

## GA를 적용한 모의 UAV의 자율착륙 테스트베드 구축

한창희<sup>†</sup>

### Generating an Autonomous Landing Testbed of Simulated UAV applied by GA

Changhee Han<sup>†</sup>

#### ABSTRACT

In case of unmanned aerial vehicles used in modern society, there has been a problem where a human operator should be still needed to control the UAV because of a lower level of autonomy. In this paper, genetic algorithm is selected as a methodology for the autonomy accomplishment and then we verify a possibility of UAV autonomy by applying the GA. The landing is one of the important classical tasks on aerial vehicle and the lunar Landing is one of the most historical events. Autonomy possibility of computer-simulated UAV is verified by landing autonomy method of a falling body equipped with a propulsion system similar to the lunar Lander. When applying the GA, the genome is encoded only with 4 actions (left-turn, right-turn, thrust, and free-fall) and applied onto the falling body, Then we applied the major operations of GA and achieved a success experiment. A major contribution is to construct a simulated UAV where an autonomy of UAV can be accomplished while minimizing the sensor dependency. Also we implemented a test-bed where the possibility of autonomy accomplishment by applying the GA can be verified.

**Key words** : Genetic Algorithm, Landing, Autonomy, Simulation, Unmanned Aerial Vehicle

#### 요 약

현대에서 사용되는 무인 비행체의 경우 자율화수준이 떨어져 사용자의 개입이 필요하다는 문제가 있다. 이에 본 연구는 자율화 달성의 방법론으로 유전자 알고리즘을 선택하고, 이것을 적용해 무인 착륙체의 자율화 가능성을 확인해 보는 것이다. 특별히 무인 비행체의 착륙은 비행체에서 매우 중요한 고전적인 과업 중의 하나이며, 착륙에 있어서 역사적 사건 중의 하나가 달착륙이라고 할 수 있다. 달 착륙선의 추동 메카니즘을 장착한 낙하체의 착륙 자율화 방법을 이용해 무인 착륙체의 자율화 가능성을 확인한다. 본 논문에서는 유전자알고리즘 적용시 계층을 단지 4가지 액션 (좌회전, 우회전, 분출, 자유낙하) 으로 인코딩하여 무인 착륙체에 적용하고, 유전자 알고리즘의 주요 연산을 접목하여 실험을 성공적으로 진행하였으며 센서에 대한 의존도를 최소화할 수 있는 무인 비행체의 모의 UAV를 제작하였고, 유전자알고리즘을 적용해 그 가능성을 확인하는 테스트 베드를 구축하였다.

**주요어** : 유전자 알고리즘, 착륙, 자율성, 시뮬레이션, 무인 비행체

## 1. 서론

본 논문을 통한 연구의 궁극적인 목적은 무인 비행체의 진정한 자율화를 이루는 데 있다. 이 목적의 시작 단계로써 본 논문에서는 자율화 달성의 방법론으로 유전자 알고리즘을 선택하였고, 이것을 적용해 무인 착륙체의 자율화 가능성을 확인해 보는 것이다. 어떤 시스템의 성공 가능성을 타진해 보는 데에는 모의(Simulation) 시스템에

\* 이 연구는 화랑대연구소의 2018년 연구활동비 지원으로 수행되었습니다.

Received: 25 January 2019, Revised: 24 March 2019,  
Accepted: 29 March 2019

<sup>†</sup> Corresponding Author: Changhee Han  
E-mail: chhan@kma.ac.kr  
Computer Science Dept., Korea Military Academy, Seoul,  
Korea

적용해 보는 것이 그 출발점일 수 있다. 또한, 무인 비행체의 행동들, 즉 이륙, 항해, 착륙 중에서 착륙의 행동에 적용해 보는 것으로 하였다. 따라서 본 논문에서는 컴퓨터 시뮬레이션을 통한 모의 무인 착륙체의 자율적 착륙에 유전자 알고리즘을 적용해, 자율 착륙의 성공 가능성을 검증해 보는 것이다.

