



Received: 2025/11/23  
Revised: 2025/12/03  
Accepted: 2026/02/11  
Published: 2026/03/31

**\*Corresponding Author:**

Woorim Jang

4F, 27, Hwangsaerul-ro 360beon-gil, Bundang-gu,  
Seongnam-si, Gyeonggi-do, Republic of Korea

Tel: +82-31-629-2939

Fax: +82-31-629-2999

E-mail: wr.jang@hanwha.com

# STANAG 4586 기반 무인기 비행경로 제어 알고리즘 설계 및 구현

## Design and Implementation for UAV Flight Path Control Algorithm Based on STANAG 4586

장우림<sup>1\*</sup>, 김신영<sup>2</sup>

<sup>1</sup>한화시스템 SW팀(항공) 선임연구원

<sup>2</sup>한화시스템 SW팀(항공) 전문연구원

Woorim Jang<sup>1\*</sup>, Sinyoung Kim<sup>2</sup>

<sup>1</sup>Engineer, SW Team(Avionics), Hanwha Systems

<sup>2</sup>Senior engineer, SW Team(Avionics), Hanwha Systems

**Abstract**

본 논문에서는 STANAG 4586 표준을 기반으로 무인항공기 비행운용모드 메시지를 정의하고 각 비행운용모드에 따른 무인항공기 비행운용 개념을 설계한다. CUCS로부터 수신받은 비행운용모드 메시지 처리 절차 설계 및 비행운용모드 메시지에 따른 비행운용 알고리즘을 설계하고 구현한다. 마지막으로 시험환경을 통해 구현된 알고리즘에 대한 검증을 수행한다.

This paper defines UAV flight operations mode messages based on the STANAG 4586 standard and designs flight operations concepts for each mode. To verify this, message processing procedures from CUCS simulators are designed, and algorithms for each flight operations mode are designed and implemented. Finally, validation of the implemented algorithm is performed through the test environment.

**Keywords**

무인항공기(UAV), 지상통제시스템(CUCS), 비행운용모드(Flight Operation Mode), 비행경로 제어(Flight Path Control), 비행경로계획(Flight Path Plan)

**Acknowledgement**

이 논문은 2019년 정부(방위사업청)의 재원으로 국방과학연구소의 지원을 받아 수행된 연구임(UC190071JD).

### 1. 서론

21세기 현대전과 민간 응용 분야에서 무인항공기(unmanned aerial vehicle, UAV)의 중요성은 급작스럽게 주목받았지만[1], 최근에는 과학기술, 교통, 통신, 물류, 구조, 항공촬영, 농업 등 다양한 민간 분야로 확대되고 있으며, 앞으로의 미래 항공산업과 시장의 성장을 주도할 것으로 예상된다[2]. 이러한 무인기 시스템의 급속한 발전과 확산에 따라 다양한 제조사와 운용 국가 간의 상호운용성(interoperability) 확보가 중요한 과제로 대두되고 있으며[3], 특히 다국적 군사 작전이나 연합 훈련에서는 서로 다른 국가의 무인기 시스템들이 효과적으로 협력할 수 있는 표준화된 통신 체계와 운용 절차가 필수적이다[4].

이와 같은 배경에서 NATO는 무인기 시스템 간의 상호운용성을 보장하기 위한 표준을 개발하게 되었고[5], STANAG 4586(NATO Standardization Agreement 4586)은 지상통제시스템(core unmanned control system, CUCS)과 무인항공기의 상호운용성을 위한 NATO 표준 인터페이스로 아키텍처, 인터페이스, 통신 프로토콜, 데이터 요소 및 메시지 형식을 정의한다[3,5].

STANAG 4586의 목적은 NATO 국가들이 정의한 운용 개념의 요구사항 충족을 위해 서로 다른 무인항공기, 즉 이기종 무인항공기 간에 필요한 상호운용성 수준(level of interoperability, LOI)을 달성하는데 구현되어야 할 인터페이스를 명시하는 것이다[3]. 따라

서 STANAG 4586은 운용 요구사항을 달성하기 위해 필요한 인터페이스를 명시하고 있으며, 이에 따르면 CUCS에서 다양한 UAV 및 각 UAV에 장착된 임무 장비 그리고 다양한 C4I 시스템과 통신하기 위한 표준 인터페이스가 구현되어야 한다[4,6].

반면, STANAG 4586에서는 무인기 시스템의 상호 운용성을 위한 포괄적인 프레임워크 제공 및 인터페이스의 구조, 메시지 형식, 데이터 요소 등을 정의하고 있으나, 구체적인 알고리즘 설계, 실제 비행 및 임무 수행을 위한 구체적인 구현과 기능 검증은 개발자들에게 맡기고 있는 실정이다[3,7].

본 논문에서는 STANAG 4586 표준을 기반으로 무인항공기 비행운용모드 메시지를 정의하고 각 비행운용모드에 따른 무인항공기 비행운용 개념을 설계한다. CUCS를 모사한 mini-GCS로부터 수신받은 비행운용모드 메시지 처리 절차를 설계하고, 각 비행운용모드 메시지에 따른 비행운용 알고리즘을 설계하고 구현한다. 마지막으로 무인항공기에 장입된 비행경로계획을 활용하여[8], mini-GCS를 통해 각 비행운용모드를 송신하여 무인기가 비행운용모드 메시지 처리 및 각 비행운용모드에 따른 비행경로 제어를 수행하도록 하고, 해당 비행 정보를 mini-GCS에게 송신하여 mini-GCS를 통해 기능에 대한 검증을 수행한다.

## 2. 무인기 비행경로 제어 알고리즘 설계 및 구현

STANAG 4586에서는 사전에 CUCS에서 임무계획을 작성하도록 함으로써 각 비행경로에 속해 있는 항로점들이나 해당 비행경로에 대한 비상착륙 계획, 임무 계획 등의 여러 가지 속성을 부여할 수 있고, 임무 지역에 다양한 비행경로들과 항로점들을 구성하여 무인항공기에 임무 장입시키면 작전 운용 등에 따라 여러 비행경로를 유연하게 이동하거나 운용할 수 있는 장점이 있다[9]. 이렇게 장입된 비행경로계획은 #2016(Vehicle Operating Mode Command) 메시지를 통해 무인항공기의 비행경로 이동, 항로점 이동, 특수 임무 비행 등을 제어하는 방식으로 활용할 수 있다[10].

최근 연구에서는 CUCS로부터 임무 장입 데이터를 수신받고, 무인항공기의 FLCC가 수신받은 임무 장입 데이터를 통해 CUCS에서 작성된 비행경로계획

을 정렬하여 관리하는 알고리즘을 다룬 바 있다[8].

본 논문에서는 FLCC 내에 정렬된 비행경로계획 데이터를 이용하여 #2016 메시지 등과 같은 비행경로를 제어하는 비행운용 관련 메시지들을 정의하고 메시지 처리 및 비행경로를 제어하는 알고리즘을 제안한다.

### 2.1 STANAG 4586 기반 비행운용 개념 설명

본 논문에서는 STANAG 4586 Edition 4 Volume II를 기준으로 비행운용 개념에 대해 설명한다[10].

#### 2.1.1 비행운용 기본 개념

무인기를 운용하기 위해서는 무인기의 수행 가능 임무 및 비행 특성에 맞는 비행경로계획 작성이 필요하며, 이는 CUCS에서 수행한다.

STANAG 4586에서는 비행경로계획 구성요소로 임무(mission), 비행경로(route), 항로점(waypoint)을 정의하고, 이와 관련된 데이터 필드들을 프로토콜에서 명시하고 있다. 임무는 여러 개로 정의할 수 있지만 무인기에는 하나만 장입이 가능하고 1개의 임무는 다수의 비행경로로 구성된다. 비행경로는 다수의 항로점으로 구성되며, 항로점은 비행경로계획 구성 요소 중에서 가장 작은 개념에 속한다.

비행경로계획 구성요소의 개념과 세부적인 특징은 Fig. 1 및 Table 1과 같다[8]. 위 기본 개념들을 토대로 작성된 비행경로계획은 정해진 절차를 통해 무인기에 임무 장입되며, 이 임무 데이터를 기반으로 비행경로 제어 모드를 통해 무인기 비행경로를 제어하게 된다. 본 논문에서는 장입된 임무 데이터를 기반으로 수신받은 비행경로 제어 모드를 통한 비행경로 제어 알고리즘을 설계한다.

