

새 때 비행 및 대형비행을 위한 다중에이전트 기반 자율 UAV 설계

Multi-Agent based Design of Autonomous UAVs for both Flocking and Formation Flight

하선호 · 지승도*

한국항공대학교 컴퓨터공학과

Sun-ho Ha · Sung-do Chi*

Department of Computer Engineering, Korea Aerospace University, Gyeonggi-do 10540, Korea

[요 약]

다수의 UAV가 다양한 임무를 수행하면서도 편대를 유지할 수 있도록 하는 집단적 지능을 갖춘 시스템을 구축하기 위해서는 AI에 관한 연구가 필수적이다. AI의 전형적인 접근 방법에는 전문가시스템을 비롯한 규칙기반의 논리 추론방식인 ‘하향식’ 접근 방법과 인공신경회로망, Flocking Algorithm과 같이 단순 개체간의 부분적 상호작용을 통해 전체적인 행동이 결정되는 ‘상향식’ 접근 방법이 있다. 기존의 Flocking Algorithm과 같은 연구에서는 개개인은 개별적인 임무를 수행 할 수 없다. 또한 UAV의 편대비행과 같은 연구에서는 편대의 부분적인 결함으로 발생하는 문제에 대해 유연하게 대처 할 수 없다. 본 논문에서는 다중에이전트 시스템을 통해 하향식 접근 방법과 상향식 접근 방법 간의 유기적 통합을 제시하고, 이를 통해, 유연한 임무수행이 가능한 편대 비행 알고리즘을 제시하였으며, 시뮬레이션을 통해 대형형성 및 충돌회피 등 유효성을 확인하였다.

[Abstract]

Research on AI is essential to build a system with collective intelligence that allows a large number of UAVs to maintain their flight while carrying out various missions. A typical approach of AI includes 'top-down' approach, which is a rule-based logic reasoning method including expert system, and 'bottom-up approach' in which overall behavior is determined through partial interaction between simple objects such as artificial neural network and Flocking Algorithm. In the same study as the existing Flocking Algorithm, individuals can not perform individual tasks. In addition, studies such as UAV formation flight can not flexibly cope with problems caused by partial flight defects. In this paper, we propose organic integration between top - down approach and bottom - up approach through multi - agent system, and suggest a flight flight algorithm which can perform flexible mission through it.

Key word : Unmanned aerial vehicle, Flocking, Multi-Agent System, Formation Flight, Swarm Intelligence.

<https://doi.org/10.12673/jant.2017.21.6.521>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 20 November 2017; Revised 24 November 2017
Accepted (Publication) 17 December 2017 (30 December 2017)

*Corresponding Author; Sung-do Chi

Tel: +82-2-300-0184

E-mail: sdchi@kau.ac.kr

I. 서론

최근 들어 다수의 UAV를 활용할 수 있는 방법에 대한 관심이 높아지고 있다. 새떼비행에 대한 연구는 Reynold[1]가 제안한 지역적인 정보를 기반으로 무리를 이루어 비행하는 Flocking Algorithm 연구가 주류를 이룬다. 이와는 달리 임무 목적에 따라 최적의 대형을 유지하면서 비행하는 Formation Flight[2][3]에 관한 연구 등이 있다. Flocking Algorithm은 단순 개체간의 부분적 상호작용을 통해 전체적인 행동이 결정되는 비행방법이다. 이를 이용해 비행 시에는 무리의 멤버들이 각각 지역적 규칙에 따라 행동하므로 상황 변화에 유연하게 대처할 수 있는 장점이 있지만, 임의적 대형유지는 어려운 단점이 있다.

한편 Formation Flight은 인간이 지정한 대형 유지를 중시하는 방식이기에 임무 수행에 적합하지만, 동시에 급격한 환경변화나 개별적으로 발생하는 부분적 문제에 대해서는 유연한 대처가 어려운 단점을 갖는다. 한편 두 방식의 장단점을 보완하는 통합 연구는 아직까지 부족한 실정이다. 따라서 본 연구에서는 두 방식의 비행이 가능한 통합형 집단비행을 제안한다. 즉 평상시 이동의 효율성을 고려하여 새 떼형 집단비행, 임무 시에는 정해진 편대형 집단비행을 가능케 하는 자율 UAV 설계방법론을 제시한다. 이를 위해 다중 에이전트를 이용한 자율 UAV를 모델링하였고, 사례연구를 통해 시뮬레이션을 수행하였다. 이를 통해 제안한 방법론의 타당성을 확인할 수 있었다.

