

## Original Article

모델 기반 설계 기법을 이용한 무인항공기의  
침입기 추적 및 충돌회피 알고리즘 설계

최현진\*, 유창선\*\*, 유혁\*\*, 김성욱\*\*, 안석민\*\*

Intruder Tracking and Collision Avoidance Algorithm Design for  
Unmanned Aerial Vehicles using a Model-based Design Method

Hyunjin Choi\*, Chang-Sun Yoo\*\*, Hyeok Ryu\*\*, Sungwook Kim\*\*, and Seokmin Ahn\*\*

## ABSTRACT

Unmanned Aerial Vehicles(UAVs) require collision avoidance capabilities equivalent to the capabilities of manned aircraft to enter the airspace of manned aircraft. In the case of Visual Flight Rules of manned aircraft, collision avoidance is performed by 'See-and-Avoid' of pilots. To obtain those capabilities of UAVs named as 'Sense-and-Avoid', sensor-system-based intruder tracking and collision avoidance methods are required. In this study, a multi-sensor-based tracking, data fusion, and collision avoidance algorithm is designed by using a model-based design tool MATLAB/SIMULINK, and validations of the designed model and code using numerical simulations and processor-in-the-loop simulations are performed.

**Key Words** : Sense-and-Avoid(탐지 및 회피), Unmanned Aerial Vehicles(무인항공기), Tracking(추적), Collision Avoidance(충돌회피), Model-based Design(모델 기반 설계)

## I. 서 론

최근 무인항공기 기술과 관련 산업의 급격한 성장이 이루어짐에 따라 이를 다양한 용도에 사용하기 위한 연구가 활발히 이루어지고 있다. 그 중에서 비행 공역을 확장하여 기존 유인항공기의 기능을 수행 또는 대체하고자 하는 연구도 포함된다. 하지만 공역을 확장하여 운용하기 위해서는 실제 항공기가 가지는 충돌회피 기능과 동등한 수준의 성능이 요구되고, 이에 따른 규정이 제정되고 있다[1].

유인항공기의 충돌회피 기술은 1950년 초부터 시작된 항공 운송의 급격한 증가에 따른 항공기 간의 공중충돌(Mid Air Collision, MAC)을 방지하고자 개발되었고, ATC(Air Traffic Control) 관제, TCAS(Traffic Collision Avoidance System), 조종사의 'See-and-Avoid'에 의한 회피 규정이 수립되어 있다[2].

무인항공기의 경우 ATC관제와 소통할 절차가 아직 수립되어 있지 않고, 대형 유인항공기에 의 무장착되는 TCAS는 가격, 중량, 성능 면에서 직접적인 적용이 어렵다. 이로 인해 무인항공기의 충돌회피 기술은 조종사의 'See-and-Avoid'와 유사한 개념의 센서를 이용한 탐지 기반의 'Sense-and-Avoid' 혹은 'Detect- and-Avoid' 방식을 채택한 연구가 활발히 수행되고 있다[3-4].

무인항공기가 공중 침입기를 탐지하기 위한 방식은 크게 TCAS, ADS-B(Automatic Dependant Surveillance Broadcast)등을 이용한 협업적 방식과 Radar, Lidar, 영상센서를 이용한

Received : 11. Oct. 2017. Revised : 28. Nov. 2017.

Accepted : 20. Dec. 2017

\* 한국항공우주연구원 선임

\*\* 한국항공우주연구원 책임

연락처자 E-mail : hjchoi01@kari.re.kr

34133 대전광역시 유성구 과학로 169-84

독립적 방식이 있고, 주로 소형화, 경량화 할 수 있는 ADS-B, 영상센서를 이용한 방식이 고려되고 있다.

본 연구에서는 ADS-B와 영상센서를 장착한 무인항공기를 고려하였고, ADS-B 장착 침입기는 송수신기의 수신 데이터를 사용하고 이를 미장착한 침입기는 영상센서를 통한 탐지 결과를 사용하여 충돌위험을 탐지하고 회피하는 시스템을 고려하였다.

해당 센서 구성에 의한 충돌회피 시스템은 침입기의 탐지 측정값을 이용한 목표물 추적 기술과 해당 목표물에 대한 충돌위험을 판단하고 회피하는 기술이 필요하다. 추적 기술은 다수 센서, 다수 목표물 환경에서 다양한 기법들이 연구되었고, 그중 대표적으로 Nearest Neighbor, Probabilistic Data Association, Multiple Hypothesis 등의 여러 방법들이 제안되었다[5]. 데이터 융합의 경우 구조적으로 중앙집중형, 분산형, 연합형 데이터 융합 구조가 연구되었다[6].