모의 자율 착륙체 제작을 위한 시작 단계로서 컴퓨터 시뮬레이션이 가능한 유인 착륙체의 제작을 거친다. 이 과정에서 제작되는 모듈 대부분은 자율 무인 모의 착륙체 제작에 그대로 사용될 뿐만 아니라, 컴퓨터 모의 착륙체를 인간 사용자가 조정해 봄으로써 착륙체를 착륙시키는 것의 난이도를 체험해 보는 부수적 효과도 얻을 수 있다.

착륙은 비행체에서 매우 중요한 고전적인 과업 중의 하나이며, 최근 들어서는 드론 등 UAV(Unmanned Aerial Vehicle)(Thiang et al., 2016; Bartak et al., 2014; Bart et al., 2014; Jin et al., 2016)의 개발 및 적용과 활용 범위가 확대됨에 따라 그 중요성이 계속하여 주목받고 있다. 특히, 이들의 적용과 활용에서 그 범위의 확대를 충족시키기 위한 필수 요소로써 무인화를 들 수 있다. 무인 드론 등 UAV는 목표를 찾아서 어느 정도의 구간까지의 항해 이후에는, 착륙하려는 지점에 정확하게 스스로 안착하는 것은 무인 비행체의 본질적 목표라 할 수 있는 무인화에 있어서 매우 중요한 요소 중 하나인 것이다.

지금까지의 드론 착륙에 관한 많은 연구(Thiang et al., 2016; Bartak et al., 2014; Bart et al., 2014; Jin et al., 2016)에서 드론을 착륙시키는 대표적인 방법은 드론을 착륙지점의 상공으로 향해시키고, 착륙지점 직상방의 위치를 유지하면서 고도를 감소시키는 것이다. 그러나 이들 연구에서 하는 착륙이 진정한 의미의 자율화가 되기에는 많은 제한점이 존재한다. 착륙지점 위에 정확히 위치하면서 고도를 낮추기 위해서는 착륙지점에서 수평적으로 오차 거리가 얼마인지, 수직적으로 얼마의 거리가 남았는지, 방향이 얼마나 틀어져 있는지를 알아야만 한다. 이를 위해 그들은 각종 센서(울트라사운드, 자이로스코프, 카메라 등)를 사용하며 이 센서를 사용하여 읽어 들인 값을 그대로 거리 오차 수정에 사용하게 된다. 이들 세 가지의 오차 수정에서의 센서 리딩이 통상적으로 가장 제한되는 것이 수평 거리 조정이다. 수평 거리 오차 수정은 Figure 1 처럼 카메라에 잡힌 화면상의 픽셀 수를 삼각함수 비례식 등에 적용하여 반복적으로 수정해 나가게 된다. 이러한 기존의 방식이 자율화를 이루기 위해서는 이러한 측량 센서의 사용이 더욱 최소화가 되어야 한다.

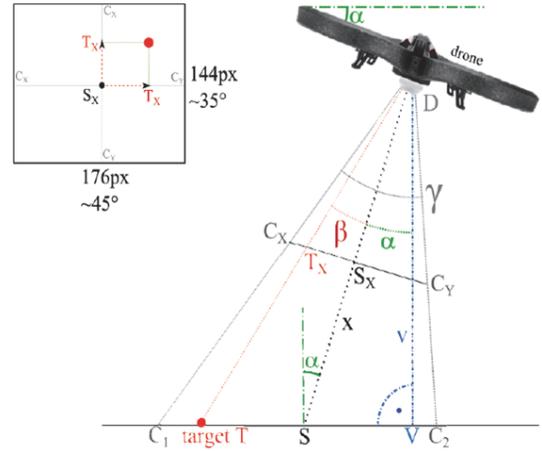


Fig. 1. Classical landing mechanism of drone with error modification for horizontal distance(Bart et al., 2014)

본 논문에서는 이러한 기존의 연구와는 다르게 센서에 의 의존을 전혀 하지 않는 방안을 모색하는 것이라고 할 수 있다. 물론 본 연구의 착륙체는 실세계가 아니고 컴퓨터의 가상 시뮬레이션 세계이므로 완벽한 비교는 불가능할 것이고, 실세계에서 운용될 비행체는 기상 등 실제 상황의 여러 변수를 반영하는 복잡성을 내포하고는 있지만, 센서 의존성을 최소화할 수 있는 방안 모색은 지향해야 한다는 점에서 본 연구의 차별성이 존재하고 이러한 점에서 본 연구의 의미가 있다고 할 수 있다.

