J. of The Korean Society for Aeronautical and Space Sciences 47(4), 266-273(2019)
DOI:https://doi.org/10.5139/JKSAS.2019.47.4.266
ISSN 1225-1348(print), 2287-6871(online)

자율정찰비행 무인항공기의 비행운영조건 고찰을 위한 비행시뮬레이션 개발

석민준[1]

Development of Autonomous Reconnaissance Flight Simulation for Unmanned Aircraft to Derive Flight Operating Condition

Min Joon Seok[1]
Defense Agency for Technology and Quality[1]

ABSTRACT

The efficiency and effectiveness of mission performance can be greatly changed according to the operating conditions such as the number of manned aircraft, flight altitude, and so on, in performing search and reconnaissance missions using a large number of small reconnaissance unmanned aerial vehicles. However, it is not easy to determine which operating conditions are most reasonable. Therefore, in this study, we developed an unmanned airplane flight simulation that can detect and identify the target while avoiding collision according to autonomous flight, suggesting a way to derive operating conditions when operating a large number of unmanned aerial vehicles.

초 록

다수의 소형 정찰용 무인항공기를 이용하여 탐색 및 정찰 임무를 수행하는데 있어서 무인항공기의 운용 대수, 비행고도 등 운용 조건에 따라 임무 수행의 효율성과 효과성은 크게 변경될 수 있다. 하지만 어떤 운용조건이 가장 합리적인지 판단하기는 쉽지 않다. 따라서 본 연구에서는 자율비행에 따라 충돌을 회피하면서 표적을 탐지 및 판별할 수 있는 무인항공기 비행 시뮬레이션을 개발하여 다수의 무인항공기 운용 시 보다 효과적이고 효율적인 운용조건을 도출할 수 있는 방안을 제시하였다.

Key Words : UAV(무인항공기), Flocking(플로킹), Autonomous Flight(자율비행), ATR(자율표적인지), Confusion Matrix(혼돈행렬)

Ⅰ. 서 론

정찰용 무인항공기는 목표지역 상공을 비행하면서 적외선 감지기나 영상카메라 등을 이용하여 적의 목표물을 식별하고 목표물에 대한 정보를 실시간으로 공유함으로서 군 작전에 효용성을 높여준다. 이러한 정찰용 무인항공기는 고고도에 운용되는 대형 무인항공기도 있지만 소규모 부대 작전수행을 위한 상대적으로 저고도에서 운용되는 소형 무인항공기도 다양하게 활용되고 있다. 저고도에서 운용되는 소형 무인항공기는 개별적 운용보다는 다수의 무인항공기를 동시에 사용하여야 신속하고 효과적인 정찰 임무를 수

† Received : October 11, 2018 Revised : February 26, 2019 Accepted : March 20, 2019
[1] Senior Researcher
[1] Corresponding author, E-mail : zoony777@gmail.com

행할 수가 있다. 하지만 다수의 무인항공기를 동시에 운용할 경우 강화된 조종통제 능력이 요구되므로 지상통제 인원이 증가될 수 있다. 따라서 다수의 소형 무인항공기는 개별적 비행통제보다는 스스로 비행하는 자율비행법칙이 적용되어야 하며 최소한의 자원을 활용한 효율적인 운용이 가능토록 할 필요가 있다. 특히 미국에서는 다수의 소형 무인항공기를 효율적으로 운용하기 위한 연구가 활발하게 진행되고 있으며, 효율적 자율비행정찰을 위한 플로킹(Flocking) 이론을 바탕으로 한 비행법칙 등의 연구결과를 볼 수 있다.

본 논문에서는 기존의 자율비행법칙을 이용하여 무인항공기가 표적을 탐지하고 표적의 종류를 판별하는 알고리즘과 여러 가지 비행조건을 비교함으로써 가능한 효율적인 비행운영조건을 도출하기 위한 비행시뮬레이션을 제시하였다.

자율비행법칙에 적용한 플로킹 기법의 가장 큰 장점은 기존의 큰 무인기 운용보다 비용이 적게 들며 통제인원이 덜 필요하다는 것이다[1]. 그리고 목표물을 식별 및 판단하는데 있어 연료 및 배터리 용량에 따른 비행시간과 영상카메라 해상도에 따른 목표물 식별 능력 등을 고려하여 비행체간의 충돌을 최소화하면서 가장 효과적이고 효율적으로 목표지역의 임의의 목표물에 대한 정찰 임무를 수행할 수 있다.

II. 본 론

2.1 자율비행법칙

2.1.1 플로킹(Flocking) 이론

플로킹은 새와 같은 일종의 생물체의 무리행동을 모델링하는 기법으로 1987년 Craig Reynolds가 처음으로 발표하였다[2]. 플로킹의 아버지로 불리는 Reynolds는 Fig. 1의 세 가지의 간단한 기본 규칙들(조타 행동들)을 이용하여 보이드(boid)라고 불리우는 존재들이 생물과 비슷한 집단행동을 취하게 됨을 보여주었다.

