J. Korean Soc. Aeronaut. Space Sci. 48(6), 419-429(2020)
DOI:https://doi.org/10.5139/JKSAS.2020.48.6.419
ISSN 1225-1348(print), 2287-6871(online)

플로킹 이론 기반 자율정찰비행 무인항공기의 탐색성능 향상에 관한 연구

김대운[1], 석민준[2], 김병수[3]

A Study on the Improvement of Searching Performance of Autonomous Flight UAVs Based on Flocking Theory

Dae Woon Kim[1], Min Jun Seak[2] and Byoung Soo Kim[3]
Defense Agency for Technology and Quality[1,2], Gyeongsang National University[3]

ABSTRACT

In conducting a mission to explore and track targets using a number of unmanned aerial vehicles(UAVs), performance for that mission may vary significantly depending on the operating conditions of the UAVs such as the number of operations, the altitude, and what future flight paths each aircraft decides based on its current position. However, studies on the number of operations, operating conditions, and flight patterns of unmanned aircraft in these surveillance missions are insufficient. In this study, several types of flight simulations were conducted to detect and determine targets while multiple UAVs were involved in the avoidance of collisions according to various autonomous flight algorithms based by flocking theory, and the results were presented to suggest a more efficient/effective way to control a number of UAVs in target detection missions.

초 록

다수의 무인항공기를 이용하여 표적을 탐색 및 추적하는 임무를 수행하는데 있어서 무인항공기의 운용 대수, 비행고도 등 운용 조건뿐만 아니라, 각 비행체들이 어떤 알고리즘을 이용해 비행경로를 결정하느냐에 따라 그 임무에 대한 성과는 크게 달라질 수 있다. 다만 이러한 표적 탐색 임무에서 자율 비행 무인항공기의 운용 방법이 어떠할 때 가장 효과적이며 효율적인지에 대한 연구는 미흡한 상태이다. 본 연구에서는 플로킹 이론을 기반을 둔 다양한 자율비행 알고리즘을 활용하여, 다수의 무인 항공기가 서로 충돌을 회피하면서 표적을 탐지하는 임무를 기반으로 비행 시뮬레이션을 수행하고 그 결과를 분석하여, 표적 탐지 임무에서의 다수의 무인항공기를 제어할 수 있는 보다 효율적/효과적인 방안을 제시하였다.

Key Words : UAV(무인항공기), Flocking(플로킹), Autonomous Flight(자율 비행), Target Detection (표적 탐지), Collision Avoidance(충돌 회피), Simulation(시뮬레이션)

Ⅰ. 서 론

현재 드론을 포함하여 수많은 무인항공기가 배달을 포함한 수송, 탐색, 환경 및 재해 감시, 산업체 감시, 농업 등의 민간분야뿐만 아니라 전장감시, 표적획득 및 추적, 피해 상황 식별 및 대테러 임무 등 군사

† Received : January 18, 2020 Revised : April 14, 2020 Accepted : April 20, 2020
[1,2] Principal Researcher, [3] Professor
[3] Corresponding author, E-mail : bskim@gnu.ac.kr

용 목적으로 활용되고 있다[1].

또한 최근에는 경로 설계(path planning)를 포함한 자율 비행 알고리즘의 발전, 카메라 등 센서 및 네트워킹 기술, 충돌 회피에 대한 연구, 고성능 배터리를 통한 체공 시간 연장 등 여러 분야의 발전을 활용하여 무인 항공기를 단독으로 쓰기보다는 다수의 무인기를 군집으로 활용하여 임무의 효율성 및 효과성을 높이고 있다[2].

이 중 저고도에서 운용되는 소형 정찰용 무인항공기를 군집으로 운용하여 도심[3], 산불 지역[4], 장애물 있는[5] 또는 없는 미확인 지역[6] 등 일명 관심 지역(area of interest)을 유전 알고리즘(neural network algorithm) 등을 이용하여 효율적으로 탐색하는 연구도 수행되고 있으며[7,8], 특히 플로킹(flocking) 이론을 자율 비행 알고리즘으로 활용하는 연구도 수행되고 있다[9-11].

본 논문에서는 다수의 고정익 무인항공기를 활용하여 정해진 관심지역에서 다수의 표적을 탐지하는 임무를 설정하여, 플로킹 이론을 기반으로 한 여러 가지 비행 패턴을 제안하고 시뮬레이션을 통해 각 성능을 도출하고 그 결과를 정리하였다.

본 논문은 크게 3개의 부분으로 구성된다.

우선 플로킹 이론 및 시뮬레이션 수행 대상이 되는 비행체 등에 대한 설명한 후, 이전 연구[10,11]에서 제시된 비행 패턴을 적용한 시뮬레이션 수행 결과를 정리하였다. 마지막으로 여러 가지 다양한 탐색 패턴의 시뮬레이션 및 플로킹 기반 군집 무인항공기의 수정된 탐색패턴을 소개 및 그 결과를 정리하고 본 논문에서 개선/제안하는 방식의 성능을 제시하였다.

II. 본 론

2.1 자율비행법칙

2.1.1 플로킹(Flocking) 이론

플로킹 이론은 새, 물고기 등과 같은 일종의 생물체의 무리 행동을 수학적으로 모델링하는 기법으로 분산된 다수의 에이전트가 충돌 없이 조화롭게 하나의 임무를 수행하도록 하는 것으로 Reynolds가 1987년 처음으로 발표하였다[12]. 이후 플로킹 기법은 협력, 충돌 및 장애물 회피와 같은 다중 에이전트 시스템의 문제점을 해결하기 위한 유망한 기술로 연구되고 있다. 특히 최근 다중 무인항공기에 대한 연구가 활발히 진행되면서 군집 드론의 위치 및 대형 제어[13], 군집드론 간 무선 네트워크 제어[14], 장애물 회피[15], 경계가 있는 환경(confined environments)에서의 최적 제어[16], 리더가 있는 군집 드론의 제어[17] 등 많은 연구가 진행되고 있다. 플로킹은 세 가지의 간단한 기본 규칙들(조타 행동들)을 이용하여 보이드(boid)를 생물과 비슷한 집단행동을 하도록 제어하였으며, 이후 지형 장애물 및 천적 무리로 인해 충돌이 발생할 경우 이를 피하기 위한 회피 규칙이 추가되어 4개의 규칙이 기본적으로 적용되고 있다.

