

論文

DOI:10.5139/JKSAS.2010.38.4.352

진화로봇공학 기반의 복수 무인기를 이용한 영역 탐색

오수훈*, 석진영**

Area Search of Multiple UAV's based on Evolutionary Robotics

Soo-Hun Oh* and Jinyoung Suk**

ABSTRACT

The simultaneous operation of multiple UAV's makes it possible to enhance the mission accomplishment efficiency. In order to achieve this, easily scalable control algorithms are required, and swarm intelligence having such characteristics as flexibility, robustness, decentralized control, and self-organization based on behavioral model comes into the spotlight as a practical substitute. Recently, evolutionary robotics is applied to the control of UAV's to overcome the weakness of difficulties in the logical design of behavioral rules. In this paper, a neural network controller evolved by evolutionary robotics is applied to the control of multiple UAV's which have the mission of searching limited area. Several numerical demonstrations show the proposed algorithm has superior results to those of behavior based neural network controller which is designed by intuition.

초 록

복수 무인기의 동시 운용을 통하여 임무 수행 효율성 제고를 꾀할 수 있으며 이를 위해서는 확장성이 용이한 제어 알고리즘을 필요로 하게 되는데 유연성, 강건성, 분산형 제어 및 자기조직화의 특징을 갖는 행동모델 기반의 무리 지능이 현실적인 대안으로 각광 받고 있다. 그러나 논리적으로 행동규칙을 설계하기 어렵다는 단점을 극복하기 위하여 최근 진화로봇공학이 무인기 제어에 적용되기 시작하고 있다. 본 논문에서는 제한된 영역을 복수의 무인기로 탐색하는 임무를 진화로봇공학을 적용하여 진화시킨 신경망제어기로 수행한 결과, 직관에 의지하여 설계된 행동모델 기반의 신경망제어기에 비하여 우수한 성능을 보임을 제시하였다.

Key Words : Multiple UAV's(복수 무인기), Evolutionary Robotics(진화로봇공학), Behavioral Model(행동모델), Swarm Robotics(무리로봇공학), Genetic Algorithm(유전알고리즘)

1. 서 론

복수의 무인기를 동시에 운용함으로써 감시/정찰 및 수색 등의 임무 수행 시간을 단축시키고, 복수의 표적을 동시에 추적할 수 있으며, 일부 무인기가 고장 나거나 파괴되더라도 나머지 무인기로 임무 완수가 가능할 뿐만 아니라 각 무인기별로 상이한 방식의 센서를 장착하면(전자광

† 2009년 12월 2일 접수 ~ 2010년 3월 24일 심사완료

* 정회원, 한국항공우주연구원

** 정회원, 충남대학교 항공우주공학과
교신저자, E-mail : jsuk@cnu.ac.kr
대전광역시 유성구 궁동 220

학, 적외선, 다중스펙트럼 등) 동일 표적에 대하여 다각적인 관측이 가능하게 되어 표적 식별 성능을 대폭 향상시킬 수 있는 등 임무 수행의 효율성을 높일 수 있기 때문에 복수 무인기의 동시 운용을 위한 제어 알고리즘 연구가 활발히 진행되고 있다[1~8].

복수 무인기의 동시 운용을 위해 필요한 제어 방식은 중앙집중형과 분산형으로 구분할 수 있다. 중앙집중형 제어시스템은 한 곳에서(지상 또는 전체를 통제하는 무인기) 모든 무인기의 현재 위치를 수집하고 다음 목표 위치를 계산하여 각각의 무인기로 전달하면, 각각의 무인기는 부여 받은 목표 위치를 추종하기 위한 제어를 수행하는 방식이다. 이는 각각의 무인기에 가장 적합한 임무를 부여함으로써 임무 수행을 최적화할 수 있다는 장점이 있으나(Teaming), 제어 대상 무인기의 수가 늘어날수록 요구되는 컴퓨터 계산량과 통신 대역폭이 지수적으로 늘어나게 되어 현실적으로는 구현하기 곤란한 방식이다(Not Scalable). 또한 이 방식의 결정적인 약점은 전체를 구성하는 무인기 중 일부가 임무 수행에 실패할 경우 전체 임무 수행에도 실패하는 결과를 초래할 수 있다는 점이다(Single Point of Failure). 이에 비하여 분산형 제어시스템은 중앙의 통제 없이 각각의 무인기가 국지 환경에 스스로 반응하며 자율적으로 비행하면서 전체적으로 목표 임무를 수행하는 방식이다(Swarming). 이 경우 임무 수행 방식을 최적화할 수 없다는 단점이 있지만 제어 대상 무인기 수가 늘어나도 컴퓨터 계산량이 늘어나지 않으며, 전역 정보에 의존하지 않고 국지 정보만을 이용하므로 통신 의존도가 낮기 때문에 시스템을 구성하는 무인기 수에 제한받지 않는 확장성(Scalability)을 가진다는 장점이 있기에 현실적으로 복수 무인기 제어에 응용 가능한 방식이다.

개미나 벌과 같은 群居 곤충(Social Insects)은 국지 정보만을 이용하여 단순한 행동규칙만을 실행함으로써 개개 구성원은 지능적이지 않지만 무리 전체로는 지능적인 행동을 표출하는데, 개미가 먹이를 운반할 때 페로몬을 이용하여 최단거리로 이동하는 행태를 보이나거나, 말벌이나 흰개미가 고도로 복잡한 구조의 집을 짓는 것 등이 그 대표적인 예이다. 무리지능(Swarm Intelligence)이란 개미, 흰개미, 꿀벌, 말벌 등의 군거 곤충이나 물고기, 새와 같이 무리를 이루는 동물의 집단행동으로부터 표출되는 지능으로 각 구성원 지능의 총합보다 월등한 지능을 무리 수준에서 표출하는 특성이 있다. 이러한 자연 현상

에 영감을 받아, 분산형 문제해결 방식이나 알고리즘 개발에 무리지능을 응용하려는 시도가 활발히 이루어지고 있으며 다양한 분야에서 성과를 보이고 있다[9]. 무리를 이루는 곤충, 새, 물고기 및 육상동물들은 서로 간 충돌을 피하면서 무리를 이룬 상태로 이동하며 즉각적으로 방향을 변경하는 능력을 가지고 있는데, 이러한 현상은 겉보기에 대단히 복잡하므로 내부적으로 복잡한 메커니즘이 작동하는 것처럼 보이지만 단순한 행동모델(Behavioral Model)만을 이용하여 시뮬레이션 가능한 것으로 입증되었다[10]. 유연성, 강건성, 분산형 제어 및 자기 조직화(Self-organized)와 같은 장점 때문에 무리지능과 행동모델을 이용한 복수 무인기 제어 연구가 활발히 전개되고 있으며, 특히 무리지능을 응용한 무리로봇공학(Swarm Robotics)이 복수 무인기 제어 방식의 대안으로 각광받고 있다[11].