착륙에 있어서 역사적 사건 중의 하나가 달착륙(Ebner et al., 2013)이라고 할 수 있다. 우리의 궁극적인 목표는 어떠한 비행체이던 그 착륙이 안전하면서도 자율적으로 수행되도록 하는 것이다. 무인 착륙의 롤모델이 되는 달 착륙을 모의한 착륙체를 통해 착륙을 실험하는 것은 매우 의미 있다고 할 수 있다. 또한, 컴퓨터 시뮬레이션상의 2차원 평면의 가로와 세로의 위치 좌표에 기초해 실험이 진행되었다. 실세계의 드론 등의 실험에서는 바람을 포함한 기상 관련 변수 등이 추가로 고려되겠지만, 본 논문의 목적이 유전자 알고리즘이 자율화를 달성할 수 있는지의 가능성을 확인하는 것이라는 관점에서 위치 좌표만의 고려는 충분하다고 본다.

2장에서는 착륙체 제작의 주요 과정을 설명한다. 3장에서는 유전자 알고리즘을 적용하여 인간의 조작을 대신해 착륙체가 스스로 착륙하도록 한다. 4장에서는 무인 자율 착륙의 실험 결과를 제시하며, 결론은 5장에서 맺도록 한다.

## 2. 착륙체 제작

드론 등 무인 비행체의 착륙에 필수적 요소인 수평 거리, 수직 거리, 수평 거리 중에서, 실세계 3차원 세상에서의 드론 비행을 컴퓨터 화면의 2차원 세계에서 착륙체로 모의함에 따라, 수평 거리 중에서 앞/뒤거리 오차 계산은 자연스럽게 빼도록 한다.

이 장에서는 착륙체와 그 시뮬레이션 제작의 주요 과정이 서술되고, 이 과정이 완료되면 인간 사용자가 착륙체의 착륙을 직접 조작해 볼 수 있게 된다. 유전자 알고리즘이 적용된 무인 착륙체를 완성하는 경로상의 중간 과정인 유인 착륙체가 제작되는 것으로, 피실험자가 조종을 경험함으로써 착륙체의 착륙이 쉽지 않음을 경험하는 효과도 얻게 된다.

### 2.1 구성요소

다음 Figure 2는 모의 착륙체의 실험환경을 나타내고 있다.

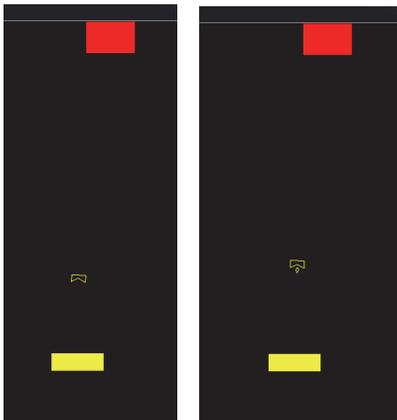


Fig. 2. Falling body experimental environment (Base, Falling body, Landing pad).

A thrust action performed in the image on the right

착륙체 모의실험 환경은 착륙체 기지, 착륙지점 패드, 착륙체로 구성되며 착륙체는 다시 착륙 본체와 불꽃으로 제작하였다. 착륙 본체의 불꽃은 평시에는 보이지 않고, 분사를 담당하는 해당키(스페이스바)를 눌렀을 때 그 분사를 표시하는 역할을 담당한다. 당연히 무인착륙체에서는 분사가 필요한 상황에서 자율적으로 실행된다.