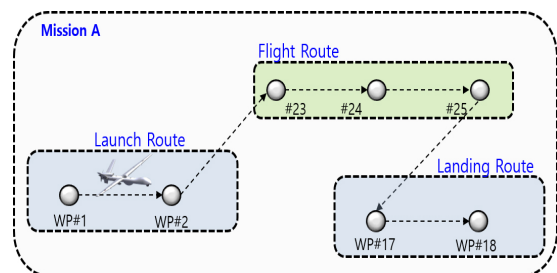


Fig. 1. Concept of the flight path plan

**Table 1.** Characteristics of flight path plan components

Component	Description
Waypoint	<ul style="list-style-type: none"> <li>The point where the UAV follows the flight</li> <li>Configuration definitions: loitering, payload control, vehicle control, vehicle-specific control</li> </ul>
Route	<ul style="list-style-type: none"> <li>Consists of multiple waypoints</li> <li>Flight path characteristics can be defined by five types (Launch, Approach, Flight, Contingency A, Contingency B)</li> <li>If a flight path includes a loop route, a separate loop route must be created.</li> </ul>
Mission	<ul style="list-style-type: none"> <li>Consists of multiple routes</li> <li>Multiple routes can be created, distinguished by the Mission ID text.</li> <li>However, only one mission can be loaded to the UAV.</li> </ul>

**2.2 STANAG 4586 메시지 분석 및 처리**

본 논문에서는 임무 장입이 완료된 상태를 가정하고 비행경로 제어에 대한 알고리즘 설계/구현 내용을 다루므로, 임무 장입에 대한 메시지 분석 내용은 다루지 않으며, 다음과 같이 비행경로 제어를 위해 필요한 메시지들을 정의하고 각 메시지 간 처리 절차를 설계하였다.

**2.3 비행경로 제어 모드 정의**

임무 장입을 완료한 후에는 CUCS를 통해서 자동 이·착륙, 항로점 이동 등에 대한 명령을 UAV로 송신하여 비행을 수행하게 된다. 이때 사용되는 메시지 그룹[10,11]은 Table 2와 같이 구분되며, 비행경로 제어 알고리즘에서 사용되는 메시지는 Table 3와 같이 정리하였다.

**Table 2.** Message group for flight path control mode

Message group	Message ID range
Flight Vehicle Command	2000-2999
Flight Vehicle Status	3000-3999
Flight Vehicle Payload-Relevant	4000-4999
Payload Status Functional	21000-23999

**Table 3.** Messages using flight path control algorithm

Message ID (message name)	Description
#2016 (Vehicle Operating Mode Command)	<ul style="list-style-type: none"> <li>Define flight path control mode, flight path Waypoints, etc.</li> </ul>
#2018 (Slave to Sensor Configuration)	<ul style="list-style-type: none"> <li>Defines the distance from the target, direction, and turning characteristics.</li> <li>Uses this message to track the target and control the flight path.</li> </ul>
#2019 (Slave to Location Configuration)	<ul style="list-style-type: none"> <li>Define enemy location information, distance, direction, turning characteristics, etc.</li> </ul>
#3001 (Vehicle Operating Mode Report)	<ul style="list-style-type: none"> <li>Message to respond to #2016</li> <li>Transmit the received Select Flight Path Control Mode to CUCS</li> </ul>
#3002 (Vehicle Operating States)	<ul style="list-style-type: none"> <li>Message to respond to #2016</li> <li>Transmit the received Commanded Speed and Commanded Waypoint Number to CUCS</li> </ul>
#3016 (Relative Route Waypoint Location Report)	<ul style="list-style-type: none"> <li>Message to respond to #2019</li> <li>Transmit the received LLA values, separation distances, look-down angles, etc. to CUCS</li> </ul>
#4001 (From To Next Waypoint States)	<ul style="list-style-type: none"> <li>Used when transmitting Latitude, Longitude, Altitude, Speed, Time, and Waypoint number for previous, current, and next route points from a drone to CUCS.</li> </ul>
#21800 (Slave To Sensor Settings Report)	<ul style="list-style-type: none"> <li>Message to respond to 2018</li> <li>Used when transmitting received Slave to Sensor Offset/Direction/Look Down Angle, etc. to CUCS</li> </ul>
#17002 (Generic Information Request)	<ul style="list-style-type: none"> <li>Message used when requesting a specific message from CUCS to the UAV.</li> </ul>

**2.4 비행경로 제어 관련 메시지 처리 절차**

위에서 분석한 메시지들을 이용하여 CUCS-UAV 간 메시지 처리 절차를 설계하였다.

Fig. 2에 나타난 바와 같이 CUCS에서 비행경로 제어를 위한 #2016 메시지를 UAV로 송신하고, 이를 수신한 UAV는 #3001, #3002 메시지를 CUCS에게 송신하여 자신이 수신한 비행경로 제어 모드, 비행체 속도 명령 등을 응답하며, CUCS에서 Slave to Sensor 비행경로 제어를 위해 #2018 메시지를 송신하면 이

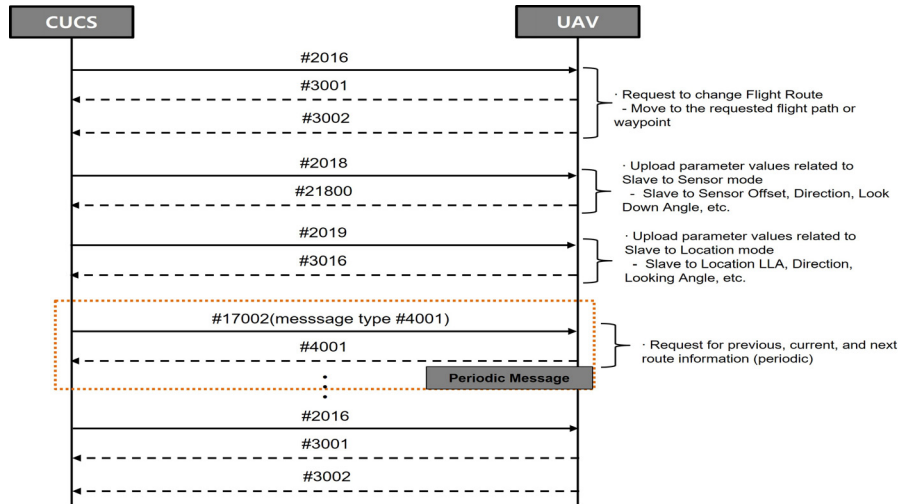


Fig. 2. Message handler procedure for flight path control algorithm

를 수신한 UAV는 #21800 메시지를 CUCS로 송신하여 자신이 수신한 Slave to Sensor offset, direction, look down angle 등을 응답한다.

CUCS에서 Slave to Location 비행경로 제어를 위해 #2019 메시지를 송신하면 UAV는 자신이 수신한 Slave to Location LLA, direction, looking angle 등을 #3016 메시지를 CUCS에게 송신하여 응답한다.

마지막으로, 비행 중인 UAV에 CUCS에서 #17002 메시지를 통해서 주기적으로 #4001 메시지를 요청하여 UAV로부터 현재 비행경로에서의 이전 항로점, 현재 항로점, 다음 항로점 정보를 요청 주기대로 수신 받으며, UAV는 CUCS에게 #17002 메시지를 요청받으면 해당 메시지의 message type 필드 데이터를 확인하여 message type에 해당 메시지를 송신한다.

2.5 비행경로 제어 모드 분석

Fig. 3와 같이 임무 장입된 비행경로계획을 작성해 볼 수 있으며, 이 계획에서는 Launch 비행경로 1개, Flight 비행경로 2개, Contingency A 비행경로 2개, Contingency B 비행경로 2개, Landing(Approach) 비행경로 1개를 정의하였다.

Fig. 3의 항로점 #2를 보면 Launch 비행경로의 마지막 항로점으로 loiter 대기 비행을 수행하는데, 이는 Landing 비행경로, Contingency A/B 비행경로를 제외한 나머지 비행경로의 마지막 항로점에서는 다음 비행경로 제어 모드를 수신할 때까지 대기 비행을 수행함을 의미한다.

이륙 이후 비행경로 제어 모드를 통해 여러 비행경로로 비행체가 이동하게 되는데, waypoint 모드의 경우 해당 모드와 더불어 이동하고자 하는 항로점을 CUCS로부터 수신하면, 수신한 해당 항로점을 시작으로 해당 비행경로에서 연결된 순서대로 이동하게 된다. 단, 이동 중에 다른 항로점으로 이동하는 명령을 수신하게 되면 그 항로점에 해당하는 비행경로로 이동하게 된다.