* 분리(Separation) : 주변 보이드들과 충돌하지 않도록 방향을 돌림

* 정렬(Alignment) : 주변 보이드들과 같은 방향을 가리키도록 함

* 응집(Cohesion) : 주변 보이드들과 평균 위치 쪽으로 방향을 돌림

* 회피(Avoidance) : 주변의 장애물이나 적과 충돌하는 것을 피함

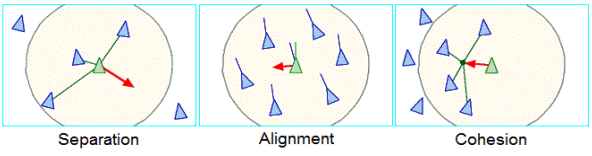

Fig. 1. Rules of Separation, Alignment, Cohesion [3]

회피(Avoidance) 규칙은 새롭게 추가된 규칙으로 지형 장애물 및 천적 무리로 인해 충돌이 발생할 경우 이를 피하기 위한 방법이다. 충돌을 피하기 위해서는 군집들이 자연스럽게 작은 군집들로 나뉘며 이렇게 떨어져나간 보이드들은 서서히 다른 주변 보이드들과 다른 군집을 형성하게 된다.

플로킹은 분산된 다수의 에이전트가 충돌없이 조화롭게 하나의 임무를 수행하도록 결집하는 행위이다. 이러한 장점으로 플로킹 기법은 컨센서스, 충돌 및 장애물 회피와 같은 다중 에이전트 시스템의 문제점을 해결하기 위한 유망한 기술로 간주된다[4].

2.1.2 자율비행법칙

플로킹 이론을 응용하여 무인항공기가 목표지역에서 임의의 목표물을 탐색하기 위한 자율비행법칙을 적용하였다. 자율비행법칙은 플로킹의 4가지 집단행동을 포함한 총 10개의 법칙(Rule)으로 구성하였고 10개의 비행법칙은 각각의 제어벡터를 생성한 후 각각의 제어벡터를 합함으로써 하나의 비행벡터를 형성한다[5]. 이렇게 형성된 비행벡터에 의해 무인항공기는 자율적으로 충돌을 방지하면서 정해진 구역 내에서 임의의 목표물의 탐색 임무를 수행한다.

- Rule 1(Separation) : 비행체 간에 거리가 일정거리 이내로 가까워지면 서로 밀치는 방향으로 제어벡터 생성

- Rule 2(Velocity Matching) : 비행체 간에 거리가 가까우면 주변 무인항공기의 평균속도에 맞추는 제어벡터 생성

- Rule 3(Flock Centering) : 최소한의 비행무리를 만들기 위한 법칙으로 일정거리 이내 주변의 비행체들의 중심위치(위치좌표의 평균값)로 향하는 제어벡터 생성

- Rule 4(Target/Waypoint Repulsion) : 비행체가 표적(Target)이나 웨이포인트(Waypoint)를 중심으로 일정한 각도로 선회비행하는 제어벡터 생성

- Rule 5(Target/Waypoint Attraction) : 비행체가 표적이나 웨이포인트로 이동하는 제어젝터 생성

- Rule 6(Stay within Boundaries) : 정해진 임무지역 바깥으로 비행하지 못하도록 통제하는 제어벡터 생성

- Rule 7(Communication Relay) : 각 비행체가 통신중계기가 되어 통신범위를 확장하는 제어벡터 생성

- Rule 8(Obstacle Avoidance) : 이미 알고 있는 비행 장애물이 있을 경우 그 장애물을 피하도록 하는 제어벡터 생성

- Rule 9(Divergence) : 타 비행체가 가까이 있을 경우 탐색의 효율성 증가를 위해 서로 발산하는 방향으로 제어벡터 생성

- Rule 10(Wander) : 비행방향과 영상카메라 조사방향을 빈번하게 변경하여 다양한 방향으로 표적을 탐색하도록 제어벡터 생성

Table 1. Binary Confusion Matrix

	Encountered Object	
Declared Object	Target	Non-Target
Target	P_{TR}	$P_{FTA/E}$
Non-Target	$1-P_{TR}$	$1-P_{FTA/E}$

Table 2. Multi Target Type Confusion Matrix

	Encountered Object		
Declared Object	TP1	TP2	Non-Target
TP1	$P_{TR1/TP1}$	$P_{TR1/TP2}$	$P_{FTA1/E}$
TP2	$P_{TR2/TP1}$	$P_{TR2/TP2}$	$P_{FTA2/E}$
Non-Target	$1-(P_{TR1/TP1}+P_{TR2/TP1})$	$1-(P_{TR1/TP2}+P_{TR2/TP2})$	$1-(P_{FTA/E}+P_{FTA/E})$

2.1.3 혼돈행렬(Confusion Matrix)