항공기의 충돌회피 기술도 다양한 연구들이 수행되었고, 기동 방식에 따라 크게 미리 지정된 기동, 역장에 의한 기동, 최적화된 기동으로 분류될 수 있다[7,8]. 다양한 충돌회피 방식 중 최근접점을 이용한 최적화된 충돌회피 방식은 일정한 거리를 회피할 수 있고 다수의 침입기에 대해 회피할 수 있는 특징을 지닌다[9].

본 연구에서는 ADS-B와 영상센서를 장착한 무인항공기의 침입기 추적 알고리즘으로 Global Nearest Neighbor(GNN) 기반 매칭을 이용한 방법을 고려하였고, 충돌회피 알고리즘으로 최근접점 기하를 이용한 충돌위험 판단 및 회피 방법을 고려하였다.

해당 알고리즘은 모델 기반 설계 도구인 MATLAB/SIMULINK 환경에서 설계하였고, 설계한 모델은 자동코드생성 기능을 이용하여 C 코드로 변환하여 충돌회피모듈에 탑재하였다.

설계된 모델은 수치시뮬레이션과 충돌회피모듈을 이용한 Processor-in-the-Loop Simulation (PILS) 환경에서 유효성을 확인하였다.

## II. 본 론

### 2.1 무인항공기 충돌회피 시스템

#### 1.1 충돌회피 시스템 구성 및 운용 개념

무인항공기 충돌회피 시스템은 Fig. 1과 같이 무인항공기에 탑재되는 ADS-B 탑재장비, 영상탐지모듈, 충돌회피모듈, RF(Radio Frequency) 통

신장비로 구성된다.

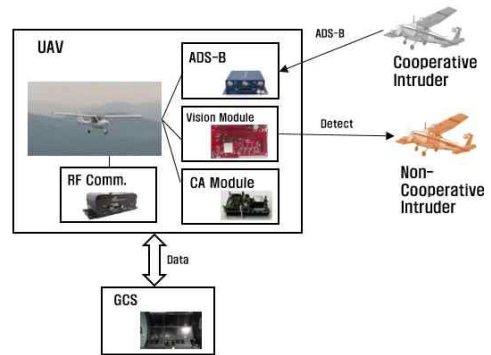


Fig. 1 UAV collision avoidance system

무인항공기 충돌회피를 위한 운용은 기본적으로 ADS-B 정보를 이용하여 주변의 침입기 유무를 판단한다. 만약, 침입기가 ADS-B를 미장착 하였거나 고장 난 경우에는 독립적인 침입기 탐지 방식으로 영상센서와 탐지모듈을 활용하여 침입기 정보 획득을 보완한다.

각각의 센서로부터 획득한 침입기 정보를 바탕으로 침입기 추적을 수행하게 되고, ADS-B와 영상탐지모듈에서 동일한 침입기라고 판단되는 정보는 데이터 융합을 통해 하나의 정보로 통합한다. 획득한 침입기 정보를 바탕으로 충돌회피 알고리즘에서 충돌위험 판단 및 충돌회피명령 생성을 수행하여 무인항공기 비행제어컴퓨터에 해당 정보를 전달하여 충돌회피를 수행한다.

추가로 충돌회피모듈에서 생성한 데이터는 지상국에서 모니터링하고, 관련 설정신호 반영을 위해 탑재통신장비를 통한 지상국과 데이터 송수신을 수행한다.

#### 2.1.2 침입기 정보 수신 장비

침입기 정보는 ADS-B 탑재장비와 카메라를 이용하는 영상탐지모듈 장비로 수신한다.

ADS-B는 송수신기를 통해 침입기의 항법정보를 수신하여 충돌회피모듈로 전달하고, 영상탐지모듈은 무인항공기 양쪽 날개에 설치된 카메라의 스테레오 영상을 통해 침입기의 항법정보를 획득하여 충돌회피모듈로 전달한다.

본 연구에서는 ADS-B로부터 침입기의 위도, 경도, 고도, 수평속력, 수직속력, heading 각 정보를 획득할 수 있고, 영상탐지모듈로부터 NED(North-East-Down) 좌표 기준의 침입기의 상대위치, 상대속도를 획득할 수 있는 시스템을 고려하였다.