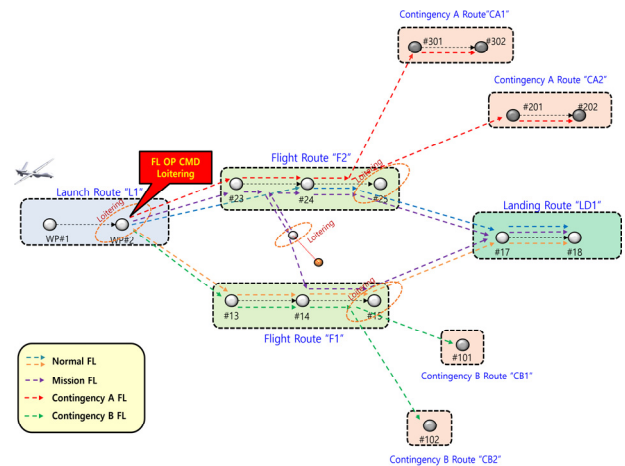


Fig. 3. Flight operation concept for the designed flight path plan

Slave to Sensor와 Slave to Location 모드의 경우, Fig. 3에 나타난 바와 같이 비행경로계획에 포함되지 않은 지점으로 이동하여 비행하게 되며, 특정 비행경로에서 비행 중(#23에서 #24로 이동 중) 해당 모드를 CUCS로부터 수신하게 되면 특정 지점으로 이동하여

Loiter 비행을 수행하며, 비행 도중 다시 비행경로 제어 모드를 CUCS로부터 수신하게 되면 특정 비행경로로 이동(#14로 이동)하게 된다.

Slave to Sensor 모드는 sensor에 종속된 모드로, sensor로부터 이동하는 관측 대상에 대한 위치정보를 주기적으로 수신하여 관측 대상과의 일정 거리에서 Loiter 비행 수행 및 추적을 수행한다.

본 논문에서는 EO/IR 임무 장비에 대해 STANAG 4586 프로토콜 기준으로 Table 4와 같이 Slave to Sensor 연동용 native message를 정의하여 관측 대상의 LLA 정보를 UAV로 송신하도록 하였다. 여기서 native message란 STANAG 4586에 정의되어 있지 않지만 설계 및 구현상 필요하여 자체적으로 정의한 메시지를 의미한다.

Slave to Location 모드는 위치에 종속된 모드로, CUCS로부터 고정된 관측 대상 위치를 수신하여 관측 대상과의 일정 거리에서 Loiter 비행을 수행한다.

Contingency A 모드는 귀환 비행경로 모드로 특정 비상 상황에 주변 활주로나 평지 등에서 비상 착륙을 수행하며, Contingency B 모드는 고장 등의 특정 비상 상황에서 민가 피해 최소화, 적으로부터 은폐 등을 위해 안전지대로 비상 착륙을 수행한다.

마지막으로 Autoland Engage 모드는 착륙 비행경로로 이동하여 착륙을 수행하는 모드이다.

**Table 4.** Field contents of native message for Slave to Sensor interface

Field no.	Filed name	Filed type
00	Presence Vector	Bitmapped
01	Time Stamp	Unsigned
02	STAREPOINT_LATITUDE	Integer
03	STAREPOINT_LONGITUDE	Integer
04	STAREPOINT_HEIGHT_WGS84	Integer

### 2.6 비행경로 제어 알고리즘 설계/구현

비행경로 제어 알고리즘은 CUCS로부터 전달받은 #2016 메시지의 비행경로 제어 모드와 이에 맞는 비행경로(route) 타입을 확인하여 UAV에 장입된 임무(mission) 데이터로부터 해당 비행경로(route) 인덱스 및 항로점 waypoint 인덱스를 탐색하는 알고리

즘이며, 해당 임무 데이터 구조는 Table 5와 같다[8].

**Table 5.** The definition of mission data structure

Structure name	Type	Variable
Mission	uint8	missionID[20]
	uint16	wpNum
	uint8	totalRtNum
	routeDat[totalRtNum]	Unsigned
Route	Route	uint8
	uint8	routeType
	uint16	initWp
	uint8	totalWpNum
	wpDat[totalWpNum]	Integer
Waypoint	Waypoint	uint8
	uint8	nextWp
	:	:

수신한 비행경로 제어 모드에 맞는 비행경로 인덱스와 항로점 인덱스에 대한 탐색이 완료되면 비행경로 인덱스, 항로점 인덱스, 비행경로 타입을 비행유도 제어기로 전달하고, 비행유도제어기에서는 해당 인덱스 정보를 통해 임무 데이터에 접근하여 위치, 속도 등의 데이터를 획득한 후 비행경로 타입을 참고하여 이에 맞는 유도제어를 수행한다.

#2016 메시지의 Select Flight Path Control Mode에 대한 정의는 Table 6과 같으며[10], 아래에서는 비행유도제어기에서 각 모드별로 필요한 데이터를 찾기 위한 탐색·연산 알고리즘을 제안한다.

**Table 6.** Definition of Select Flight Path Control Mode for #2016

Flight path control mode	Value
Waypoint	11
Autoland Engage	19
Launch	21
Slave to Sensor	22
Fly Contingency A	23
Fly Contingency B	24
Slave to Location	25

자동 이륙 모드는 #2016 메시지의 Select Flight Path Control Mode 값이 '21'인 경우 UAV에서는 Launch Mode라고 판단한다. 해당 모드 값을 입력받으면 비행경로 제어 알고리즘에서는 임무 데이터의 비행경로 데이터셋에서 비행경로 타입이 'Launch'인 비행경로를 찾아 해당 인덱스를 저장하며, 이를 비행 유도제어기에 전달한다. 이에 대한 알고리즘은 Fig. 4와 같다.

항로점 모드는 #2016 메시지의 Select Flight Path Control Mode 값이 '11'이면 UAV에서는 Waypoint Mode라고 판단하며, CUCS에서는 해당 모드 제어를 위한 이동할 항로점인 Commanded Waypoint 정보도 포함하여 메시지를 송신한다. 해당 모드 값을 입력받으면 비행경로 제어 알고리즘에서는 임무 데이터의 비행경로 데이터셋으로부터 비행경로 타입이 'Flight'인 비행경로를 찾아 해당 인덱스를 탐색하

고, 해당 비행경로에 Commanded Waypoint가 존재하는지 탐색한다. Commanded Waypoint가 존재하는 항로점 인덱스 탐색이 완료되면, 해당 비행경로 인덱스와 항로점 인덱스를 비행유도제어기에 전달한다. 이에 대한 알고리즘은 Fig. 5와 같다.

STANAG 4586 Edition 4 Volume II에서는 Fig. 7과 Fig. 8과 같이 Slave to Sensor(센서 종속) 모드 및 Slave to Location(위치 종속) 모드에 대해 정의하고 있다.

Slave to Sensor 모드는 해당 모드로 비행하기 위해 Fig. 2와 같이 #2018와 #2019 메시지를 통해 Slave to Sensor 모드와 Slave to Location 모드를 위한 파라미터값을 UAV에 우선 장입해야 하며, 해당 메시지 장입 완료 후 #2016의 Select Flight Path Control Mode를 Slave to Sensor Mode(22) 또는 Slave to Location Mode(25)를 UAV에 송신하여 해당 비행경