행동모델 기반의 제어를 개발하기 위해서는 로봇(무인기)이 센서를 통해 인지하는 상황별 대응 규칙을 설계해야 하는데, 로봇이 센서를 통해 받아들이는 정보와 그에 대응하는 방식에 대한 관점이(Proximal Point of View) 설계자가 외부에서 바라보는 로봇의 반응에 대한 인식의 관점과(Distal Point of View) 다르기 때문에 논리적으로 행동규칙을 설계하는 방법은 존재하지 않는다. 로봇이 센서를 통해 정보를 받아들이면 이에 대해 반응함으로써 센서 정보가 바뀌게 되고 이는 다시 다른 형태의 로봇의 반응을 야기하는 과정이 반복됨으로써 로봇과 환경이 서로 긴밀하게 상호작용하는 형태가 되는데 이러한 내부 작용을 제어기 설계자는 관찰할 수 없고, 단지 외부로 드러난 로봇의 반응만을 볼 수 있기 때문에 센서와 반응간 또는 로봇과 환경간의 정확한 인과관계를 파악하는 것이 불가능하다. 즉, 어떤 센서 인식에 대해서 어떤 행동규칙을 대응시켜야(Sensory-Motor Mapping) 원하는 제어를 설계할 수 있는지 추론하기 곤란하다는 것이다[12].

이러한 행동모델의 단점에 대한 대안으로 제시된 진화로봇공학(Evolutionary Robotics)의 장점은 제어기 설계자가 로봇의 세세한 행동모델(센서-반응간 규칙)을 설계할 필요가 없다는 것이다. 행동모델이 유전알고리즘(Genetic Algorithm)을 통하여 신경망제어기의 진화를 통해 내재적으로 구현되는 방식이기 때문에, 설계자의 직관에 의지하지 않고 로봇과 주변 환경의 상호작용을 그대로 포용한 제어를 구현하게 하는 것이다. 지금까지는 주로 로봇용 제어기 개발 수단으로 연구되어 왔으나, 최근 무인기의 제어기 개발에 적용하려는 연구가 시도되고 있다.

Richards et al.[13]은 일정 영역을 복수의 무인기로 탐색하는 문제에 유전프로그래밍(Genetic Programming)을 적용함으로써 동일한 지역을 중복 탐색할 확률이 최소화되도록 무인기의 방위각 결정 분지도(Decision Tree)를 진화시킬 수 있음을 시뮬레이션으로 확인하였다.

Ruini와 Cangelosi[14]는 진화로봇공학을 이용하여, 4대의 무인기에 대한 상대 거리와 방위각이 주어지는 목표물을 제거(근처에 접근하여 자폭)하는 임무 수행을 위해 1개의 은닉층을 가지는 피드포워드 신경망제어기를 진화시켰다. 신경망제어기는 목표물까지의 거리와 방위각 및 가장 인접한 장애물까지의 거리를 입력 받아서 장애물을 회피하며 목표물에 접근하기 위한 무인기의 방위각 및 폭발 여부를 출력하는 구조이다.

Nardi et al.[15]는 초소형(총중량 195g의 Hirobo XRB Model) 무인헬기의 컬렉티브 및 싸이클릭 제어를 위한 신경망제어기를 유전알고리즘을 이용하여 개발하였다. 보편적인 신경망 구조가 아닌 비례미분 제어기와 유사한 구조를 가지는 신경망제어기의 각 입력에 대한 가중치를 유전알고리즘을 이용하여 결정하였다. 요방향 제어 관련 가중치만 우선 진화시키고 이후에 모든 가중치를 함께 진화시키는 점진적 진화(Incremental Evolution) 기법을 적용하여 목표 경로점을 성공적으로 통과하는 제어기를 진화시키는데 성공하였다.

Floreano et al.[16]는 20대의 소형 무인기로 반경 550m 이내의 조난자를 자동 탐색하고, 조난자를 발견하면 구조본부와의 임시 통신 네트워크를 자동으로 구성하는 무인기용 신경망제어기를 유전알고리즘을 이용하여 개발하였다. 시뮬레이션으로 가장 성공적인 제어기가 탑재된 무인기들의 비행경로를 분석한 후 이를 2개의 간단한 제어원리로 재구성하여, 이를 적용한 무인기의 성능이 성공적임을 여러 조건하에서 시뮬레이션으로 입증하였다. 진화된 신경망제어기의 핵심을 추출하여 이를 다시 보편적인 제어기로 재구성하는 독특한 개념을 제시하였다.

Barlow et al.[17]은 유전 프로그래밍을 이용하여 레이더 위치를 찾아서 추적하는 무인기용 제어기를 개발하였다. 유사하중이 제한적인 무인기에 경량, 저가형 센서를 장착함으로써 오차가 큰 레이더 신호세기와 방위각을 입력 받는 경우에도, 레이더를 추적하기 위한 무인기의 롤각을 출력하는 제어기를 성공적으로 진화시킬 수 있었으며 특히 점진적 진화기법을 적용할 경우 일반적인 진화알고리즘을 적용할 경우 대비 월등한 성능을 나타냄을 제시하였다.

본 논문에서는 제한된 영역을 복수의 무인기로 탐색하는 임무를, 직관에 의지하여 설계된 행동모델 기반의 신경망제어기로 수행한 결과와 진화로봇공학을 적용하여 진화시킨 신경망제어기로 수행한 결과를 비교하여 제시하였다. 지금까지 진화로봇공학을 무인기용 제어기에 적용한 연구는 지나치게 단순화한 항공기 운동모델(고도 및 속력을 고정시키고 선회율 제한 정도만 고려)을 이용하였으며 다른 종류의 알고리즘에 기초한 제어기와의 성능비교가 수행되지 않아서, 무인기용 제어기 개발에 있어서 진화로봇공학의 상대적인 유용성을 제시하지 못하였다. 본 논문에서는 3자유도 질점 항공기 운동방정식을 적용함으로써 보다 실제에 가까운 항공기 운동을 고려하였고 특히 행동모델 기반 제어기와의 성능 비교를 통하여 진화로봇공학을 이용하여 진화시킨 제어기의 상대적인 유용성을 확인하였다.

