

Original Article

<https://doi.org/10.12985/ksaa.2019.27.3.001>
ISSN 1225-9705(print) ISSN 2466-1791(online)

ROS와 픽스호크를 활용한 이기종 무인 이동체간 통합 시뮬레이션 환경 구축

김형민*, 이대우**

Integrated Simulation Environment for Heterogeneous Unmanned Vehicle using ROS and Pixhawk

Hyeong-Min Kim*, Dae-Woo Lee**

ABSTRACT

Cooperative systems among various unmanned vehicles are widely used in various field and emerging. Unmanned vehicles are able to operate various missions without operator onboard and they are highly stable. Collaborative work of multiple unmanned vehicles is emphasized due to the difficulty of recent missions such as SEAD (Suppression of the Enemy Air Defenses), MUSIC (Manned Unmanned Systems Integration Capability), golden-time in the rescue mission. In this study, ROS and Pixhawk were proposed as a method of construction of a collaboration system and framework for an integrated simulation environment for heterogeneous unmanned vehicles is proposed. Totally 5 unmanned vehicles were set for the simulation for the observation of illegal fishing boats. This paper shows the feasibility of the cooperative system using ROS and Pixhawk through the simulation and the experiment.

Key Words : Unmanned Vehicle(무인 이동체), Cooperative System(협업 시스템), Open source(오픈소스), Pixhawk(픽스호크), ROS(로봇 운영체제)

1. 서 론

세계적으로 무인 이동체에 대한 관심이 높아짐에 따라 관련 시스템 개발의 중요성이 부각되어 왔다.

무인 이동체는 유인 이동체에 비해 인명피해

를 줄일 수 있다는 장점이 있고, Imdoukh(2017), 김경배(2016), Mogili(2018), Bechtsis(2017), Bechtsis(2017), Specht(2017), Wernli(2000) 등의 연구에서 위험지역의 수색, 정찰, 팽핑 등 다양한 임무의 수행으로 관련 분야의 연구가 활발히 진행되고 있음을 확인할 수 있다.

하지만 다양한 임무를 수행하면서 임무난이도가 증가함에 따라 협업 시스템의 필요성이 부각되고 있다. 한동기(2017)에 따르면 미 국방부는 다수의 무인 이동체를 이용하여 적의 중요 시설을 타격하는 SEAD(Suppression of enemy air defense)

Received : 4. Apr. 2019. Revised : 10. May. 2019.
Accepted : 11. Jul. 2019

* 부산대학교 항공우주공학과

** 부산대학교 항공우주공학과 교수

연락처 E-mail : baenggi@pusan.ac.kr

연락처 주소 : 부산광역시 금정구 부산대학로 63번길 2

임무, 지상운용자와 무인 차량 및 무인기를 활용하여 적을 수색하고 타격하는 MUSIC(Manned Unmanned Systems Integration Capability) 임무, 조난자구조 등에서 골든타임을 놓치지 않기 위한 다수의 무인 이동체의 협업에 대한 필요성을 제시하였다.

이에 본 연구에서는 픽스호크와 ROS(Robot Operating System)를 이용하여 불법 어선 감시를 위한 협업 시스템을 제시하였다. 픽스호크와 ROS는 무인 이동체 및 로봇 개발에 활발히 사용되고 있다. Sinisterra(2017)는 픽스호크를 이용한 무인 수상정 시스템을 구성하였으나, ROS가 아닌 Dronekit을 사용하여 구성 ROS에 비해 확장성이 떨어지며, Braga(2017), López(2017), Portugal (2019)은 ROS를 이용하여 SLAM, 다수 로봇에 대한 적용을 하였으나, 이 연구의 대부분은 단일 플랫폼에 최적화 되어 있는 패키지를 사용함으로써 확장성에 한계가 있었다. 또한 윤원근, 조인호, 김태식(2016)은 픽스호크 및 ROS와 같은 오픈소스를 활용하여 국내 및 해외 시장 진출을 시도해야 함을 시사하고 있다.

협업 시스템에 사용된 무인 수상정은 무인기들의 모션의 개념으로 정찰 임무를 수행하며, 무인기들은 무인 수상정의 위치를 중심으로 사각형 대형으로 유지, 불법 어선 포착 임무가 발생하면 선회를 통해 임무를 수행한다. 무인 수상정은 픽스호크 및 라즈베리파이(Raspberry pi)를 탑재하여 하드웨어를 구성하였고, 무인기는 Gazebo 시뮬레이션 프로그램을 이용하여 구성하였다. 이들은 MAVLink를 ROS 패키지화한 MAVROS를 이용하여 통신하며, 임무 알고리즘은 직접적인 패키지 구성을 통해 각 무인기들에 적용하였다. 또한 MATLAB/SIMULINK를 이용하여 임무의 전체

적인 시나리오를 구성하였으며, 실제 운항 실험을 통하여 구성된 시스템의 실현 가능성을 검토하였다.

제시된 협업 시스템은 불법 어선 감시뿐만 아니라, 다수 무인기를 이용한 정찰, 감시 및 재난 현장에서의 무인기와 무인 차량의 협업 시스템으로도 확장될 수 있다.

II. 본 론

2.1 무인 수상정 시스템

2.1.1 하드웨어 구성

본 연구에서는 취미용 무선 조종 낚시보트를 개량하여 무인화 하였다. 선체 후방에는 방향 조절을 위한 러더 및 추력 조절을 위한 스톱들이 장착되었으며, 제원은 Table 1과 같다.

Table 1. Boat specification

List	Specification
Length	1,000 mm
Width	350 mm
Height	320 mm
Weight	13 kg
Motor	450 w BLDC motor

시스템 구성을 위하여 Fig. 1의 좌측 그림과 같이 전방에 픽스호크 2.1과 호환되는 RTK Here GNSS 센서를 부착하였고, 수동 조작을 위한 RC 수신기를 선체 중앙부에 부착하였다. 선체 후방부에는 ROS 통신을 위한 WIFI 수신기

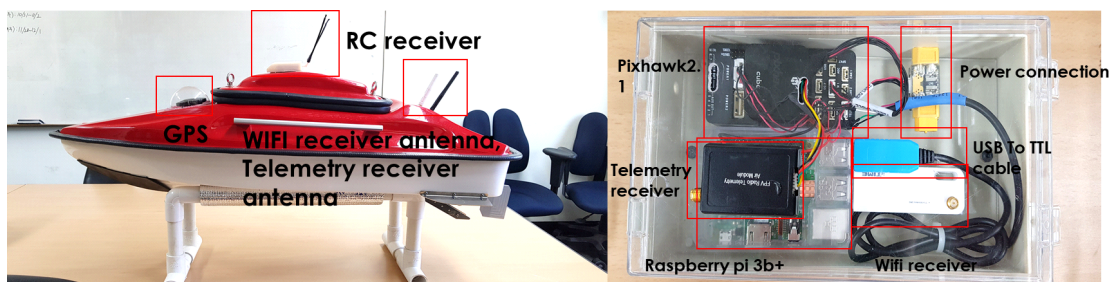


Fig. 1. USV system configuration

및 ROS 통신이 단절되는 경우를 대비하여 픽스호크와 직접 통신이 가능한 433 MHz 대역의 텔레메트리 수신기를 부착하였다.