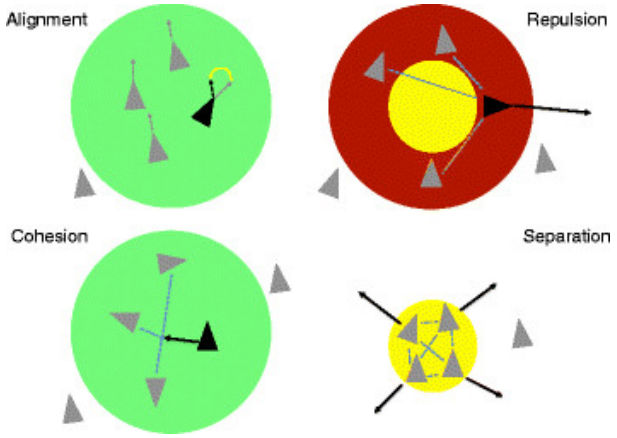

Fig. 1. Illustration of the Alignment, Repulsion Cohesion, Separation [18]

* 정렬(Alignment) : 주변 보이드들과 같은 방향과 속도를 유지하도록 작용

* 응집(Cohesion) : 무리가 이루어지도록 주변 보이드들과 평균 위치 방향으로 작용

* 분리(Separation) : 주변 보이드들과의 충돌 회피를 위해 각 보이드와 분리되는 방향으로 작용

* 회피(Repulsion) : 주위의 장애물과 충돌을 피하도록 작용

Figure 1은 각 기본 규칙을 도식화한 것이다.

2.1.2 자율비행 로직

플로킹 이론을 이용하여 다중 비행체의 자율 비행을 통한 표적 탐지 임무를 연구하기 위한 시뮬레이션은 매트랩(Matlab) 프로그래밍 언어로 개발되었다.

소스 프로그램은 크게 4개의 부분 - ① 초기값 설정-비행체 및 표적의 수, 비행 고도 및 시간, 전체 탐지영역의 크기 등 시뮬레이션 파라미터를 설정, ② 전체 시뮬레이션 제어하는 메인(main), ③ 플로킹 룰 계산-플로킹 이론을 응용하여 각 비행체의 다음 위치를 결정하는 자율비행 법칙을 적용, ④ 표적 확인-비행체의 카메라 FOV 내에 표적이 들어오면 표적임을 확인하는 로직으로 구성되며, 하위 40개의 프로그램 코딩 파일로 시뮬레이션을 구성하였다.

각 비행체는 정보를 바탕으로 각 비행체는 가중치가 적용된 플로킹 비행법칙에 따라 다음 타임 스텝의 속도 벡터를 계산하는데, 이때 자율비행법칙을 플로킹의 4가지 집단행동을 포함한 총 10개의 법칙에 따라 속도 벡터가 생성된다. 이후 이 벡터들의 총합이 다음 차례의 각 비행체의 속도 벡터이다. 이 값은 각 비행체의 속도, 가속도, 각속도 등에 대해 물리적 한계(limitation)를 적용하고 최종적으로 다음 스텝의 속도 벡터를 갱신하게 된다. 이렇게 구성된 비행 알

Table 1. Program Logic Discipline

name	Description
Initialize	Set the program initials
Main	Control the whole simulation between each step
Rule_calcu	Build flocking rule based on each UAV's position
Rule 1~10	Calculate the velocities of each rule
Target_decla	Check whether target is real when captured within camera's FOV
Confirm	Another UAV Confirms that target is true or false

고리즘에 의해 무인항공기는 분산되어(decentralized) 자율적으로 충돌을 방지하면서 정해진 구역 내에서 임의의 목표물의 탐색 임무를 수행할 수 있다.

Table 1은 프로그램의 로직 중 주요한 기능을 수행하는 파일에 대한 개략적인 설명이다.

2.1.3 플로킹 이론을 활용한 비행법칙 확장

다음은 10개의 비행법칙에 대한 설명이며 각 법칙에 대한 세부 수식은 kaiser 논문[9]에 세부적으로 제시되어 있다.

- Rule 1 분리(Separation) : 일정거리 내에 있는 비행체 간에 충돌을 방지하기 위해 상호 밀어내는 방향으로 벡터($\vec{u}_{1,j}$)를 생성.

- Rule 2 속도 매칭(Velocity Matching) : 군집 내 비행체 간 정렬을 위해 방향과 속력을 비슷하게 유지하는 벡터($\vec{u}_{2,j}$)를 생성.

- Rule 3 군집 센터링(Flock Centering) : 이웃 비행체간의 군집(비행무리)을 만들기 위해 비행체들의 중심위치로 향하는 벡터($\vec{u}_{3,j}$)를 생성.

- Rule 4 표적/경로점 반발(Target/Waypoint Repulsion) : 비행체가 표적이나 경로점에 가까울수록 지수 함수 형태로 밀어내는 벡터($\vec{u}_{4,j}$)를 생성. rule 5와 함께 비행체가 표적(Target)이나 경로(Waypoint)를 중심으로 일정한 각도로 선회 비행하도록 작용.

- Rule 5 표적/경로점 인력(Target/Waypoint Attraction) : 비행체가 표적이나 경로점에 가까울수록 지수 함수 형태로 당기는 벡터($\vec{u}_{6,j}$)를 생성.

- Rule 6 임무 지역 내 유지(Stay within Boundaries) : 정해진 임무 지역 내에서 비행을 유지하도록 통제하는 벡터($\vec{u}_{7,j}$)를 생성.

- Rule 7 통신 릴레이(Communication Relay) : 각 비행체가 통신 중계기가 되어 통신범위를 확장하도록 하는 벡터($\vec{u}_{7,j}$)를 생성.

- Rule 8 장애물 회피(Obstacle Avoidance) : 이미 알고 있는 장애물이 있을 경우, 그 장애물을 회피하도록 하는 벡터($\vec{u}_{8,j}$)를 생성.

- Rule 9 분산(Divergence) : 탐색의 효율성 증가를 위해 일정거리 이내의 비행체간 서로 발산하는 방향으로 벡터($\vec{u}_{9,j}$)를 생성.

- Rule 10 배회(Wander) : 비행방향과 영상카메라 조사방향을 빈번하게 변경하여 다양한 방향으로 표적을 탐색하도록 벡터($\vec{u}_{10,j}$)를 생성.

2.1.4 무인항공기 위치 및 속도 계산

앞서 설명한 10개의 rule을 통해 각 비행체의 다음번 속도 벡터를 생성하는 방식으로 플로킹 이론을 활용하여 다수의 비행체를 제어한다. 이때 j 번째 비행체의 새로운 위치 $\vec{q}_j(t+1)$는 식 (1)로 나타낼 수 있다.

$$\vec{q}_j(t+1) = \vec{q}_j(t) + \vec{p}_j(t+1)\triangle t \tag{1}$$

이때, $\vec{q}_j(t)$는 j 번째 비행체의 현재 위치 즉, $\vec{p}_j = \triangle\vec{q}_j$이며, $\vec{p}_j(t+1)$는 j 번째 비행체의 새로운 속도이다.

식 (1)은 다시 다음과 같이 풀어쓸 수 있다.

$$\vec{q}_j(t+1) = \vec{q}_j(t) + (\vec{p}_j(t) + \vec{u}_j(t+1))\triangle t \tag{2}$$

즉, $\vec{p}_j(t+1)$는 j 번째 비행체의 현재 속도 $\vec{p}_j(t)$에 새로운 속도 벡터 $\vec{u}_j(t+1)$가 업데이트된 값이며, 이때 $\vec{u}_j(t+1)$는 앞에서 설명한 10개의 rule에 의해 1 step 동안의 변화된 속도 벡터의 총합이며, 식 (3)과 같이 나타낼 수 있다.

$$\vec{u}_j(t) = \sum_{r=1}^{10} \omega_r u_{r,j} \tag{3}$$

식 (3)의 ω_r는 r 번째 법칙(rule)의 가중치(Table 3에 명시)이며, $\vec{u}_{r,j}$ 벡터는 j 번째 비행체의 r 번째 법칙 속도 벡터 변화량으로 이는 2.1.5절의 비행체의 속도 변화량의 실제 물리적 한계($\vec{u}_{real}$)보다 작다.

