

# 예제에 기반한 날개짓 비행의 제어방법\*

원정담<sup>0</sup>, 박종호, 김관유, 이제희  
서울대학교

(jungdam, jh.park, kwanyu, jehee)@mrl.snu.ac.kr

## How to Train Your Dragon: Example-Guided Control of Flapping Flight

Jungdam Won<sup>0</sup>, Jongho Park, Kwanyu Kim, Jehee Lee  
Seoul National University

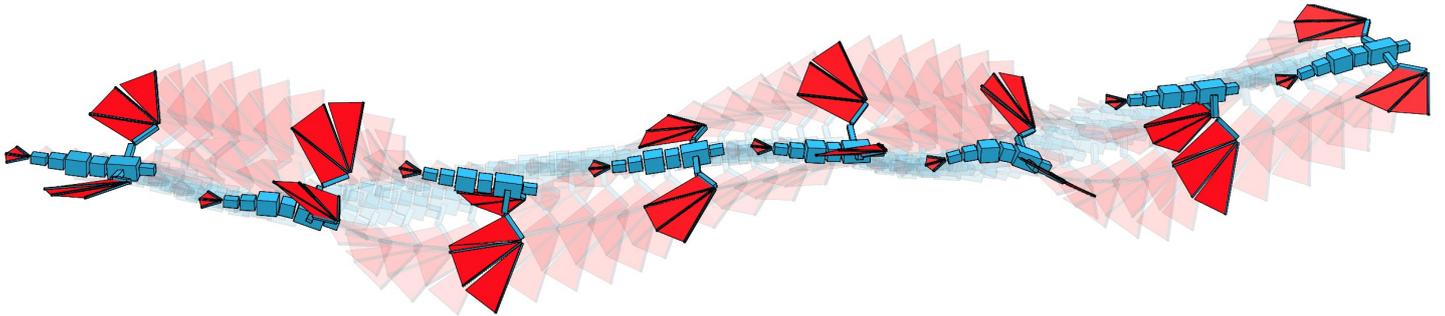


그림 1. 키 프레임과 날개짓 비행 동작

### 요약

물리 시뮬레이션 상에서 가상 캐릭터의 제어를 설계하는 것은 물리 기반 애니메이션을 제작하는 방법 중 하나로 컴퓨터 그래픽스 분야에서 많은 관심을 받아온 문제이다. 본 논문에서는 물리 시뮬레이션 환경에서 복잡한 날개짓 비행을 하는 가상 캐릭터의 제어를 학습시키는 새로운 강화학습 알고리즘을 제안한다. 제안 알고리즘은 기존의 알고리즘들에 비해 수렴 속도가 빠르며 일부 환경에서는 최종 성능도 향상되었다. 또한 사용자가 키 프레임을 제공함으로써 기반이 되는 동작을 직접 지정해줄 수 있고, 대화형 인터페이스를 통해 실시간으로 날개짓 스타일을 전환할 수도 있다. 실험에서는 물리 시뮬레이션 환경에서 목표 지점과 장애물이 주어졌을 때 장애물과의 충돌을 피하며 목표 지점으로 날아가는 동작을 학습하는 문제를 통해 학습 제안한 알고리즘의 성능을 검증하였다.

### 1. 서론

물리 시뮬레이션 되는 가상 캐릭터를 위한 제어를 설계하는 것은 물리 기반 애니메이션을 제작하는 방법 중 하나로 컴퓨터 그래픽스 분야에서 많은 관심을 받아온 문제이다. 제어기는 가상 캐릭터의 모델과 애니메이션

상의 환경에 따라 그에 적합한 종류의 운동 기술을 갖추어야 하는데, 특히 공중에서의 날개짓 비행 동작을 학습하는 것은 캐릭터의 근육 작용과 중력, 공기 저항의 상호작용을 고려해야하는 복잡한 문제이다. 또한 캐릭터의 모델이 높은 자유도를 갖는다면 이는 더욱 어려운 문제이다[1]. 본 논문에서는 물리 시뮬레이션 환경에서 복잡한 날개짓 비행을 하는 가상 캐릭터의 제어를 학습시키는 새로운 강화학습 알고리즘을 제안한다. 제안 알고리즘의 특징 및 장점은 다음과 같다:

1. 기존의 강화학습 알고리즘에 비해 수렴 속도가 빠르며 일부 환경에서는 더 높은 최종 성능을 보인다.
2. 사용자는 키 프레임을 제공함으로써 학습하고자 하는 날개짓의 기본적인 동작을 직접적으로 지정해줄 수 있다.
3. 대화형 인터페이스를 통해 날개짓 스타일을 실시간으로 변경할 수 있다.