정찰용 무인항공기가 하나의 표적을 발견했을 경우 자율표적인지(ATR, Autonomous Target Recognition) 기능에 의해 표적의 타입(Type)을 자동으로 판단하게 되는데, 자율표적인지는 사람의 간섭 없이 목표물을 탐지, 분류, 인지, 식별하는 기계적 기능을 말한다. 자율표적인지는 그 능력에 따라 어느 정도의 오류가 발생될 확률이 존재하며 표적으로 인지하지 못할 수도 있다[6]. 진짜 표적(Real Target)을 발견했을 때 진짜 표적으로 판단할 확률을 P_{TR},(Probability of target report), 진짜가 아닌 가짜 표적(False Target)을 진짜 표적으로 판단할 확률을 $P_{FTA/E}$(Probability of false target attack given false target encounter)라 했을 때, 한 가지의 타입만 있는 표적의 경우 Table 1과 같이 2×2 행렬(Matrix)로 표현할 수 있고 표적의 타입이 여러 가지 일 경우 Table 2와 같이 확대하여 표현할 수 있다. 이때 각 열의 합은 1이 되어야 한다[7].

2.2 상태전이 및 비행법칙 할당

2.2.1 무인항공기 상태전이를 통한 표적 탐색 알고리즘

목표지역에서 표적 탐색을 위한 비행 알고리즘은 Fig. 2와 같이 비행체의 상태(State)로 정의하였다. 비행 알고리즘은 State 0 '비행체 이륙'으로 시작하여 State 1 '이륙 후 분산비행 및 표적 탐색', State 2 '표적 식별 및 판단', State 3 '귀환', State 4 '귀환 장소 선회비행 및 착륙', State 5 '추락'으로 마무리 된다. 다만 각 State에서 언제라도 충돌이 발생할 경우 State 5 '추락'으로 변경되어 더 이상의 탐색비행 임무에서 제외된다.

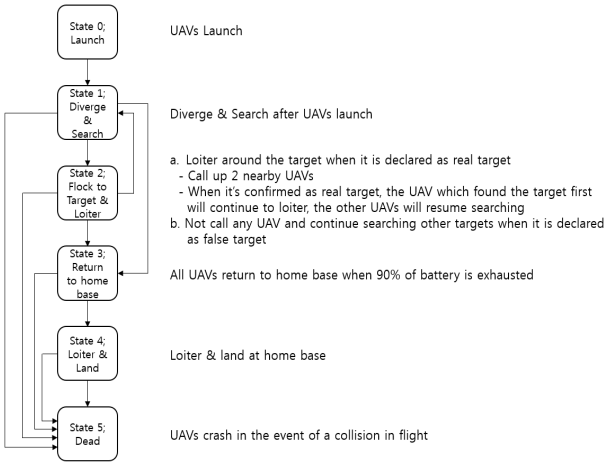

Fig. 2. State Transition Diagram

각 State별 정의는 다음과 같다.

- State 0 : 이륙(Launch)
  각 무인항공기 이륙과 함께 시뮬레이션 시작
- State 1 : 발산 및 탐색(Diverge & Search)
  이륙한 무인항공기는 State 1로 즉시 변경되고 Fig. 3과 같이 표적을 찾기 위해 발산하듯 비행을 시작함.
- State 2 : 표적으로 이동 및 선회비행(Flock to Target & Loiter)

State 2는 State 2-1, 2-2, 2-3으로 나눌 수 있다. State 1인 무인항공기가 하나의 표적을 찾으면 혼돈행렬에 따라 표적이 진짜 표적인지 가짜 표적인지 판단하고 State 2-1로 전이된다. 만약 그 표적을 진짜 표적(Real Target(T1 또는 T2))으로 판단할 경우 State가 1인 가장 가까이 있는 타 무인항공기를 최대 2대까지 그 표적으로 호출한다. 반대로 가짜 표적으