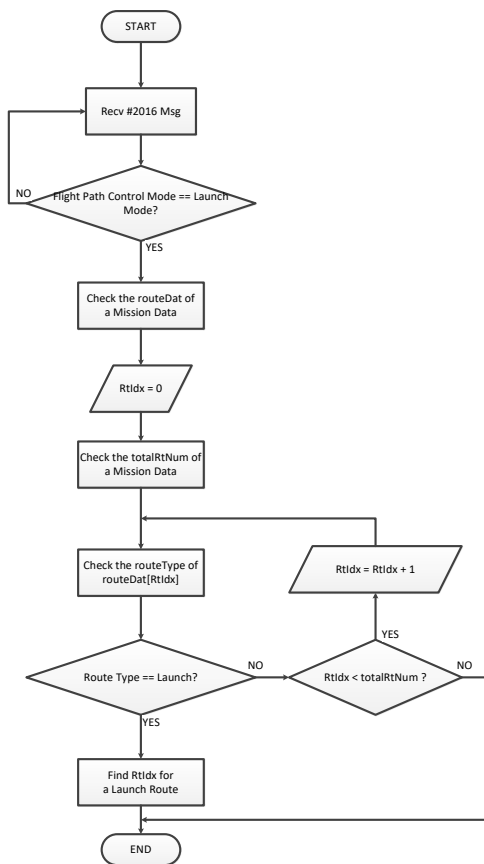


Fig. 4. Flight path control algorithm for Launch Mode

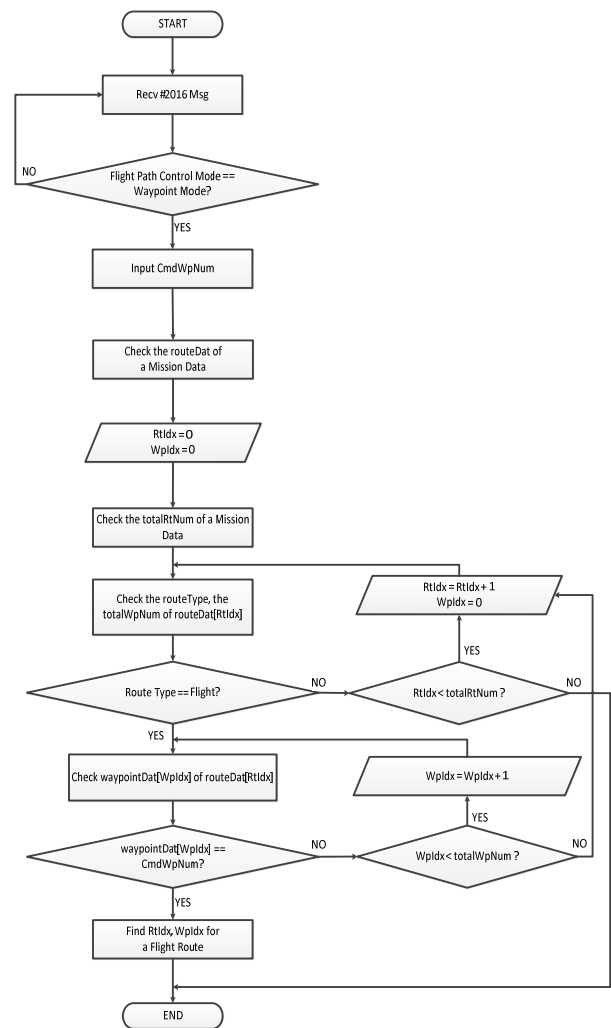


Fig. 5. Flight path control algorithm for Waypoint Mode

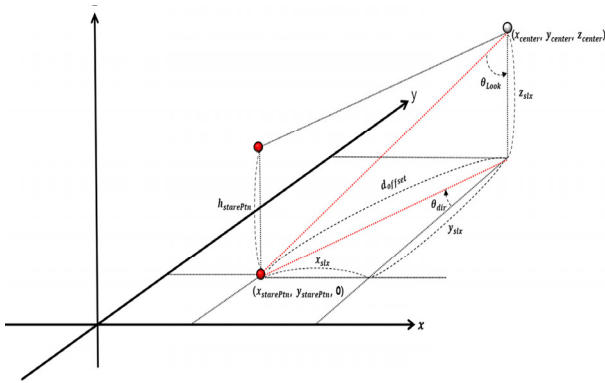


Fig. 6. The simplification for the formulation of Slave to Sensor Mode based on x-y-z coordinate system

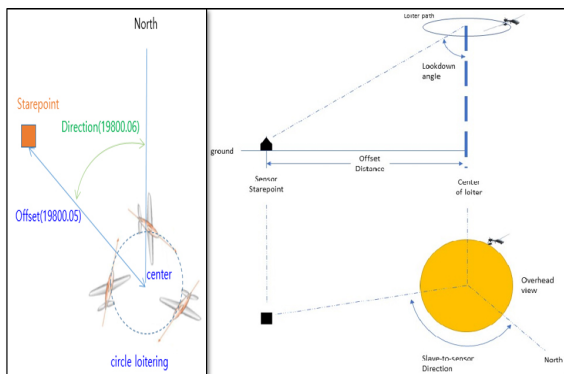


Fig. 7. Flight operation concept for Slave to Sensor mode

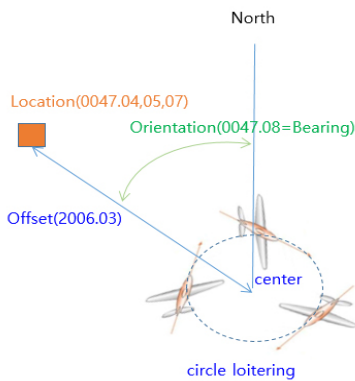


Fig. 8. Flight operation concept for Slave to Location mode

로로 동작하도록 제어한다.

UAV는 EO/IR 센서로부터 타겟의 LLA 위치정보를 일정 주기로 수신하고 #2018 메시지에서부터 받은 Slave to Sensor offset, direction, look down angle 등을 이용하여 UAV의 loiter 비행에 필요한 center point의 LLA 정보를 산출해야 한다. 반면 Slave to Location 모드의 경우 타겟의 위치정보를 전달하는 주체가 CUCS로 변경되어 CUCS로부터 위치정보를 수신할 때 고정 타겟에 대한 위치정보를 받기 때문에

#2019 메시지를 통해 Slave to Location latitude/longitude/altitude, Slave to Location offset, orientation, looking angle 등의 위치정보 데이터가 1번만 UAV에 장입되어 UAV loiter 비행을 위해 필요한 center point의 LLA 값을 산출하게 된다.

Center point 좌표는 Geodetic 좌표계, ECEF 좌표계, NED 좌표계 순으로 변환하여 NED 좌표계 시준으로 산출하였으며, 산출된 좌표값은 역으로 좌표 변환하여 비행유도제어기에는 LLA 값을 전달하였다.

Figs. 7-8은 식 (1) – 식 (6)와 같이 표현할 수 있다. 식 (5)에서는  $\theta_{Look}$ 이  $0^\circ$ 인 경우  $h_{center}$ 가  $d_{offset}$ 이 되며,  $\theta_{Look}$ 이  $90^\circ$ 인 경우에는  $h_{center}$ 는  $h_{starePtn}$ 와 같은 값을 가진다.

- 센서 종속/위치 종속 중심점 좌표 계산:

$$x_{center} = x_{starePtn} + x_{slx} \tag{1}$$

$$y_{center} = y_{starePtn} + y_{slx} \tag{2}$$

$$x_{slx} = d_{offset} \sin(\theta_{dir}) \tag{3}$$

$$y_{slx} = d_{offset} \cos(\theta_{dir}) \tag{4}$$

$$z_{slx} = \frac{d_{offset}}{\tan(\theta_{Look})} \tag{5}$$

$$z_{center} = \begin{cases} z_{slx} & \theta_{Look} < 90^\circ \\ h_{starePtn} & \theta_{Look} = 90^\circ \end{cases} \tag{6}$$

여기서,  $x_{center}$ ,  $y_{center}$ ,  $z_{center}$ 는 UAV center point 좌표값,  $x_{starePtn}$ ,  $y_{starePtn}$ ,  $z_{starePtn}$ 은 타겟의 위치 좌표값,  $d_{offset}$ 은 Slave to Sensor offset(Slave to Location offset),  $\theta_{Dir}$ 은 Slave to Sensor direction(orientation),  $\theta_{Look}$ 은 Slave to Sensor look down angle (Slave to Location looking angle)이다.

EO/IR로부터 주기적으로 수신하는 타겟의 위치값과 식 (1) – 식 (6)을 통해 UAV의 center point를 주기적으로 업데이트하여 비행유도제어기에 전달하며, 선회 비행 수행에 필요한 파라미터값인 loiter radius, loiter duration, commanded speed도 비행유도제어기에 전달한다. 이에 대한 알고리즘은 Figs. 9-10과 같다.