2.2 비행법칙 등 시뮬레이션 배경

2.2.1 비행체 및 탑재체

시뮬레이션에 활용될 비행체는 RQ-11B로서 날개길이가 1.5 m 미만인 소형 무인기이다. 통상 150 m 이하의 고도에서 시간당 40~80 Km(25~50 mile)의

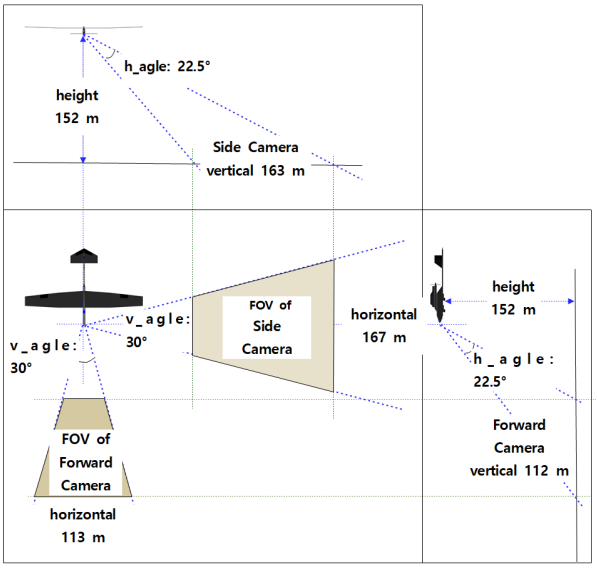

Fig. 2. FOV of RQ-11B's dual camera

속도로 60~90분 동안 비행할 수 있다. kaiser 논문[9]에서는 비행체의 가속 및 감속을 위한 가속도 성능은 초당 3.2 Km(2 mile), yaw축 회전 각속도는 10°/sec의 가속도 및 최대 속도 72 Km/h(45 mile/h)의 한계를 설정하였다.

비행체에 운용되는 탑재체는 전방 및 측면의 2중 전자광학카메라(dual EO camera)를 보유하고 있으며, 각 카메라의 시야각(Field of View)은 전방 및 측면카메라 공히 수평(30°), 수직(22°)이다.

또한 시뮬레이션 결과의 현실성을 추가하기 위해 탑재체의 판별 능력에 대해 각 카메라에 포착된 표적을 포착하지 못하는 에러(missing error)(표적이 카메라의 시야각 내로 포함되더라도 20%의 missing error를 추가하여 구현)를 이전 논문과 같이 80% 확률로 부여하였으며, 상기의 모든 수치는 시뮬레이션에 반영되었다. Fig. 2는 고도 150 m(500 ft)로 수평 비행하는 상황에서 수평 및 수직 시야각이 적용된 전방 및 측면의 시야각을 도시한 그림이다.

2.2.2 무인항공기 상태(state) 변경

본 연구에서 각 비행체는 상황 및 조건에 따라 비행체 상태(state)가 변경되면서 임무를 수행한다. 이륙(상태 0)한 항공기는 초기에 설정된 지점(waypoint)로 이동(상태 1)하고 이후 표적을 탐지하기 위해 비행(상태 2-1)한다. 표적을 탐지한 경우에는 표적 주위를 선회(상태 2-2)하며, 배터리 시간의 90%의 시간이 소요된 이후에는 정해진 귀환 지점으로 복귀(상태 3)하여 귀환지점에서 선회(상태 4)한다. 임무 중 어떠한 상태에서도 비행체끼리 충돌하는 경우 추락 상태로 전이되어 임무가 중지(상태 5)된다.

Table 2는 이륙 시점부터 비행체가 가지는 각 상태를 정리한 자료이다.

Table 2. States Discipline

State	Description
0: take-off	Take off every 30 seconds at the takeoff position
1: diverse	Move to the initial route, when the initial route is specified
2: detect & loiter	Target searching (2-1), loitering around the target (2-2)
3: return	Return to home base when 90% of battery time
4: landing	loitering around the return point
5: dead	Dead when colliding with another vehicle

Table 3. Rule Allocation by states

State \ Rule	1	2	3	4	5	6	7	8	9	10
0: take-off	1	1	1	0	0	1	1	1	1	1
1: diverse	1	0	0	1	1	0	1	0	0	1
2: detect & loiter	1	1	0	1	1	1	1	1	0	0
3: return	1	1	0	1	1	1	1	1	0	0
4: landing	1	1	0	1	1	1	1	1	0	0
5: dead	0	0	0	0	0	0	0	0	0	0

2.2.3 각 상태별 비행법칙 적용

이전 연구[11]에서는 Table 3과 같은 비행법칙을 적용하였다. 각 상태에서는 이전 절(2.1.3)의 Rule 1~10까지의 각 가중치 w_r을 1과 0으로 명시하였다.

III. 비행 시뮬레이션 수행 및 결과

3.1 시뮬레이션 공통 조건

본 논문에서는 자율정찰비행 무인항공기의 비행운용 조건에 따른 표적의 탐지 확률, 평균 탐지 시간, 충돌 및 충돌할 뻔한(이하, Near miss) 수를 정량적으로 시뮬레이션하고 그 결과를 확인하고자 한다.

비행 시뮬레이션은 다음과 같다. 각 비행체는 30초 간격으로 이륙하여 1개 또는 2개소의 초기 위치로 이동(이전 논문[11]과 같이 본 논문에서도 초기 경로로의 이동은 미설정하여 상태 1은 미실행)하며, 주어진 탐지 영역 내에서 비행하며 표적을 탐지를

시작한다. 비행고도는 500ft로 고정이며, 시뮬레이션 시간은 비행시간은 60분, 마지막 6분 동안은 지정된 위치로 귀환한다.

비행체가 탐지할 표적은 이동하지 않는 고정형 표적 3개이며, 전체 임무 영역은 가로, 세로 각 약 4.8 Km(3 mile)인 정사각형으로 설정하였다.

만약 1개의 비행체가 표적을 탐지하면 인접한 2개의 비행체를 호출하고 이 2개의 비행체중 최소 1개가 표적임을 재확인하여 실제 표적으로 확인되며, 이후 최초 표적을 발견한 비행체는 시뮬레이션이 끝날 때까지 표적 상공을 선회하여 표적을 계속 감사하고 호출된 2개의 비행체는 계속 탐색 임무로 복귀한다. 각 비행체는 비행 중 다른 비행체와 충돌이 발생할 수 있는데, 두 비행체간의 최단 거리가 날개 크기(1.5 m)의 1배 이내로 근접하는 경우 방향에 상관없이 충돌 발생으로 판정하고 상태 5로 변경하고 비행을 중지한다. 날개 크기의 3배 이내로 근접할 경우 Near miss로 판정하고 그 비행체는 계속 임무를 수행하도록 설정하였다.