우리는 물리 시뮬레이션 환경에서 목표 지점과 장애물이 주어졌을 때 장애물과의 충돌을 피하며 목표 지점으로 날아가는 동작을 학습하는 문제를 통해 제안한 알고리즘의 성능을 검증하였다.

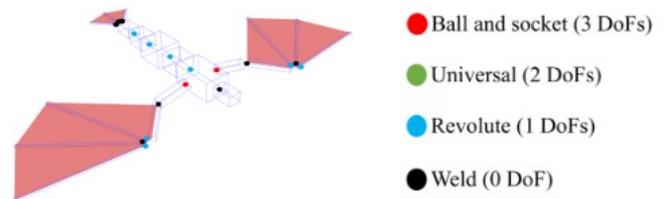


그림 2. 날개짓 비행 캐릭터의 구조

\* 구두 발표논문

\* 본 논문은 요약논문 (Extended Abstract) 으로서, 본 논문의 원본 논문은 SIGGRAPH Asia. 2017 에 제출 되었음.

\* 본 연구는미래창조과학부 및 정보통신기술진흥센터의 SW컴퓨팅산업원천기술개발사업(SW스타랩)의 연구결과로 수행되었음 (IITP-2017-0536-20170040).

## 2. 날갯짓 비행 제어기 학습 문제

### 2.1 제어기 학습 구성 요소 정의

캐릭터는 표현하고자 하는 가상의 비행 생물체를 단순화한 형태로 골격과 관절 그리고 막으로 구성된다 (그림 2 참조). 골격에는 부분마다 무게가 있어 중력이 작용하며 관절은 돌림 힘을 발생시킬 수 있는 가상의 모터이다. 그리고 막에는 속도에 따라 유체에 의한 항력과 양력이 작용한다. 우리는 [1]에서 제안한 방법과 비슷하게 간단한 유체역학식을 이용하여 캐릭터의 물리 시뮬레이션 환경을 구축하였다. 추가적으로 캐릭터에게는 무작위로 생성된 장애물과 사용자가 사전에 지정한 목표 지점이 주어지게 된다 (그림 3 참조).

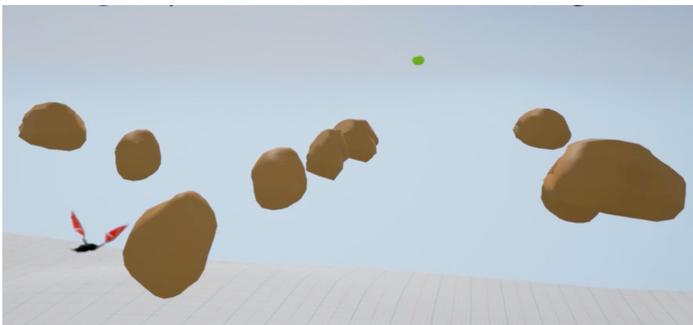


그림 3. 날갯짓 비행 환경. 캐릭터와 목표지점 (초록색 구) 사이에 총 8개의 장애물이 놓여 있는 모습.

캐릭터의 목적은 주어진 환경에서 장애물과의 충돌을 피하면서 지정된 목표 지점에 최대한 가깝게 이동하는 것이다. 이 과정에서 캐릭터는 균형을 잡으며 원하는 방향으로의 속력을 발생시키는 날갯짓 비행 동작을 학습하게 된다.

### 2.2 학습 방법

제어기의 입력은 캐릭터의 물리적 상태와 주변 환경에 대한 인식이고, 출력은 캐릭터가 다음으로 취해야 하는 자세이다. 본 논문에서 제어기는 심층 신경망(Deep Neural Network)으로 구성하였으며, 이 신경망의 학습은 본 논문에서 새롭게 제안된 진화적 심층 Q 학습(Evolutionary Deep Q Learning)을 통해 진행된다.