II. Research of Group Flight

2-1 플로킹(flocking) 알고리즘

Reynold는 새들의 집단행동을 시뮬레이션 한 boid라는 프로그램을 통해 Flocking algorithm을 소개하였다. 무리를 이루며 비행하기 위해 필요한 세 가지 요소로써 Flock Centering, Collision Avoidance, Velocity Matching 이라는 3가지 규칙을 제안하였으며, 이를 기반으로 다양한 연구가 진행되고 있다. Flocking Centering 이란 무리에 속한 개체가 무리의 중심으로 모이려고 하는 규칙, Collision Avoidance는 무리 내에 다른 개체와 가까워 졌을 때, 무리의 밖으로 나가려는 규칙, Velocity Matching은 무리 전체의 이동 방향으로 이동하려는 규칙이다. Reza[4]는 기존의 Flocking Algorithm을 그래프 이론을 기반으로 기존 Flocking Algorithm에서 발생하는 무리에 속한 개체가 무리에서 소외되는 현상인 regular fragmentation 현상을 해결하기 위해 기존 Flocking Algorithm을 처리하는 알파 에이전트에 충돌회피, 무리로 합류할 수 있도록 방향을 보정해주는 가상의 베타, 감마 에이전트를 통해 새로운 알고리즘을 제시하였고 그 성능을 평가하였다. 하지만 이러한 방법은 무리 전체를 유지하려는 규칙으로만 행동하므로 각 개체들이 독립적으로 개별 임무를 수행할 수 없다는 단점을 갖는다.

2-2 다중 에이전트 구조

에이전트란 사용자를 대신해서 사용자가 원하는 작업을 스스로 해결할 수 있는 소프트웨어를 말한다. 에이전트는 스스로 환경의 변화를 인지하고 그에 대응하는 행동을 취한다[5].

다중에이전트 시스템은 하나의 에이전트로 해결하지 못하는 복잡한 문제의 해결을 위하여 여러 에이전트간의 협동을 위해 필요한 시스템이다. 편대의 UAV가 자율적으로 복잡한 임무를 수행할 수 있는 시스템을 설계하는 경우에는 문제영역이 크기 때문에 다중에이전트 시스템이 적합하다.

다중 에이전트 구조는 주로 인공지능이나 국방 M&S 분야에서 사용되는 설계 방법론이다. 정찬호 등[6](2009)은 에이전트의 의사결정 수준을 높여 복잡한 상황 묘사를 하기 위해 정보장고, 함장 및 함대사령관 등 역할에 따라 지능적 기능을 수행할 수 있는 계층 구조적 다중 에이전트 시스템을 성공적으로 제안하였다. 또한 유용준 등[7](2010)은 실제 사람을 대체할 수 있는 컴퓨터생성 가상전투객체를 효과적으로 모델링하고, 모델링 된 가상전투객체가 자율성을 갖추고 목표 지향적 행위를 할 수 있는 방안으로 HEAP 기반의 지능 에이전트 기법을 제안하여 많은 연구에서 검증 받은 바 있다. 본 연구에서 도입하는 다중 에이전트 아키텍처는 HEAP 원칙을 기반으로 설계되었다.

III. Multi Agent Based Flocking Algorithm

3-1 다중 에이전트 구조 설계

본 연구에서 제시하는 다중에이전트 기반 UAV 편대비행 개념은 그림 1 과 같이 요약된다. UAV의 하드웨어는 현재의

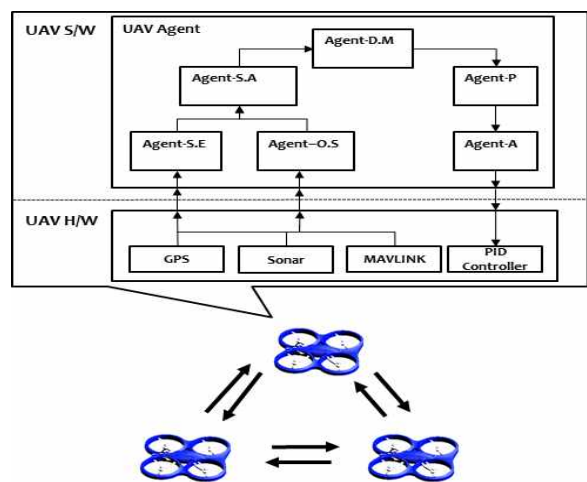


그림 1. 다중에이전트 기반 UAV 편대비행 개념도
Fig. 1. Multi-agent based UAV flight flight concept map.