각 비행 알고리즘의 시뮬레이션은 각 조건에서 100회 수행하며 그 결과를 분석하였다.

3.2 플로킹 이론을 적용한 비행법칙 적용

3.2.1 이전 시뮬레이션 수행 배경

Figure 3은 이전 연구[11]에서 사용된 비행법칙을 적용한 3개의 비행체의 궤적 전체를 평면도에 투사한 모습이다. 서론에서도 명시한 바와 같이 플로킹 알고리즘은 각 비행체는 현재의 자기 및 타 비행체의 위치를 기반으로 각 비행체 다음 스텝의 속도 벡터를 생성하게 되는데, 너무 근접한 비행체에 대해서는 멀어지며(rule 1) 주어진 탐색범위(4.8*4.8 Km(3*3 mile)) 내에서(rule 6) 배회하며(rule 10) 이동하게 된다. 이러한 rule들에 의한 경로의 선택은 Fig. 3과 같은 비행체의 무작위적인 궤도를 가지며, 다음 속도 벡터 생성 시 자기 및 타 비행체의 이전 위치를 고려하지 않으므로 한 구역에만 집중적, 중복적인 탐색이 발생할 가능성과 함께 비행체가 많아질수록 급격히 충돌 가능성이 높아질 수 있다는 예측이 가능하다.

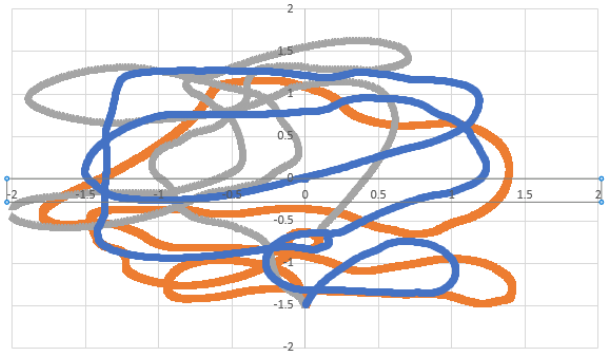

Fig. 3. Trajectories of original Flocking Algorithm

Table 4. Results of original flocking algorithm

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	82.3	85.0	89.6
AVER. detection Time(sec)	884	805	761
PROB. of Collision(%)	12	18	20
PROB. of Near miss(%)	15	23	41

3.2.2 기존 시뮬레이션 수행 결과 및 분석

Table 4는 이전 연구와 동일한 비행 알고리즘을 적용하여 시뮬레이션 수행한 결과를 정리한 것이다. 비행체는 각각 8, 10, 12개를 운용하여 60분 동안 다수의 표적을 탐지하는 상황을 반복적으로 100회 시뮬레이션 수행한 결과를 누적하고 그 평균을 구하였다.

표의 각 항목은 다음과 같다. 본 논문의 모든 시뮬레이션에서는 공통으로 총 3개의 표적을 설정하므로 탐지 수는 100회 시뮬레이션에서 얻어진 평균 탐지 개수이며, 탐지 확률을 백분율을 병기하였다. 탐지 평균 시간은 각 표적이 첫 번째 비행체에 의해 탐지되고 및 주위의 비행체에 의해 재확인(confirm)되어 표적으로 확정된 시각의 평균을 계산하였다(탐지하지 못한 표적이 발생하는 경우, 해당 표적의 탐지 시각은 총 비행시간(1800 sec)로 대체하였다). 충돌 및 Near miss 발생 확률 역시 100회 반복 시뮬레이션 동안 발생한 총 충돌 횟수 및 Near miss 발생 횟수를 누적하여 총 시뮬레이션 횟수(100회)로 나눈 값으로 1회 임무 당 충돌 및 Near miss 발생 확률이다.

Table 4에서의 시뮬레이션 결과는 앞서 비행체수가 많아질수록 탐지성능은 향상(탐지된 표적 수는 많아지고, 탐지에 소요된 평균 시간은 작아짐)되는 반면, 충돌 가능성(충돌 및 Near miss 확률)은 커짐을 알 수 있다.

비행체 12개의 시뮬레이션의 경우 총 표적의 90%에 근접하는 탐지 확률을 보이는 반면 비행체 8대인 경우 탐지 확률은 82%까지 떨어진다.

충돌 가능성의 경우, Table 4의 마지막 12개의 비행체가 총 100번의 시뮬레이션 중 20회의 충돌 및 41회의 Near miss가 발생하여 각각의 단일 임무에서 충돌 확률 20% 및 Near miss 확률 41%가 된다. 반면 비행체 수가 작은 경우 충돌 및 Near miss 확률이 작아짐을 볼 수 있어, 탐색에 동원되는 비행체 수가 클수록 충돌 가능성이 커짐을 알 수 있다.

3.2.3 기존 플로킹 이론 적용과 비행 영역 분할

앞 절의 시뮬레이션 결과는 탐지 비행체 수를 증가 시킬수록 탐지 확률은 높아지고 평균 탐지시간을

줄어드는 대신, 충돌 및 Near miss 확률이 비약적으로 증가함을 볼 수 있으며, 이는 운용되는 비행체의 수에 따라 탐지 성능과 충돌 가능성은 비례하는 결과를 보임을 알 수 있다.

이는 앞서도 분석한 바와 같이 기존의 플로킹 이론에 기반을 둔 자율 비행 알고리즘의 경우, 각 비행체의 탐지 구역을 미리 설정하지 않고 전체 구역을 임의로 비행하며 탐지하기 때문으로 분석된다.

이에 따라 플로킹 이론을 자율 비행법칙으로 그대로 사용하되 각 비행체의 탐지 구역을 임의로 구분하여 배정하면, 탐지 성능은 유지하면서 충돌확률을 줄어 들 수 있는지 확인을 위해 아래와 같이 비행영역을 임의로 구분하여 할당하는 2개의 새로운 시뮬레이션을 수행하였다.

각 비행체의 임무 영역을 할당을 제외하고 이전 시뮬레이션과 모든 조건은 동일하게 두되, 첫 번째는 탐지 구역을 아래, 위로 2개로 구분(가로 4.8 * 세로 2.4 Km의 2개의 탐지구역으로 설정)하고 각 탐지 구역에 절반에 해당하는 비행체를 각각 할당하였으며, 두 번째로는 각 4분면으로 구분(2.4*2.4 Km의 4개의 탐지구역으로 구분)하여 총 비행체를 4등분하여 탐지 구역을 설정하고 시뮬레이션을 수행하였다.

Table 5 및 Table 6은 각 시뮬레이션 결과이다.