Contingency A는 #2016 메시지의 Select Flight Path Control Mode를 Contingency A Mode(23)로

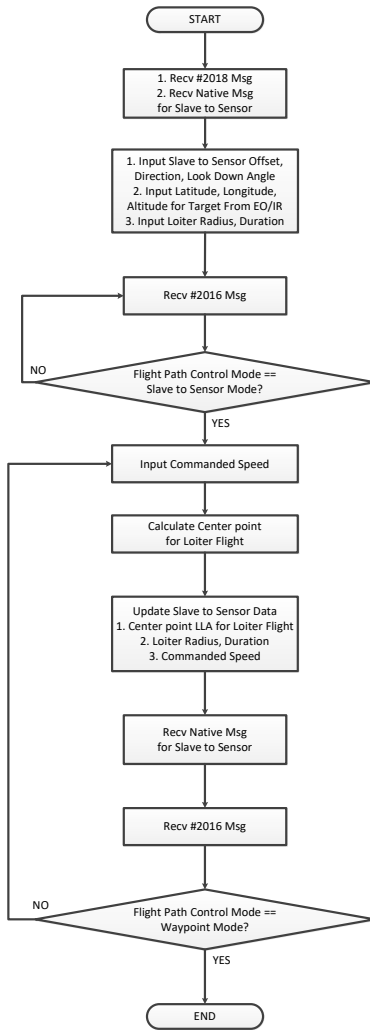


Fig. 9. Flight path control algorithm for Slave to Sensor mode

Contingency B는 Contingency B Mode(24)로 UAV에 장입하며, 비행경로계획에서 여러 비상착륙 포인트들이 존재하므로 CUCS에서는 비행경로의 초기 항로점(initial waypoint)을 Commanded Waypoint 정보로 포함하여 메시지를 송신한다.

Contingency A Mode 또는 Contingency B Mode 값을 입력받으면, 비행경로 제어 알고리즘에서는 임무 데이터의 비행경로 데이터셋으로부터 비행경로 타입이 ‘Contingency A’ 또는 ‘Contingency B’이며 비행경로의 초기 항로점이 Commanded Waypoint와 일치하는 비행경로 인덱스를 탐색한다. 탐색이 완료되면 해당 비행경로 인덱스를 비행유도제어기에 전달한다. 이에 대한 알고리즘은 Fig. 11과 같다.

자동 착륙 모드는 #2016 메시지의 Select Flight Path Control Mode를 Autoland Engage Mode(19)

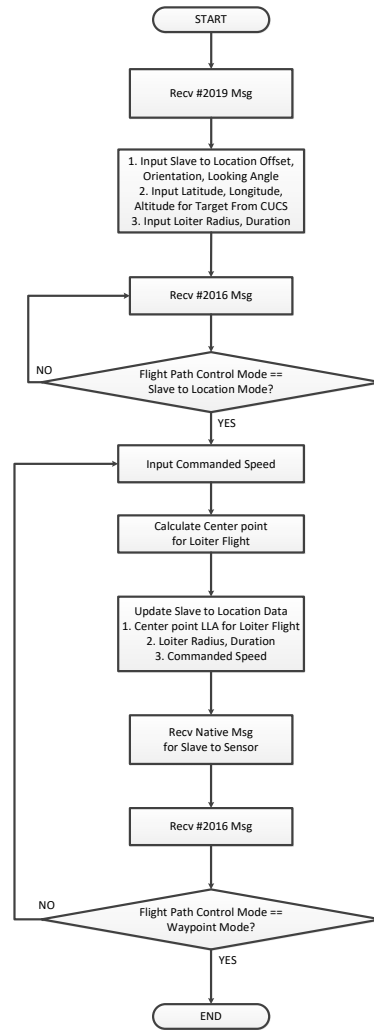


Fig. 10. Flight path control algorithm for Slave to Location mode

로 UAV에 장입한다. 해당 모드 값을 입력받으면 비행경로 제어 알고리즘에서는 임무 데이터의 비행경로 데이터셋으로부터 비행경로 타입이 ‘Approach’인 비행경로를 찾아 해당 인덱스를 탐색하며, 해당 비행경로 인덱스를 비행유도제어기에 전달한다. 이에 대한 알고리즘은 Fig. 12와 같다.

### 3. 비행경로 제어 알고리즘 검증

Fig. 13과 같이 구성된 통합 연동 시험환경을 활용하여 비행경로 제어 알고리즘에 대한 검증을 수행하였다[8].

본 시험환경에서 mini-GCS에 TC(TeleCommand) 메시지를 송신할 경우 mini-GCS → 탑재데이터링크 제어장비 → AFDX 스위치 → 통합연동장비 → UAV

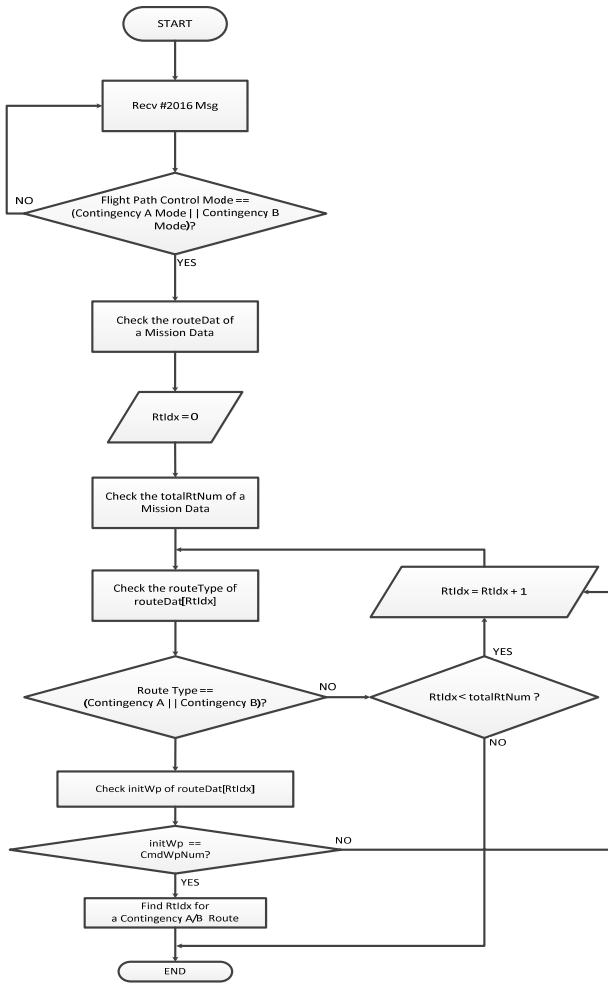


Fig. 11. Flight path control algorithm for Contingency A/B mode

(FLCC 모의 장비) 순으로 메시지 데이터가 이동한다. UAV에서 TM(Telemetry) 메시지를 송신하면 UAV (FLCC 모의 장비) → 통합연동장비 → AFDX 스위치 → 탑재데이터링크 제어장비 → mini-GCS 순으로 메시지 데이터가 이동한다.

여기서 AFDX란 Avionics Full-Duplex Switched Ethernet의 약자로, 항공 전자 장치를 연결하고 통신하기 위한 네트워크 프로토콜로서 항공산업에서 신뢰성 높은 데이터 통신을 제공하기 위해 개발된 기술을 일컫는 용어이며, 이에 대한 구성은 AFDX 스위치, AFDX 엔드 시스템, 이더넷 케이블로 되어있다.

알고리즘 구현 검증은 mini-GCS에서 #2016 TC 메시지를 송신한 후 각 #3001, #3002 등 비행경로 제어 기능별 매칭되는 메시지와 UAV로부터 주기적으로 송신되는 #4001 TM 메시지를 통해서 확인하는 방식으로 수행하였다.