### 2.1.3 충돌회피모듈

충돌회피모듈은 Fig. 2와 같이 DSP 보드를 이용한 충돌회피컴퓨터, 무인항공기의 위치 및 자세 정보를 위한 GPS와 INS의 항법 컴퓨터, GPS 안테나, 전원 관리를 위한 파워 보드로 구성된다.

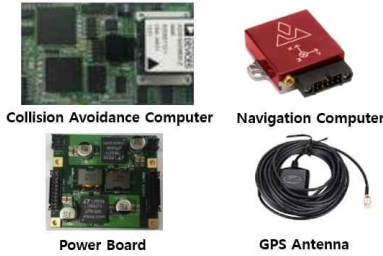


Fig. 2 Components of the collision avoidance module

충돌회피모듈은 내부 GPS/INS 항법 컴퓨터를 통해 항법정보를 수신 받고 ADS-B 탑재장비와 영상탐지모듈로부터 침입기 정보를 수신 받으며 무인기로 충돌회피명령을 지상국으로는 충돌회피 관련 정보를 전달하도록 구성되어 있다.

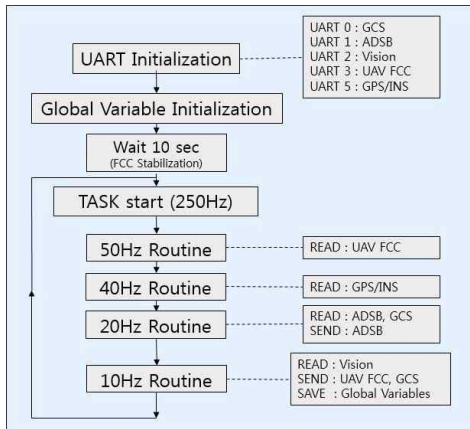


Fig. 3 Operational program of the system

해당 기능을 위해 RS-232c를 이용한 시리얼통신을 사용하고 충돌회피모듈 내부 프로그램은 Fig. 3과 같이 통신포트와 전역변수를 초기화하고 각자 정해진 주기에 따라 각 기능이 실행되도록 구성하였다.

## 2.2 침입기 추적 기법 설계

### 2.2.1 이중 센서 침입기 추적 및 데이터 융합 기법

ADS-B와 영상탐지모듈은 서로 다른 주기마다 수신값을 이용할 수 있기 때문에 각각의 정보에 대한 침입기 추적 기법의 설계가 필요하다.

이를 위해 Fig. 4와 같이 각각의 센서에 대해 추적 기능을 따로 구성하고 중앙에서 데이터융합을 수행하는 분산형 구조를 바탕으로 추적 및 데이터융합 기법을 설계하였다.

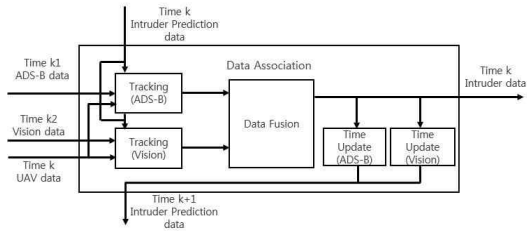


Fig. 4 Diagram of tracking and data fusion

설계한 추적 및 데이터융합 기법은 ADS-B와 영상에 대한 각각의 Tracking, Time Update 기능과 두 추적값에 대한 데이터 융합을 수행하는 Data Fusion 기능으로 구성되고 각각의 기능은 메인 프로세스 주기에 따라 실행된다.

Tracking과 Time Update 기능을 위해 각각의 센서에 대해 N개의 침입기 추적 버퍼와 N개의 버퍼 각각에 대해 측정값과의 매칭 상태를 파악할 수 있는 파라미터를 사용하였다. 추적 버퍼는 {Confirmed set, Tentative set}으로 나누어 추적 신뢰성에 따라 확정값과 잠재값을 구분하는 방식 [10]을 사용하였고, 매칭 관련 파라미터 p1, p2를 다음과 같이 구성하였다.

$$\{p1 = \text{연속 매칭 수}, p2 = \text{연속 비매칭 수}\}$$

Tracking 기능은 매 주기마다 실행 된다. 이때, 센서의 측정값이 없을 때는 현재 추적 버퍼의 상태값을 그대로 반환하고 추적 중인 버퍼값의 p2를 증가시킨다. 측정값이 있을 때는 측정값과 기존 추적 버퍼의 값을 비교하여 일치성이 판단되는 측정값을 해당 추적 버퍼에 업데이트하고 해당 버퍼의 p1을 증가시키고 p2의 초기화를 수행한다. 측정값과 매칭되지 않는 추적 버퍼에 대해서는 p1을 초기화하고 p2를 증가시킨다.