### 2.2 액션

비행체 착륙의 오차 보정을 위해 필요한 3요소인 수직

거리, 수평 거리, 회전 거리를 보정하는 것으로, 본 착륙체의 액션으로는 ‘좌로 회전’, ‘우로 회전’, ‘분사’의 단지 3가지만을 생성하면 되는 것이다. 수직 거리는 통상 무인 비행체가 자유낙하를 하는 것이므로 본 논문에서도 자유 낙하가 되도록 하였다. 회전 거리의 보정은 ‘좌로 회전’과 ‘우로 회전’의 액션이 담당하게 되며, 키보드의 좌, 우 화살표에 그 기능을 부여하였다. 수평 거리는 좌/우회전과 분사의 조합으로 가능하게 하였다.

```

if (f_space==true) {
    double ShipAcceleration =(THRUST_PER_SECOND
    * TimeElapsed) / Mass;
    Velocity.x -= ShipAcceleration * Math.sin(Rotation);
    Velocity.y -= ShipAcceleration * Math.cos(Rotation);
    JetOn = true;
}
if(f_left==false && f_right==true) {
    Rotation -= ROTATION_PER_SECOND * TimeElapsed;
    if (Rotation < -PI){Rotation += 2*PI;}
}
if(f_left==true && f_right==false) {
    Rotation += ROTATION_PER_SECOND * TimeElapsed;
    if (Rotation > 2*PI){Rotation -= 2*PI;}
}
    
```

Fig. 3. Major module for left-rotate, right-rotate, and thrust

이러한 액션에 의해서 객체가 자유롭게 움직이도록 하려면 변환(Transformation) 모듈이 작성되어야 한다(Han et al., 2017).

회전, 스케일링, 이동의 변환은 각각 다음과 같다.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x*\cos\theta-y*\sin\theta & x*\sin\theta+y*\cos\theta & 1 \end{bmatrix},$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} * \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x*k & y*k & 1 \end{bmatrix},$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix} = \begin{bmatrix} x+a & y+b & 1 \end{bmatrix}$$

변환을 위처럼 행렬식으로 표현하면 세 가지 요소를 한 번에 묶어서 집합적으로 처리 할 수 있어 연산의 효율성을 높일 수 있다. 이때 처리의 시퀀스상에서 이동 변환이 마지막에 놓이도록 한다.

착륙체가 기지에서 떨어져 착륙지점으로 내려오는 거리는 화면의 y축 길이로써 약 600픽셀이다. 지면에 도달하는 시간은 다음의 식으로 구해질 수 있다.

$$d = ut + at^2/2, \quad u: \text{초기 속도}, a: \text{가속도}$$

지구의 중력은  $9.8m/s^2$  이므로 600픽셀을 내려오는데 걸리는 시간은 11초로 상당히 긴 시간이다. 한편 만약 이것을 달(중력 1.63)에서의 실험으로 간주한다면 무려 27초 정도로 더 오랜 시간이 걸리게 된다. 착륙체가 낙하하는데 걸리는 시간을 현실감 있게 느끼도록 하기 위해 상수 값을 도입하였다. 이때 지구와 달의 중력 차이를 통한 시뮬레이션의 비교를 위한 계산이므로 1픽셀은 1m로 간주 하였다.

### 2.3 착륙 성공 검사

착륙체가 일정한 높이에 도달하면 착륙 패드에 성공적인 안착을 했는지를 검사하게 된다. 이때 착륙 패드의 중심으로부터 착륙체 중심의 거리가 일정한 범위 안에 있어야 하며, 속도와 회전 거리도 일정한 값 이내여야 성공적인 착륙으로 간주한다.

이와 같은 일련의 과정들이 끝나게 되고 그 착륙이 안착인지 잘못된 착륙인지가 판단된 이후, 리셋 버튼을 작동시키면 착륙 패드가 임의의 위치로 변경되며 낙하 조정을 다시 해 볼 수 있다.

## 3. 유전자 알고리즘 적용

이번 장에서는 유전자 알고리즘을 2장에서 제작된 유인 착륙체에 적용해 자율 무인 모의 착륙체를 완성하게 된다. 이를 위한 모듈들을 세부적으로 설명해 본다.