II. 본 론

2.1 무인기 운동모델 및 제어알고리즘

무리를 이루는 각 무인기의 제어는 Anderson과 Robbins[18]가 행동모델을 이용한 무인기의 편대비행 제어에 활용한 것으로, 아래와 같은 3자유도 질점 운동방정식 기반의 항공기 제어 알고리즘을 적용하였다. 이후 식에서 하첨자 c 는 명령값을 의미한다.

$$\dot{V} = g \left[\frac{T-D}{W} - \sin\gamma \right] \quad (1)$$

$$\dot{\gamma} = \frac{g}{V} (n \cos\mu - \cos\gamma) \quad (2)$$

$$\dot{\chi} = \frac{g n \sin\mu}{V \cos\gamma} \quad (3)$$

$$\dot{x} = V \cos\gamma \cos\chi \quad (4)$$

$$\dot{y} = V \cos\gamma \sin\chi \quad (5)$$

$$\dot{h} = V \sin\gamma \quad (6)$$

여기서 D , T , V , W 는 각각 무인기의 항력, 추력, 속도 및 중량을, x , y , h 는 위치 및 고도를, γ , μ , χ 는 경로각, 경사각 및 방위각을 의미하고 g 는 중력가속도, n 은 하중배수를 나타낸다. 항공기의 추력 및 선회 동특성은 다음과 같은 1차 미분방정식으로 모델링하며 τ_b , τ_n , τ_t 는 각각 경사각, 하중배수 및 추력 시상수를 의미한다.

$$\dot{T} = \frac{T_c - T}{\tau_t} \quad (7)$$

$$\dot{\mu} = \frac{\mu_c - \mu}{\tau_b} \quad (8)$$

$$\dot{n} = \frac{n_c - n}{\tau_n} \quad (9)$$

속력, 경로각 및 방위각 제어 특성은 다음과 같은 1차 미분방정식에 따르길 원하는 것으로 가정하고 $\omega_v, \omega_\gamma, \omega_\chi$ 는 각각 속력, 경로각 및 방위각 제어기의 대역폭을 의미하며 하첨자 d는 요구값(desired)을 의미한다.

$$\dot{V}_d = \omega_v (V_c - V) \quad (10)$$

$$\dot{\gamma}_d = \omega_\gamma (\gamma_c - \gamma) \quad (11)$$

$$\dot{\chi}_d = \omega_\chi (\chi_c - \chi) \quad (12)$$

식(1)~(3) 및 (10)~(12)로부터

$$T_c = D + \frac{\omega_v W (V_c - V)}{g} + W \sin \gamma \quad (13)$$

$$n_c \cos \mu_c = \frac{\omega_\gamma V (\gamma_c - \gamma)}{g} + \cos \gamma = a \quad (14)$$

$$n_c \sin \mu_c = \frac{\omega_\chi V (\chi_c - \chi) \cos \gamma}{g} = b \quad (15)$$

를 얻을 수 있다. 또한 식 (14),(15)로부터

$$n_c^2 = a^2 + b^2 \quad (16)$$

$$\mu_c = \tan^{-1} \left(\frac{b}{a} \right) \quad (17)$$

를 얻을 수 있다.

본 연구에서는 고도차를 이용한 충돌회피 방식의 배제를 위하여 항상 수평선회 제어가 수행되는 것으로 가정하였다($\dot{\gamma}=0$). 이상의 모델을 이용한 무인기 제어 알고리즘은 다음과 같으며 하첨자 't+1'은 다음 시간 구간 값을 의미한다.

- a) 항공기 초기 상태 Trim 계산
- b) 행동모델(2.3항)에 근거한 $\vec{V}_c (V_c, \gamma_c, \chi_c)$ 설정
- c) 식(13)~(17)로부터 T_c, n_c, μ_c 계산
- d) 식 (7)~(9)로부터 $(T, n, \mu)_{t+1}$ 계산
- e) 식 (1)~(6)으로부터 $(V, \gamma, \chi, x, y, h)_{t+1}$ 계산
- f) 변화된 항공기 위치에서 새로운 \vec{V}_c 설정
- g) 새로운 행동모델 적용시까지 c)~f)항 반복

중 량	11 kg
날개면적	0.55 m ²
최대추력	13 kg
실속속도	15 m/s
최대 양력계수	1.57
최소 양력계수	-0.33
최대 경사각	45도
항력 모델	$C_D = 0.0434 + 0.0278 C_L^2$
τ_t, τ_b, τ_n	1.5 sec
ω_v	0.3 sec ⁻¹
ω_γ	0.2 sec ⁻¹
ω_χ	0.5 sec ⁻¹
트림 속도	25 m/s
트림 고도	1000 m

무인기 모델은 Unmanned Dynamics사가 공개한 AeroSim Blockset[19]에 포함된 6자유도 에어로존데(Aerosonde) 모델을 이용하여 3자유도 질점 모델을 위한 파라미터를 상기와 같이 추출하여 적용하였다. (C_D : 항력계수, C_L : 양력계수)

2.2 시뮬레이션 환경 및 신경망 구조

탐색영역은 4km × 4km 크기의 정방형으로 200m × 200m 크기의 400개 셀로 분할한 페로몬 지도(Pheromone Map)에 대응시켰다. 페로몬 지도의 좌측 및 하부 방향으로 200m 폭의 경계 외부는 완충지역으로 무인기가 본 지역에 진입하는 것은 허용되나 완충지역 외부로 이동시에는 파괴시킴으로써 무인기가 탐색 영역을 과도하게 벗어나지 않도록 유도하였다. 페로몬 지도의 상부 및 우측은 지형지물 등의 장애물 지역으로 무인기가 이 방향으로 탐색영역을 이탈하면 장애물과 충돌한 것으로 가정하여 파괴시킨다. 단 두 가지 경계 이탈의 경우 모두 선회에 필요한 공간 소요로 인하여 잠시 경계를 벗어난 경우는 파괴시키지 않았다(Fig. 1).

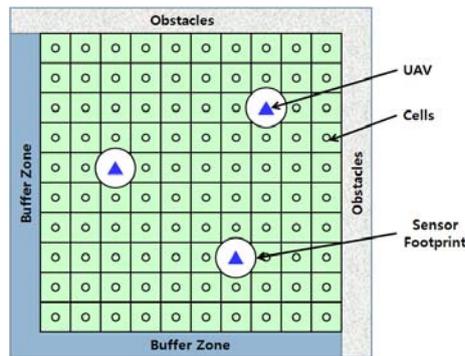


Fig. 1. Pheromone Map

페로몬 지도는 무리지능의 대표적 사례인 개미가 페로몬을 이용하는 방식을 복수 무인기 제어에 응용하기 위한 수단으로, 이동 경로 상에 페로몬을 분비하는 개미의 방식을 항공기에는 적용할 수 없기 때문에 관심지역의 가상 페로몬 분포 상태를 표시할 수 있는 페로몬 지도 개념을 이용한다[3~7]. 즉, 탐색영역에 대응하는 가상의 페로몬 지도를 설정하고 무인기가 탐색한 횟수에 비례하여 해당 셀의 페로몬 강도를 증가시키는 동시에 페로몬회피 행동모델을 적용함으로써, 해당 셀을 반복 탐색할 가능성을 최소화하여 대상 영역을 효율적으로 탐색하는 개념이다.

초기에 10대의 무인기 위치 및 방위각은 무작위로 설정하고 이후 각 무인기들은 2.3항에 설명할 행동모델에 의하여 다음 이동할 목표 셀의 중심부(셀 중심 기준 20m 반경의 원)로 유도되며, 목표 셀의 중심부에 도달할 때까지는 행동모델을 고려하지 않고 2.1항의 알고리즘을 이용한 유도만 진행된다(Fig. 2). Fig. 3에 2.1항과 2.2항의 무인기 제어 및 유도 모델 블록선도를 나타내었다.