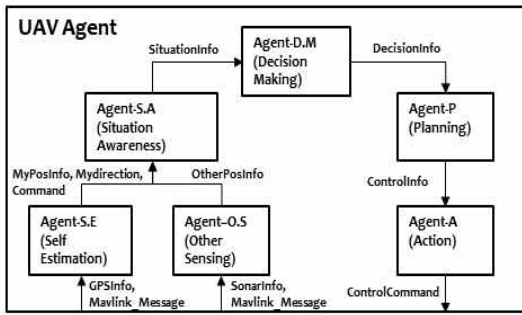


그림 2. UAV 편대비행을 위한 다중에이전트 구조
Fig. 2. Multi-agent architecture for UAV formation flight.

위치를 측정하기 위한 GPS와 자기 주변의 물체를 인식하기 위한 Sonar 센서, UAV간의 통신에 필요한 Mavlink, UAV의 자세와 이동을 제어하는 PID Controller로 구성된다. UAV 편대가 스스로 행동하기 위한 소프트웨어로써 다중에이전트 구조가 적용되었다.

본 연구에서 제안하는 Multi-Agent Architecture는 그림 2와 같다. 다중 에이전트는 Self Estimation, Sensing, Situation Awareness, Decision Making, Planning, Action 등 여섯 개의 에이전트로 구성된다. 다중 에이전트는 UAV의 하드웨어와 정보를 주고받으며 UAV 개개인의 자율적인 행동을 실현해주며 동시에 UAV 무리 전체를 유지할 수 있도록 작동한다.

1) 자기 추정 에이전트(self estimation)

Self Estimation 에이전트는 자신의 상태를 추정하는 에이전트이다. SE에이전트는 UAV의 GPS 센서, Mavlink를 통해 자신의 위치에 대한 정보와 리더의 명령에 관한 정보를 받아 SA 에이전트에게 전달한다. 그림 3 과 같이 SE에이전트는 리더의 명령 여부를 탐지하고 그 후 자신의 위치를 탐지한다.

2) 주변 인식 에이전트(Other Sensing)

Other Sensing 에이전트는 자신의 주변의 상태를 판단하는 에이전트이다. OS에이전트는 UAV의 Sonar 센서와 Mavlink를 통해 자기 주변에 있는 물체에 관한 정보와 편대(리더)에 관한 받아들인다. 그림 4 와 같이 기본적으로 자기 주변물체들의 정보를 탐지하며 만약 가용 탐지범위 내에 UAV가 존재하면 UAV의 위치와 이동방향에 대한 정보를 얻고, 존재하지 않으면 리더와 직접 통신을 하여 리더의 위치와 이동방향에 대한 정보를 얻어 자신의 이웃한 UAV로 인식하게 한다.

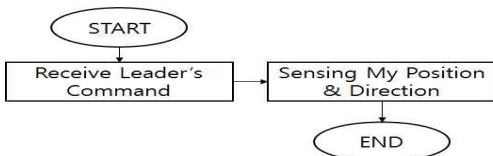


그림 3. Self-estimation에이전트 동작 절차
Fig. 3. Self-estimation agent operation procedure.

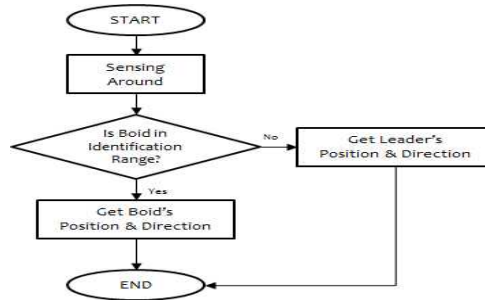


그림 4. Other sensing 에이전트 동작 절차
Fig. 4. Other sensing agent operation procedures.

3) 상황인식 에이전트 (situation awareness)

Situation Awareness 에이전트는 나에 대한 정보와 주변의 정보를 이용하여 상황을 인식하는 에이전트이다. 그림 5 와 같이 리더의 명령의 유무에 따라 Formation Flight/Flocking Flight 모드가 결정되고 UAV의 위치가 자신의 충돌범위(collision range) 내에 Close 상황(UAV가 근접한 상황), UAV의 위치가 자신의 충돌범위 밖, 인식범위 안에 있으면 Normal 상황(기존의 Flocking 규칙대로 가면 되는 상황), UAV의 위치가 인식범위 밖이면 Far 상황(UAV의 무리에 가까워 져야하는 상황)으로 인식한다.

4) 의사결정 에이전트 (decision making)

Decision Making 에이전트는 인식한 상황에 대해 적절한 행동을 결정하는 에이전트이다. 그림 6 에서 처럼 가장 먼저 모드에 따라 Formation Flight/ Flocking Flight을 결정하게 된다. Formation Flight로 결정이 되면 Close 상황에는 Avoid(회피) 결정을 내리고 나머지 상황에는 리더가 결정해주는 방향을 따른다. Flocking Flight가 결정 되면 상황에 따라 Close 이면 Avoid로, Normal 이면 Follow Boids 로, Far 이면 Follow Boids Faster로 결정된다.