Time Update 기능은 매 주기마다 현재의 상태값을 기준으로 프로세스 주기만큼 상태값을 외삽하여 업데이트하고, p1 값의 조건에 따라 Tentative set에서 Confirmed set으로 추적 버퍼

업데이트를 수행하고 p2 값의 조건에 따라 추적 삭제의 삭제를 수행하도록 구성하였다.

Data Fusion에서는 ADS-B 추적 버퍼와 영상 추적 버퍼의 일치성을 판단하여 일치하는 버퍼의 상태값은 융합된 데이터값을 전달하고 일치하지 않는 버퍼 상태값은 각각의 버퍼값을 전달하도록 구성하였다.

Tracking, Time Update, 및 Data Fusion 기능에 대한 Pseudo 코드를 Fig. 5~Fig. 7에 정리하였다. 본 연구에서 Tracking을 위한 매칭과 데이터 융합을 위한 매칭은 Euclidean 거리를 이용한 GNN 기반 매칭 기법을 사용하였다[10].

2.2.2 침입기 추적 모델 구성

제안된 침입기 추적 및 데이터 융합 기법은 SIMULINK 환경에서 Fig. 8과 같이 모델링하였다. ADS-B와 영상에 대한 추적 버퍼는 Data Storage Memory 블록을 이용하였고 코드 생성 시 전역변수로 생성하도록 설정하였다. 각각의 센서에 대한 Tracking, Time Update 기능과 Data Fusion 기능은 MATLAB function 블록을 이용하여 Fig. 5~Fig. 7의 Pseudo 코드를 적용하여 스크립트를 구성하였다.

```

void Tracking(void)
{
    if (Number of intruder != 0) {
        Track_ConfirmedAndTentative( inputs )
        Track_Init_Tentative( inputs )
    } else {
        increase p2 of tracks
    }
}
void Track_ConfirmedAndTentative( inputs )
{
    check association with tracking buffers
    (Confirmed set, Tentative set)
    if (association == true) {
        increase p1 count of the track
    }
}
void Track_Init_Tentative( inputs )
{
    add new intruder to the buffer
    (Tentative set)
}
    
```

Fig. 5 Pseudo code of Tracking

```

void Time_Update(void)
{
    if (p2 of the track in the Confirmed set < Down condition) {
        remain
    } else {
        move to the Tentative set
    }
    if (p1 of the track in the Tentative set > Up condition) {
        move to the Confirmed set
    } else {
        if (P2 of the track in the Confirmed set < Del condition) {
            remain
        } else {
            delete
        }
    }
}
    
```

Fig. 6 Pseudo code of Time\_Update

```

void Data_Fusion(void)
{
    Check matching between ADS-B Tracks and Vision Tracks
    if (matching == 1) {
        update fusion states
    } else {
        update single-measurement states
    }
}
    
```

Fig. 7 Pseudo code of Data\_Fusion

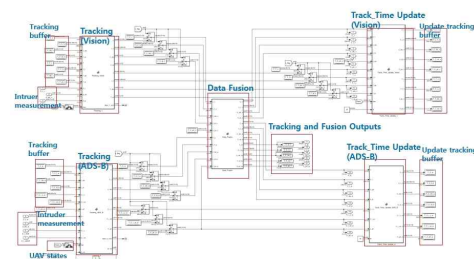


Fig. 8 SIMULINK model of tracking and data fusion

2.3 충돌회피 기법 설계

2.3.1 충돌회피 파라미터 설정

ROA(Rules-of-the-Air)[11]를 고려한 무인항공기의 충돌회피를 위해 Near miss 수평거리 500ft (152.4m) 이상을 유지하는 수평충돌회피 상황을 고려하였다. 이에 대한 충돌회피 한계값을 정의하여 침입기의 상태가 이를 침범할 때 회피를 수행하도록 파라미터를 설정하였다.

무인항공기가 수평 선회로 충돌위험을 벗어나는 시나리오에 대해 Fig. 9와 같은 안전반경을 침범하지 않고 회피하는 충돌회피 기하를 고려하였다.