무인기가 목표 셀의 중심부에 도달하면 해당 셀은 탐색된 것으로 간주하여 페로몬 강도를 1만큼 증가시킨다. 탐색영역 내부 셀의 페로몬 강도 초기값은 1이며, 탐색영역 외부 셀의 페로몬 강도는 100으로 설정함으로써 무인기가 탐색영역을 벗어나거나 장애물과 충돌하지 않도록 유도한다.

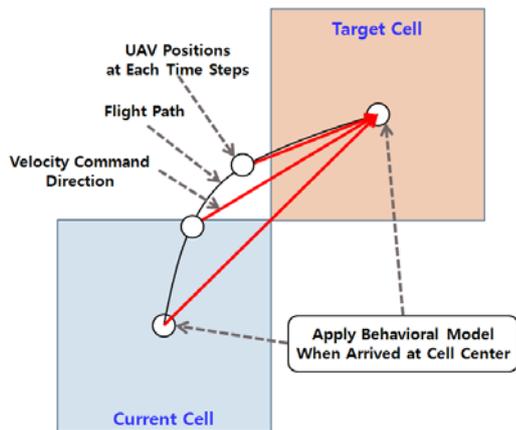


Fig. 2. Guidance Logic

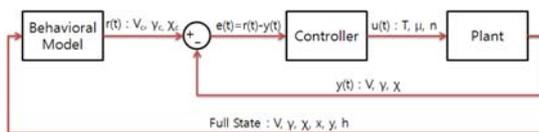


Fig. 3. Control Block Diagram

무인기간 거리가 30m 이하일 경우 충돌이 이루어진 것으로 가정하여 해당 무인기는 모두 파괴시키며 본 과정을 다음과 같은 시뮬레이션 정지 조건 중 하나라도 만족할 때까지 반복 수행한다.

- 최대 계산 수행시간(1,000 초)에 도달
 - 100% 영역 탐색
 - 충돌 및 경계이탈로 인한 모든 무인기 파괴
- 모든 무인기는 실시간으로 페로몬 지도 정보와 다른 무인기의 위치 정보에 접근 가능하다고 가정하였다(완전한 분산형 제어방식의 구현을 위해서는 각 무인기가 직접 지나온 셀에 대한 페로몬 지도 정보만 보유하다가 통신 가능 거리 이내로 접근한 무인기간에 위치 정보와 페로몬 지도 정보를 교환함으로써 다른 무인기의 위치 정보와 페로몬 지도 정보를 갱신해 나가는 분산형 페로몬 지도[3] 방식의 구현이 필요하며 본 논문의 알고리즘에 쉽게 적용 가능하다).

본 논문에서는 퍼셉트론과 피드포워드의 2가지 신경망이 이용되었으며 피드포워드 신경망 제어기의 구조는 Fig. 4와 같은데 퍼셉트론의 경우에는 은닉층 없이 입력층과 출력층의 2개 층으로만 구성된다. 퍼셉트론 신경망 출력함수는 출력을 이진부호화(Binary Coding)하기 위하여 식 (18)과 같이 하드리밋(Hard Limit) 함수를 사용하였는데 y 는 출력벡터, p 는 입력벡터, W 는 가중치 행렬, b 는 바이어스 벡터를 나타낸다.

$$y = \text{hardlim}(W \times p + b) \quad (18)$$

피드포워드 신경망의 경우에도 식 (19), (20)과 같이 은닉층과 출력층 모두 하드리밋(Hard Limit) 함수를 적용하였으며 하첨자 1은 은닉층, 하첨자 2는 출력층을 나타낸다.

$$y_1 = \text{hardlim}(W_1 \times p + b_1) \quad (19)$$

$$y_2 = \text{hardlim}(W_2 \times y_1 + b_2) \quad (20)$$

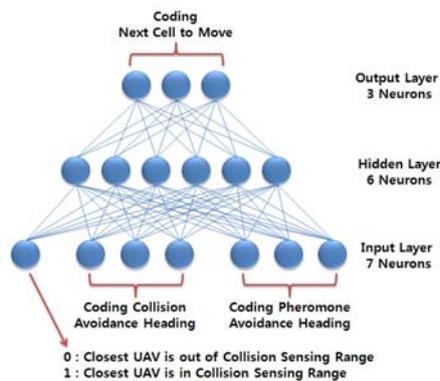


Fig. 4. Structure of the Feedforward Neural Net Controller

2.3 행동모델

본 시뮬레이션에 이용한 행동모델은 충돌 및 페로몬 회피의 2가지로 구성된다.

충돌회피 : 자신으로부터 가장 인접한 무인기와의 거리가 500m 이내면 서로 반대 방향으로 이동한다. 충돌감지거리는 너무 작으면 충돌위험이 증가하고, 너무 크면 효율적인 탐색이 어려워지므로 적절한 값을 설정해야 하는데 파라미터 시뮬레이션을 통하여 가장 높은 영역탐색율을 보이는 500m로 설정하였다.

페로몬회피 : 충돌감지거리 내에 다른 무인기가 존재하지 않을 경우, 주변 셀들의 페로몬 강도 분포를 이용하여 페로몬회피 벡터를 구하고 해당 방향의 셀로 이동한다(Fig. 5). 주변의 페로몬 강도 분포가 대칭을 이루어 페로몬회피 벡터의 크기가 0인 경우 주변 8개의 셀 중 무작위로 이동한다.

본 행동모델을 퍼셉트론 학습법 및 역전파 학습법을 이용하여 각 신경망에 학습시킴으로써 신경망의 가중치와 바이어스가 결정되는데, 입력층은 가장 인접한 무인기와의 거리가 충돌감지거리 이내일 경우 1, 초과일 경우 0을 지시하는 한 뉴런과, 충돌회피 및 페로몬회피 방위각을 이진부호화하는 각 3개의 뉴런을 합하여 총 7개의 뉴런으로 구성된다(Fig. 4). 회피 방위각은 행동모델에 의해 결정되는, 다음에 이동할 목표 셀의 방위각을 지시하는 것으로 0도에서 315도까지 45도 간격으로 8개의 방위각이 존재하는데 이를 Fig. 6과 같이 3개의 뉴런에 이진부호화 하였으며 출력층은 다음 이동할 목표 셀의 방위각을 동일한 방식으로 3개의 뉴런에 이진부호화하였다. 신경망 학습을 위한 패턴은 첫째 입력 뉴런 값이 1일 경우 충돌회피 방위각이, 첫째 입력 뉴런 값이 0일 경우 페로몬회피 방위각이 출력되는 패턴이다. 예를 들어 입력이 [1 0 0 1 1 0 0]이라면 가장 가까운 무인기가 충돌감지거리 이내에 존재하고, 행동모델에 의한 충돌회피 방위각은 45도(0 0 1), 페로몬회피 방위각은 180도(1 0 0)라는 의미이며 이 입력에 대한 신경망 출력은 다른 무인기가 충돌감지거리 내에 존재하므로 [0 0 1], 즉 충돌회피 방위각이 출력되도록 학습시킨다. 피드포워드 신경망의 은닉층 뉴런수는 충돌회피 및 페로몬회피 행동모델을 역전파 학습법으로 완벽하게 학습시킬 수 있는 최소 뉴런 수인 6개로 구성하였다. 퍼셉트론의 경우 충돌회피 및 페로몬회피 행동모델 패턴이 선형 분리 불가능하므로 완벽하게 학습시킬 수 없다. 학습이 완료되어 가