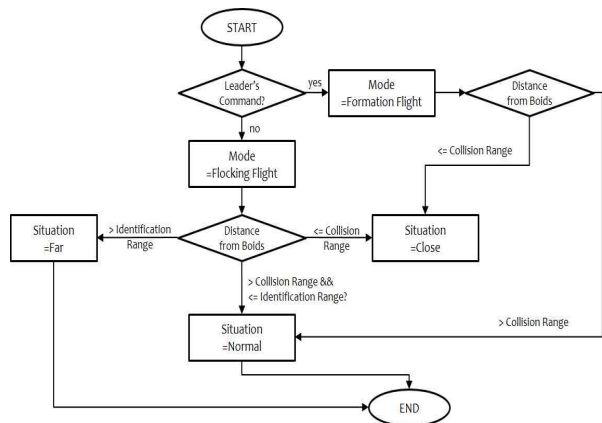


그림 5. Situation awareness 에이전트 동작 절차
Fig. 5. Situation awareness agent operation procedure.

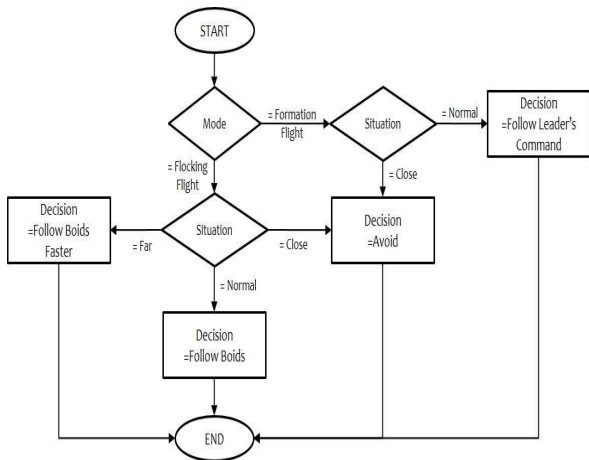


그림 6. Decision Making 에이전트 동작 절차
 Fig. 6. Decision Making agent operation procedure.

5) 계획 에이전트 (planning)

Planning 에이전트는 Decision Making 에이전트에서 결정한 행동을 수행하기 위한 계획(연산)을 수립하는 에이전트이다. 본 연구에서는 Flocking Algorithm을 (1)과 같이 표현하였다. (1)에서 V 는 다음 이동할 방향을 의미하며, v_a 는 Velocity Matching 규칙을 표현하는 정렬벡터, v_c 는 Flock Centering 규칙을 표현하는 응집벡터, 마지막으로 v_s 는 Collision Avoidance 규칙을 표현하는 분리벡터이다. K_a, K_s, K_c 는 Flocking Algorithm의 각 규칙들의 가중치 조절을 위해 곱해주는 계수이다.

$$V = K_a \cdot v_a + K_c \cdot v_c + K_s \cdot v_s \quad (1)$$

그림 7 과 같이 Decision Making 에이전트에서 결정 된 정보를 바탕으로 Follow Leader's Command 일 경우 리더가 UAV의 정렬하는 힘을 일정한 위치로 정해주기 때문에, K_a 의 값과 이동속도를 증가시키고 K_s, K_c 의 값을 감소시켜 응집, 분리하는 힘의 방해 없이 편대비행을 하기 쉽도록 해준다. Follow Boids일 경우 Flocking 규칙대로 K_a, K_s, K_c 의 값을 1로 정하여 기존의 Flocking Algorithm 규칙을 준수한다. Avoid일 경우 K_s 의 값을 증가시키고 K_a, K_c 의 값을 감소시켜 회피하는 힘을 증가시켜준다. Follow Boids Faster일 경우 K_a, K_c 의 값과 이동속도를 증가시키고, K_s 의 값을 감소시킴으로써 이웃 UAV의 무리에 더욱 빠르게 합류할 수 있게 해준다. 이렇게 정해진 K_a, K_s, K_c 값을 Flocking Algorithm의 각 규칙에 적용하여 다음 이동방향을 계산한다.