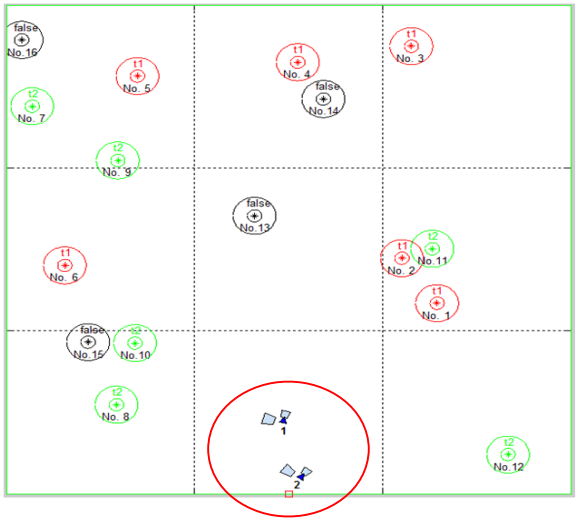

Fig. 3. A Scene of Diverge & Search After Launch

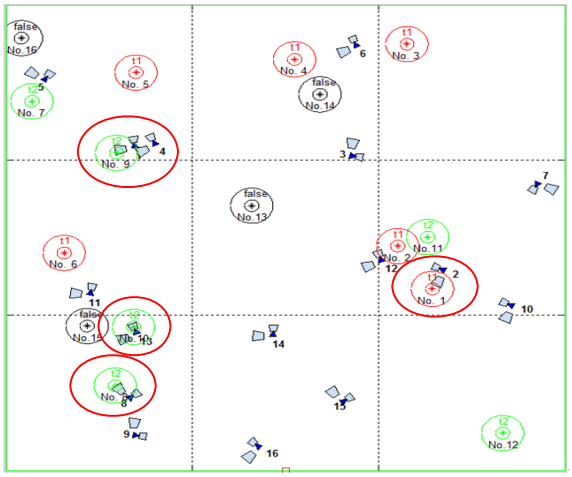

Fig. 4. A Scene with Some Real Target Confirmed

로 판단할 경우 다시 State 1으로 돌아가며 그 표적은 무시된다. 처음 표적을 발견한 무인항공기가 타 무인항공기를 호출하면 호출하거나 호출당한 무인항공기는 모두 State 2-2가 되고 표적을 탐색하여 표적의 종류를 판단한다. 만약 최소 2대 이상의 무인항공기가 동일한 표적 타입으로 판단하면 그 표적은 2대 이상의 무인항공기가 판단한 표적의 종류로 확정된다. 그 표적이 진짜 표적으로 확정된 경우 호출한 무인항공기는 State 2-3이 되며 그 표적 주변을 계속해서 선회비행을 하고 나머지 무인항공기는 State 1으로 돌아가며 그 표적을 벗어나 다른 표적을 찾기 위해 비행을 계속한다. 반대로 그 표적이 가짜 표적으로 확정된 경우 호출하거나 호출당한 모든 무인항공기는 다시 State 1로 변경되며 다른 표적을 찾기 위한 비행을 계속한다.

- State 3 : 귀환(Return to homebase)

배터리의 90%를 소모하였을 경우 모든 무인항공기는 Fig. 5와 같이 지정된 웨이포인트로 귀환한다.

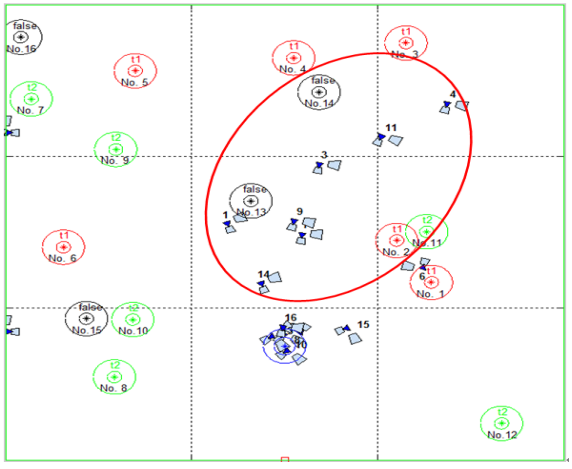

Fig. 5. A Scene of Return to Homebase

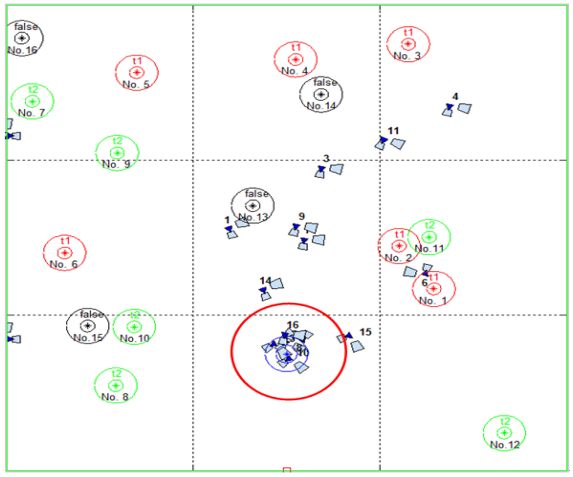

Fig. 6. A Scene of Loiter & Landing Ready in Homebase

- State 4 : 선회비행과 착륙(Loiter & Land)

웨이포인트로 귀환한 무인항공기는 선회비행을 하다가 착륙을 한다.

- State 5 : 추락(Dead)

무인항공기가 어떤 state에서라도 서로 충돌하여 추락한 경우 Fig. 7과 같이 그 무인항공기는 시뮬레이션의 좌측으로 이동되고 더 이상 시뮬레이션에 영향을 미칠 수 없게 된다.