Table 5. Results of Target box 2 division

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	77.3	84.6	88.3
AVER. detection Time(sec)	933	850	802
PROB. of Collision(%)	6	6	12
PROB. of Near miss(%)	12	21	25

Table 6. Results of Target box 4 division

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	81.3	84.6	89
AVER. detection Time(sec)	850	853	780
PROB. of Collision(%)	6	5	11
PROB. of Near miss(%)	9	13	18

시뮬레이션 결과 전체 탐색 구역을 단순히 2분할 또는 4분할하여 각 비행체에 분할된 탐지 구역을 설정하면, 영역을 분할하지 않은 이전 결과에 비해 충돌 및 Near miss 확률은 절반 이하로 낮아짐을 알 수 있다. 그러나 탐지 성능은 변함이 없거나 오히려 약간 낮아지는데, 이는 기존 영역에 비해 각 비행체가 할당된 탐지 구역이 작아지면서 전체 임무 영역(4.8*4.8 Km) 밖으로 벗어나 있는 시간이 길어지고, 또한 이에 따라 각 비행체의 속도가 낮아지기 때문(임무 범위 밖으로 나간 비행체의 경우 rule 6에 의해 감속하기 때문)으로 추론할 수 있으며, 즉 단순히 임무 영역을 세분화하는 경우 충돌 성능이 일부 좋아질 수 있음을 알 수 있었다.

3.2.4 case study - Rule1 parameter 최적화

위 절에서는 비행법칙을 변경하지 않은 상태에서 단순히 임무 영역을 세분화한다고 성능이 향상되지 않음을 알 수 있었다. 이어서 Rule1 분리하는 벡터를 최적화함으로써 성능 향상이 가능한지 확인하였다.

j 번째 비행체의 rule 1에 의해서 만들어지는 속도 벡터 $\vec{u}_{1,j}$은 다음과 같다.

$$\vec{u}_{1,j} = \frac{c_{1a}}{|N_j(d_1)|}\left(\sum_{i\in N_j(d_1)} \phi(\vec{q}_j - \vec{q}_i)/\triangle t\right), j \neq i \qquad (4)$$

식 (4)의 d_1는 rule 1의 분리를 위한 거리(이하, separation size) 변수로 기존 연구에서 76 m(250 ft)로 설정되었다. $N_j(d_1)$은 j 번째 비행체와의 거리가 d_1보다 작은 타 비행체의 총 대수이며, c_{1a}, c_{1b}는 rule 1의 상수로서 이전 참조논문[9,11]에서 각각 1과 5280로 설정된 값을 우선 그대로 사용하였다.

이때 비행체 간 거리에 따라 발생되는 벡터 함수 $\phi(d)$는 아래와 같다.

$$\phi(d) = \begin{cases} c_{1b}\left(1 - \dfrac{d_{i,j}}{d_1}\right)^2, & i \in N_j(d_1) \\ 0 & , else \end{cases} \qquad (5)$$

즉, $\phi(d)$는 separation size 범위 밖에서는 항상 0이며, separation size 범위 이내일 경우 이웃된 비행체 간의 거리($d_{i,j}$) 및 separation size의 비율의 제곱에 반비례한다.

이에 따라 본 논문에서는 separation size, d_1 및 상수 c_{1b}을 아래와 같이 4개의 경우로 변경하여 기존 플로킹 이론과 비교하여 성능 개선이 있는지 확인해 보았다.

기존 시뮬레이션과 동일하게 비행체 8개, 10개 및 12개를 활용하였고, separation size를 기존의 약 1/2배 및 2배로, 벡터 크기를 좌우하는 상수 c_{1b} 역시 기존의 약 1/2배 및 1.5배로 설정하였다. 각 가중치(d_1, c_{1b})의 변동에 따른 성능의 영향성의 추이를 보기 위

해 임의로 선정하였으며, 시뮬레이션을 수행한 결과는 다음과 같다.

case1 : separation size, d_1 150, 가중치 c_{1b} 5820
case2 : separation size, d_1 600, 가중치 c_{1b} 5820
case3 : separation size, d_1 600, 가중치 c_{1b} 8200
case4 : separation size, d_1 600, 가중치 c_{1b} 3800

Table 7 ~ Table 9의 시뮬레이션 결과를 분석하면 다음과 같다.

Table 7. Results of Rule1 parameters variation (8 UAV)

# of UAV	case1	case2	case3	case4
PROB. of detection(%)	84.7	81.3	82.7	81.3
AVER. detection Time(sec)	844	866	886	866
PROB. of Collision(%)	6	2	7	5
PROB. of Near miss(%)	26	12	11	11

Table 8. Results of Rule1 parameters variation (10 UAV)

# of UAV	case1	case2	case3	case4
PROB. of detection(%)	81.3	88	83.3	90
AVER. detection Time(sec)	866	782	856	777
PROB. of Collision(%)	2	4	7	8
PROB. of Near miss(%)	12	17	21	17

Table 9. Results of Rule1 parameters variation (12 UAV)

# of UAV	case1	case2	case3	case4
PROB. of detection(%)	93	90.6	90.6	88
AVER. detection Time(sec)	759	734	721	749
PROB. of Collision(%)	26	9	10	10
PROB. of Near miss(%)	52	26	16	22

기존의 플로킹 이론을 적용한 이전 연구에서의 결과(Table 4, N=12) 대비하여 separation size를 기존보다 2배로 한 모든 경우(case 2~4)에서 탐지확률은 비슷한 수준을 유지하고, 충돌확률 및 Near miss 확률이 거의 절반가량으로 작아지며, 반면 속도 벡터 가중치 c_{1b} 경우에는 그 변경(0.5배, 1.5배)에도 결과에 큰 차이를 보이지 않음을 알 수 있다.

이는 탐색임무에서 각 비행체가 통상적으로 최대속도(본 논문에서는 72 Km/h)에 근접해서 탐색 임무에 운용되기 때문에 Rule 1에 의해서 만들어지는 속도벡터 $\vec{u}_{1,j}$는 separation size가 큰 경우에 더 잘 반영되며, separation size를 축소하는 경우 2.2.1에서 언급한 가속도 및 각가속도의 물리적인 한계에 의해 충돌 회피에 불리한 이류로 분석된다.

3.2.5 기존 플로킹 이론 적용 결과 분석

Figure 4에서 Fig. 7까지는 12개의 비행체에 대해 지금까지 시뮬레이션 수행한 여러 조건별 시뮬레이션 결과를 그래프로 도시한 것이다.

Figure 4는 각 조건에서의 표적 탐지 확률을, Fig. 5에서는 표적을 탐지하는 평균 시간을 도시한 것으로 rule 1의 separation size를 45 m(150 ft)로 기존 연구에 비해 절반으로 줄였을 때 가장 큰 탐지 확률을 가짐을 알 수 있다. Fig. 7 및 Fig. 8에서는 충돌확률 및 Near miss 확률을 도시하였다. 투입되는 비행체 수가 작을수록, separation size를 크게 설정하는 경우 낮은 충돌 확률 및 Near miss 확률을 가짐을 알 수 있다. Fig. 6 및 Fig. 7에서는 1회 시뮬레이션 수행 시 발생하는 충돌 및 Near miss 확률을 도시하였다. 투입되는 비행체 숫자가 작을수록, separation size를 180 m(600 ft)로 크게 설정하는 경우에서 더 작은 충돌 및 Near miss 확률을 가짐을 알 수 있다.

Figure 4에서 Fig. 7에서도 확인할 수 있는 결과는 다음과 같이 소결론으로 요약할 수 있다.

