

# DroneKit-SITL과 초음파센서를 활용한 드론 자율주행 가상화 구현

현만석, 최광훈, 김재훈

아주대학교 컴퓨터공학과

{bws2958, saarc, jaikim}@ajou.ac.kr

## Drone Self-Running Virtualization Using DroneKit-SITL and Ultrasonic Sensor

Hyun Man Seok, Choi Kwang Hoon, Kim Jai-Hoon

Department of Computer Engineering, Ajou University

요약

드론에 대한 관심이 증가하면서, 여러 분야에서 드론의 활용에 대한 연구가 진행되고 있다. 더불어 기체 제어에 대한 오픈 소스 플랫폼이 제공되면서 하드웨어 없이 소프트웨어만으로도 기체 제어 시뮬레이션을 수행할 수 있게 되었다. 이번 연구에서는 ArduPilot 기반의 기체 제어 API인 DroneKit와 Software Simulator인 SITL Simulator를 활용하여 드론 제어를 시뮬레이션한다. 더불어 Raspberry Pi와 초음파센서를 활용하여 기체가 이동 중에 장애물을 만나는 조건을 부여하고 회피 및 목적지 이동에 대한 경로 설계 알고리즘을 시뮬레이션으로 구현해본다.

### I. 서론

드론에 대한 관심이 증가하면서, 여러 분야에서 드론의 활용에 대한 연구가 진행되고 있다. DJI, Parrot 등의 드론 제조사들은 드론의 판매뿐만 아니라 기체 제어를 위한 API도 제공한다. 시중에 판매되는 완성형 드론들은 프레임이 안정적이고 제공되는 API들의 완성도가 높지만 일반 개발자들이 개발을 목적으로 기체를 직접 커스터마이징 하기에는 제한이 있다. 한편 Pixhawk, APM 등 DIY로 제작이 가능한 드론의 경우에도 오픈소스로 기체 제어 및 소프트웨어 시뮬레이션 프로그램이 제공되고 있다. DIY로 개인이 드론을 제작할 경우, 개인의 목적에 따라 기체의 커스터마이징이 용이하나 적절한 시뮬레이션을 거치지 않고 비행을 할 경우, 완성형 드론에 비해 비행 중에 추락할 가능성이 높다는 단점이 있다. 본 연구에서는 ArduPilot 기반의 드론에 대한 기체 제어 API인 DroneKit와 Software Simulator인 SITL Simulator를 활용하여 드론 제어를 시뮬레이션한다. 더불어 Raspberry Pi와 초음파센서를 활용하여 기체가 이동 중에 장애물을 만나는 조건을 부여하고 해당 조건에서의 기체 제어 및 회피에 대한 경로 설계 알고리즘을 시뮬레이션으로 구현한다.

### II. 본론

#### 1) DroneKit<sup>[1]</sup>

3D Robotics사에서 제공하는 Open API로써 MAVLink<sup>[2]</sup> protocol로 드론과 GCS(Ground Control Station)를 연결하여 기체의 상태 정보 및 동작 명령을 내릴 수 있다. 본 논문에서는 Python 프로그래밍 언어를 바탕으로 DroneKit API를 활용하여 드론 제어 코드를 제작 및 테스트하였다.

#### 2) DroneKit-SITL<sup>[3]</sup>

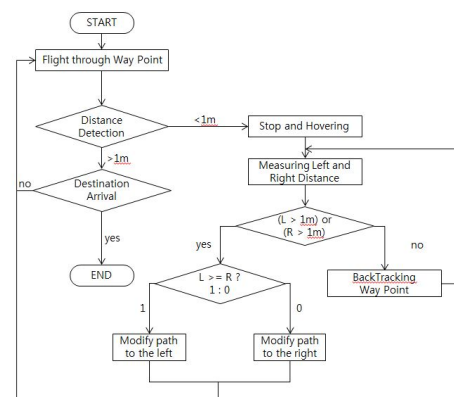
DroneKit-SITL(Software In The Loop)은 Mac OS X, Linux, Window 환경에서 DroneKit API로 제작한 드론 제어 코드를 시뮬레이션해주는 Simulator이다. DroneKit API와 마찬가지로 DroneKit-SITL도 오픈소스로 제공되며, Binary build는 x86 Architecture에서 가능하며, ARM Architecture(e.g. for RPi)에서는 사용이 불가능하다.