Fig. 1의 우측 그림은 무인 수상정의 내부 시스템 구성도이다. 방수 엔클로저 내부에 주 프로세서인 픽스호크 2.1 및 라즈베리파이 3b+가 USB To TTL 케이블을 이용하여 연결되어 있다. 전원은 배터리에서 파워 모듈을 통하여 5V 정격 전압으로 픽스호크 및 라즈베리파이에 각각 공급된다. 또한 라즈베리파이의 기본 WIFI 모듈은 송수신 거리 한계를 고려하여 이를 확장하기 위한 별도의 WIFI 수신기를 연결하였다.

2.1.2 픽스호크와 라즈베리파이를 이용한 제어 시스템 구성

본 연구에서는 Companion Computer로 라즈베리파이를 사용하였다. 일반적으로 라즈비안 운영체제를 주로 사용하나, ROS를 구동하기 위하여 Ubuntu Mate를 사용하여 ROS Kinetic 버전을 구동하였다. 본 연구에서는 라즈베리파이 3b+를 사용하여 픽스호크와 USB To TTL 케이블을 이용한 시리얼 통신을 사용하였다. 또한 지상통제시스템으로는 Mission Planner를 사용하였고, 라즈베리파이의 WIFI Module을 이용하여 UDP 통신으로 지상통제시스템과 통신하였다. 무인 수상정 시스템의 구성도는 Fig. 2와 같다.

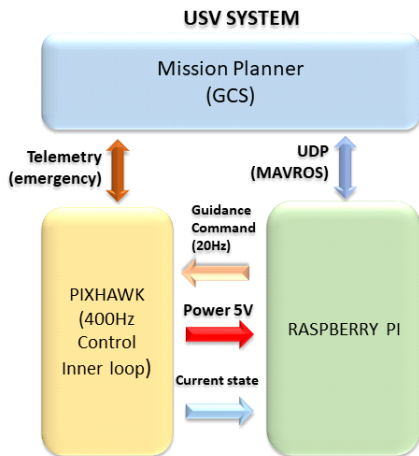


Fig. 2. USV system diagram

Fig. 3은 무인 수상정에서 ROS 통신을 구현하기 위한 launch 파일의 스크립트 내용이다. 사용자는 fcu_url 및 gcs_url 부분을 설정하게 되는데, fcu_url에는 픽스호크와 라즈베리파이가 Serial 통신을 하므로 해당 주소를 입력하고, gcs_url에는 지상통제시스템을 사용할 컴퓨터의 IP주소 및 UDP 포트를 지정한 후 launch 파일을 실행하게 되면 ROS통신이 시작된다.

```
<launch>
<!-- vim: set ft=xml noet : -->
<!-- example launch script for ArduPilot based FCU's -->

<arg name="fcu_url" default="/dev/ttyACM0:57600" />
<arg name="gcs_url" default="" />
<arg name="tgt_system" default="1" />
<arg name="tgt_component" default="1" />
<arg name="log_output" default="screen" />
<arg name="fcu_protocol" default="v2.0" />
<arg name="respawn_mavros" default="false" />

<include file="$(find mavros)/launch/node.launch">
<arg name="pluginlists_yaml" value="$(find mavros)/launch/apm_pluginlists.yaml" />
<arg name="config_yaml" value="$(find mavros)/launch/apm_config.yaml" />

<arg name="fcu_url" value="$(arg fcu_url)" />
<arg name="gcs_url" value="$(arg gcs_url)" />
<arg name="tgt_system" value="$(arg tgt_system)" />
<arg name="tgt_component" value="$(arg tgt_component)" />
<arg name="log_output" value="$(arg log_output)" />
<arg name="fcu_protocol" value="$(arg fcu_protocol)" />
<arg name="respawn_mavros" default="$(arg respawn_mavros)" />
</include>
</launch>
```

Fig. 3. Script contents of the launch file for ROS communication in the USV

2.2 무인기 시스템

2.2.1 시뮬레이션 프로그램을 이용한 무인기 시스템 구성

픽스호크는 ROS 및 Gazebo를 지원하기 때문에 실제 실험 전에 펌웨어를 시뮬레이션하여 활용 가능성에 대한 검증이 가능하다. 픽스호크는 멀티콥터 드론, 일반 고정익 무인기, 표준 VTOL, Tailsitter VTOL, Rover 등을 지원할 수 있으며, 본 연구에서는 멀티콥터 드론 모델을 사용하였다.

본 연구에서 활용한 무인기는 3DR Robotics社의 iris 모델이다. 총 4대의 무인기를 활용하였고, launch 파일을 이용하여 초기 위치, 자세, 시뮬레이션 프로그램과의 UDP 통신 포트, 지상 통제시스템과의 UDP 통신 포트 및 함께 실행시킬 패키지 등을 설정하였다. 설정에 활용된 일부 파일 스크립트의 내용은 Fig. 4와 같다.

```
<?xml version="1.0"?>
<launch>
  <!-- MAVROS posix SITL environment
  launch script -->
  <!-- launches Gazebo environment and 2x:
  MAVROS, PX4 SITL, and spawns vehicle -->
  <!-- vehicle model and world -->
  <arg name="est" default="ekf2"/>
  <arg name="vehicle" default="iris"/>
  <arg name="world" default="$(find
  mavlink_sitl_gazebo)/worlds/empty.world"/>
  <!-- gazebo configs -->
  <arg name="gui" default="true"/>
  <arg name="debug" default="false"/>
  <arg name="verbose" default="false"/>
  <arg name="paused" default="false"/>
  <!-- Gazebo sim -->
  <include file="$(find
  gazebo_ros)/launch/empty_world.launch">
    <arg name="gui" value="$(arg gui)"/>
    <arg name="world_name" value="$(arg
  world)"/>
    <arg name="debug" value="$(arg
  debug)"/>
```

```
<arg name="verbose" value="$(arg
  verbose)"/>
  <arg name="paused" value="$(arg
  paused)"/>
</include>
<!-- UAV1 -->
<group ns="uav1">
  <!-- MAVROS and vehicle configs -->
  <arg name="ID" value="1"/>
  <arg name="fcu_url"
  default="udp://14541@localhost:14581"/>
  <!-- PX4 SITL and vehicle spawn -->
  <include file="$(find
  px4)/launch/single_vehicle_spawn.launch">
    <arg name="x" value="0"/>
    <arg name="y" value="0"/>
    <arg name="z" value="0"/>
    <arg name="R" value="0"/>
    <arg name="P" value="0"/>
    <arg name="Y" value="0"/>
    <arg name="vehicle" value="$(arg
  vehicle)"/>
    <arg name="mavlink_udp_port"
  value="14561"/>
    <arg name="mavlink_tcp_port"
  value="4561"/>
    <arg name="ID" value="$(arg ID)"/>
  </include>
  <!-- MAVROS -->
  <include file="$(find
  mavros)/launch/px4.launch">
    <arg name="fcu_url" value="$(arg
  fcu_url)"/>
    <arg name="gcs_url" value="" />
    <arg name="tgt_system" value="$(eval
  1 + arg('ID'))"/>
    <arg name="tgt_component"
  value="1"/>
  </include>
</group>
```