다중 비행체를 활용하여 표적 탐지 임무를 수행 시 이전 연구에서 사용된 일반적인 플로킹 이론을 활용하는 경우, 여러 가지 변수를 활용하여 그 효율성을 높이는 시뮬레이션을 수행하였지만, 결국 탐지 확률을 높이기 위해서는 충돌 및 Near miss 확률이 높아지는 것을 감수해야 함을 있음을 알 수 있다.

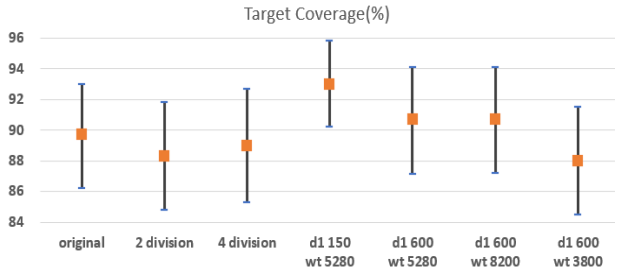

Fig. 4. Probabilities of Target Coverage (12 UAV)

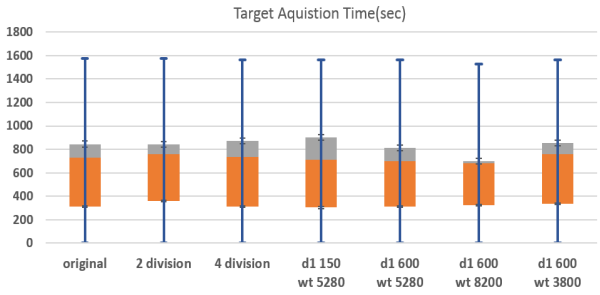

Fig. 5. Time of Target Coverage (12 UAV)

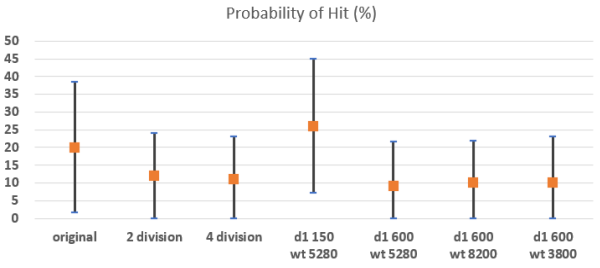

Fig. 6. Probabilities of Collision (12 UAV)

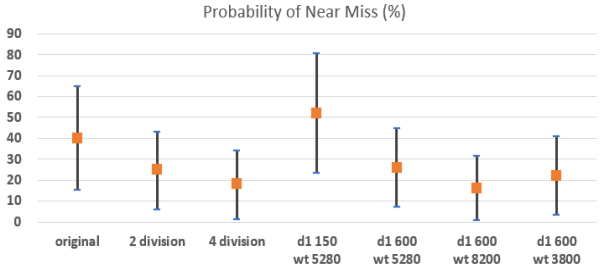

Fig. 7. Probabilities of Near Miss (12 UAV)

이후 장에서 이러한 단점을 해소하여 탐지 성능은 높이면서 동시에 충돌 가능성을 줄이기 위해 수정된 알고리즘을 제안한다.

3.3 개선된 플로킹 이론을 적용한 비행법칙 적용

3.3.1 시뮬레이션 조건

이번 절에서 새롭게 제안하는 방식은 탐색 구역을 더 세부적으로 구분하여 각 비행체에 1:1 할당하되 할당되는 세부 탐색 구역 자체가 시간에 따라서 이동하는 방식이다. 또한, 각 비행체는 플로킹 비행 법칙 10개의 rule 중 rule 1(separation) 및 rule 6(stay within boundary)만 사용하여 그 탐색 구역 내에서 탐색 구역의 테두리를 따라 이동하며 탐지하는 방식으로 다른 비행체들과의 충돌확률을 낮추고자 하였다. Table 10은 이 장에서 제시하는 state 2의 rule 할당을 표시한 것이다.

Table 10. New Rule Allocation

State \ Rule	1	2	3	4	5	6	7	8	9	10
Old 2: detect & loiter	1	1	0	1	1	1	1	1	0	0
New 2: detect & loiter	1	0	0	0	0	1	0	0	0	0

3.3.2 시뮬레이션 조건

본 연구에서 첫 번째로 제시하는 시뮬레이션 수행한 패턴은 선회 반경이 큰 비행체의 탐색임무에 적합하다는 일명 잠보니 패턴(zamboni pattern)으로서 [19], 가로 2.4 Km(1.5 mile), 세로의 크기를 약 800~1,600 m(0.5~1 mile)의 임무 영역을 비행체 수만큼 생성하여 각 비행체에 해당하는 임무 영역으로 할당하고, 이 임무 영역은 시간에 따라 연속적으로 이동시킨다. 이 경우 비행체는 임무 영역의 외곽을 따라 비행하며 탐색하며 결과적으로 스프링 형태의 비행경로를 가진다. 이전 절과 마찬가지로 비행체를 8~12개로 운용하는 경우를 상정하였으며 좌·우측 반평면으로 구분하여 각각 절반씩(4~6개)의 비행체 각각을 시간에 따라 움직이는 탐색구역으로 1:1로 할당하였다.

Figure 8은 제시된 이동하는 임무 영역을 각 비행체 3대의 비행경로를 표시한 그림이다. 3개의 각 비행체가 다른 색깔로 스프링 형태로 탐색하며 탐색 구역들이 중첩되고 변경됨을 확인할 수 있다.

이렇게 사전에 운용자에 의해 각 비행체를 각 하위 탐색 구역을 할당하고 시간에 따라 그 탐색 경로를 이동시키면 각 비행체의 충돌 확률을 낮추는 동시에 탐색구역 전체를 누락 없이 임의로 할당할 수 있는 장점이 있다.

또한 이러한 비행체에 대해 사전에 탐색 구역을 할당하는 경우는 운용 비행체의 미래 경로가 피답지 대상에게 확인 및 회피되지 않는 일반적인 표적을 탐색하는 임무에만 사용이 가능하며, 표적이 비행체를 식별하고 회피 또는 은폐할 수 있는 상황은 고려하지 않는다.

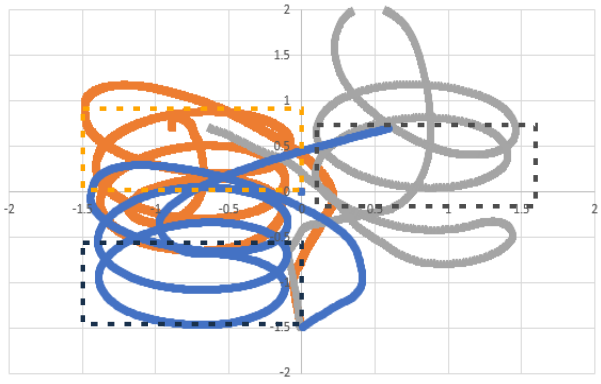

Fig. 8. New Trajectories of Moving Target box

Table 11. Result of Moving Target box algorithm

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	77.6	79.7	85.3
AVER. detection Time(sec)	975	974	888
PROB. of Collision(%)	1	3	9
PROB. of Near miss(%)	5	16	12

3.3.3 시뮬레이션 결과

각 탐색구역을 결정하는 방식(탐색구역의 형상(크기), 각 탐색 구역 간 배치 간격, 탐색구역의 이동 패턴 및 속도 등)이 탐색성능 및 충돌성능에 직접적인 영향을 주는 변수가 되며, error and trial method를 통해 도출된 탐색구역의 파라미터는 다음과 같다.