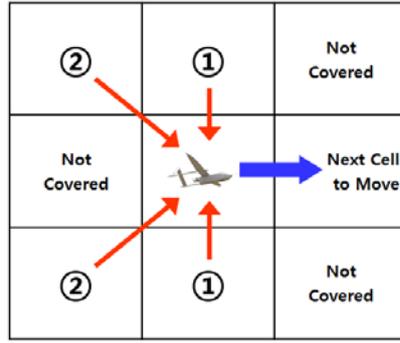


Fig. 5. Pheromone Avoidance Algorithm (numbers in cells mean pheromone strength)

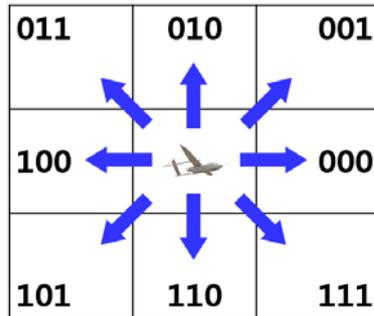


Fig. 6. Binary Coding of Headings

중치와 바이어스가 결정된 신경망 제어기를 이용하여 임의의 초기치(무인기의 초기 위치 및 방위각)에 대하여 100회의 시뮬레이션을 수행하였다.

2.4 유전알고리즘

진화로봇공학은 설계자의 직관에 의존하여 행동모델을 설계하는 대신에 유전알고리즘[20]을 이용하여 신경망의 가중치와 바이어스를 진화시키는 방식이다. 2.3항의 행동모델 기반 제어는 신경망에 직관적으로 설계된 행동모델을 학습시키는 방식인데 반하여, 진화로봇공학 기반 제어는 신경망에 주변환경에 따라 결정되는 7개의 뉴런 입력값을 제공하는 점은 동일하지만 이를 이용하여 신경망의 출력값을 결정하는 가중치와 바이어스가 유전알고리즘을 통하여 자동으로 결정된다는 점이 상이하다. 즉, 최적의 적합도(Fitness)를 나타내는 신경망 제어기의 가중치와 바이어스 값을 시뮬레이션을 통하여 스스로 진화시켜 나가는 방식으로 신경망 제어기가 결정된다.

무작위로 설정된 초기 가중치와 바이어스를 갖는 신경망제어기를 사전에 설정된 개체 수 (Population) 만큼 생성하여 제 1세대

(Generation)를 구성하고 시뮬레이션을 통해 높은 적합도를 보이는 개체들을 선별한 후, 복사(Elitism) 및 돌연변이(Mutation)를 이용하여 다음 세대를 구성하는 방식을 일정 세대 동안 반복함으로써 주어진 적합도를 극대화시키는 신경망의 가중치와 바이어스를 도출한다. 본 시뮬레이션의 목적은 일정 시간동안 최대한 많은 영역을 탐색하는 데 있으므로 적합도는 영역탐색율(Coverage, 전체 400개의 셀 중 탐색된 셀의 비율)로 설정하였다.

유전 알고리즘에 적용된 변수의 설정치는 개체 수 100, 최대 세대 수 200, 엘리트(Elite) 수 20, 돌연변이 확률 0.1이며 초기 가중치와 바이어스는 -0.1과 0.1 사이의 무작위 값으로 설정하였다. 최대 세대 수에 도달하거나, 현 세대로부터 이전 50세대의 적합도 평균값이 현 세대로부터 20세대 이전에서 50세대 이전까지 30세대의 적합도 평균치에 비하여 0.5% 이상 증가하지 않으면 진화가 한계상태에 이른 것으로 보고 진화과정을 중지시킨다. 한 세대를 구성하는 각 100개의 개체에 대하여 시뮬레이션을 수행하고(무인기들의 초기 상태에 의존하는 신경망제어기가 진화되지 않도록, 각 개체에 대하여 무인기들의 초기 위치 및 방위각이 무작위로 설정되는 10회의 시뮬레이션을 수행한 후 그 평균값을 적합도로 설정하였다) 적합도가 가장 높은 20개의 엘리트를 선택하여 4배수로 복사함으로써 나머지 80 개체를 생성한다. 20개의 엘리트와 돌연변이가 적용된 80개의 복사 개체로 다음 세대를 구성하는 100개의 개체가 생성되며 돌연변이는 각 가중치 및 바이어스에 0.1의 확률로 -1과 1사이의 무작위 값을 더해주는 방식으로 적용되었다.

일반적인 유전알고리즘에서 다음 세대를 구성하기 위해 주로 이용되는 교차(Crossover)는 신경망제어기의 가중치를 진화시키는 경우에는 적용하지 않는 것이 일반적이므로[12] 엘리트와 돌연변이를 이용하여 다음 세대를 구성하게 되었다. 본 시뮬레이션이 매번 임의의 초기치에 대하여 수행된 10회의 평균치를 적합도로 설정하는 통계적 방식이기 때문에 적합도의 순위를 결정론적으로 지정할 수 없으며 따라서 적합도에 따라 선택확률이 정해지는 룰렛휠이나 랭크 등의 선택 방식은 본 연구에 적용하기에는 부적합하여, 상위 20%의 적합도를 기록한 개체를 모두 엘리트로 지정하고, 이를 4배수로 복사한 뒤에 돌연변이를 적용한 80 개체와 함께 다음 세대를 구성하는 방식을 적용하였다. 이는 참고문헌 12에서 로봇의 신경망 제어기를 진화시키는 사례에서 빈번

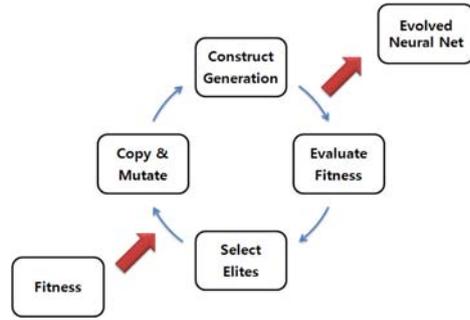


Fig. 7. Evolutionary Process

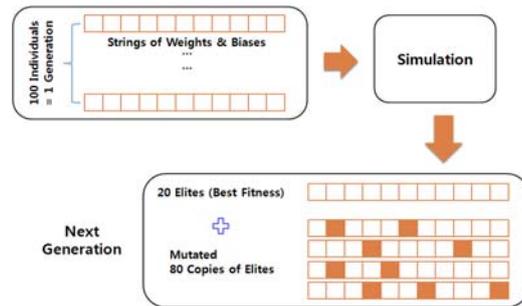


Fig. 8. Genetic Algorithm

하게 제시된 방식이며 참고문헌 14, 15에서도 동일한 방식이 적용되었다. Fig. 7 및 8에 이상의 유전 알고리즘을 이용한 신경망제어기의 진화과정을 도식화하였다.