#### 3) Raspberry Pi<sup>[4]</sup>, HC-SR04<sup>[5]</sup>

이번 연구에서 사용할 초음파센서(HC-SR04)의 데이터 값을 읽기 위해 라즈베리파이를 사용하였다. UART 방식을 사용하는 일반적인 FTDI 케이블로는 HC-SR04의 데이터 값을 읽을 수 없기 때문에 라즈베리파이와 초음파센서를 연결하여 1차적인 센서 정보를 읽었다. 이후, TCP socket 통신을 통해 시뮬레이션을 진행하는 PC에 센서 데이터를 전달하였으며, 해당 데이터를 바탕으로 시뮬레이터에서 장애물과 드론 사이의 거리를 인식하여 기체 제어 및 이동경로를 자율적으로 수정하는 알고리즘을 설계하였다.

#### 4) 센서 데이터 기반의 장애물 인식과 회피 알고리즘 생성

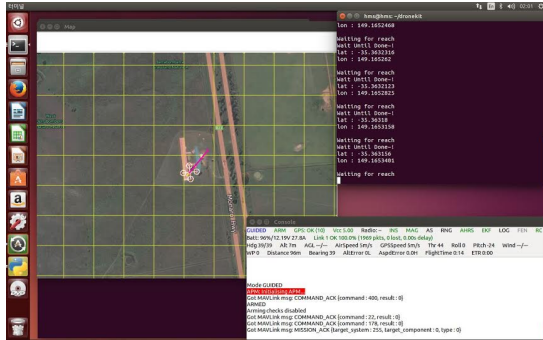
본 연구에서 전진주와 송진담, 빌딩 같은 복잡도가 낮은 물체를 장애물로 설정하였다. HC-SR04 초음파 센서는 카메라, 레이더와 달리 대상물 측정 범위가 넓지 않아 복잡도가 높은 형태의 장애물을 정확히 포착하기에 부적합하다. <그림1>은 평면 공간상에서 거리 탐색을 기반으로 하는 회피 알고리즘을 나타낸다. 실내에서 SITL Simulator를 활용하여 거리를 측정하기 때문에 장애물 인식 거리를 1m로 설정하였다. 실제 상황에서는 기체 속력과 드론의 명령어 인식 및 동작에 대한 delay time을 고려하여 거리의 재설정 필요하다.



<그림1> 거리 탐색을 기반의 평면 공간 회피 알고리즘(L:left, R:right)

### 5) DroneKit-SITL을 활용한 드론 제어

초음파 센서를 활용하여 드론이 자율적으로 장애물을 회피하는 알고리즘을 테스트하기 위해 DroneKit-SITL Simulator를 사용한다. 이는 시뮬레이션 없이 실제 드론 하드웨어를 사용하여 테스트하는 경우, 예측되지 않은 요소들에 의해 기체가 추락하여 사고를 유발할 수 있다는 점에서 보다 안정적이고, 경제적인 실험이다.



<그림2> DroneKit-SITL을 활용한 드론 제어 시뮬레이션

### 6) 센서 데이터 수집

시뮬레이션을 진행하는 PC에서는 echo와 trigger 신호를 활용하는 HC-SR04 초음파 센서의 값을 직접 읽을 수가 없다. 따라서 GPIO 핀으로 HC-SR04 센서값을 읽을 수 있는 Raspberry Pi를 활용하여 거리 데이터를 획득한다. 이후 Raspberry Pi를 네트워크에 연결한 뒤, TCP 소켓 통신을 활용하여 시뮬레이션을 진행하고 있는 PC에 센서값을 전달한다. 본 연구에서는 충돌 위험 거리를 50cm 미만으로 가정하고 소프트웨어로 시뮬레이션을 진행하는 드론과 장애물 사이의 거리가 기준 거리 미만이면 충돌 위험 상황으로 자동적으로 인식하여 경로를 재설정하도록 한다.