각 비행체의 탐색구역은 가로 및 세로를 2.4 * 1.2 Km(1.5 * 0.75 mile), 탐색구역의 이동 속도를 2.4 Km/h(1.5 mile/h), 탐색구역 간의 배치 간격 160 m(0.1 mile)로 설정할 때 가장 좋은 성능을 확인하였으며 그 결과는 Table 11과 같다.

시뮬레이션 확인 결과, 기존의 플로킹 이론을 사용하여 전체 탐색 구역을 지정한 결과에 비해 충돌 횟수 및 Near miss 횟수는 비약적으로 작아짐을 알 수 있으나, 탐색성능이 상대적으로 낮음을 알 수 있다. 이는 앞서 전체 탐색구역(4.8*4.8 Km)을 단순히 2분할 및 4분할한 결과에서 분석한 바와 같이 잘게 쪼개진 탐색 구역을 벗어나는 확률이 높아질수록, 비행체 속도가 줄어들고 표적이 없는 구역에서 선회할 시간이 늘어나는 이유로 추론 가능하다.

3.4 비행 영역 재설정

이전 절에서 사용된 탐색구역의 세부 분할, 할당 및 이동의 경우, 각 비행체는 좌, 우로 나누어진 2개의 하위 탐색구역(2.4 * 1.2 Km) 내에서만 시간에 따라서 연속적으로 이동하며 표적을 탐지한다. 또한 표적으로 탐지하고 다른 비행체에 의해 표적으로 확정된 경우에는 최초 탐지한 비행체의 표적 선회 및 충돌로 인해 임무에서 제외된 비행체에 할당된 탐색영역은 더 이상 다른 비행체에 의해 탐색되지 않는다. 또한 여러 차례의 시뮬레이션을 통해 한 지역을 여러 비행체에 의해 중복 탐색되도록 탐색구역을 설정하여야 탐지성능이 높아짐을 확인할 수 있다.

이러한 시퀀스는 탐색구역은 임의로 할당하지만, 표적은 랜덤하게 발생하는 이유로 인해 다음과 같은 여러 가지 상황에서 탐색성능을 하락시키는 단점을 가질 수 있다.

Table 12. Results of Target box Reallocation

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	81.3	85.3	87.3
AVER. detection Time(sec)	993	938	849
PROB. of Collision(%)	4	5	11
PROB. of Near miss(%)	13	11	17

1. 좌 또는 우측에 표적이 몰려있을 경우
2. 표적이 어느 특정한 구역 내에서 몰려서 있는 경우
3. 비행체가 충돌해서 해당 구역의 탐색 임무를 더 이상 수행할 수 없는 경우

상기와 같은 상황에서 임의의 영역에 사전 할당된 비행체가 임무를 수행하지 못하는 경우, 그 영역에 속한 표적의 경우 탐지 확률이 극적으로 떨어질 수 있는 단점이 있으며 실제 Table 11의 결과처럼 탐지성능이 다소 작아지는 것으로 확인된다. 이러한 단점을 보완하기 위해서 임무 중 각 비행체의 위치 및 상태 정보를 기반으로 각 비행체의 탐지 구역을 변경하고 현 위치를 기반으로 다시 할당하였다.

탐지된 표적을 선회하는 비행체(state 2-2)나 충돌로 인해 탐색 임무가 불가능한 비행체(state 5)가 발생하는 경우 및 이로 인해 좌측 및 우측의 탐색하는 비행체의 숫자가 2개 이상 차이나는 경우에는 전체 비행체의 탐색 영역 및 탐색 임무 가능한 비행체를 재식별하고 탐색지역을 재할당함으로써 확률적으로 탐색성능을 높이는 방식을 제안하였으며, 그 결과는 Table 12와 같다.

그 결과 비행 영역을 재설정하기 전 시뮬레이션 결과에 비해 탐지성능은 일정 부분 좋아졌으나, 기존 플로킹 이론을 적용한 연구에 비해서는 아직 낮은 수준이며, 충돌 가능성 또한 탐색 영역을 할당하기 전보다 커짐을 알 수 있다.

3.4.1 탐색 성능 및 충돌 가능성 개선

탐지 성능 향상 및 충돌 가능성을 줄이기 위해 이전 2.3.3절에서 제시한 rule 1의 separation size를 152 m(600 ft)로 설정하고 탐색구역 재할당을 모두 적용하여 시뮬레이션 한 결과는 Table 13과 같다. 이는 이전 플로킹 이론을 적용한 시뮬레이션 결과 및 이전 절의 다른 결과와 비교하여 동등 이상의 수준의 탐지 성능을 유지하면서도 충돌 가능성은 최대 1/4 수준으로 줄일 수 있음을 확인할 수 있다.

Figure 9부터 12까지는 이동하는 비행 영역을 분할 할당 및 재할당을 통한 각 시뮬레이션 결과이다.

Table 13. Results of the final proposal

# of UAV	N=8	N=10	N=12
PROB. of detection(%)	80.3	86	92.3
AVER. detection Time(sec)	975	910	810
PROB. of Collision(%)	0	1	4
PROB. of Near miss(%)	4	8	15