2.5 시뮬레이션 결과 및 검토

본 논문에서는 충돌회피 및 페로몬회피 행동 모델을 퍼셉트론(Perceptron) 및 피드포워드(Feedforward) 신경망에 학습시킨 직관에 의한 신경망제어기(2.3항)와 진화로봇공학을 이용하여 진화시킨 신경망제어기(2.4항)의 비교 시뮬레이션을 통하여 진화로봇공학의 유용성을 제시하고자 하였으며, 이를 위해 4가지 시뮬레이션을 비교 수행하였다(Fig. 9). 경우 I 및 II는 퍼셉트론 및 피드포워드 신경망에 충돌회피 및 페로몬회피 행동 모델을 학습시킨 것으로 경우 II는 충돌회피 및 페로몬회피 논리가 완벽하게 학습된 것이며 경우 I은 신경망 구조상 완벽한 학습이 불가능하다. 경우 III 및 IV는 각각 퍼셉트론 및 피드포워드 신경망을 유전 알고리즘으로 진화시킨 것이며, 4가지 시뮬레이션 케이스를 Table 1에 정리하였다.

2.1항의 제어 알고리즘을 이용하여 2.2항의 시뮬레이션 환경에서 수행된 4가지 시뮬레이션 결과를 Table 2에 정리하였다. I, II의 경우는 학습

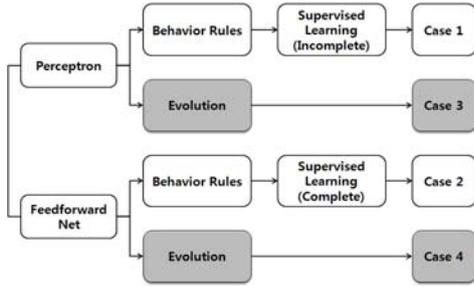


Fig. 9. Simulation Cases

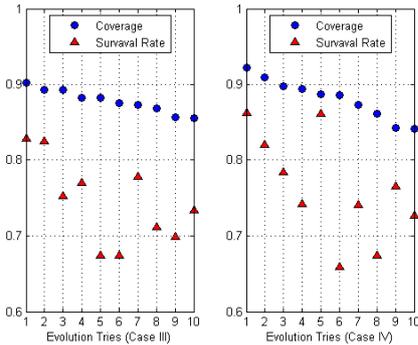


Fig. 10. 10 Evolved Results of Case III & IV (Averaged over 100 Tries)

Table 1. Simulation Cases Summary

Case	Net Structure	Weight & Bias
I	Perceptron	Supervised Learning
II	Feedforward Net	
III	Perceptron	Genetic Algorithm
IV	Feedforward Net	

Table 2. Simulation Results Summary (Averaged over 100 Tries)

CA/PA : Collision/Pheromone Avoidance

Case	Coverage	Survival Rate	Average CA Error	Average PA Error
I	71.5 %	45.9 %	64.7 deg	16.9 deg
II	84.6 %	83.0 %	0 deg	0 deg
III	90.2 %	82.9 %	74.5 deg	35.2 deg
IV	92.2 %	86.2 %	70.3 deg	30.2 deg

이 완료된 신경망을 이용하여 100회의 시뮬레이션을 수행한 후 그 평균값을 제시한 것이고, III, IV의 경우는 각각 10회의 진화를 시도하여 100회의 시뮬레이션 평균값(Fig. 10, 적합도 순위로

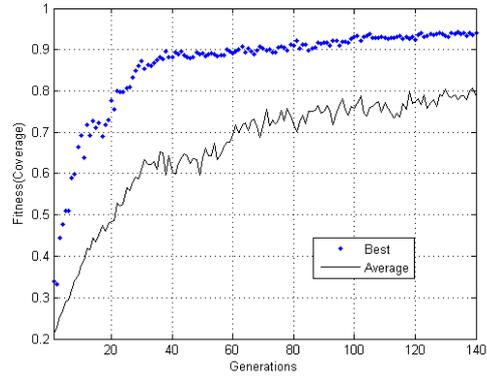


Fig. 11. Fitness Trend

정렬)이 가장 높은 적합도를 보이는 신경망의 결과를 제시한 것이다. Fig. 11은 세대가 진행됨에 따라 유전 알고리즘에 의해 신경망제어기의 성능이 진화되는 것을 보여주는 것으로 본 논문에서 수행된 대표적인 사례를 나타낸 것이다.

생존율(Survival Rate)은 10대의 무인기 중 충돌이나 영역 경계를 벗어남으로써 파괴되지 않고 시뮬레이션 종료 시점까지 살아남은 무인기의 비율을 의미한다. 적합도에 생존율을 추가시키면 생존율이 1에 가까운 신경망제어기를 진화시킬 수 있으나 영역탐색율은 저하되므로, 본 연구에서는 생존율에 관계없이 영역탐색율을 극대화시키는 신경망 제어기를 구현하고자 하였다.

평균충돌회피오차는 64가지 입력 패턴의 조합(충돌감지거리 이내에 다른 무인기가 존재할 경우에 대하여, 충돌회피 방위각 8가지 × 페로몬회피 방위각 8가지)에 대한 오차(충돌회피 행동모델에 의한 방위각과 신경망이 출력하는 방위각의 차이)의 평균값이며, 평균페로몬회피오차도 충돌감지거리 이내에 다른 무인기가 존재하지 않을 경우에 대하여 같은 개념으로 계산된 값이다.

Fig. 12는 각 경우의 시뮬레이션 예(비행궤적 및 페로몬지도)를 표시한 것이다. 비행궤적에서 음영으로 표시된 셀은 각 무인기의 초기 위치이며, 페로몬지도상의 음영표시는 시뮬레이션 종료 시점의 페로몬 강도를 농담으로 표시한 것인데, 값이 1인 경우 한 번도 탐색되지 않은 셀이며 이후 방문 횟수에 따라 해당 셀 값이 1씩 증가한다. 경우 I을 제외하고는 전반적으로 고르게 전 지역이 탐색되었음을 확인할 수 있다.