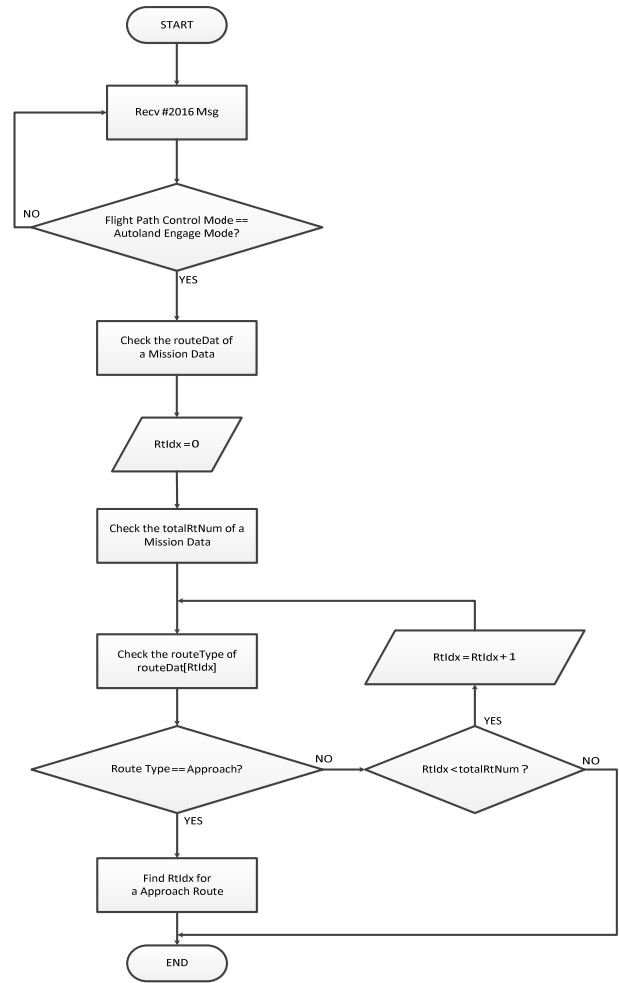


Fig. 12. Flight path control algorithm for Approach mode

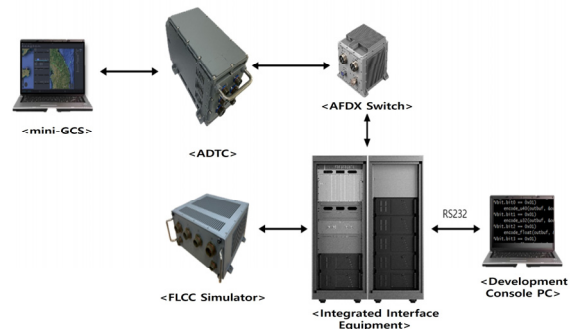


Fig. 13. Configuration of integrated interface test environment

### 3.1 임무 시나리오 설정

Fig. 14과 같이 임무 시나리오를 설정하였으며, 비행경로의 구성은 Table 7과 같다.

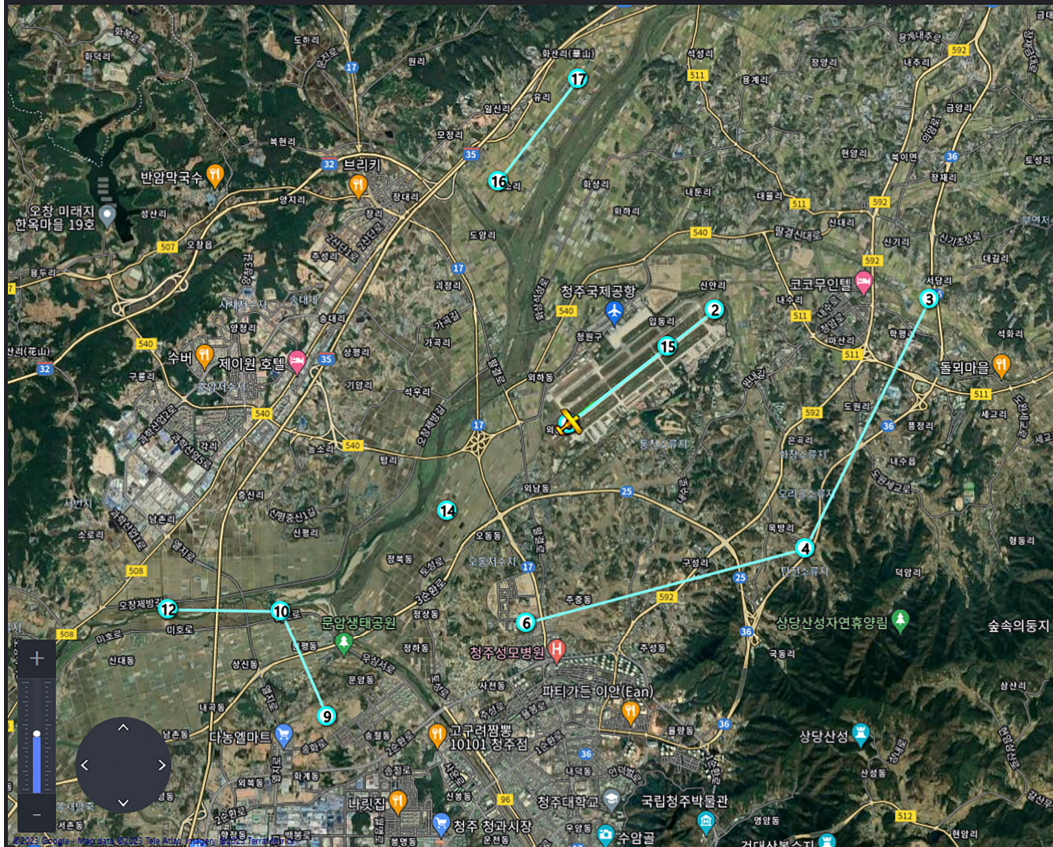


Fig. 14. Mission scenario map to verify the flight-path control algorithm

Table 7. A mission scenario table to verify the flight-path control algorithm

Route name	The number of Waypoint	Waypoint configuration and flight path order
LaunchRt	2	1 → 2
FlightRt1	3	3 → 4 → 6
FlightRt2	3	9 → 10 → 12
ContigARt1	2	16 → 17
ContigBRt1	1	14
ApproachRt	2	8 → 15

임무 시나리오는 1개의 이륙 비행경로(LaunchRt), 착륙 비행경로(ApproachRt), 2개의 비상착륙 비행 경로(ContigARt1, ContigBRt1), 2개의 항로점 비행 경로(FlightRt1, FlightRt2)로 구성하였다. 하나의 임무에 이륙 비행경로와 착륙 비행경로는 각 1개씩만 존재할 수 있으며, 그 외에 비상착륙 비행경로, 항로점 비행경로 등은 다수의 비행경로 구성이 가능하다. 해당 임무 시나리오를 통해 자동 이륙 제어 기능,

항로점 제어 기능, 위치(센서) 종속 제어 기능, 비상 착륙 제어 기능, 자동 착륙 제어 기능에 대한 알고리즘 검증을 수행하였다.

### 3.2 자동 이륙(Launch) 제어기능 검증

Mini-GCS로 Fig. 15과 같이 #2016 메시지 필드 중 Flight Path Control Mode를 ‘Launch’로 선택하여 FLCC 모의 장비에 장입하였으며, 정상적으로 자동 이륙 모드가 인가된 다음 Fig. 16과 같이 UAV가 LaunchRt 비행경로의 마지막 항로점을 향해 이동하여 비행경로의 마지막 항로점에서 다음 비행경로 제어 명령을 받기 전까지 대기 선회 비행을 수행한다.

Fig. 17과 같이 #4001 메시지인 From To Next Waypoint States를 통해 확인하였으며, 2번 항로점 위치에서 다음 항로점은 존재하지 않으므로 Next Waypoint의 정보가 모두 ‘0’로 나오게 되며, Table 7과 Fig. 17에서 LaunchRt 비행경로의 항로점 구성이 일치함을 확인하여 자동 이륙 제어 기능이 수행됨을 확인하였다.

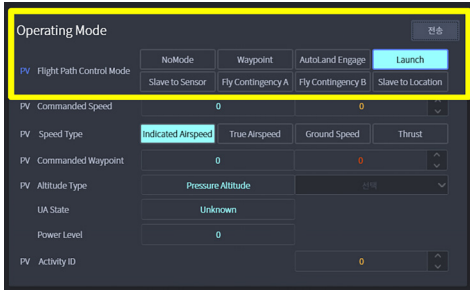


Fig. 15. Field data setting of #2016 message for Launch mode



Fig. 16. Scenario map screen to verify Launch mode

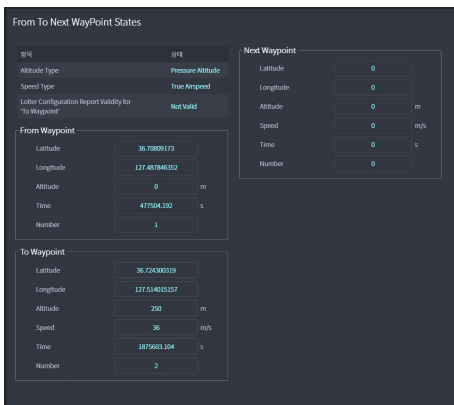


Fig. 17. From To Next Waypoint Status screen of Launch mode

3.3 항로점(Waypoint) 제어 기능 검증

Mini-GCS로 Fig. 18과 같이 #2016 메시지 필드 중 Flight Path Control Mode를 ‘Waypoint’로 선택하고 Commanded Waypoint에 항로점 번호를 입력하여 FLCC 모의 장비에 장입한다. 메시지가 정상 수신되면 FLCC 모의장비로부터 #3002를 통해 수신된 항로점 번호를 응답받고 수신받은 항로점 번호는 입력 항로점 좌측에 하늘색 글자체로 표기된다.

항로점 모드가 인가되면 Fig. 19과 같이 FLCC 모의 장비에서는 해당 비행경로의 항로점을 탐색한 후

UAV가 해당 항로점으로 이동하게 되며, 해당 데이터는 Fig. 20의 #4001 메시지를 통해 확인하였다.