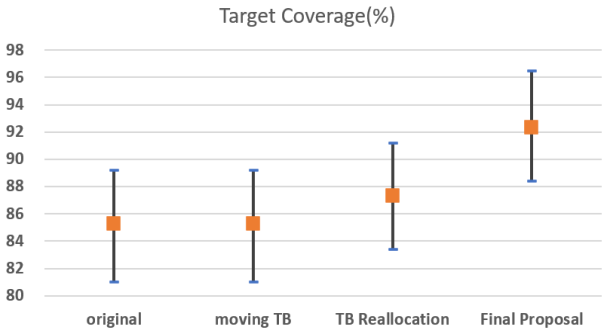

Fig. 9. Probabilities of Target Coverage

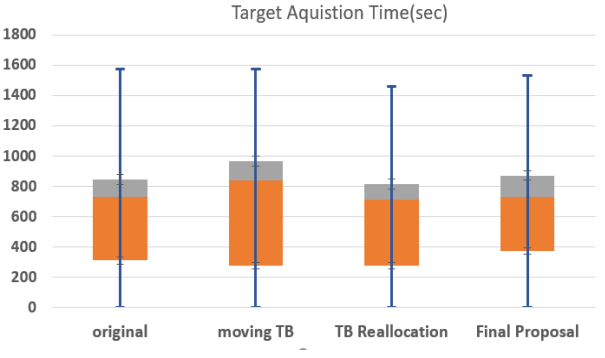

Fig. 10. Time of Target Coverage

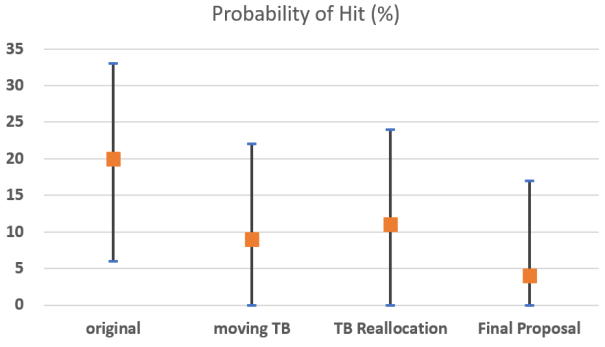

Fig. 11. Probabilities of Collision

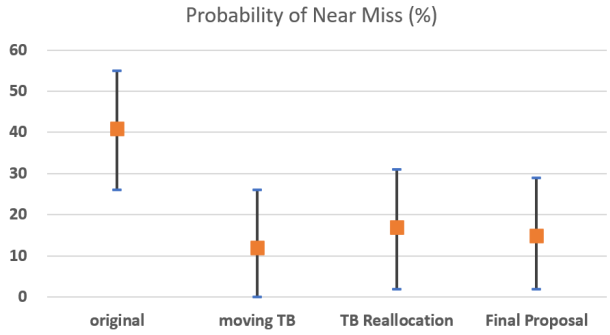

Fig. 12. Probabilities of Near Miss

Ⅳ. 결 론

본 연구에서는 플로킹 이론을 기반으로 다수의 무인항공기를 자율 비행 알고리즘을 구성하였으며 이를 통해 표적을 탐지하는 시뮬레이션을 수행하였다. 표적의 탐지 확률 및 평균 탐지시간을 탐색 성능으로, 충돌확률 및 Near miss 확률을 충돌 회피 성능으로 설정하고, 각각 8개, 10개, 12개의 비행체를 활용하여 3개의 표적을 탐지하는 시뮬레이션을 수행하고 그 성능을 확인하였다.

각 비행체에 대해 시간에 따라 움직이는 탐색 구역을 세부적으로 설정하고, 임무 중 변경되는 각 비행체의 상태에 따라 이를 재할당하는 등의 방법을 통해 이전의 연구에서 제시된 플로킹 이론만 활용한 정찰용 군집 무인항공기의 자율비행 알고리즘과 비교해서 탐색 성능은 유지하면서 충돌 회피 성능을 비약적으로 높일 수 있음을 확인하였다.

다만 비행체 수, 탐색구역의 형상 및 크기, 이동속도, 탐색영역 재할만 방법 등 많은 변수들이 복합적으로 엮여서 탐색 성능 및 충돌 회피 성능에 영향을 주고 있어서 본 연구에서는 최적화된 해답은 찾지 못하였으며, 이후 인공신경망을 활용한 최적화 방안 등 추가적인 연구를 통해 다중 무인기의 탐색 임무에 대한 최적화 연구가 필요할 것으로 판단된다.

References

1) Shakhatreh, H., Sawalisch, A. H., Ap-Falasha, A., Dou, Z., Almadia, E., Khalif, I. and Guarani, M., “Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges,” *IEEE Access*, 7, 2019, pp. 48572~48634.

2) Shackler, R., Ap-Garda, M. A., Badrawi, A., Mohamed, A., Khanate, T., Ap-All, A. and Guarani, M., “Design challenges of multi-UAV systems in cyber-physical applications: A comprehensive survey, and future directions,” IEEE *Communications Surveys*

and Tutorials, Vol. 21, No. 4, 2019, pp. 3340~3385.

3) Gen, L., Hang, Y. F., Wang, J. J., Foh, J. Y. and Ted, S. H., "Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms," *IEEE International Conference on Control and Automation (INCA)*, IEEE, June 2013, pp. 828~833.

4) Career, D. W., Beard, R. W., Mcluhan, T. W., Eli, S. M. and Menhir, R. K., "Forest fire monitoring with multiple small UAVs," *American Control Conference,* IEEE, June 2005, pp. 3530~3535.

5) Xi, X., Yang, L., Men, W., Cai, Q. and Fun, M., "Multi-Agent Coverage Search in Unknown Environments with Obstacles: A Survey," *Chinese Control Conference(CRC)*, IEEE, July 2019, pp. 2317~2322.

6) Yang, Y., Miami, A. A. and Polycarpous, M. M., "Decentralized cooperative search by networked UAVs in an uncertain environment," *American Control Conference*, IEEE, Vol. 6, June 2004, pp. 5558~5563.

7) Pirie, R. R., Eli, X. R. and Dealable, R., "UAV route planning for joint search and track missions—An information-value approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 48, No. 3, 2012. pp. 2551~2565.

8) Roxburghe, V., Tarbouchi, M. and Lambenté, G., "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 1, 2012, pp. 132~141.

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

10) Jones, P. J., "Cooperative area surveillance strategies using multiple unmanned systems," *Doctoral dissertation*, Georgia Institute of Technology, 2009.

11) Sock, M. J., "Development of Autonomous Reconnaissance Flight Simulation for Unmanned Aircraft to Derive Flight Operating Condition," *Journal of The Korean Society for Aeronautical and Space Sciences*, Vol. 47, No. 4, 2019, pp. 266~273.

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

13) Join, G. and Wang, L., "Multi-Agent Flocking with Angle-based Formation Shape Control," *IEEE Transactions on Automatic Control*, Vol. 65, Issue 2, February 2020, pp. 817~823.

14) Dak, F., Chen, M., Wei, X. and Wang, H., "Swarm Intelligence-Inspired Autonomous Flocking Control in UAV Networks," *IEEE Access*, Vol. 7, 2019, pp. 61786~61796.

15) Zhao, W., Chu, H., Hang, M. and Sun, T., "Flocking Control of Fixed-wing UAVs with Cooperative Obstacle Avoidance Capability," *IEEE Access*, Vol. 7, 2019, pp. 17798~17808.

16) Vasarhelyi, G., Virago, C., Somorjai, G., Nepusz, T., Eiben, A. E. and Vicsek, T., "Optimized Flopping of Autonomous drones in Confined Environments," *Science Robotics*, Vol. 3, Issue 20, eaat3536, 2018.

17) Ludo, X., Eli, X., Eli, S., Jang, Z. and Gaud, X., "Flocking for Multi-agent System with Optimally Rigid Topology Based on Information Weighted Karman Consensus Filter," *International Journal of Control, Automation and Systems*, Vol. 15, No. 1, 2017, pp. 138~148.

18) Narcomania, A., Baker, R., Descale, R., Matchers, B., Kolokotronis, S. O., Kreiswirth, B. and Planet, P. J., "Clusterflock: a flocking algorithm for isolating congruent phylogenetic dataquest," *Technical Note*, Geoscience, December 2016.

19) Aramco, J. F., Suit, P. B. and Sousa, J. B., "Multiple UAV area decomposition and coverage," *IEEE symposium on computational intelligence for security and defense applications*, CISDA, IEEE, April 2013, pp. 30~37.