경우 I의 퍼셉트론은 충돌회피 및 페로몬회피 알고리즘을 완전하게 학습하지 못했기 때문에 서로 간 충돌 및 장애물 충돌(경계 이탈)에 의하여 많은 무인기가 파괴되고 이에 따라 낮은 영역탐

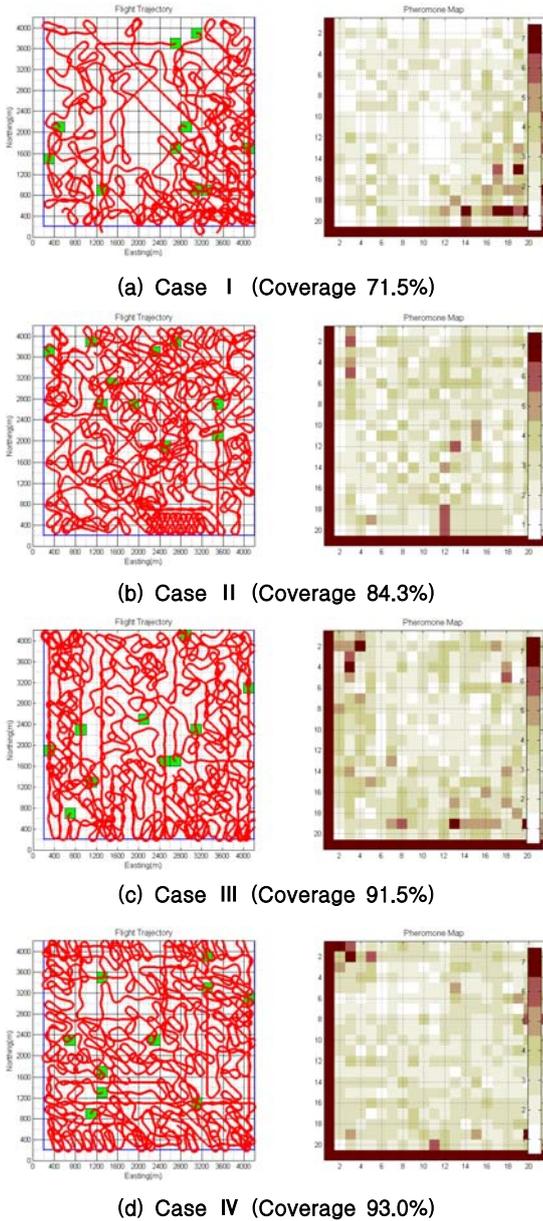


Fig. 12. Simulation Examples of 4 Cases

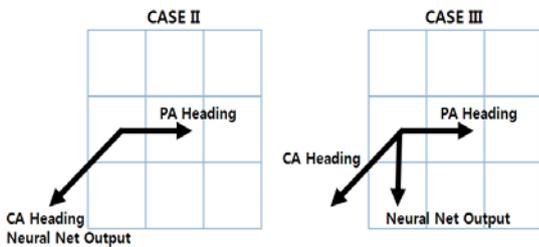


Fig. 13. Comparison of Case II & III

색을 나타내었다. 반면에 경우Ⅱ의 피드포워드 신경망은 충돌회피 및 페로몬회피 알고리즘을 완전하게 학습하였으므로 84.6%의 상대적으로 높은 영역탐색율을 나타내었다. 충돌회피 알고리즘을 완전하게 학습했는데도 생존율이 낮은 이유는 영역의 경계 위치에서 충돌회피 방위각이 영역 외부로 지향할 경우 행동모델에 따라 영역 외부로 이동하여 해당 무인기가 파괴되기 때문이다. 충돌회피 알고리즘 구현에 실패하면 2대의 무인기가 파괴되고 페로몬회피 구현(경계 외부 셀의 페로몬 강도를 100으로 설정하여 경계 이탈 방지를 유도)에 실패하면 1대의 무인기만 파괴되므로 충돌회피가 페로몬회피에 우선순위를 갖는 논리는 합리적이다.

경우Ⅲ의 퍼셉트론은 동일한 신경망 구조를 가짐에도 불구하고 행동모델 기반의 경우Ⅰ에 비하여 상당한 성능향상을 보이고 있으며 충돌회피 및 페로몬회피 알고리즘을 완전하게 학습한 경우Ⅱ와 비교해도 약간 높은 성능을 보인다. 이는 진화로봇공학의 효용성을 보여주는 것으로 직관에 의한 행동모델 기반의 제어기보다 유전알고리즘에 의하여 무인기간 및 무인기와 환경간 상호작용을 스스로 고려함으로써 진화된 가중치와 바이어스를 갖는 신경망제어기가 더 나은 성능을 나타냄을 보여주는 것이다. 경우Ⅲ의 평균충돌회피오차 및 평균페로몬회피오차가 경우Ⅱ와 비교할 수 없을 정도로 큰데도 더 높은 영역탐색율을 나타내는 것은 Fig. 13에서 알 수 있듯이 경우Ⅲ의 신경망제어기는 충돌 위험을 감지하면 경우Ⅱ와 같이 충돌회피 방향으로 이동하는 방위각을 출력하지 않고, 대부분 충돌회피와 페로몬회피 방위각 사이의 타협적인 방위각을 출력한다. 이를 통하여 앞에서 설명했듯이 영역의 경계 위치에서 충돌회피를 위한 이동 방향이 영역 외부로 지향할 경우 무인기가 파괴되는 경우Ⅱ 제어기의 단점을 극복할 수 있는 것이다. 직관적인 논리는 고려하기 어려운 해결책을 진화로봇공학 기법을 통하여 개발할 수 있다는 것을 알 수 있다.

행동모델 기반의 신경망제어기는 퍼셉트론과 피드포워드 신경망의 성능차이가 크지만(71.5 → 84.6%), 진화된 신경망제어기의 경우 피드포워드 신경망제어기의 성능 개선 정도가 미미함을 알 수 있다(90.2 → 92.2%). 이는 진화로봇공학의 한계를 보여주는 것으로 즉, 성능을 최적화시킬 수 있는 신경망의 구조를 예단하기 곤란하다는 것이다.

진화된 신경망의 시뮬레이션 과정에서 입출력 뉴런값의 히스토리를 저장하여 분석하면 입력충

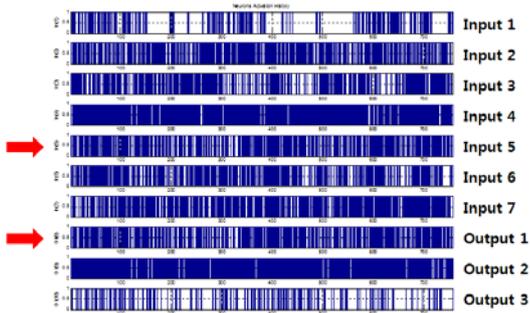


Fig. 14. Neuron Activation History (Black 1, White 0)

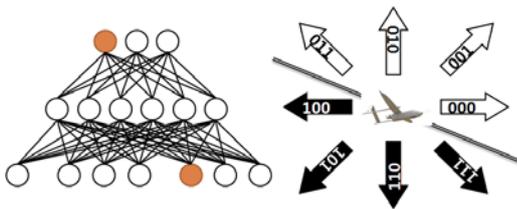


Fig. 15. Evolution of Pheromone Avoidance

의 5번 뉴런(페로몬회피 방위각의 최상위 비트)과 출력층의 1번 뉴런(신경망 제어기 출력의 최상위 비트) 간의 상관관계가 거의 모든 경우에 1 또는 1에 가까운 값을 가지도록 진화되었음을 관찰할 수 있다. 이는 페로몬회피 헤딩 입력과 신경망 출력의 오차가 135도 이내로 유지됨을 의미하는 것으로 진화 과정에 행동모델에 대한 대응 정보가 주어지지 않았음에도 불구하고 페로몬회피 본능이 진화를 통해 스스로 구현되었음을 의미하는 것이다(Fig. 14, 15). Ruini와 Cangelosi [14]는 방위각 부호화에 있어서 그레이부호화(Gray Coding) 방식의 결과가 이진부호화 방식에 비하여 더 좋다고 제시하였으나 본 연구에서는 이진부호화 방식이 더 좋은 결과를 나타내었으며 그 이유는 Fig. 15에서 알 수 있듯이 페로몬회피 본능의 구현에 있어서 이진부호화 방식이 더 유리하기 때문이다.