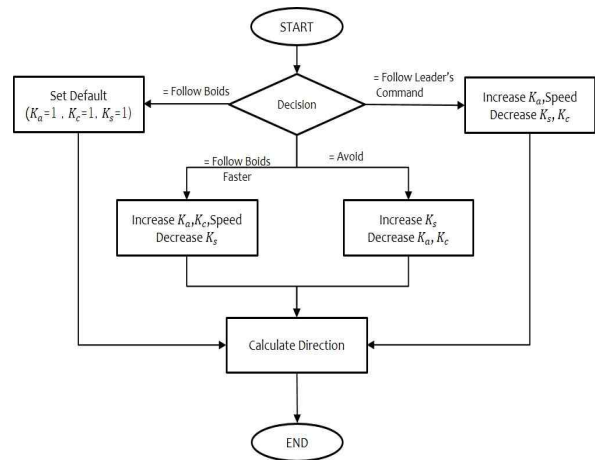


그림 7. Planning 에이전트 동작 절차
 Fig. 7. Planning agent operation procedure.

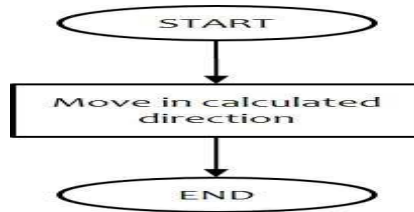


그림 8. Action 에이전트 동작 절차
 Fig. 8. Action Agent operation procedure.

6) 행동 에이전트 (action)

Action 에이전트는 그림 8 과 같이 Planning 에이전트에서 계산된 방향으로 UAV를 움직이는 에이전트로써, UAV의 하드웨어에게 직접 명령을 내려 UAV이 움직일 수 있도록 동작한다.

IV. 사례연구

4장에서는 제시된 다중 에이전트 구조를 통해 상황에 맞게 변하는 집단비행 알고리즘에 대한 타당성을 검증하기 위한 시뮬레이션을 진행하였다.

시뮬레이션에서 총 9대의 UAV 무리가 정해진 초기 위치로부터 목표 위치로 이동한다. 무리에 속한 UAV들은 무작위로 일정 간격을 유지하며 비행한다. 비행하는 동안 상부로부터 대형 전환에 관한 명령을 받게 되며, Line - Column - Flocking의 순서로 대형 전환을 시뮬레이션을 진행하였다.

그림 9-a ~ 그림 9-d 는 UAV 편대의 대형이 변하는 동안 시간별로 UAV의 위치와 방향을 보여 준다. 각 그림에서 삼각형의 길쭉한 꼭지점은 UAV의 방향을 나타낸다. 검정색 삼각형은 무리 전체의 방향을 잡아주는 리더를, 주황색 삼각형은 리더를 따라가는 UAV를 각각 나타낸다. t 는 시뮬레이션이 시간을 나타낸다.

그림 9-a 에서 시뮬레이션 진행 동안 UAV 편대의 개체들 간에 충돌은 없었으며, t=13일 때의 편대의 모양(line)을 보면 주어진 임무를 효과적으로 수행하였음을 알 수 있다. 또한 그림 9-b 에서는 Line의 형태에서 다시 기본적인 Flocking대형으로 돌아오는 것을 잘 보여준다.

그림 9-c 는 초기의 Flocking 대형에서 Column로 변하는 모습을 보여준다. 이 과정에서도 충돌은 없었으며, Line 대형으로 전환할 때 (t=13) 보다 시간이 많이 (t=39) 소모되었고 원하는 편대가 이뤄지는 동안 그만큼 이동거리도 많았다. 그 이유는 Column 대형을 형성하는 동안 개체들 간의 충돌을 회피하는데 시간이 다소 소모 되었고 또한 편대를 형성하는 데에 필요

한 이동거리도 상대적으로 멀었기 때문인 것으로 추정된다.

그림 10 은 t=0, t=5일 때의 각 UAV의 위치, 방향을 보여주며, 그 때 각 Agent 상태를 보여준다. 리더 UAV은 정해진 경로를 향해 등속도로 비행하도록 설정하였고, t=5일 때 진행방향에 터널이 있다고 인식하도록 가정하였다. 리더가 터널의 존재를 인식하면 나머지 UAV들에게 Line 대형으로 전환하라는 명령을 내린다. 이때 각 UAV은 ‘Leader-is-Commanding’ 명령임을 인식하며 대형비행 모드로 전환하게 된다.

종합적으로 그림 9 ~ 그림 10 을 통해, UAV 대형이 전환되는 동안에도 큰 충돌 없이 안정적으로 대형을 유지하는 것을 확인할 수 있다.