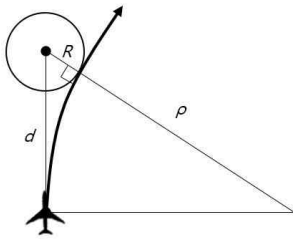


Fig. 9 Geometry of a circular region avoidance

무인항공기 선회 비행 시의 속력과 수평 각속도를  $V, \dot{\psi}$ 이라 하면, 무인항공기가 특정 위치를  $R$ 만큼의 거리를 두고 회피하기 위한 최소거리  $d$ 를 다음과 같이 구할 수 있다.

$$\dot{\psi} = \frac{g}{V} \tan \phi \tag{1}$$

$$\rho = \frac{V}{\dot{\psi}} = \frac{V^2}{g \tan \phi} \tag{2}$$

$$d = \sqrt{R^2 + 2\rho R} \tag{3}$$

여기에서  $g$ 는 중력가속도,  $\phi$ 는 무인항공기를 각이다.

회피를 위한 최소거리  $d$ 는 무인항공기의 속력이 증가하면 비례적으로 증가하는 관계를 가진다. 최소거리  $d$ 를 무인항공기 속력으로 나누면 Fig. 10과 같이 제한된 값의 범위를 가지게 되고, 이를 바탕으로 회피를 위해 필요한 최소시간을 정의할 수 있다.

본 연구에서 수평거리를 확보하기 위한 충돌회피 한계값은 시간조건으로 설정하였고, 이 조건은 위의 충돌회피를 위한 최소시간과 Fig. 10을 고려하여 18초로 설정하였다. 무인기와 침입기 사이가 가장 근접하기까지의 시간 또는 상대 거리에 속력을 나눈 값이 한계값 범위인 18초보다 작아지면 충돌회피를 수행하도록 구성하였다.

### 2.3.2 최근접점 기하 기반 위험판단 기법

무인항공기의 충돌위험 판단 및 충돌회피 기법은 Fig. 11과 같은 최근접점(Closest-Point-of-Approach, CPA) 기하 기반 충돌회피 기법을 고려하였다. 최근접점 기하는 무인항공기와 침입기의 위치와 속도를 바탕으로 충돌상황을 분석하고 해석적인 충돌회피 솔루션을 획득할 수 있다[9, 12].

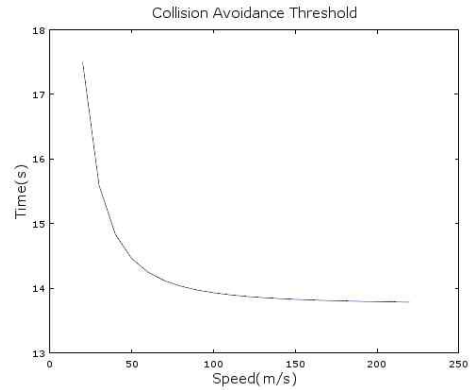


Fig. 10 Time-based collision avoidance threshold

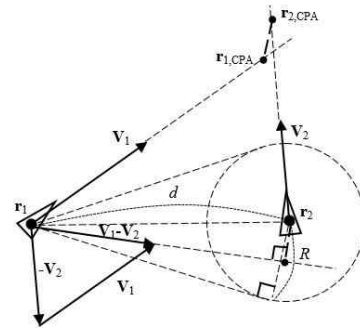


Fig. 11 Geometry of CPA

무인항공기의 수평 위치와 속도를  $r_1, V_1$  침입기의 수평 위치와 속도를  $r_2, V_2$  로 정의하고 상대 위치를  $\tilde{r} = r_1 - r_2$ , 상대속도를  $\tilde{V} = V_1 - V_2$ , 상대거리를  $d = \|\tilde{r}\|$  로 정의하면, 무인항공기와 침입기의 수평 최근접시간  $t_{CPA}$  과 최근접점  $r_{1,CPA}, r_{2,CPA}$ 은 다음과 같이 구할 수 있다.

$$t_{CPA} = -\frac{\tilde{r}^T \tilde{V}}{\tilde{V}^T \tilde{V}} \tag{4}$$

$$r_{1,CPA} = r_1 + V_1 t_{CPA} \tag{5}$$

$$r_{2,CPA} = r_2 + V_2 t_{CPA} \tag{6}$$

최근접점 기하에 의하면 무인항공기와 침입기가 가장 근접했을 때의 거리가 Near miss를 위한 거리보다 작은 경우 충돌위험이 존재한다고 판단할 수 있다. 여기에 충돌회피 한계값을 침범할 때 충돌회피를 수행하도록 구성하기 위해서 Table 1의 조건으로 충돌위험 판단을 구성하였다. 여기에서 CA\_R은 Near miss 거리에 측정오차에 의한 마진을 추가하여 170m, CA\_t는 충돌

회피 한계값 18초로 정의하였고,  $d_{CPA}$ 는 최근접 점에서의 침입기의 상대거리로 정의하였다.