2.2.2 상태별 비행법칙 할당

무인항공기 비행 임무 중 State별 비행법칙이 Table 3과 같이 할당되어 적용된다. State별로 법칙이 '1'이면 활성화 된 것이고 '0'이면 활성화가 되지 않은 것이다. 예를 들어, 무인항공기가 state 1인 경우 법칙 1, 2, 3, 6, 7, 8, 9, 10에 의한 제어벡터를 생성하고 이들 제어벡터의 합에 의해 비행경로가 결정된다. 즉, state 1에서는 표적을 탐지하지 못한 상태에

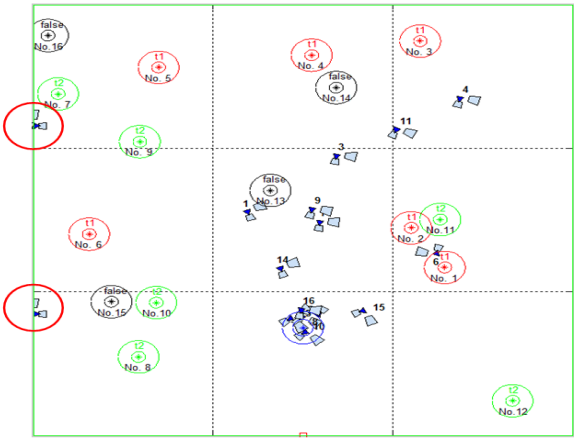

Fig. 7. A Scene in Which 2 UAVs Were Excluded from The Simulation Due to Crash after Collision

Table 3. Rule Allocation by States

State \ Rule	1	2	3	4	5	6	7	8	9	10
1: Diverge & Search	1	1	1	0	0	1	1	1	1	1
2: Flock to Target & Loiter	1	1	0	1	1	1	1	1	0	0
3: Return to homebase	1	1	0	1	1	1	1	1	0	0
4: Loiter(& Land)	1	1	0	1	1	1	1	1	0	0
5: Dead	0	0	0	0	0	0	0	0	0	0

서 표적탐색을 위한 자율비행 상태이므로 표적으로 이동하여 선회비행을 하도록 하는 법칙 4와 5가 적용되지 않는다. 그리고 state 2, 3, 4의 경우는 무인항공기가 표적이나 해당부대로 최대한 신속하게 이동하는 상태이므로 법칙 3에 의해 주변 무인항공기와 무리를 만들거나 법칙 9, 10에 의해 발산이나 빈번한 방향전환이 필요 없다.

State 2, 3, 4의 법칙이 동일하게 적용되는 부분은 state에 따라 각 법칙에서 적용하는 선회비행 반경을 달리 할 수 있으므로 따로 구분해 놓았으며 state 2는 state 2-1, 2-2, 2-3을 모두 포함한다.

2.3 비행 시뮬레이션

2.3.1 시뮬레이션 개발 목적

본 시뮬레이션은 임의의 임무 지역에서 적의 다양한 표적 타입의 정보를 효과적이고 효율적으로 획득하기 위한 무인항공기의 운용 조건을 도출하는데 있다.

2.3.2 시뮬레이션 로직

시뮬레이션은 매트랩(Matlab) 프로그래밍 언어로 개발되었다. 소스코드에는 무인기 대수, 타입별 표적의 수, 비행고도, 비행법칙의 가중치, 타임스텝(time step)의 길이와 같이 거의 모든 시뮬레이션 파라메타들이 초기화되는 형상(Configuration) 함수를 포함한다. “Main”이라는 이름의 시뮬레이션 제어함수는 시뮬레이션을 위한 원 타임 스텝(one time step)을 대표하는 일련의 이벤트를 반복적으로 발생시켜 전체 시뮬레이션이 작동할 수 있도록 통제한다. 시뮬레이션에서 각 무인항공기는 자신의 현재 위치, 속도, 선회율(Turn Rate), 포메이션(Formation) 그리고 배터리 상태 등의 정보를 포함하고 통신범위 이내의 다른 비행체와 자신의 정보를 공유한다. 이러한 정보를 바탕으로 각 비행체는 가중치가 적용된 비행법칙에 따라 다음 타임 스텝에서 요구되는 위치로의 벡터를 계산하고 그 지점으로 도달하기 위한 가속도를 즉각 적용한다.

표적은 진짜 표적(붉은색 Type1(T1), 초록색 Type2(T2))과 가짜 표적(검은색 Type(False))으로 구분하여 시뮬레이션에 사전에 입력이 가능하며 무인기가 탐색 비행 중 임의의 표적을 발견하면 표적의 타입을 혼돈행렬에 의해 판별한다. 발견된 표적은 고정되어 있거나 이동 중일 수 있으며 이동 중인 표적을 발견하였을 경우 이동 중인 표적을 따라가며 표적판별 임무를 수행한다. 이때 최초 발견한 비행체가 발견한 표적을 진짜 표적으로 판별한 경우 가장 가까운 2대의 다른 무인기를 호출하고 최종적으로는 3대의 무인기의 판별 결과에 따라 2대 이상이 동일하게 판별한 표적 타입을 최종 표적 타입으로 결정한다.