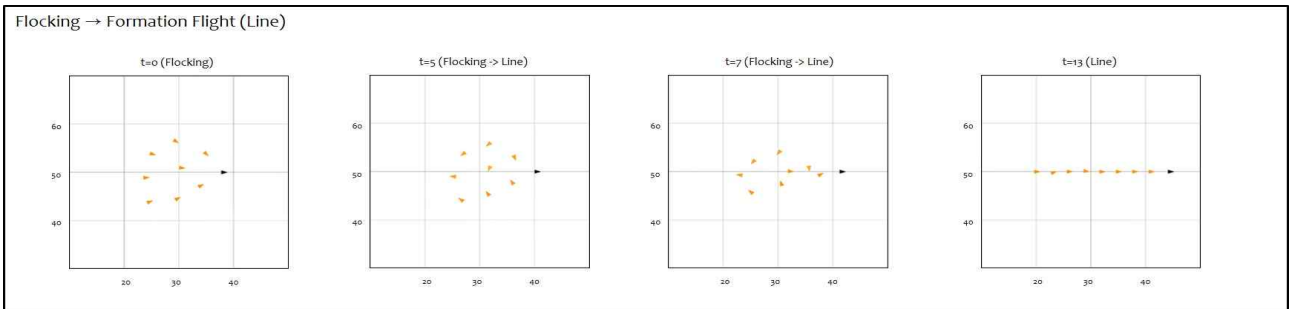


그림 9-a. Flocking대형에서 Line대형으로 전환 시 시간별 위치
Fig. 9-a. Time position when switching from Flocking Formation to Line Formation.

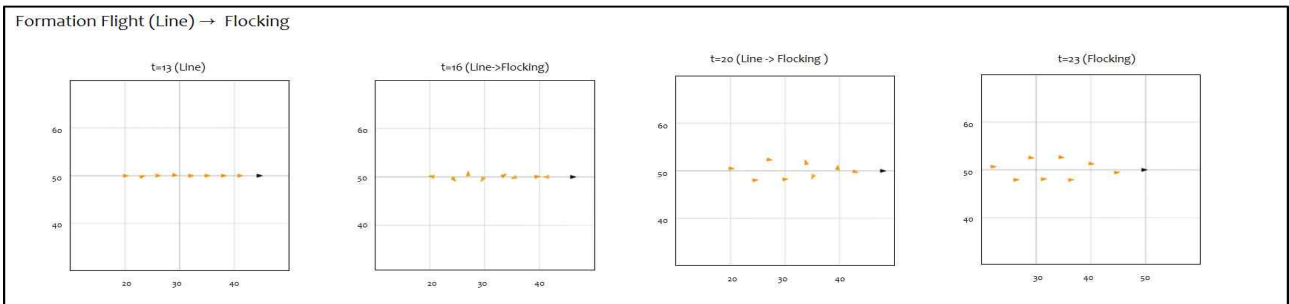


그림 9-b. Line대형에서 Flocking대형으로 전환 시 시간별 위치
Fig. 9-b. Time position when switching from Line Formation to Flocking Formation.

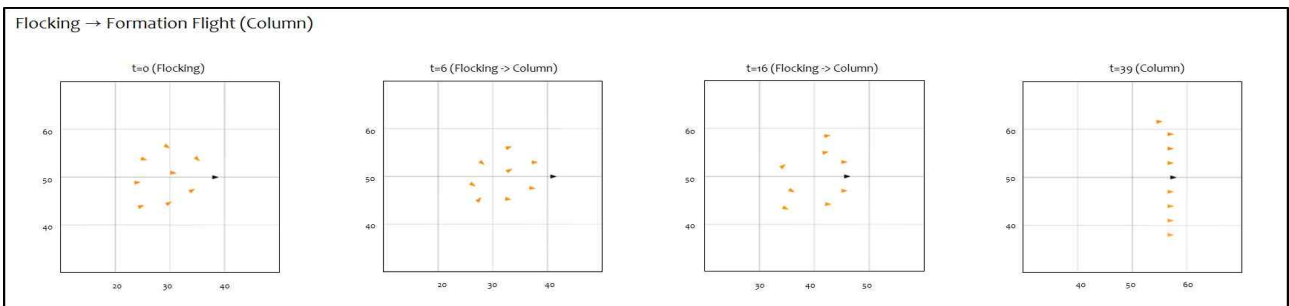


그림 9-c. Flocking대형에서 Column대형으로 전환 시 시간별 위치
Fig. 9-c. Time position when switching from Flocking Formation to Column Formation.