### 3.1 게놈 인코딩

착륙체의 일생 주기 동안의 삶의 행위를 결정짓는 유전적 형질을 정하는 것이 이번 절의 핵심 사항이다. 착륙체가 탄생할 때 소유하고 있는 모든 액션은 2장 착륙체의 제작 섹션에서 언급된 좌로회전, 우로회전, 분사와 자유낙하이다. 착륙체가 어떻게 행위를 하며 일생을 살아간냐의 또 다른 요소는 이 4가지 행동 각각을 얼마만큼의 기간 발현시키느냐고 할 수 있다. 따라서 행동(Action; A)과 기간(Duration; D)의 두 요소를 하나의 Gene으로 간주한다. 실험적으로 게놈의 길이는 20개의 Gene으로 한다.



Fig. 4. Genom data structure

### 3.2 후보자 그룹 선택

학습 훈련에 참여하는 인구 구성의 게놈의 양은 100으로 한다. 이 인구 구성은 매 세대를 거치면서 우성 인자 중심으로 새롭게 개선된다. 이때 그러한 우성 인자를 선택하는 방법으로 룰렛 휠 선택법을 사용한다. 각 게놈의 적응도의 크기가 선택을 받는 데 확률로써 사용되게 된다. 즉 적응도가 크면 클수록 선택받을 확률도 증가하게 된다. 이렇게 하여 다음 세대에 더욱 더 적응력이 우수한 염색체를 물려 줄 수 있게 된다.

### 3.3 교차 및 돌연변이

새로운 개체를 만드는 기본 연산으로 부모 염색체의 형질을 교차시킨다. 이러한 교차에도 여러 방식이 존재하는데 본 연구에서는 다중점 교차 연산을 사용한다.

교차의 기본 연산은 첫 번째 자녀는 첫 번째 부모 두 번째 자녀는 두 번째 부모의 형질을 상속받게 된다. 그러나 각 gene에서 지정된 변화율에 부합되면 각 자녀는 다른 쪽 부모의 형질을 상속받는 것으로 변화된다. 결과적으로 자녀는 양쪽 부모의 형질을 고르게 가져갈 개연성이 증대된다.

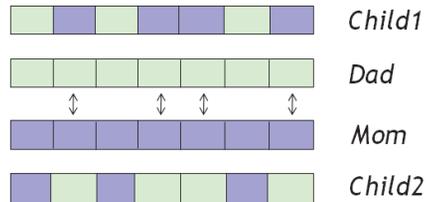


Fig. 5. Mutate mechanism

돌연변이 연산도 새로운 자녀 생성에 유의미한 영향을 주게 되는데, 본 논문에서는 이 변화율을 0.01로 하여 각 gene에 대해 이 변화율 이내 해당하면 현재의 액션과 기간을 임의의 값으로 변화시킨다.

### 3.4 게놈 디코딩 및 목적함수

지금까지의 과정 즉 gene 인코딩, 새로운 우성 염색체의 선택, 교차와 돌연변이 등 세대의 발전을 위한 일련의 노력이 마무리되었다면 세대를 구성하고 있는 염색체를 실제로 행위 하도록 할 시점에 와 있게 된다.

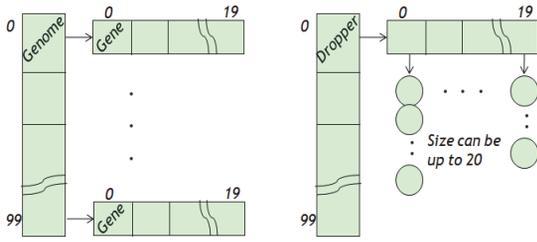


Fig. 6. Genom encoding and decoding

좌측은 게놈이 인코딩된 모습이다. 100개의 인구 각각은 20개의 유전형질을 소유하게 된다. 우측은 이것이 낙하 착륙체로 디코딩된 모습이다. 액션의 크기는 최대 20으로 각 gene의 액션의 크기는 자신이 소유한 기간의 크기만큼 생성된다.

예를 들어 첫 번째 낙하체는 좌로회전을 3회로 시작하고, 두 번째 낙하체는 우로회전을 7회 시작하는 것처럼 다양하게 자신들의 고유 행위를 하게 된다.