2번 항로점에서 대기한 후 항로점 제어 명령을 받았기 때문에 From Waypoint 정보가 2번 항로점 정보로 업데이트되고, FlightRt1 비행경로의 4번 항로점에 대한 항로점 제어 명령을 받아 To Waypoint 정보는 4번 항로점 정보로, 다음 항로점인 6번 항로점이 Next Waypoint 정보로 업데이트되는 것을 확인할 수 있다. 따라서 Table 7과 Fig. 20 간 FlightRt1 비행경로의 항로점 구정이 일치함을 확인하여 정상적으로 항로점 제어 기능이 수행됨을 확인하였다.

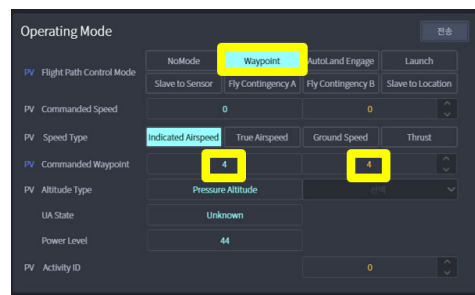


Fig. 18. Field data setting of #2016 message for Waypoint mode



Fig. 19. Scenario map screen to verify waypoint mode

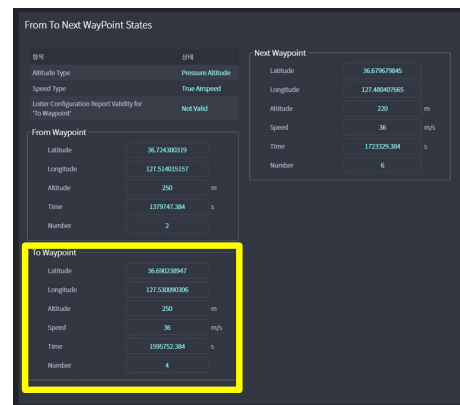


Fig. 20. From To Next Waypoint Status screen of Waypoint mode

### 3.4 Slave to Sensor/Location(센서/위치 종속) 제어 기능 검증

센서 종속 제어 모드와 위치 종속 제어 모드는 같은 수식을 통해 산출된 center point를 찾아 비행경로 제어를 수행하므로 본 연구에서는 위치 종속 모드를 통해 기능 검증을 수행하였다. 해당 검증 시나리오는 UAV가 2번 항로점에서 3번 항로점으로 이동하는 중 위치 종속 모드를 인가받아 해당 관측 임무를 수행하는 것으로 설정하였으며, Fig. 21과 같이 mini-GCS에 입력하여 #2019 메시지를 FLCC 모의 장비로 장입하였다. 이때 입력한 파라미터 값은 Table 8과 같다.

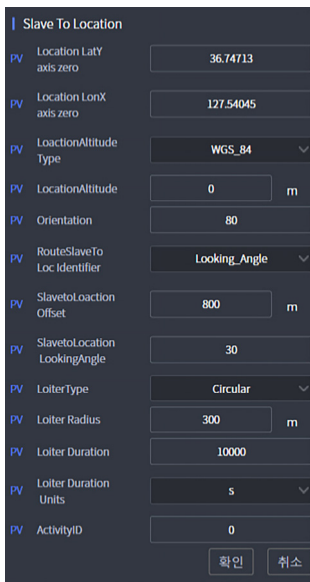


Fig. 21. Setting parameter values of #2019 message from mini-GCS

Table 8. Parameter values of #2019 message for Slave to Location mode setting

Filed name	Value
Location Latitude Y axis zero	36.74713
Location Longitude X axis zero	127.54045
Location Altitude	0
Orientation	80
Slave to Location Offset	800
Slave to Location Looking Angle	30
Loiter Radius	300
Loiter Duration	10,000
Loiter Duration Units	s

비행경로 제어를 위하여 #2016 메시지 필드 중 Flight Path Control Mode를 ‘Slave to Location’으로 선택하여 FLCC 모의 장비에 장입하였고, 3번 항로점으로 이동 중인 UAV는 Fig. 22와 같이 타겟 위치 부근으로 이동하여 Slave to Location 파라미터를 통해 산출된 center point를 중심으로 선회 비행을 수행하였다.

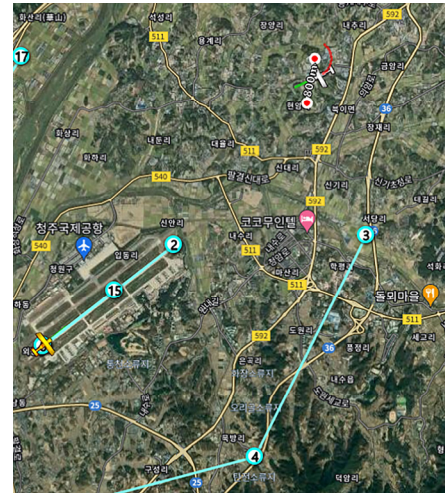


Fig. 22. Scenario map screen to verify Slave to Location mode

선회 비행 시간(loiter duration)이 초과된 후에는 이전에 이동 중인 항로점으로 이동해야 하기 때문에 Fig. 23에 나타난 것과 같이 Next Waypoint는 이전 이동 중인 항로점인 3번 항로점으로 설정되었으며, Slave to Location의 경우 별도의 항로점 번호가 아닌 임의로 생성된 항로점이므로 ‘0’번을 부여하도록 설계하였다.

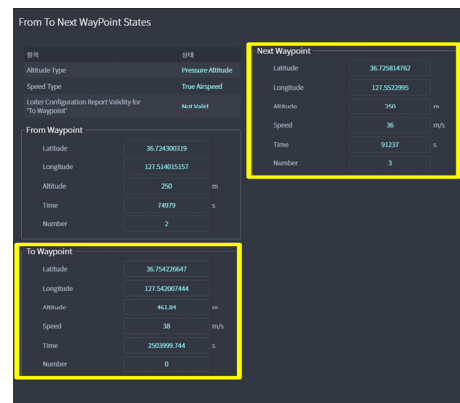


Fig. 23. From To Next Waypoint Status screen of Slave to Location mode

비행경로 마지막 항로점의 Next Waypoint 번호는 STANAG-4586에 따르면 ‘0’으로 설정하도록 정의되어 있으나[10] Slave to Location의 경우는 항로점 번호에 대한 정의가 없어 ‘0’번 항로점을 부여하는 것으로 설계하였다.

현재 항로점 위치정보는 식 (1) – 식 (6)을 통해 산출된 center point의 좌표값이다. 해당 좌표값이 정확히 산출되었는지 확인하기 위해 계산을 통해 교차검증을 수행하였고, 계산 시 사용된 파라미터 값은 Table 9과 같다.

**Table 9.** Parameter values to calculate the center point required for loiter flight

Filed name	Unit	Value
Reference latitude	deg	36.70809173
Reference longitude	deg	127.487846352
Reference altitude	m	0
Target latitude	deg	36.74713
Target longitude	deg	127.54045
Target altitude	m	0
Target North	m	4,340.86
Target East	m	4,689.30
Target Down	m	0

식 (3)과 식 (4)를 통해  $x_{slx}$ ,  $y_{slx}$  값을 계산하면 각각 787.846202(m), 132.918442(m)가 되며, 이를 바탕으로 식 (1)을 통해 center point의  $x_{center}$ ,  $y_{center}$  값을 구하면 5128.706202(m), 4822.218542(m)가 나온다.

식 (5)에 Table 8의 Slave to Location Offset 값과 Slave to Location Looking Angle 값을 넣어 계산하면 center point의 고도값인 461.88(m)이 산출된다. 최종적으로 center point를 LLA로 좌표변환하면 (36.754226[deg], 127.542007[deg], 461.88[m])이 산출되며, 이를 Fig. 26의 현재 항로점 데이터와 비교하면 약간의 오차가 존재하지만 유사한 값이 산출됨을 확인할 수 있었다.

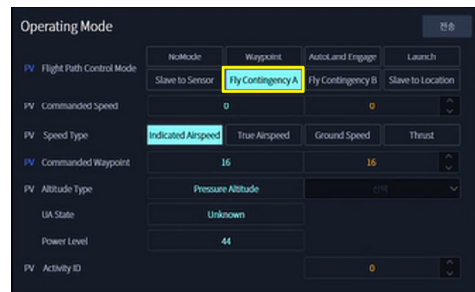
Fig. 22과 같이 #2016을 통해 Slave to Location 모드가 UAV로 인가되면 산출된 center point를 중심으로 선회 비행을 수행하는 것을 확인할 수 있으며, center point와 target의 위치가 지도상 거리를 측정하여 Slave to Location으로 입력한 값 800 m

와 일치함을 확인하였다.