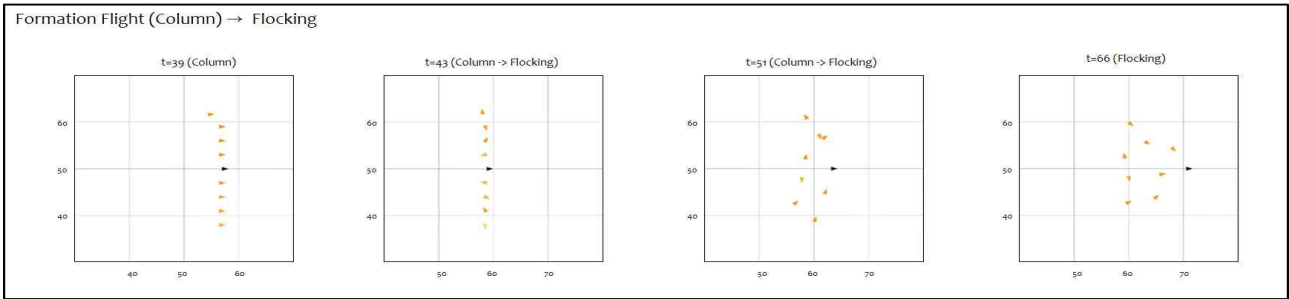


그림 9-d. Column대형에서 Flocking대형으로 전환 시 시간별 위치
 Fig. 9-d. Time position when switching from Column Formation to Flocking Formation.

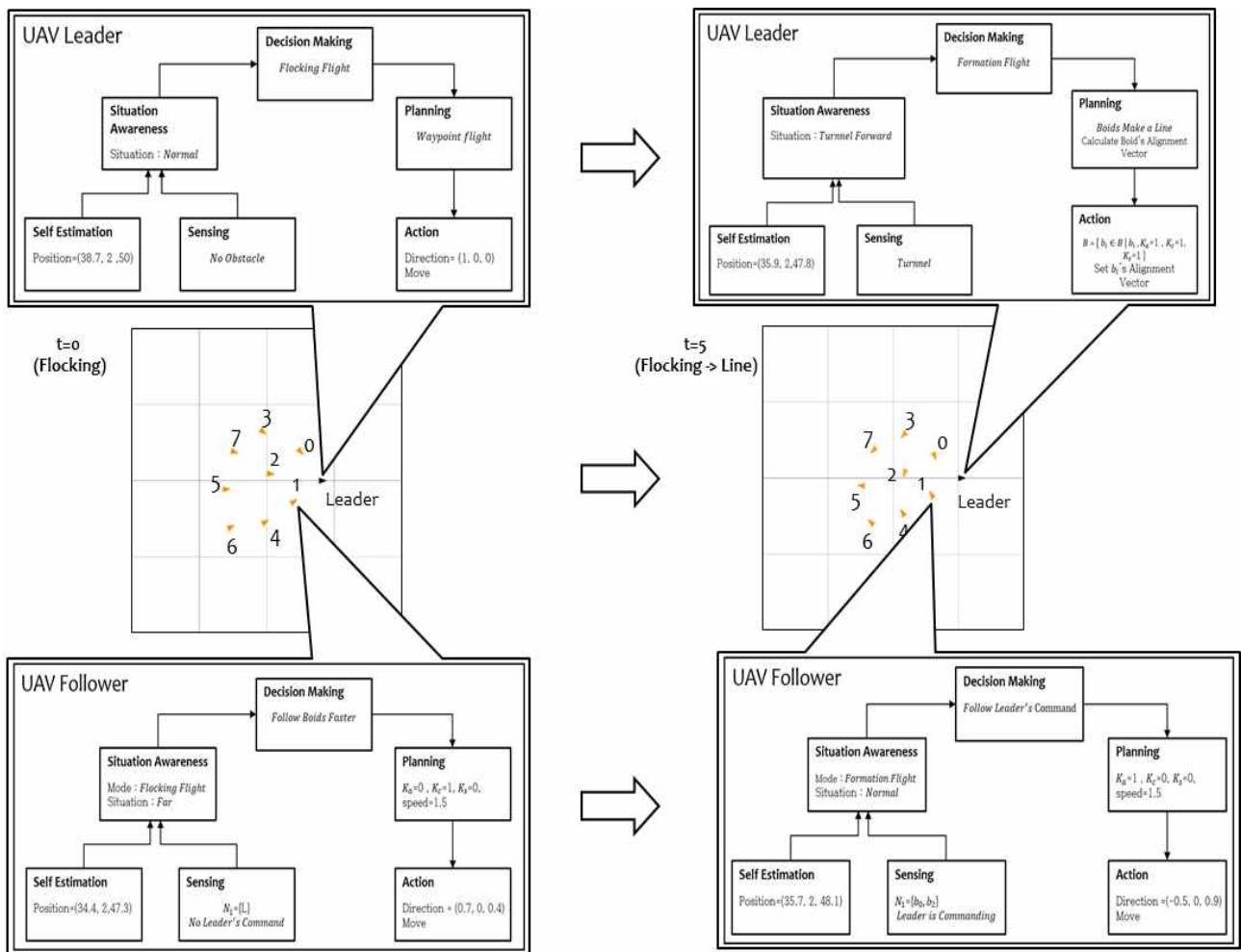


그림 10. 대형 전환을 위한 에이전트의 세부적 작동 과정의 일부
 Fig. 10. Part of the detailed operation of the agent for Formation change.