Fig. 4. Part of UAV launch file script

Fig. 4의 파일을 통해 설정이 완료된 무인기들에 대해 파일을 실행시키면 Fig. 5와 같이 Gazebo에서 실행된다.

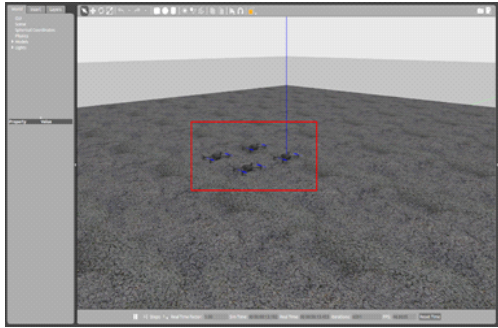


Fig. 5. Gazebo with 4 UAVs

2.3 ROS를 이용한 무인 수상정 및 무인기 간 협업 시스템

본 연구에서 사용되는 협업 시스템에서 무인 수상정은 무인기들의 모션 개념으로 운항하며, 무인 수상정의 위치를 중심으로 4대의 무인기들이 사각형의 대형을 유지, 불법 어선 감시 임무를 수행하였다.

2.3.1 ROS

협업 시스템은 라즈베리파이에 구축된 ROS를 이용하여 각 무인기, 무인 수상정을 패키지로 구성하였다. 이때 ROS의 구성 요소인 노드, 토픽, 메시지를 이용하여 무인 이동체간 통신을 한다. 노드는 프로세서 중 실행 가능한 최소단위의 용어이다. 이 노드들은 메시지를 기반으로 통신하게 된다. 패키지는 하나 이상의 노드들을 실행시키기 위한 도구이며, 여러 노드들을 한번에 실행시키기 위하여 노드들을 하나의 패키지로 구성한다. 또한 메시지는 정수, 부동 소수점과 같은 변수 형태로서 노드 간의 통신을 위해 설정되는 요소이다. 또한 토픽은 ROS에서 통신할 때 사용되는 메시지 통신의 방법을 일컫는다. 이때 송신부를 Publisher, 수신부를 Subscriber라 한다. 이에 대한 간략한 설명은 Table 2와 같다.

Table 2. ROS components

Node	Package
Minimum available unit	Tools to run one or more nodes
Message	Topic
Variables set for communication between nodes	A method of message communication used in ROS

2.3.2 MAVROS

MAVROS는 MAVLink에서 ROS 통신을 위하여 MAVLink를 ROS 패키지로화한 것이다. MAVROS 패키지는 MAVLink를 사용하는 다양한 시스템들에 대하여 통신 드라이버를 제공하며, UDP 통신을 통하여 지상통제 시스템에서도 사용이 가능하다.

본 연구에서는 Fig. 6과 같이 무인 수상정 및 무인기들 간 MAVROS 통신을 사용하였다. MAVROS는 여러가지 토픽을 제공하는데, 사용자가 시스템 구성에 필요한 토픽들을 송신 및 수신한다. 본 연구에서 주로 사용된 토픽은 clock, local_position/pose, set_position/local 등이 있다.



Fig. 6. Real-simulation communication

이는 실제 시스템과 시뮬레이션 간의 연동이 가능하게 해주는 부분으로, Fig. 6과 같이 local_position/pose는 기체가 운항을 위하여 아밍(arming)된 부분을 기준점인 (0, 0, 0)좌표에 해당하는 홈 포인트(Home Point)로 설정하여 NED 좌표계를 기준으로 하여 상대적 거리를 나타내는 토픽이다. 이를 이용하여 무인 수상정의 홈 포인트를 이용, 무인 수상정과 무인기들이 한 공간에 있는 것으로 모사한다.

2.3.3 MATLAB / SIMULINK를 이용한 사용자 정의 토픽 구성

본 연구에서는 MAVROS가 제공하는 기본 토픽 이외에 MATLAB/SIMULINK를 이용하여 사용자 정의 토픽을 구성하였다. 이 토픽은 불법 어선감시 임무 모사를 위해 구성된 토픽이며, 목표물 감지 여부를 알리는 Mission_chk 토픽, 감지된 목표물의 위험도를 나타내는 Target_risk 토픽, 무인기에서 임무 수행 시 타 무인기로 도움을 요청할 시 사용되는 UAVx_help 토픽을 구성하였고, 이에 대한 설명은 Table 3와 같으며 Fig. 7과 같이 구성되었다.

Table 3. Mission usage parameter

Parameter	Contents	Value
Mission_chk	Target detection	0 or 1
Target_risk	The risk of detected targets	0~2
UAVx_help	Whether to request for help from a UAV	0 or 1

Fig. 7에서 좌측 부분은 모드 및 전체 시나리오의 신호를 조정하는 블록으로써 모드는 임무를 수행하기 위한 임무 모드 및 임무를 수행하지 않고 운항을 하는 일반 모드로 구성되어 있다. 우측 부분은 각 토픽에 대한 블록이며, 토픽의 타입 및 송신하는 부분으로 구성되어 있다.

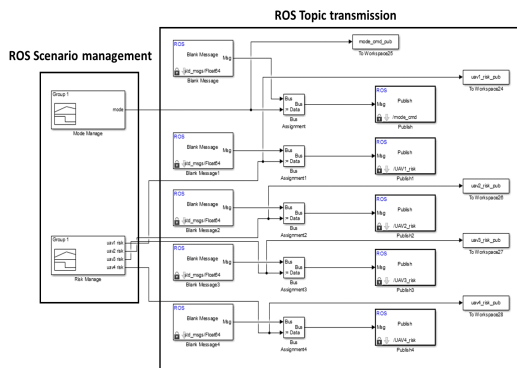


Fig. 7. Construction of user defined topics using MATLAB/SIMULINK

2.3.4 토픽의 송수신 시뮬레이션

토픽들은 라즈베리파이에서 형성된 ROS 네트워크를 통해 송수신되며, 송신된 데이터와 수신된 데이터를 비교하여 정상적인 송수신 여부를 판별하였다. 시뮬레이션 결과는 Fig. 8과 같고, 토픽의 송수신은 ROS 네트워크를 통해 정상적으로 이루어짐을 확인할 수 있다. Fig. 8에서 publish는 송신을, subscribe는 수신을 의미한다.