각 염색체가 달성해야 할 목적 함수는 다음과 같다.

$$F(d, s, r) = \{ (d < 20) \ \&\& \ (s < 0.5) \ \&\& \ (r < \pi/16) \}$$

여기서 d는 착륙 패드와의 거리, s는 스피드, r은 회전 각을 의미한다.

## 4. 착륙 실험 및 결과

### 4.1 기본 액션 결과

기본 액션인 좌로회전, 우로회전, 분사의 결과 이미지이다.

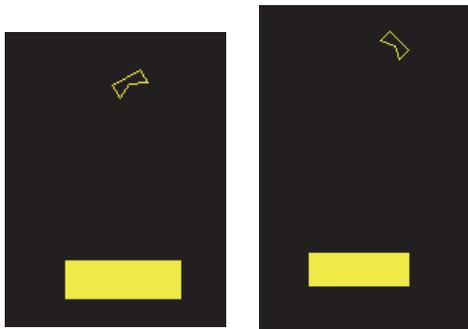


Fig. 7. Actions of left-rotate and right-rotate.  
The thrust can be checked in the Fig. 2

좌, 우로 이동은 기본액션인 회전과 분사가 조합하여 함께 작용하는 경우로서, 회전한 해당 방향으로 착륙체가

이동한다.

### 4.2 유인 착륙 결과

모의 무인 착륙체의 제작을 직접 실시하지 않고 유인 착륙체를 먼저 제작하고 실험을 진행한 이유로는 무인 착륙체의 제작과 유인 착륙체의 제작에는 공통된 모듈이 대부분이며, 시뮬레이션 제작자를 포함해 인간 실험자가 착륙 메카니즘을 체험토록 하는 것은 착륙과정의 이해에 필수적이라고 할 수 있다.

랜딩이 성공했을 때와 실패했을 때에 관한 결과 이미지이다. 실패의 상황은 2.3항에서 설명한 3가지 tolerance의 값 중에서 하나라도 허용치를 벗어나는 경우이다.

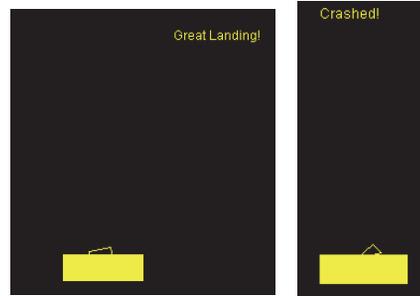


Fig. 8. Success landing on the left, and failure one on the right

Table 1. Result of manned landing

th	1	2	3	4	5	6	7	8	9	10
sec	51	46	33	53	35	29	49	38	38	67
o/x	o	x	o	o	x	x	x	o	x	o
th	11	12	13	14	15	16	17	18	19	20
sec	38	44	46	45	56	50	61	34	45	30
o/x	x	x	x	x	x	o	o	o	o	x

사용자가 얼마나 훈련을 통해 학습되었는가에 따라 달라지겠지만, 본 실험은 저자가 한 결과로써 약 45%의 성공률을 보인다. 성공한 경우의 평균 착륙 시간은 48초 정도로 다소 긴 시간이 걸렸다. 이러한 과정을 통해 사용자가 비행체의 성공적 착륙을 위한 조종이 결코 쉬운 것이 아니라는 사실을 경험하게 하는 부가적인 결과를 얻을 수 있었다.

### 4.3 무인 착륙 결과

10회의 실험을 진행했다. 한 회의 실험은 400세대를 기준으로 하여 훈련을 시키는 것으로 하였다. 이 한 에포크 내에 성공하지 못하면 시스템이 스스로 리셋이 되며, 임의의 랜딩 패드가 선정되고 새로운 에포크 (400 세대)

로 진입하여 주어진 새로운 세상에 대해 학습 훈련에 다시 돌입하는 것으로 하였다.