**Table 1 Decision conditions for collision avoidance**

Decision	Condition
Approach	$t_{CPA} > 0$
CA_path_in	$(d_{CPA} \leq CA\_R$ or $d \leq CA\_R)$
CA_threshold_in	$(t_{CPA} \leq CA\_t$ or $d/V \leq CA\_t)$ and $CA\_path\_in == 1$
CA_maneuver	$(Approach == 1$ and $CA\_threshold\_in == 1)$

**2.3.3 충돌회피 해**

무인항공기의 충돌회피는 최근접점 기하를 바탕으로 침입기의 시선각 방향으로 좌표변환을 한 원뿔방정식으로부터 해를 구하였다[9, 12].

시선각 좌표축 기준으로 현재 상대위치는  $\tilde{r}_{LOS} = (d, 0)$ 이고, 무인항공기의 충돌회피 솔루션 속도벡터를  $V_{1,LOS}^s$  침입기의 현재 속도벡터를  $V_{2,LOS} = (u_2, v_2)$ 로 하면  $V_{1,LOS}^s$ 는 아래와 같다.

$$V_{1,LOS}^s = V_{2,LOS} + \begin{bmatrix} \beta h \\ \pm h \end{bmatrix} \quad (7)$$

여기서  $\beta \equiv \sqrt{(d^2 - R^2)/R^2}$ 이고, 원뿔 형태의 기하에 따라  $h$ 가 양수일 때는 우측, 음수일때는 좌측으로 회피방향이 결정된다. 본 연구에서는 ROA의 우측 회피를 준용하여 우측 방향인  $+h$ 를 선택하여 충돌회피 솔루션을 구하였다.

무인항공기 기준속력을  $V_{1,s}$ , 침입기의 속력을  $V_2$ 로 정의하면  $h$ 는 다음 이차방정식의 양수인 해를 이용하여 결정할 수 있고, 이에 대한 충돌회피 솔루션 속도벡터를 얻을 수 있다.

$$(1 + \beta^2)h^2 + 2(u_2\beta + v_2)h + V_2^2 - V_{1,s}^2 = 0 \quad (8)$$

충돌회피 방향  $\psi_{CA}$ 은  $V_{1,LOS}^s$ 의 x성분 속도  $V_{1,LOSx}^s$ 와 y성분 속도  $V_{1,LOSy}^s$ 의 arctan 각도에 침입기의 시선각 각도  $\psi_{LOS}$ 를 더하여 구할 수 있다.

$$\psi_{CA} = \psi_{LOS} + \text{atan}(V_{1,LOSy}^s / V_{1,LOSx}^s) \quad (9)$$

이를 바탕으로 충돌위험 판단 및 충돌회피명령 생성 알고리즘을 구성하였고, 해당 알고리즘은 Fig. 12에 Pseudo 코드로 정리하였다.

```

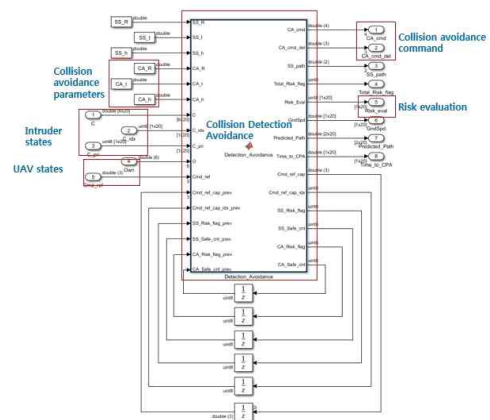
void Collision_Detection_Avoidance(void)
{
    Initialize variables and outputs
    for (number of intruder == 1)
    {
        Calculate Intruder's path
        Evaluate risk
        if (Self separation condition for each intruder == 1)
        {
            generate self separation path
        }
        if (Collision avoidance condition for each intruder == 1)
        {
            generate collision avoidance command
        }
    }
    if (Self separation condition for intruders == 1)
    {
        update self separation path
    }
    if (Collision avoidance condition for intruders == 1)
    {
        update collision avoidance command
    }
}
    
```

**Fig. 12 Pseudo code of Collision\_Detection\_Avoidance**

**2.3.4 충돌회피 모델 구성**

제안된 충돌회피 알고리즘은 SIMULINK 환경에서 Fig. 13과 같이 모델링하였다. 충돌회피 파라미터와 침입기 항법 정보, 충돌회피 무인기 항법정보를 입력으로 받고 MATLAB function 블록에서 알고리즘 연산을 수행하여 위험 존재 및 충돌회피 명령을 출력하는 구조로 구성하였다.