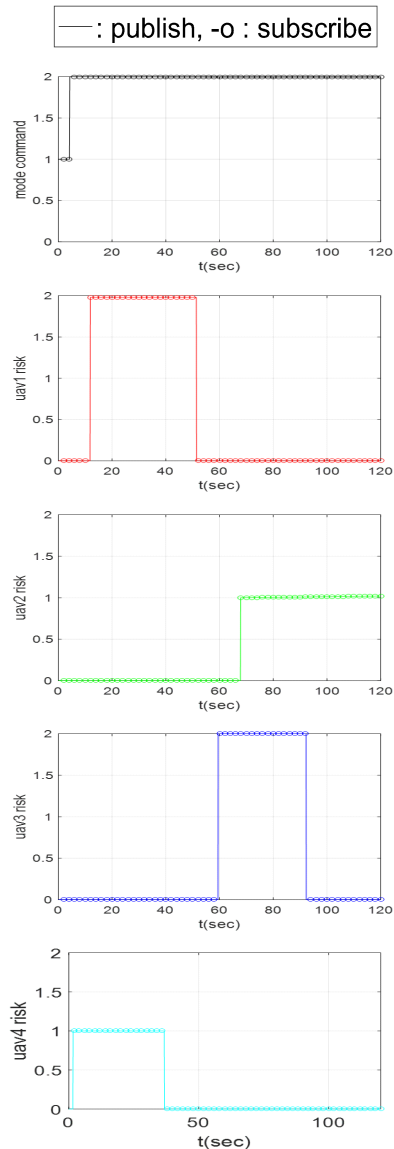


Fig. 8. The result of user defined topic transmission/reception simulation

Fig. 8을 통해 송신 Data와 수신 Data의 값이 정확히 일치함을 확인할 수 있었다.

2.3.5 임무 알고리즘 구성

본 연구에서 사용된 임무 알고리즘은 Fig. 9와 같고, 기본 편대는 각 무인기간 일정 간격을 유지하며, Fig. 10과 같이 사각형의 대형을 유지한다. Fig. 7에서 구성된 모드 변경 토픽에서 값이 1이 되면 임무를 수행하지 않고 기본 편대를 유지하고 운항만 수행하는 운항모드로 운항하며, 값이 2로 변환되면 임무를 수행하는 임무 모드로 전환된다. 이때 목표물이 포착되면 해당 목표물에 위험도에 따른 임무 수행 여부를 결정한다.

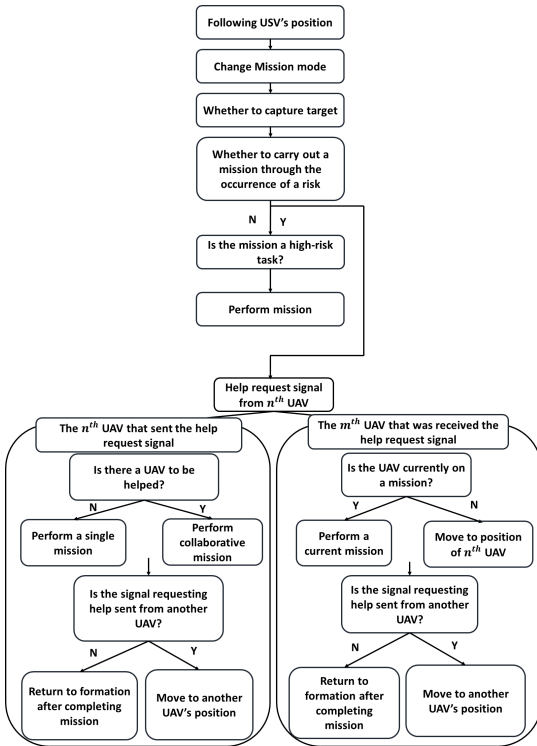


Fig. 9. Mission algorithm

목표물의 위험도가 0으로써 불법어선이 아닌 일반어선 또는 부유물로 판명되면 임무를 수행하지 않고 기본 편대를 유지하며 위험도가 1이면 규모가 크지 않은 불법어선으로 간주하여 타 무인기에 도움 요청을 하지 않고 단독 임무를

수행한다. 목표물의 위험도가 2가 되면 대규모 불법어선으로 인지하고, 타 무인기에 도움 요청 신호를 송신하게 되고, 현재 임무를 수행 중이지 않은 무인기들은 도움 요청 신호에 따라 요청을 송신한 무인기의 위치 이동하여, 협업 임무를 수행한다.

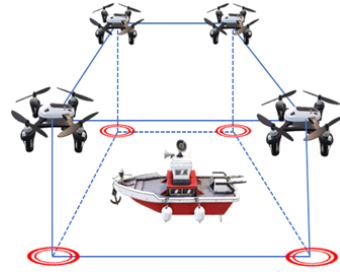


Fig. 10. Basic formation

본 연구에서는 총 4대의 무인기를 설정하여 임무수행 지점은 0~4개가 발생할 수 있다고 가정하였다.

2.4 협업 시스템 실험 및 결과

2.4.1 시나리오 설정

전체 시나리오 설정은 운항 실험 시작 전 Fig. 7의 왼쪽 블록을 통해 Fig. 11 및 12와 같이 설정하였다. 총 실험 시간은 800sec로 설정하였고, 전체 시나리오에 대한 모드 명령 및 각 무인기의 위험도 설정은 Fig. 11 및 12와 같다.