2.3.3 표적 판별 능력 설정

미 육군의 통신-전자사령부(CECOM, Communications-Electronics Command) 산하의 야간투시경전자센서부(NVESD, Night Vision and Electronic Sensors Directorate)에서 1980년 초기부터 현재까지 75개 이상의 자율표적인지 시스템을 평가한 결과에 따르면 자율표적인지 시스템 탐지능력의 주요 결정요소는 시스템의 신호대잡음비, 해상도 그리고 클러터 수준이다. 여기서 클러터는 표적과 상관없는 나무나 바위 등 모든 물체를 말한다[6]. 참고문헌[6]에 따르면 자율표적인지 시스템의 탐지능력은 신호대잡음비가 증가함에 따라 비례적으로 상승하다가 신호대잡음비가 어느 수준에 도달하면 일정한 성능을 유지한다. 그리고 해상도의 경우도 비례적으로 탐지능력이 증가하는 경향이 있으나 클러터 수준에 따라 해상도가 더 좋더라도 탐지능력은 떨어지기도 한다.

무인항공기에 장착된 영상카메라는 고도가 증가함에 따라 지상으로부터의 거리는 증가하여 지상에 대한 해상도는 지수함수적으로 떨어지므로 자율표적인지 기능에 따른 표적 타입에 대한 판별능력은 떨어지며, 때로는 가짜 표적을 진짜 표적으로 또는 진짜 표적을 가짜 표적으로 오인하는 경우를 고려하여 임의로 표적 판별 능력을 Tables 4, 5, 6과 같이 혼돈행렬로 구성하였다. 이는 영상카메라를 포함한 자율표적인지 시스템의 실제 성능에 따라 결정될 사항이지만 여러 가지 가변 요소에 따라 달라지는 판별능력의 특성을 고려하면 그 값을 특정할 수가 없으므로 이 시뮬레이션에서는 임의의 판별능력을 설정하였다.

Table 4. Confusion Matrix at altitude 300 ft

	T1	T2	False
T1	0.9	0.07	0.03
T2	0.07	0.9	0.03
False	0.03	0.03	0.94

Table 5. Confusion Matrix at altitude 500 ft

	T1	T2	False
T1	0.8	0.15	0.1
T2	0.15	0.8	0.1
False	0.05	0.05	0.8

Table 6. Confusion Matrix at altitude 1,000ft

	T1	T2	False
T1	0.7	0.2	0.15
T2	0.2	0.7	0.15
False	0.1	0.1	0.7

2.3.4 오차 설정

실제 비행을 최대한 모사하기 위하여 여러 가지 오차를 시뮬레이션에 적용하였다. 즉, 비행체의 위치 오차, Heading 각도 오차, GPS 오차에 의한 표적 위치정보 오차가 적용되었다. 이런 오차들은 무인항공기의 방향, 속도 그리고 표적 탐색에 일부 영향을 미칠 수 있다.

2.3.5 Hit 및 Nearmiss 설정

다수의 무인기가 표적 인근에서 군집비행을 할 경우 서로 충돌하여 추락할 수도 있고 충돌하지는 않지만 서로간의 거리가 매우 가까워 충돌위험이 발생할 수 있다. 따라서 비행의 안정성을 평가하기 위한 충돌과 충돌위험의 발생 횟수를 모니터링 할 수 있도록 시뮬레이션을 구현하였다. 시뮬레이션에서 충돌은 Hit로, 충돌위험은 Nearmiss로 명명하였다. Hit의 조건은 가장 가까운 두 대의 무인항공기의 중심간 거리가 비행체의 날개길이(Wingspan)보다 가까울 경우 Hit로 정의했으며, Nearmiss의 조건은 두 대의 무인항공기의 중심간 거리가 비행체의 Wingspan의 1~3배 거리에 있는 경우로 설정하였다.

2.4 시뮬레이션 결과

2.4.1 시뮬레이션 조건

전체 표적 수를 임의로 16개로 설정하였다. 표적의 각 타입 별로는 T1 6개, T2 6개, False 4개로 진짜 표적은 총 12개이고 가짜 표적은 4개로 설정하였다. 비행체의 수와 비행고도는 Table 7과 같이 비행체 수 12, 16, 20대 각각에 대해 비행고도를 300, 500, 1000ft로 가변하면서 시뮬레이션을 수행하였다. 각각의 시뮬레이션 조건별로 50회의 반복 시험을 수행하였고, 배터리 용량에 따른 시뮬레이션 시간동안에 비행체의 Hit, Nearmiss 수, 진짜표적탐색율(Real Target Coverage) 그리고 진짜 표적에 대한 판별오류 정도(False Target Declaration/Real Target Covered)를 확인하였다.