### 3.5 비상착륙(Contingency A/B) 제어 기능 검증

비상착륙 제어 모드도 센서/위치 종속 모드와 마찬가지로 Contingency A와 Contingency B의 운용 방식의 차이만 있을 뿐 비행경로 제어 알고리즘에서의 동작은 동일하므로, 본 연구에서는 Contingency A를 기준으로 기능 검증을 수행하였다.

Mini-GCS로 Fig. 24와 같이 #2016 메시지 필드 중 Flight Path Control Mode를 ‘Fly Contingency A’로 선택하고 Commanded Waypoint에 항로점 번호를 입력하여 FLCC 모의 장비에 장입하였다.



**Fig. 24.** Field data setting of #2016 message for Contingency A mode



**Fig. 25.** Scenario map screen to verify the Contingency A mode

Fig. 25와 같이 입력된 16번 항로점을 찾아 비상착륙 모드를 확인할 수 있었다. 그리고 Fig. 26와 같이 UAV는 16번 항로점을 지나게 되어 16번 항로점이 이전 항로점으로 업데이트되며 현재 항로점이 17번으로 변경되고, 이때 착륙 시퀀스가 동작하여 UAV는 감속 후 정지하게 된다. Mini-GCS로 비상착륙 모드가 인가되고 항로점 번호가 FLCC 모의 장비로 장입되면, UAV는 해당 비상착륙 비행경로인 ContigARt1을 탐색하고, 비상착륙도 착륙 모드 중 하나이므로 정상적으로 착륙 시퀀스가 동작함을 확인하였다.

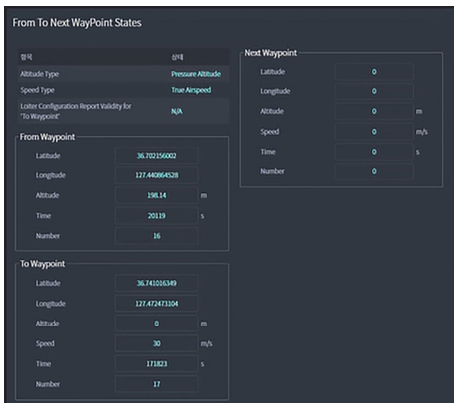


Fig. 26. From To Next Waypoint Status screen of Contingency A mode

### 3.6 자동 착륙(Autoland Engage) 제어 기능 검증

Mini-GCS로 Fig. 27과 같이 #2016 메시지 필드 중 Flight Path Control Mode를 ‘Autoland Engage’로 FLCC 모의 장비로 장입하였으며, Fig. 28과 같이 기존 비행경로에서 벗어나 일정한 거리를 이동한 후에 착륙 비행경로인 ApproachRt의 초기 항로점을 향해 heading 방향을 정렬하고 Fig. 29과 같이 15번 항로점 부근에 착륙을 수행하게 되며, 착륙이 완료된 후에

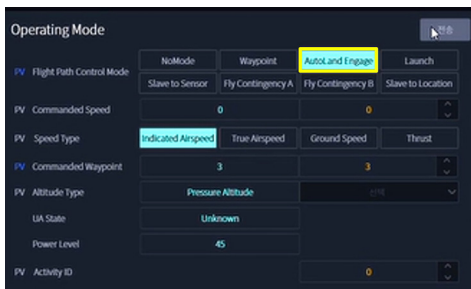


Fig. 27. Field data setting of #2016 message for Autoland Engage mode

는 Fig. 30의 From Waypoint Status에 마지막 항로점인 15번 latitude, longitude, number 등의 정보를 업데이트하고 비행을 종료하게 된다.

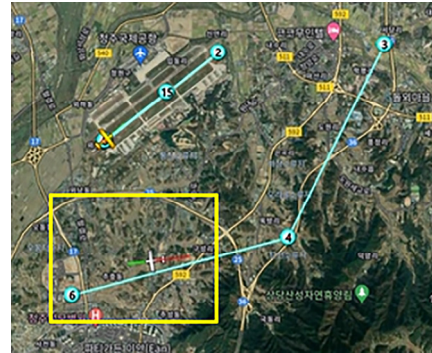


Fig. 28. Scenario map screen to verify Autoland Engage mode (1)



Fig. 29. Scenario map screen to verify Autoland Engage mode (2)

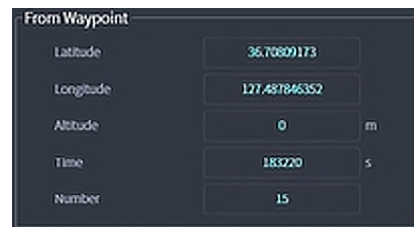


Fig. 30. From Waypoint Status screen of Autoland Engage mode

## 4. 결론

본 논문은 STANAG 4586에서 제시하는 운용 개념에 맞는 UAV의 비행운용 개념을 구체화하고 이에 대한 비행경로 제어 알고리즘을 설계하였으며, 다양한 비행경로 제어 모드에 대한 개념들을 구체화하고 이에 대한 알고리즘을 설계 및 구현하여 검증 환경을 토대로 작성된 임무 데이터를 기반으로 구현된 알고리즘에 대해 이를 검증하였다.

CUCS를 모의한 mini-GCS를 통해 UAV에 메시지들을 송신 및 UAV로부터 메시지들을 수신하고 이를 구현된 비행경로 제어 알고리즘을 통해 UAV 내에 탑재된 비행유도제어기가 비행경로 제어 명령에 따른 의도된 비행 제어에 대해 검증을 수행하여 무인기의 비행경로 제어 알고리즘이 정상적으로 동작함을 확인하였다.

이를 기반으로 비행경로 제어와 관련된 메시지들의 필드 활용 및 다양한 시나리오를 구체화한다면 본 논문에서 제시한 비행경로 제어 외에 UAV의 다양한 비행경로 제어가 가능할 것으로 예상되며, 이와 관련된 운용개념 분석 및 설계 등에 대한 추가적인 연구가 필요하다.

## 참고문헌

- [1] SangYong Yun & SangJung Park, 'A Study of Foreign Military UAV Development Trend and Operational Cases for the UAV Developments of Republic of Korea Armed Forces,' Strategic Studies, VOL. 25, NO. 1, 2018, pp. 205-232.
- [2] Association for Unmanned Vehicle Systems International, Update on STANAG 4586 Published, AUVSI News, 2017.
- [3] Mário Monteiro Marques, STANAG 4586 - Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability, NATO Science and Technology Organization.
- [4] J. T. Platts, M. L. Cummings, & R. J. Kerr, Applicability of STANAG 4586 to Future Unmanned Aerial Vehicles, NATO Research and Technology Organization, 2007.
- [5] NATO Standardization Office, STANAG 4586 Edition 4: Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability - AEP-84 Edition A, 2017.
- [6] Barış Kayayurt & İhsan Yayla, Application of STANAG 4586 Standard for Turkish Aerospace Industries UAV Systems, IEEE Conference Publication, 2013.
- [7] J. A. Smith & R. B. Johnson, 'Interoperability Challenges in Modern UAV Systems: A STANAG 4586 Perspective,' International Journal of Unmanned Systems Engineering, VOL. 15, NO. 3, 2022, pp. 45-62.
- [8] Woo-Rim Jang, Sinyoung Kim, Min-Young Kim, & Seungmok Lee, 'Design and Implementation for the Mission Loading Data Processing and the Flight Path Planning Generation Algorithm Based on Stanag-4586,' Journal of the Korean Society for Aeronautical & Space Sciences, VOL. 52, NO. 7, 2024, pp. 539-550.
- [9] Gyeong Rae Nam, Jeonghwan Go, Cheolhee Kwon, & Soyoung Jeong, 'A Study on the Analysis and Improvement of STANAG 4586 / MAVLink Protocol for Interoperability Improvement of UAS,' Journal of the KIMST, VOL. 23, NO. 6, 2020, pp. 618-638.
- [10] NATO, AEP-84 Volume II, Edition A, Version 1, 2017.
- [11] JongHun Lee, Taesan Park, KilYoung Seong, Gyeong Rae Nam, & Jung Ho Moon, 'Interoperability Design and Verification of Small Drone System Applying STANAG 4586,' Journal of Aerospace System Engineering, VOL. 16, NO. 6, 2022, pp. 74-80.