<그림3> HC-SR04 초음파센서를 활용한 거리 측정

```
Distance : 344.2 cm
Distance : 257.2 cm
Distance : 184.9 cm
Distance : 184.1 cm
Distance : 185.0 cm
Distance : 184.2 cm
Distance : 184.2 cm
Distance : 184.2 cm
Distance : 183.8 cm
Distance : 184.7 cm
Distance : 183.8 cm
Distance : 183.9 cm
Distance : 184.3 cm
Distance : 184.4 cm
Distance : 184.3 cm
Distance : 184.3 cm
Distance : 183.8 cm
Distance : 183.8 cm
Distance : 183.8 cm
Distance : 184.3 cm
Distance : 184.3 cm
Distance : 183.8 cm
```

### 7) DroneKit SITL을 활용한 장애물 인식 및 회피 시뮬레이션

<그림4>는 1)~6)의 내용을 바탕으로 제작한 DroneKit-SITL 시뮬레이션 프로그램의 동작 모습이다. (4.1)은 지정된 목적지로 기체가 이동 중에 전방의 장애물을 감지하여, 제자리에 정지 및 호버링을 한 뒤, 좌우 거리 탐색을 통한 회피 경로 설정을 나타낸다. (4.2)는 회피 경로를 설정하여 장애물을 회피하고 사전에 지정된 목적지로의 자율주행을 나타낸다.

시뮬레이션 프로그램을 사용하지만 드론의 위치와 경로는 실제 하드웨어와 연동 시에 사용하는 구글 지도 API를 그대로 사용하고 있으며, 기체의 이동 속도에 따른 고도와 속도 변화를 Console 화면을 통해 확인할 수 있다.



<그림4> 장애물 인식 및 좌우 회피 경로 탐지(4.1), 회피 경로 설정 및 목적지 이동(4.2)

## III. 결론

본 논문에서는 DroneKit-SITL Simulator와 초음파센서를 활용하여, 센서 데이터와 연계한 드론의 자율주행을 가상으로 구현하였다. DroneKit-SITL Simulator의 활용은 기체 제어 실험의 안전성을 높여주었다. 실제로 하드웨어를 제작하여 다양한 센서들과 연결한 드론 제어 실험의 경우, 다양한 요인들에 의해 기체가 추락하는 사고가 발생하였다. 드론 제작 시에 Calibration이 제대로 되지 않거나 비행 중에 센서 오작동에 의한 예상치를 벗어난 데이터가 입력되어 드론이 비행 범위를 벗어나거나 추락하는 사고가 발생하였다. 하지만 DroneKit-SITL을 활용하여 실제 비행 이전에 발생할 수 있는 위험요소에 대한 시뮬레이션을 함으로써 본 비행 실험에서 드론이 임무 수행 중에 추락하는 사고를 줄일 수 있었다. 뿐만 아니라 드론이 추락하지 않음으로써 경제적인 효율성도 높일 수가 있었다.

본 논문에서는 장애물 감지를 위하여 초음파 센서를 사용하였지만, 드론과 연계하여 사용될 수 있는 센서는 다양하다. 이와 관련한 추후 연구로는 적외선 센서 및 레이더를 활용한 침입자 감지 및 추적, 열화상 카메라와 풍속 감지 센서를 활용한 산불 진행 방향 및 피해 예측 프로그램 등이 있다.

## ACKNOWLEDGMENT

이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No. 2015R1D1A1A01060034)과 정보통신기술진흥센터의 ICT유망기술개발지원사업(2016-0-00546(R7723-16-0033))의 결과임.

## 참 고 문 헌

- [1] DroneKit, (<http://dronekit.io/>)
- [2] MAVLink, (<http://qgroundcontrol.org/mavlink/start>)
- [3] DroneKit-SITL, ([http://python.dronekit.io/develop/sitl\\_setup.html](http://python.dronekit.io/develop/sitl_setup.html))
- [4] Raspberry Pi, (<https://www.raspberrypi.org/>)
- [5] HC-SR04, (<https://www.sparkfun.com/products/13959>)
- [6] Meier, L., Tanskanen, P., Fraundorfer, F., Pollefeys, M.: Pixhawk: A system for autonomous flight using onboard computer vision. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 2992 - 2997 (2011)