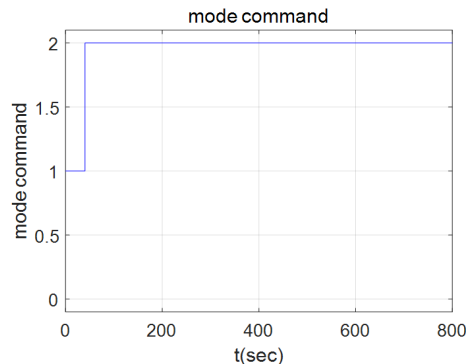


Fig. 11. Full scenario mode command settings

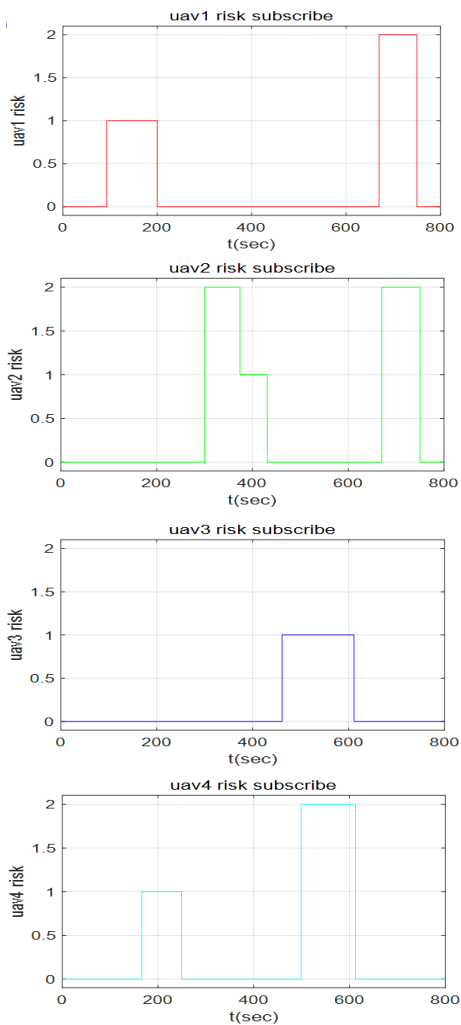


Fig. 12. UAVs risk setting of all scenarios

운항 실험은 부산에 위치한 화명 생태공원 요트 계류장에서 수행하였으며, 초기 수동 운항에서 자동 운항으로 변경되는 시점부터 시간, 무인기 및 무인 수상정의 위치, 위험도 값들을 측정하였다.

Fig. 13은 운항 실험에서 무인 수상정의 자동 운항 중의 모습을, Fig. 14는 무인기들의 임무 수행 모습을 나타낸다. (a)는 시작시의 시점이며, (b)는 임무 중인 상황, (c)는 또 다른 임무를 수행할 때, (d)는 임무 수행을 마치고 복귀할 때의 시점을 나타낸다.

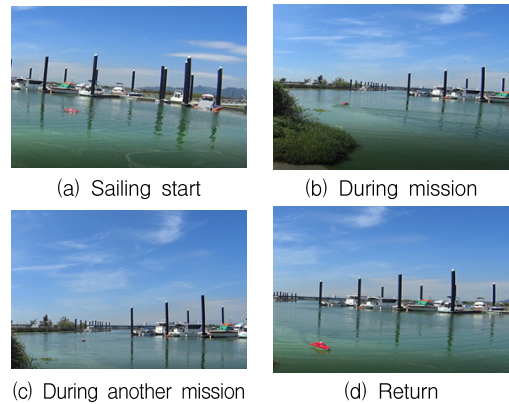


Fig. 13. Sailing experiment

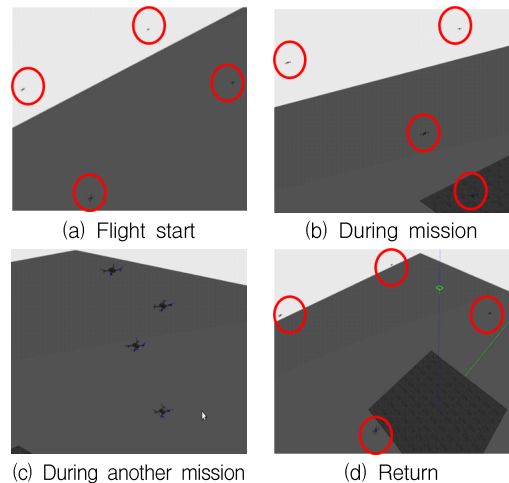


Fig. 14. Gazebo simulation

2.4.1.1 무인기 단독 임무 수행

기 설정된 시나리오에 의거하여 4대의 무인기들이 기본 편대를 구성하며, 무인 수상정의 위치를 추종하던 중 약 90sec에서 (15,-15) 부근 위치에서 1번 무인기에 일반 임무가 발생하였다. 고위험 임무가 아니었으므로 타 무인기에 도움을 요청하지 않았고, 단독 임무를 수행하였으며, 200sec에서 위험도가 0으로 전환되었으므로 다시 기본편대로 복귀하여 무인 수상정의 위치를 추종하였다. 또한 약 170sec에서 (-15, 20) 부근 위치에서 4번 무인기에서 일반 임무가 발생하였고, 역시 단독으로 임무를 수행하였으며, 250sec에서 위험도가 0으로 전환되어 4번 무인기 또한

기본 편대로 복귀하여 무인 수상정 위치를 추종하였다. Fig. 15는 위 실험 동안 무인기와 무인 수상정의 궤적을 도식한 것이며, 주어진 위험상황에 따른 임무를 수행하였음을 확인할 수 있다. Fig. 16은 전체 시나리오의 위험도 중 단독 임무 수행 시 해당 구간의 무인기간 위험도 수준을 그래프화 한 내용이다. Fig. 17은 무인 수상정과 무인기들 사이의 상대 고도 그래프이며, 무인기들이 10m로 이륙 후 무인 수상정과 7m 간격으로 일정한 고도를 유지함을 확인하였다. 또한 Fig. 18에서는 각각 무인기와 무인 수상정의 상대 위치를 그래프화 한 내용이다. 임무를 수행하지 않는 무인기는 15m의 일정한 상대거리를 유지하고 있음을 확인할 수 있었다.

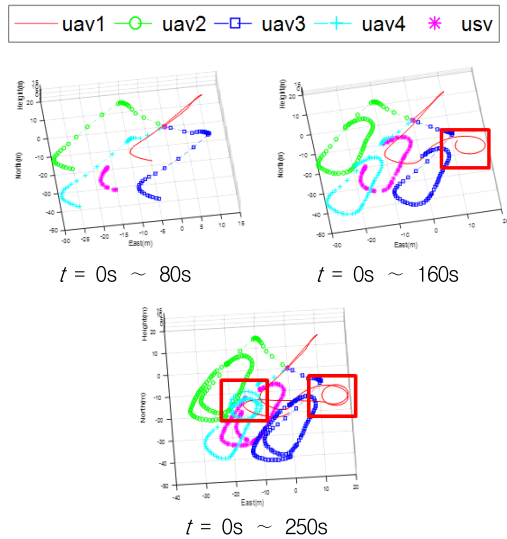


Fig. 15. Trajectory when performing single mission

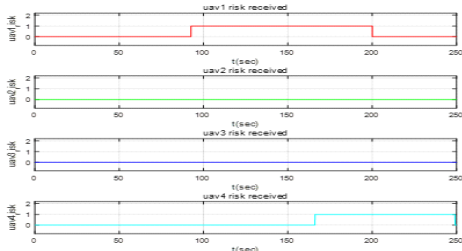


Fig. 16. A graph of the risk between UAVs when performing single mission

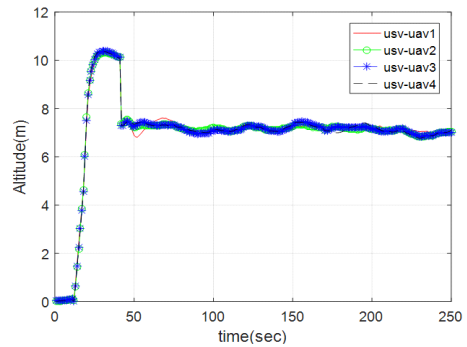


Fig. 17. A graph of the altitude between each UAV and USV (case 1)