Table 7. Simulation Conditions

Task No.	UAVs (aircraft)	Altitude (ft)	Tests (times)
1	12	300	50
2	12	500	50
3	12	1,000	50
4	16	300	50
5	16	500	50
6	16	1,000	50
7	20	300	50
8	20	500	50
9	20	1,000	50

2.4.2 시뮬레이션 결과

시뮬레이션 결과는 각 시뮬레이션 조건별로 50회의 반복 시험을 수행한 결과를 평균과 95% 신뢰도 구간을 적용하여 그래프에 표현하였다. 확인 결과 Fig. 8과 Fig. 9와 같이 비행체의 수가 증가할수록 평균 Hit 수와 Nearmiss 수가 증가하였고, 비행고도가 높을수록 영상카메라의 조사범위가 넓어져 표적이나 웨이포인트에서의 선회비행 반경이 길어져 Hit와 Nearmiss는 감소하는 경향이 보였다. 그러나 무인기가 12대에서 16대로 증가하였을 때보다 16대에서 20대로 증가하였을 경우 500ft 고도에서는 Hit 수가 대폭 증가하였고 1,000ft 고도에서는 Nearmiss 수가 대폭 증가한 점은 한정된 임무영역 내에서 무인기 수가 어느 수준 이상으로 증가하면 충돌의 위험성이 급격히 커질 수 있음을 의미한다.

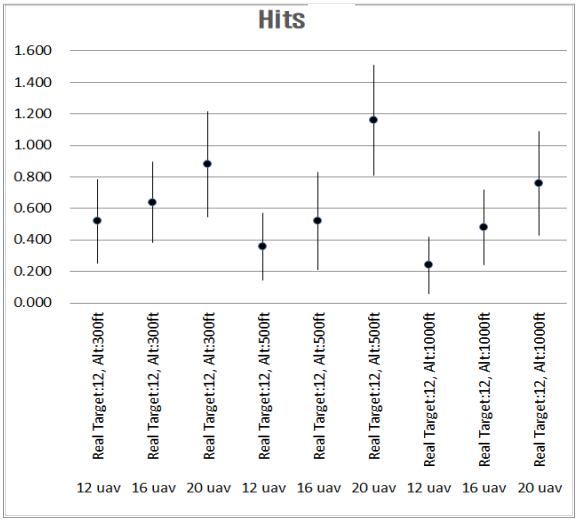

Fig. 8. Average Number of Hits

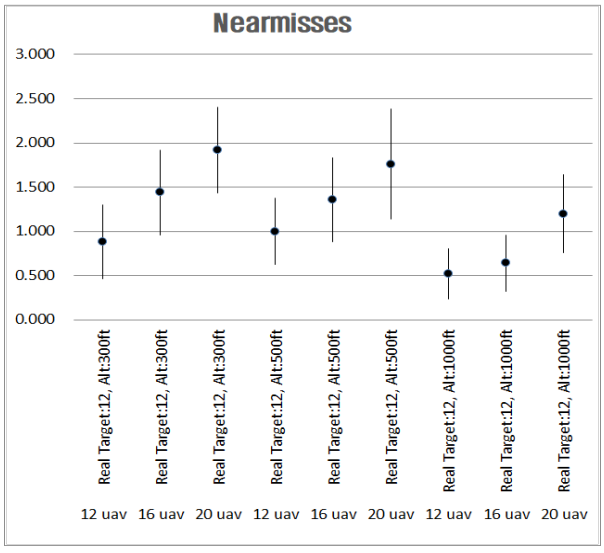

Fig. 9. Average Number of Nearmisses

Figure 10에서는 비행체의 수가 증가할수록 또는 비행고도가 높을수록 진짜표적 탐색율이 유의하게 증가함을 알 수 있다. 또한 300ft 고도에서는 무인기 대수 증가에 비례하여 진짜표적 탐색율이 증가하였으나, 500ft 및 1,000ft 고도에서는 무인기가 12대에서 16대로 증가하였을 때보다 16대에서 20대로 증가하였을 경우 진짜표적 탐색율의 증가율이 감소한 것을 알 수 있다. 마찬가지로 한정된 임무영역 내에서 무인기 수가 어느 수준에 도달하면 진짜표적 탐색율이 감소할 수 있음을 의미한다.

반면, Fig. 11과 같이 고도에 따른 진짜 표적에 대한 판별오류는 혼돈행렬에서 고도가 높을수록 판별능력을 떨어뜨림으로써 고도가 높을수록 판별오류가 증가하였지만 유의한 차이는 없었다.