또한 Flocking 알고리즘 기반으로 비행가능하고 또한 리더에 의한 대형비행이 가능함을 확인할 수 있었다. 시뮬레이션 조건에서는 리더가 팔로워들의 모든 위치를 알고 있기 때문에 대형형성에 대해서 본 논문에서 제시한 방법론의 타당성을 확인할 수 있었다. 만약, 실세계에서 통신의 문제로 인해 팔로워가 리더의 명령을 받지 못하더라도 자신 주변의 정보로 플로킹 알고리즘을 통해 충돌 없이 비행할 수 있다. 이는 대형비행만을 하는 방법에 비해 집단에 대한 안정성이 높다고 볼 수 있으며, 플로킹 알고리즘만으로 비행하는 방법에 비해 사용자가 원하는 임무수행을 할 수 있다는 장점이 있다.

V. 결 론

본 연구에서는 Reynold가 제안한 지역적인 새떼비행을 기반으로 임무기반의 편대비행까지 가능한 통합 집단비행 UAV를 제안하였다.

다수의 UAV 들이 상황을 인식하고 그에 맞는 행동을 스스로 하기 위한 다중 에이전트 구조를 설계하였으며, 이를 통해 Flocking Algorithm의 규칙을 준수하면서 동시에 각자의 상황에 맞는 행동을 함으로써, 유연한 임무수행이 가능한 편대 비행 알고리즘을 제시하였다.

또한 임무중심의 하향식 AI기법과 Flocking Algorithm을 통한 상향식 AI간의 유기적 통합을 제시하였다. 상향식 접근 방법과 하향식 접근 방법의 장단점을 상호보완 해줌으로써 개체의 부분적 결함에 대해 전체 집단의 안정성이 강하며 급격한 환경변화나 결함으로 발생하는 문제에 대해 유연하게 대처 할 수 있으며 논리적인 추론을 통해 무리에 속한 각각의 개체마다 목적 지향적 행위를 실현할 수 있었다.

본 연구에서는 다수의 UAV들이 Flocking Algorithm을 지키면서 원하는 형태의 편대를 생성할 수 있는 에이전트를 설계하였는데 향후 과제로써 각각의 UAV들이 더욱 복잡한 임무를 수행할 수 있도록 에이전트를 설계함으로써 다수의 UAV들의 자율성 단계를 높일 예정이다.

Acknowledgments

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음" (IITP-2017-10-006)

References

- [1] Reynolds, C. W. "Flocks, herds and schools: A distributed behavioral model," ACM SIGGRAPH computer graphics 21.4 pp. 25-34, 1987
- [2] Zelinski, S, Koo, T. J, & Sastry, S. "Hybrid system design for formations of autonomous vehicles," in *Decision and Control, 2003 Proceedings 42nd IEEE Conference on*. Vol. 1. IEEE, 2003.
- [3] S. W. Min, and H. W. Nam. "A formation flight control of UAVs using ZigBee," in *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*. IEEE, 2016.
- [4] Olfati-Saber, R "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on automatic control* 51.3 pp 401-420, 2006
- [5] C. J. Min. "[Technical Commentary] overview and research directions of agents." 15.3, pp.7-16, 1997
- [6] C. H. Jung, Y. J. You, H. E. Ryu, J. S. Lee, J. I. Kim, and S. D. Chi, "Multi-platform warship M&S system using the hierarchical multi-agent system," *Journal of the Korea Society for Simulation* 18.4, pp.117-125, 2009.
- [7] Y. J. You, S. D. Chi, and J. I. Kim. "A study of HEAP-based intelligent agent applied to warship combat simulation," *Journal of the Korea Society for Simulation* 19.4 pp. 281-289, 2010.



하 선 호 (Sun-ho Ha)

2016년 2월 : 한국항공대학교 소프트웨어학과 학사
2016년 3월 ~ 현재 : 한국항공대학교 컴퓨터공학과 석사과정
※ 관심분야 : 드론, 인공지능, 편대비행, 시뮬레이션



지 승 도 (Sung-do Chi)

1982년 2월 : 연세대학교 전기공학과 졸업
1984년 2월 : 연세대학교 전기공학과 석사
1991년 : 아리조나대학교 전기전산공학 박사
1992년 ~ 현재 : 한국항공대학교 소프트웨어학과 교수
※ 관심분야 : 무인자율시스템, 지능시뮬레이션, 자율에이전트