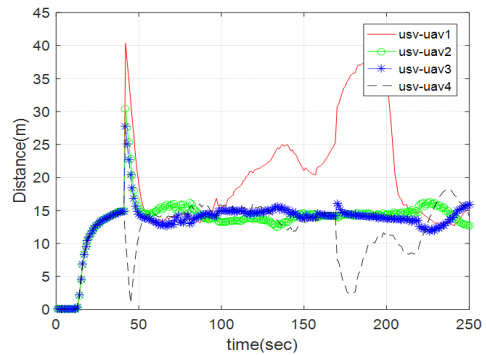


Fig. 18. A graph of the distance between each UAV and USV (case 1)

2.4.1.2 단일 무인기 고위험에서 일반 임무 전환 발생

실험 개시 후 300sec 시점에서 (3,3) 부근에 2번 무인기에서 위험도 2 상황이 부여되었다. 2번 무인기는 고위험 임무가 발생하였으므로 타 무인기에 도움 요청 신호를 1로 송신하여 도움을 요청하였다. 타 무인기들은 도움 요청 신호를 수신 받게 되고, 현재 임무를 수행 중이지 않은 1, 3, 4번 무인기는 2번 무인기의 위치로 이동하여 협업 임무를 수행하였다. 이후 370sec 부근에서 2번 무인기의 위험도가 1로 전환되어 고위험 임무가 아닌 일반 임무로 전환하였고, 2번 무인기는 도움 요청신호의 값을 0으로 전환하였다. 이후 1, 3, 4번 무인기는 기본 편대로 복귀하여 무인 수상정의 위치를 추종하였다. Fig. 19 및 20은 위 상황에 대한 궤적 및 위험도 그래프

이며, 각 무인기들은 충돌 회피를 위해 고도를 변화시켰음을 Fig. 21을 통해 확인할 수 있었다. 또한 Fig. 22를 통해 각 무인기들과 무인 수상정의 상대 위치를 통해 370sec 부근에서 2번 무인기의 위험도가 1이 되었을 때 나머지 무인기들은 상대 거리를 15m로 유지하는 것을 확인할 수 있었다.

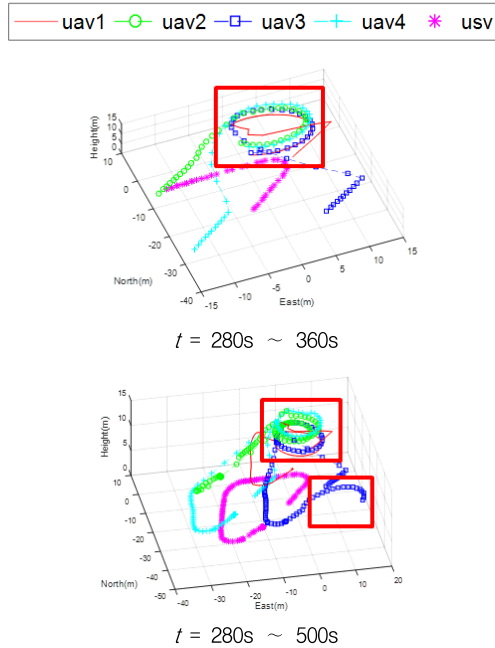


Fig. 19. The trajectory in case of a mission change from high-risk mission to general mission

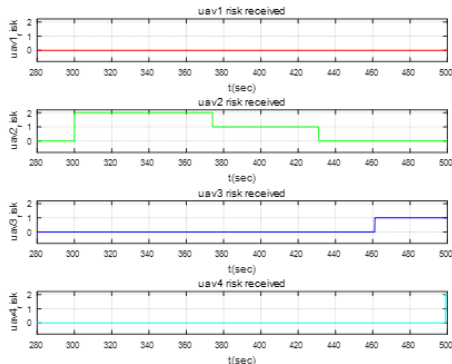


Fig. 20. A graph of the risk between UAVs when performing mission change case

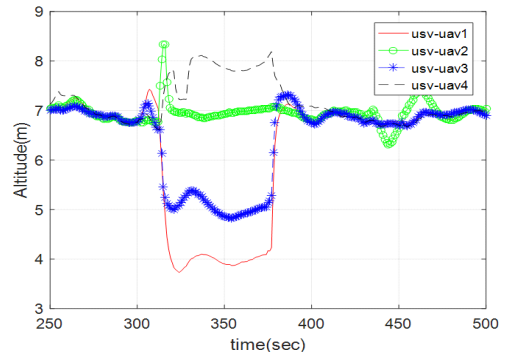


Fig. 21. A graph of the altitude between each UAV and USV (case 2)

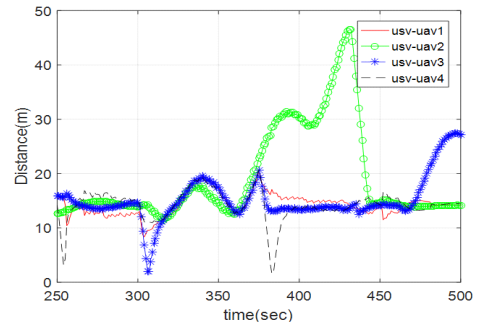


Fig. 22. A graph of the distance between each UAV and USV (case 2)

2.4.1.3 임무 수행 중 타 무인기 고위험 임무 발생

이 경우는 3번 무인기에서 460sec 시점에 (-15, -40) 부근에서 위험도 1의 일반 임무가 발생, 단독으로 임무 수행 중 500sec시점, (10, -30) 부근에서 4번 무인기에 위험도 2의 고위험 임무가 발생한 경우이다. 이때 3번 무인기는 임무를 수행 중에 있으므로 수행 중인 임무를 지속적으로 수행, 임무를 수행하고 있지 않던 1, 2번 무인기가 4번 무인기 위치로 이동하여 함께 임무를 수행하였다. 620sec 부근에서 3, 4번 무인기의 위험도가 0으로 전환되자 다시 기본 편대로 복귀하여 무인 수상정의 위치를 추종하였다. Fig. 23 및 24에 대한 무인기들의 임무수행 경로와 위험도 수준을 나타내고 있다. 이때 1, 2, 4번 무인기는 충돌을 회피하기 위하여 고도를 변화시켰음을 Fig. 25를

통해 확인할 수 있었다. 또한 Fig. 26에서 620sec 이후부터 기본 대형을 유지하며 15m의 상대 거리를 유지함을 확인할 수 있었다.