설계된 충돌회피 알고리즘 모델은 침입기 추적 및 융합 모델과 통합하여 Fig. 14와 같이 구성하였고, 입력력 단위를 맞춰 통합모델 자체를 프로그램에서 하나의 함수로 실행되도록 하였다.



**Fig. 13 SIMULINK model of collision avoidance**

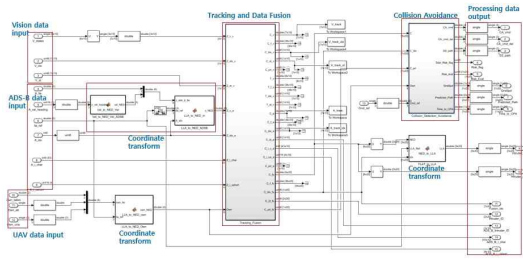


Fig. 14 SIMULINK model of integrated collision avoidance system

## 2.4 수치 시뮬레이션 및 PILS 실험

### 2.4.1 수치 시뮬레이션

설계된 추적, 데이터 융합, 충돌회피 알고리즘을 검증하기 위해 무인항공기와 침입기의 움직임을 묘사한 수치 시뮬레이션 환경을 Fig. 15와 같이 구성하였다.

무인항공기와 침입기의 운동모델은 질점모델을 가정하였고, 침입기가 무인항공기의 정면과 측면에서 접근하는 상황에 대한 시뮬레이션을 수행하였다.

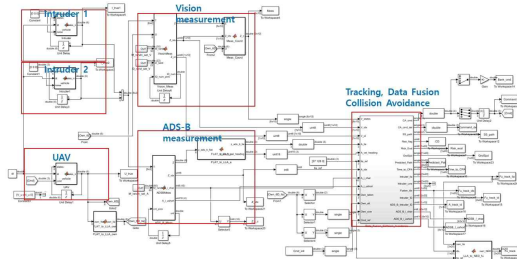


Fig. 15 Environment of Numerical Simulation

침입기가 정면과 측면에서 동시에 접근하는 상황에 대해 Fig. 16을 통해 침입기 추적 결과와 무인항공기의 충돌회피 결과를 확인할 수 있다. 그림에서 데이터융합 정보는 영상과 ADS-B가 융합된 경우만을 표시하였다. 시뮬레이션에서 침입기와 무인항공기 사이의 최소 거리는 169.05m로 CA\_R 값 170m에 근접함을 확인할 수 있다.

### 2.4.2 PILS 실험

PILS 실험을 위해 무인항공기와 침입기의 항법정보를 모사하여 GPS/INS, ADS-B, 영상탐지 모듈 신호를 충돌회피모듈로 전달하고, 충돌회피 모듈의 충돌회피명령을 무인항공기의 운동에 반

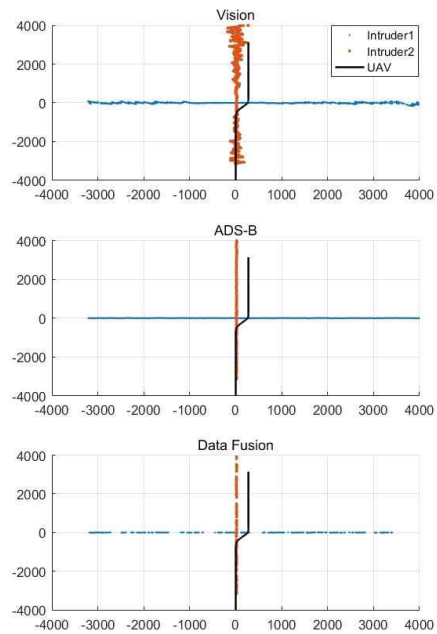


Fig. 16 Tracking/Data Fusion/Collision Avoidance result

영하고 관련 정보를 모니터링 할 수 있는 실험 환경을 Fig. 17과 같이 구성하였다. 해당 환경은 충돌회피모듈을 위한 5채널의 RS-232c 통신 인터페이스를 구현하여 ADS-B, 영상탐지모듈, GPS/INS, 무인항공기, 지상국 정보를 송수신한다.