III. 결 론

복수 무인기로 영역탐색율을 극대화시키는 문제에 행동모델 기반의 신경망제어기와 진화로봇공학을 이용하여 진화시킨 신경망제어기를 적용한 결과 다음과 같은 결론을 도출하였다.

- 퍼셉트론 신경망제어기의 경우 행동모델 기반의 신경망제어기에 비하여 진화된 신경망제어

기가 월등한 성능을 보이는데, 무인기간 및 무인기와 환경간 상호작용을 고려할 수 있는 진화로봇공학의 특성상 직관적으로 추론하기 어려운 해결책을 이용할 수 있기 때문이다.

- 행동모델 기반의 신경망제어기와는 달리 진화된 신경망제어기는 퍼셉트론과 피드포워드 신경망의 성능 차이가 크지 않았다. 이는 제어기가 논리적으로 구현되지 않고 주어진 환경과의 상호작용을 통한 진화 과정에서 스스로 조직화되고 (Self-Organized) 창발되는(Emergent) 진화로봇공학의 특성상 최종 진화된 신경망제어기의 성능을 예상하기 곤란하고 따라서 논리적인 접근이 곤란하다는 한계를 의미하는 것이다.

- 그러나 이러한 한계에도 불구하고 복수 무인기를 이용한 영역 탐색 문제에 있어서, 논리적인 행동규칙을 설정하기 어려운 행동모델 기반 제어기의 단점을 설계자의 직관에 의지하지 않고 주변 환경과의 상호작용을 그대로 포용할 수 있는 장점을 가진 진화로봇공학 기법을 이용하여 극복할 수 있음을 확인하였다.

본 논문에서는 3자유도 항공기 운동모델을 적용하였으나 수평선회 가정을 도입하였기 때문에 실제로는 2자유도 운동모델을 적용한 것과 큰 차이가 없으며, 실제로 수평선회 제한을 없애더라도 시뮬레이션 결과에는 큰 차이가 없음을 확인하였다. 이는 본 논문에 적용한 충돌회피 및 페로몬회피 알고리즘 자체가 2자유도에 기반한 것에 따른 한계로, 3자유도 운동모델 적용의 효과를 위해서는 3자유도 기반의 충돌 및 페로몬회피 알고리즘을 적용해야할 것으로 판단되어 향후 이 부분으로 연구를 확장할 계획이다.

후 기

이 연구는 지식경제부 지원으로 수행하는 21세기 프론티어 기술개발사업(스마트무인기기술개발)의 일환으로 수행되었습니다.

참고문헌

- 1) Crowther, B. and Riviere, X., "Flocking of Autonomous Unmanned Air Vehicles", 17th Bristol UAV Conference, 2002.
- 2) Kuiper, E. and Tehrani, S. N., "Mobility Models for UAV Group Reconnaissance Applications", Wireless and Mobile Communications, 2006.

- 3) Erignac, C. A., "An Exhaustive Swarming Search Strategy based on Distributed Pheromone Maps", AIAA Infotech@Aerospace 2007 Conference and Exhibit, 2007.
- 4) Parunak, V. D., Purcell, M., and O'Connell, R., "Digital Pheromones for Autonomous Coordination Of Swarming UAV's", First AIAA Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference, 2002.
- 5) Bamberger, R. J., Watson, D. P., Sheidt, D. H., and Moore, K. L., "Flight Demonstrations of Unmanned Aerial Vehicle Swarming Concepts", *Johns Hopkins APL Technical Digest*, Vol. 27, No. 1, 2006.
- 6) Sauter, J. A., Matthews, R. M., Parunak, V. D., and Brueckner, S. A., "Performance of Digital Pheromones for Swarming Vehicle Control", 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems(AAMAS05), 2005.
- 7) Gaudiano, P., Shargel, B., Bonabeau, E., and Clough, B. T., "Swarm Intelligence: a New C2 Paradigm with an Application to Control of Swarms of UAVs", 8th ICCRTS Command and Control Research and Technology Symposium, 2003.
- 8) Vincent, P. and Rubin, I., "A Framework and Analysis for Cooperative Search Using UAV Swarms", Association for Computing Machinery(ACM) Symposium on Applied Computing, 2004.
- 9) Bonabeau, E., Dorigo, M., Theraulaz, G., *Swarm Intelligence : from Natural to Artificial Systems*, Oxford University Press, 1999, pp. 1~23.
- 10) Reynolds, C. W., "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Computer Graphics*, Vol. 21, No. 4, 1987, pp. 25~34.
- 11) 오수훈, "무리지능을 이용한 복수 무인기 제어", *항공우주산업기술동향지*, 제7권 제1호, 2009, pp. 141~152.
- 12) Nolfi, S. and Floreano, D., *Evolutionary Robotics; The Biology, Intelligence, and Technology of Self-Organizing Machines*, 1st Ed., MIT Press, 2000.
- 13) Richards, M. D., Whitley, D., and Beveridge, J. R., "Evolving Cooperative Strategies for UAV Teams", Genetic and Evolutionary Computation Conference(GECCO), 2005.
- 14) Ruini, F. and Cangelosi, A., "Distributed Control in Multi-Agent Systems: A Preliminary Model of Autonomous MAV Swarms", 11th International Conference on Information Fusion, 2008.
- 15) Nardi, R. D., Holland, O., Woods, J., and Clark, A., "SwarMAV: A Swarm of Miniature Aerial Vehicles", 21st Bristol UAV Systems Conference, 2006.
- 16) Floreano, D., Hauert, S., Leven, S., and Zufferey J. C., "Evolutionary Swarms of Flying Robots", International Symposium on Flying Insects and Robots, 2007.
- 17) Barlow, G. J., Oh, C. K., and Grant, E., "Incremental Evolution of Autonomous Controllers for Unmanned Aerial Vehicles using Multi-objective Genetic Programming", IEEE Conference on Cybernetics and Intelligent Systems, 2004.
- 18) Anderson, M. R. and Robbins, A. C., "Formation Flight as a Cooperative Game", Proceedings of the AIAA GNC Conference and Exhibit, 1998, pp. 244~251.
- 19) <http://u-dynamics.com/aerosim/>
- 20) Mitchell, M., *An Introduction to Genetic Algorithms*, 1st Ed., MIT Press, 1998.