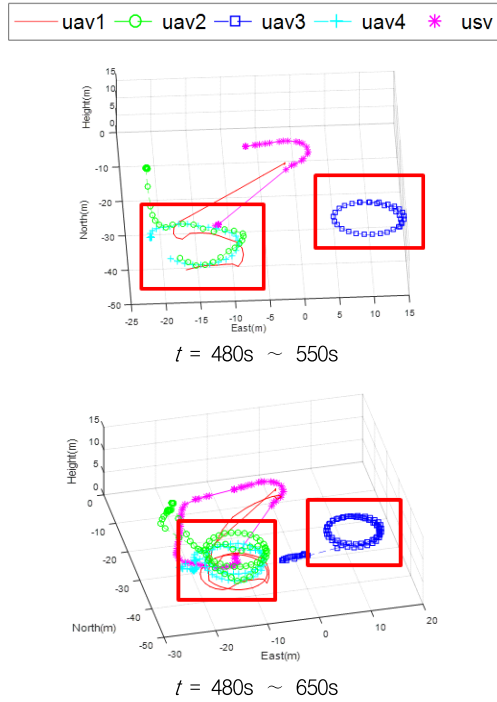


Fig. 23. The trajectory in case high-risk mission during performing mission

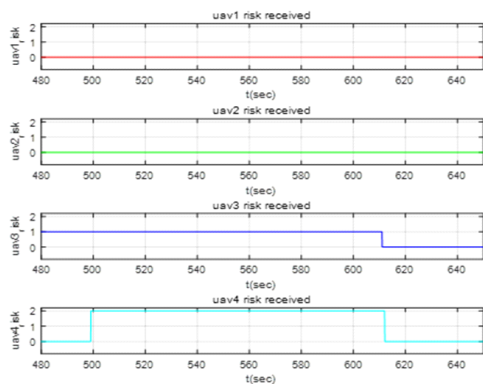


Fig. 24. A graph of the risk between UAVs when a high-risk mission occurs during performing mission

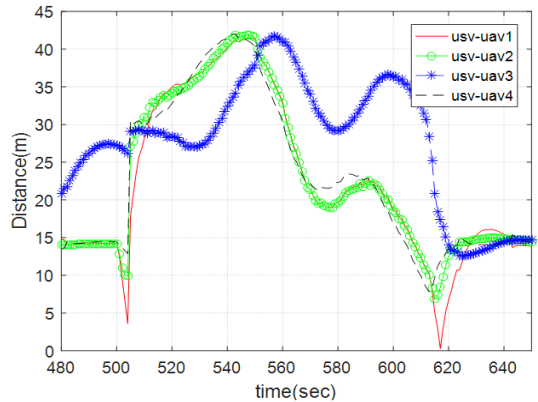


Fig. 26. A graph of the distance between each UAV and USV (case 3)

2.4.1.4 다수 무인기 고위험 임무 발생

이 경우는 1번 무인기와 2번 무인기에서 670sec 시점에서 각각 (0,-10), (20,-10) 부근에서 위험도 2의 고위험 임무가 발생한 경우이다. 1번 무인기와 2번 무인기는 각각 도움 요청신호를 송출하였고, 3번 및 4번 무인기는 1번과 2번 무인기의 도움 요청을 수신, 도움을 요청한 무인기가 2대이므로 각각 다른 무인기의 위치로 이동하여 협업을 수행하였다. 기본 Coupling 관계는 1-2번, 3-4번 무인기로 설정이 되어 있었으나, 이 경우는 1번과 2번 무인기에서 도움을 요청하므로 Coupling 관계를 변화시켜 1-3번 및 2-4번 무인기가 Coupling 관계로 설정되었다. Fig. 27 및 28은 이에 대한 각각의 무인기의 궤적 및 위험도 수준을 나타내고 있다. Fig. 29에서는 각 무인기들과 무인 수상정 사이의 상대 고도를 그래프화한 내용으로 전체적으로 6.8m 기준으로 0.8m 이내로 유지하고 있음을 확인할 수 있었고, Fig. 30을 통해 약 770sec 이후부터는 임무 수행 후 귀환을 위해 사각 대형을 이루기 위해 15m의 간격을 유지하는 것을 확인할 수 있었다.

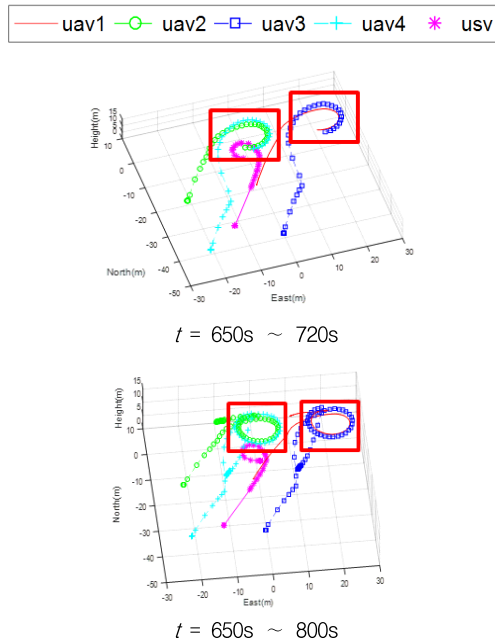


Fig. 27. The trajectory in case of high-risk missions

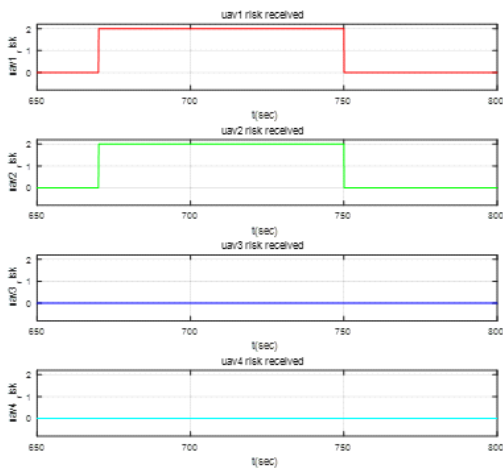


Fig. 28. A graph of the risk between UAVs when performing high-risk missions

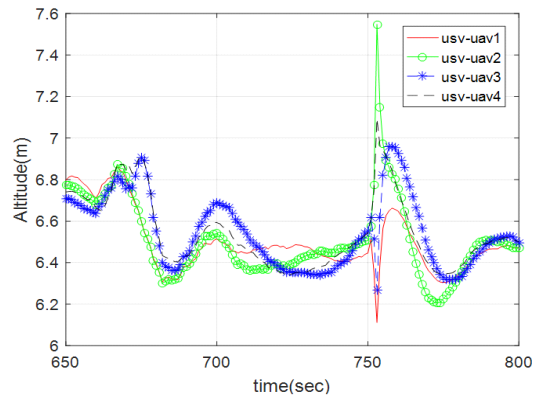


Fig. 29. A graph of the altitude between each UAV and USV (case 4)