제안된 학습 알고리즘은 크게 두 부분으로 나뉘는데, 첫 번째는 학습에 필요한 정보를 얻어 재생 메모리[2]라 불리는 저장 공간에 축적하는 것이다. 이때 기존의 강화 학습 알고리즘은 현재 자신이 최선이라고 알고있는 동작에 임의의 노이즈를 추가함으로써 새로운 상황에 대한 정보를 얻는데, 우리는 여기에 CMA-ES[3] 기법을 추가하여 단순한 노이즈로는 얻을 수 없는 새로운 정보를 추가적으로 얻도록 하였다. 두 번째는 재생 메모리에 저장된 정보를 이용하여 제어기 모델을 구성하는 파라미터를 수정하는 단계이다. 업데이트 알고리즘은 연속적 배우-비평가 학습 오토마톤(Continuous Actor-Critic Learning Automaton) 방법을 이용하였다. 전체 학습 과정은 이 두 단계를 번갈아가며 반복적으로 수행함으로써 진행된다.

## 3. 결과 및 결론

우리는 DART[4]를 기반으로 캐릭터와 물리 시뮬레이션 환경을 구현하였고 TensorFlow를 이용하여 심층 신경망과 강화학습 알고리즘을 구현하였다. 물리시뮬레이션의 높은 계산부하로 학습에 GPU를 사용하지 않았고, 일반적인 데스크탑(Intel i7 Core)기준으로 대략 20~30 시간 정도가 학습하는데 소요되었다. 추가적으로 OpenAI Gym에서 제공하는 환경을 이용하여 기존의 강화학습 알고리즘과 본 논문에서 제안하는 진화적 심층 Q 학습 알고리즘의 성능을 비교하였다 (그림 4 참조). 제안 알고리즘은 기존의 강화학습 알고리즘들에 비해 학습속도가 빠르며 일부 환경에 있어서는 최종 성능도 향상되었음을 확인할 수 있다.

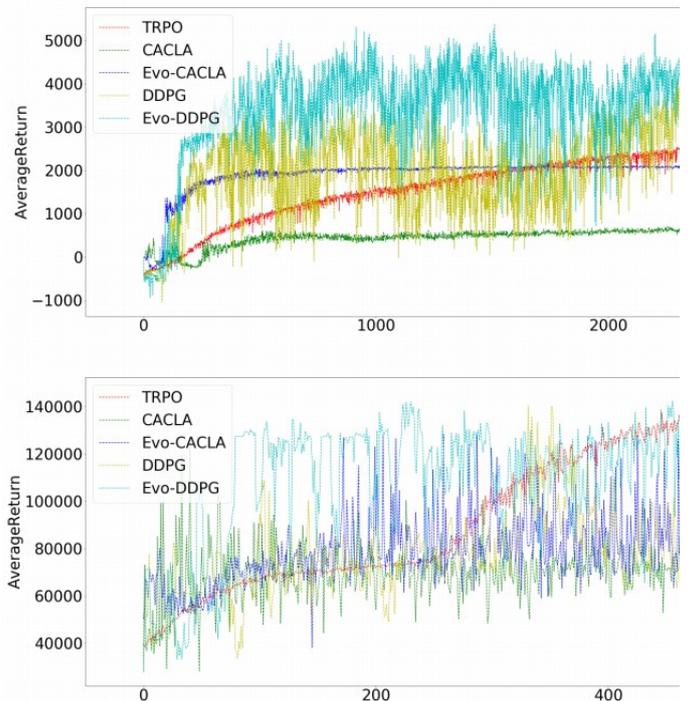


그림 4. OpenAI Gym 환경에서의 알고리즘 성능 비교. 가로축은 학습의 진행정도, 세로축은 성능을 뜻한다. Evo라고 표시되어 있는 것이 본 문에서 제안한 진화적 기법을 이용한 학습 방법이다.

## 참고문헌

- [1] Ju E., Won J., Lee J., Choi B., Noh J. and Choi M. G., A Data-driven control of flapping flight, *ACMTrans. Graph.*, 32, 151:1–151:12.
- [2] Hansen, N. and Ostermeier, A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*. 312 – 317.
- [3] Mnih V., et al. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).
- [4] DART, 2012. Dart: Dynamic animation and robotics toolkit. <https://dartsim.github.io/>.