따라서 비행체 수가 많을수록 또는 비행고도가 높

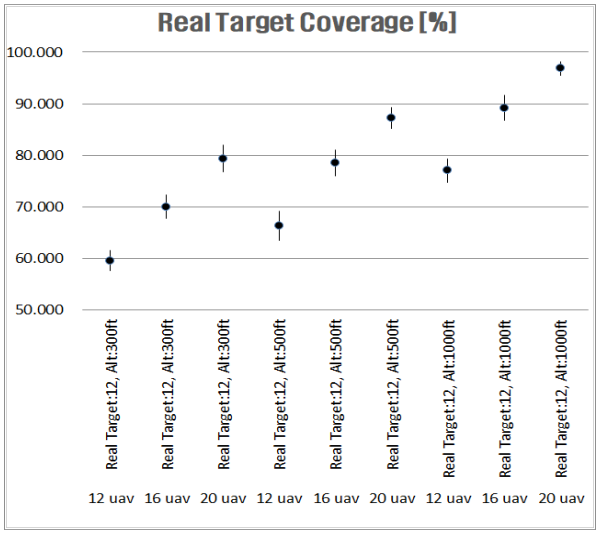

Fig. 10. Average of Real Target Coverage

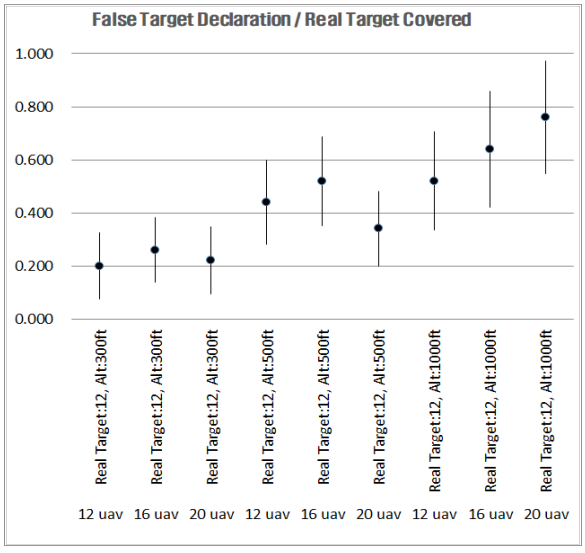

Fig. 11. The Average Number of False Target Declarations for Confirmed Real Targets

을수록 진짜 표적에 대한 탐색율은 증가하지만 비행체가 많아지면 충돌가능성이 높아지고 고도가 높으면 표적 판별력이 떨어진다. 대신에 고도가 높으면 충돌가능성도 감소하므로 실제 무인항공기 운영 시 충돌가능성과 표적 판별력을 상호 고려한 고도설정이 필요할 것으로 판단된다.

III. 결 론

플로킹 이론을 바탕으로 한 무인항공기 자율비행 시뮬레이션을 활용하여 임의의 표적 수에 따른 비행체 수 및 고도 등 특정 비행 조건하에서 비행체의 충돌을 최소화 하면서 목표지역 내에서의 표적 탐색 임무를 얼마나 효율적으로 수행할 수 있는 지를 분석해 보았다. 본 시뮬레이션은 다수 무인항공기의 정찰임무 수행을 위한 가용한 자원 내에서 가장 효율적이고 효과적인 자원 운용의 솔루션을 제공할 수 있을 것으로 판단된다. 다만, 본 논문에서 제시한 시뮬레이션은 이미 알고 있는 고정된 장애물과 군집 내의 무인항공기에 대한 충돌회피 기능만 고려가 되었으며 임의의 이동 장애물에 대한 충돌회피 기능은 고려되지 않은 한계점이 있으므로 그에 대한 추가 연구가 필요할 것으로 고려된다.

References

1) Lambach, J. L., “Integrating UAS Flocking Operations with Formation Drag Reduction,” Master’s Thesis, Air Force Institute of Technology,

Wright Patterson AFB, OH, March 2014.

2) Reynolds, C. W., "Flocks, herds and school: A distributed behavioral model," ACM SIGGRAPH *Computer Graphics*, Vol. 21, No. 4, July 1987, pp.25~34.

3) Kwon, I. K., and Lee, S. Y., "Fuzzy flocking method for boid's ecosystem behavior modeling," *Proceedings of the Korean Institute of Information Scientists and Engineers,* Vol. 30, No. 2, October 2003, pp.73~75.

4) Xu, H., and Carrillo, L. R. G., "Distributed Near Optimal Flocking Control for Multiple Unmanned Aircraft Systems," *2015 International Conference on Unmanned Aircraft Systems(ICUAS)*, June 2015, pp.879~885.

5) Kaiser, J. N., "Effects of Dynamically Weighting Autonomous Rules in an Unmanned Aircraft System(UAS) Flocking Model," Master's Thesis, *Air Force Institute of Technology*, Wright Patterson AFB, OH, September 2014.

6) Ratches, J. A., Walters, C. P., Buser, R. G., and Guenther, B. D., "Aided and Automatic Target Recognition Based Upon Sensory Inputs From Image Forming Systems," *IEEE Transactions on Pattern Analysis and Intelligence*, Vol. 19, No. 9, September 1997, pp.1004~1019.

7) Schulz, C. S., Jacques, D. R., and Pachter, M., "Cooperative Control Simulation Validation Using Applied Probability Theory", *Theory and Algorithms for Cooperative Systems*, 2004, pp.481~498.