**Table 2.** Result of unmanned landing

th	1	2	3	4	5	6	7	8	9	10
epoch	101	272	111	-	193	242	-	94	-	135
o/x	o	o	o	x	o	o	x	o	x	o

4, 7, 9회를 제외한 7 번의 경우 한 에포크 내에서 학습 훈련이 성공되었다. 학습에 필요한 세대의 수는 평균 164 세대였다. 하나의 에포크에서 성공하지 못한 경우에도 바로 다음 에포크 내에서는 성공하였다. 성공하지 못한 경우를 보면 착륙 패드의 위치가 스크린의 좌측이나 우측 모서리 근처에 생기는 경우였다. 즉 착륙 패드의 위치가 낮아 가지에서 수평적으로 먼 위치의 경우이다. 이러면 학습에 어려움이 있는 것은 사실 자연스러운 현상인 것이다. 목표지점이 수평적으로 먼 경우는 그 방향으로 UAV가 좀 더 항해하여 이동한 후 착륙 메카니즘이 발생하는 것이 더욱 더 타당한 경우를 의미하기 때문이다. 이러한 점까지를 고려한다면 이번 연구의 학습 훈련이 매우 성공적이며 건전한(robustness) 결과를 가져왔다고 할 수 있다.

### 5. 결론 및 향후 연구

본 논문에서는 무인 비행체의 자율 착륙 연구를 컴퓨터 시뮬레이션 할 수 있는 모의 착륙체를 제작하였다. 컴퓨터 모의 착륙체를 인간 사용자가 조정해 봄으로써 자율 무인 착륙체 제작의 난이도를 체험해 보는 부수적 효과도 얻을 수 있었다.

계를 단지 4가지 액션 (좌회전, 우회전, 분출, 자율 낙하)으로 인코딩하여 무인 착륙체에 적용하고, 유전자 알고리즘의 주요 연산을 접목하여 실험을 성공적으로 진행하였다. 본 논문의 가장 큰 성과는 센서에 대한 의존도

를 최소화할 수 있는 무인 비행체의 모의 UAV를 제작하였고, 유전자알고리즘을 적용해 그 가능성을 확인하는 테스트 베드를 구축하였다는 것이다. 따라서 실세계의 UAV의 자율화에 이정표를 제시함과 동시에 진정한 Unmannedness를 달성하는데 이바지하였다 할 수 있다.

### References

Thiang, It Nun, Dr.LuMaw, and Hla Myo Tun (2016) "Vision-Based Object Tracking Algorithm With AR. Drone", *Intl, Journal of Scientific & Technology Research*, Vol. 5, Iss. 6.

Bartak, Roman, Andrej Hraško, and David Obdržalek (2014) "A Controller for Autonomous Landing of AR.Drone", *26th Chinese Control and Decision Conference (CCDC)*.

Bart, Roman, Andrej Hraško, and David Obdržalek (2014) "On Autonomous Landing of AR.Drone: Hands-on Experience", *Proc. of the 27th Intl' Florida Artificial Intelligence Research Society Conference*.

Jin, Shaogang, Jiyang Zhang, Lincheng Shen, and Tengxiang Li (2016) "On-board Vision Autonomous Landing Techniques for Quadrotor: A Survey", *Proceedings of the 35th Chinese Control Conference*.

Ebner, Marc, John Levine, Simon M. Lucas, and Tom Schaul, Tommy Thompson, and Julian Togelius (2013) "Towards a Video Game Description Language", *Dagstuhl Publishing, Wadern*, pp. 85-100.

Han, Changhee, K.M. Kim, D.H. Yoo, and Y.M. Eum (2017), "Framework integration specification necessary for simulation object rendering", *Fall conference of Korean Entertainment Industry Association*. (한창희, 김경민, 유대훈, 엄영문 (2017), "시뮬레이션 객체 렌더링 필수 프레임워크 통합 명세", *한국엔터테인먼트산업학회 2017 추계학술대회*)



**한 창 희** (chhan@kma.ac.kr)

1990 육군사관학교 이학사  
 1994 미국 시라큐스대학교(Syracuse Univ.) 전산학 석사  
 2004 미국 남가주대학교(Univ. of Southern California) 전산학 박사  
 2009 미국 Texas A&M University 방문연구원  
 2005~ 현재 육군사관학교 정교수

관심분야 : 모델링&시뮬레이션, 지능형시스템, 기계학습