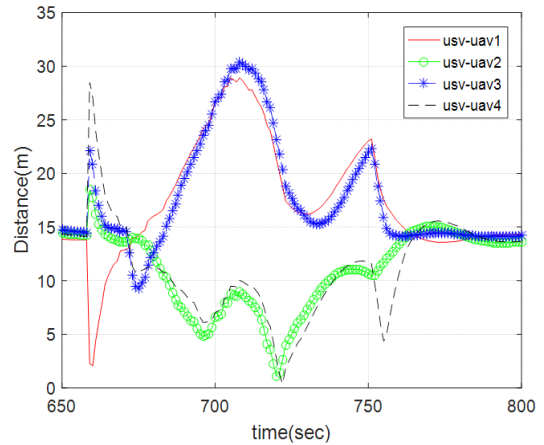


Fig. 30. A graph of the distance between each UAV and USV (case 4)

총 4가지의 경우를 통해 임무를 수행 중이지 않을 경우 기본 사각 대형을 유지하며 무인 수상정과 무인기 사이의 상대거리 15m, 상대 고도 약 3m로 유지하고 있음을 확인하였으며, 이를 통해 시뮬레이션 환경 구축에 문제점이 없음을 확인하였다.

III. 결 론

본 연구에서 ROS 통신을 이용하여 이기종 무인 이동체 간의 통합 시뮬레이션 환경을 구축하고, 픽스호크 및 라즈베리파이를 이용한 시스템

및 시뮬레이션 프로그램을 이용하여 협업 프레임 워크의 실현 가능성을 확인하고자 하였다.

본 연구에서 사용된 하드웨어는 픽스호크 2.1 및 라즈베리파이 3b+이며, 소프트웨어로는 Ardupilot 및 ROS를 사용하였다. 무인기 시스템은 Gazebo 시뮬레이터를 이용하여 구성하였고, PX4 펌웨어와 ROS package를 이용하여 임무 알고리즘을 설계 및 적용하였다.

또한 무인 수상정과 무인기들의 협업 프레임 워크를 구성하기 위하여 MAVROS 패키지를 이용하였으며, 위치 공유 및 무인 수상정 시스템과 시뮬레이터를 연동하기 위한 토픽을 사용하였다. 사용자가 직접 토픽을 생성하고 적용 가능함을 확인하기 위하여 MATLAB/SIMULINK를 이용하여 mode_cmd, Target_risk, UAVx_help topic을 생성, 송수신하여 사전에 설정한 시나리오대로 임무수행이 정상적으로 이루어지는지 검증하였다.

본 연구에서 제시한 통합 시뮬레이션 환경 구축 및 무인 수상정 시스템과 시뮬레이터 무인기들 사이의 토픽 송수신을 통해 실제 환경에서의 ROS 네트워크의 실현 가능성을 검증하고, 추락과 같은 위험도가 높은 무인 이동체는 시뮬레이터를 이용하였다. ROS 네트워크 및 구성된 시스템의 실현 가능성만을 검증하므로 기 연구된 편대 또는 협업 알고리즘이 아닌, 간단한 위치 명령만을 이용하였다. 본 실험을 통해서 ROS 네트워크를 이용하여 위치 명령만을 통해서도 사각형의 편대를 이루고 있으며, 편대를 유지할 시 일정한 상대 거리 및 상대 고도를 유지하며, 임무를 수행하고 있음을 확인할 수 있다. 또한 위치 명령뿐만 아니라, 속도, 가속도 등의 값을 명령 값으로 전달할 수 있으므로 기 연구된 편대 또는 협업 알고리즘을 사용하여 이기종 무인 이동체 간의 협업 시스템 개발 기간을 더욱 앞당길 것으로 판단한다.

References

- [1] Imdoukh, A., et al., "Semi-autonomous indoor firefighting UAV. Advanced Robotics (ICAR)", 2017 18th International Conference on. IEEE, 2017.
- [2] Kim, G., Lee, S., and Park, H., "Design and development of integrated video transmission receiver for aerial photograph using drone". Proceedings of Symposium of the Korean Institute of communications and Information Sciences, 2016, pp.130-131.
- [3] Mogili, Rao, U. M., and Deepak, B. B. V. L., "Review on application of drone systems in precision agriculture." Procedia Computer Science 133, 2018, pp.502-509.
- [4] Bechtsis, D., et al., "Scheduling and control of unmanned ground vehicles for precision farming: A real-time navigation tool." CEUR Workshop Proceedings. 2017.
- [5] Specht, C., Emilian, Ś., and Mariusz, S., "Application of an autonomous/unmanned survey vessel (asv/usv) in bathymetric measurements." Polish Maritime Research, 24(3), 2017, pp.36-44.
- [6] Wernli, and Robert, L., "Low cost uuv's for military applications: Is the technology ready?", Space And Naval Warfare Systems Center San Diego Ca, 2000.
- [7] Han, D., Ryu, H., Kim, J., and Park, B., "Current status of technology for autonomous cooperative operation of multiple-heterogeneous unmanned vehicles." Proceeding of the Korean Society for Aeronautical and Space Sciences (KSAS) conference, 2017.
- [8] Sinisterra, A., Manhar, D., and Nicholas, K., "A USV platform for surface autonomy." OCEANS - Anchorage, 2017. IEEE, 2017.
- [9] Braga, R. G., et al., "UAV swarm control strategies: A case study for leak detection." Advanced Robotics (ICAR), 18th International Conference on. IEEE, 2017.
- [10] López, E., et al., "A multi-sensorial simultaneous localization and mapping (SLAM) system for low-cost micro aerial vehicles in GPS-denied environments." Sensors, 17(4), 2017, p.802.

-
- [11] Portugal, D., Luca, I., and Alessandro, F., "A ROS-based framework for simulation and benchmarking of multi-robot patrolling algorithms." Robot Operating System (ROS). Springer, Cham, 2019, pp.3-28.
- [12] Youn, W. K., Cho, I. H., and Kim, T. S., "Development status of open-architecture based flight control/mission computer for small UAV". Current Industrial and Technological, 2016.
- [13] <https://mavlink.io/en/>