무인항공기와 침입기의 운동모델은 질점모델로 구현하였고, 무인항공기가 진행하는 방향 기준으로 침입기가 접근하는 시나리오를 생성하여 충돌회피를 테스트할 수 있도록 구성하였다.

해당 시뮬레이션 환경으로 수행한 침입기 회피 결과는 Fig. 18과 같고, 접근하는 침입기를 우측으로 회피한 결과를 확인할 수 있다.

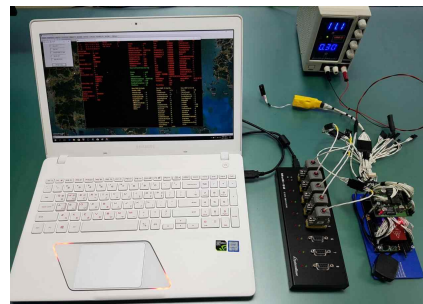


Fig. 17 PILS with Collision avoidance Module Environment



Fig. 18 A collision avoidance result of PILS test

### III. 결 론

본 연구는 무인항공기의 공중충돌 방지를 위해 ADS-B와 영상센서를 이용한 충돌회피 시스템을 모델기반 설계 도구 MATLAB/SIMULINK를 사용하여 설계하였다. 제안된 충돌회피 시스템은 이종 센서 기반 추적 및 데이터 융합 기술을 통해 침입기를 추적하고, Near miss 거리 이상을 확보할 수 있는 충돌위험 판단 및 충돌회피 기술을 통해 도출된 충돌회피명령을 무인항공기에 전달한다. 설계된 충돌회피 시스템은 수치 시뮬레이션 및 PILS 환경에서의 시뮬레이션으로 유효성을 확인하였다. 본 연구를 바탕으로 향후 다양한 시나리오에 대한 실험 및 구성품 통합 비행시험을 수행하여 설계된 시스템의 검증을 수행할 계획이다.

### 후 기

본 연구는 산업자원부 산업기술혁신사업 중 “ADS-B기반 무인항공기 충돌회피시스템 개발”(10059035) 사업의 지원을 받아 수행되었습니다.

### Reference

- [1] RTCA DO-365, *Minimum Operational Performance Standards for Detect and Avoid Systems*, RTCA Inc., 2017.
- [2] Federal Aviation Administration, Advisory Circular 90-48D, *Pilot's Role in Collision Avoidance*, U.S. Department of Transportation, 2016.
- [3] Zeitlin, A. D., "Sense & Avoid Capability Development Challenges," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 25, No. 10, 2010, pp. 27-32.
- [4] Angelov, P., *Sense and Avoid in UAS: Research and Application*, Wiley, Chichester, United Kingdom, 2012.
- [5] Bar-Shalom, Y., and Li. X. R., *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs, CT, 1995.
- [6] Gao, Y., Krakiwsky, E. J., Abousalem, M. A., and McLellan, J. F., "Comparison and Analysis of Centralized, Decentralized, and Federated Filters," *Journal of the Institute of Navigation*, Vol. 40, No. 1, 1993. pp. 69-86.
- [7] Kim, J., and Kang, J., "A Study on Conflict Detection and Resolution for Aircraft Separation Assurance in a Free Flight Environment," *Journal of the Korean Society for Aviation and Aeronautics*, Vol. 18, No. 3, 2010, pp. 27-33.
- [8] Lee, D.-Y., and Kang, J., "Comparative Analysis of Free Flight Conflict Detection and Resolution Algorithms," *Journal of the Korean Society for Aviation and Aeronautics*, Vol. 19, No. 4, 2011, pp. 83-90.
- [9] Choi, H., Kim, Y., Lee, Y., and Kim, E. T., "A Reactive Collision Avoidance Algorithm for Multiple Midair Unmanned Aerial Vehicles," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 56, No. 1, 2013, pp. 15-24.
- [10] Sinha, A., Ding, Z., Krrubharajan, T., and Farooq, M. "Track Quality Based Multitarget Tracking Approach for Global Nearest-Neighbor Association," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 2, 2012, pp. 1179-1191.
- [11] ICAO Annex 2, *Rules of the Air*, International Civil Aviation Organization ,2005
- [12] Choi. H., "One-to-One Collision Situation Analysis and Avoidance Based on the Closest-Point-of-Approach Geometry for Unmanned Aerial Vehicles," *2017 KSAS spring conference*, Samcheok, South Korea, Apr. 2017