. GPS-Spoofing 공격 시 AGSM 상황**



[그림 15] AGSM을 통한 드론 비행 모니터링 (공격 상황)

GPS-Spoofing 공격이 감지되면, 출발지와 현재 드론의 위치 정보 사이의 기울기가 출발지와 목적지 사이의 기울기와 큰 차이가 나기 때문에 4-5의 코드에 의하여 “Bad moving” 이 출력되고 이 횟수가 연속 15회 이상 발생하게 되면 [그림 15] 와 같이 AGSM이 공격 상황으로 판단하여 “Abnormal moving detected” 라는 경고문을 수차례 출력한다. 사용자는 이를 통해 QgroundControl이 제대로 지원되지 않는 상황에서도 공격 상황임을 판단할 수 있게 된다.

**6. 결론**

**6.1. 기대효과**

해당 연구를 통해서 드론의 보안 문제를 환기시킴으로 드론 운용자들(기업 및 민간)에게 드론 해킹에 대한 경각심을 일깨워 줄 수 있으며, PX4 사용자들에게도 시각적인 자료를 제공할 수 있게 될 것이다.

또한 드론 운용자들은 AGSM을 사용하여 GPS-Spoofing 공격을 탐지할 수 있게 될 것이며, 이를 통해 GPS-Spoofing 공격에 대한 즉각적인 대처가 가능 할 것으로 보인다.

마지막으로, 해당 연구의 과정을 PX4 오픈소스 커뮤니티에 공유함으로써 기존에 부족했던 드론 입문자들을 위한 자료가 생성될 것이며 다른 Anti module 제작에 참고자료가 될 수 있을 것이다.

**6.2. 추후 연구 방향**

추 후에 AGSM으로 인해 공격 패턴이 발견되면 사용자의 즉각적인 입력을 받아 드론을 강제로 특정 위치로 이동시키는 대처 기능을 추가하고자 한다. 또한 GPS 센서가 아닌, 고도계, 카메라와 같은 다른 센서의 보안 취약점을 분석하고 이에 맞는 Anti Module을 제작하고자 한다. 그리고, 비정상 신호가 식별된 시점에서 드론의 재가동 여부를 판단하는 Module을 제작하여, mission 상태뿐만 아니라, user가 직접 control 하는 상황에서도 GPS-Spoofing을 감지할 수 있도록 하고자 한다.

**7. 참고문헌**

[1] QGroundControl User Guide, Dronecode, 2021.04.26. 접속,

https://docs.qgroundcontrol.com/master/en/index.html

[2] PX4 Autopilot User Guide Introduction, PX4 Autopilot 2021.06.14. 접속, <https://docs.px4.io/master/en/>

[3] 조승민, “드론 보안에 적용된 암호기술 현황”, 정보보호학회지 vol.30-2, 2020.4

[4] A. Koubâa, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," in *IEEE Access*, vol. 7, pp.

87658-87680, 2019

[5] 류해원, 최성한, 하일규, “드론 운용의 보안 위협과 대응 방안", 한국정보처리학회, 2018.10

[6] 서진범, “GPS 스니핑을 이용한 안티 드론 알고리즘”, 한국정보통신학회, pp. 63 –66, 2019.05.23