

드론 시뮬레이션 기술

Drone Simulation Technologies

이수전 (S.J. Lee, soojeonlee@etri.re.kr)

무인이동체연구실 책임연구원

양정기 (J.G. Yang, jgyang@etri.re.kr)

무인이동체연구실 연구원

이병선 (B.S. Lee, lbs@etri.re.kr)

무인이동체연구실 책임연구원/실장

ABSTRACT

The use of machine learning technologies such as deep and reinforcement learning has proliferated in various domains with the advancement of deep neural network studies. To make the learning successful, both big data acquisition and fast processing are required. However, for some physical world applications such as autonomous drone flight, it is difficult to achieve efficient learning because learning with a premature A.I. is dangerous, cost-ineffective, and time-consuming. To solve these problems, simulation-based approaches can be considered. In this study, we analyze recent trends in drone simulation technologies and compare their features. Subsequently, we introduce Octopus, which is a highly precise and scalable drone simulator being developed by ETRI.

KEYWORDS 드론 시뮬레이션, 다중 드론, 기계학습, 머신러닝, A.I.

1. 서론

지난 몇 년간 드론 산업계는 급격히 성장 중이며 다양한 분야로부터 큰 관심을 받고 있다. 수많은 서비스 시나리오, 애플리케이션, 운영 개념 등이 활발히 논의되고 구현되고 있으며, 또한 그 다양성과 다이내믹함으로 인해 한 해 수백 개의 회사들이 시장으로 진입하기도 또는 사라지기도 한다. 다만 최근 드론 산업계로부터 몇 가지 중요한 경향이 관

찰된다.

첫째, 드론 하드웨어보다 드론 소프트웨어에 대한 투자가 강화되고 있다[1]. 이는 서비스와 데이터가 보다 중요해짐을 의미한다. 일례로, 드론 하드웨어 벤더의 경우 사용자에게 독자적인 토탈 솔루션을 제공하지 못하므로, 서로 다른 기술을 가진 회사 간 전략적 파트너십이 활발히 이루어지고 있다.

둘째, 머신러닝 기술 발전에 따라 사용자들은 기

* DOI: <https://doi.org/10.22648/ETRI.2020.J.350408>

* 본 연구는 한국전자통신연구원 연구운영비지원사업의 일환으로 수행되었음[20ZS1200, 인간 중심의 자율지능시스템 원천기술 연구].



본 저작물은 공공누리 제4유형

출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

©2020 한국전자통신연구원

존 레거시 시스템이 전혀 제공하지 못하는 수준의 똑똑한 서비스를 원하게 되었다. 드론의 응용 분야가 매우 다양해짐에 따라 수동 제어를 통한 임무 수행이 비효율적이거나 뚜렷한 한계를 보일 수 있기 때문이다. 예를 들어, 광활한 면적의 대단위 태양광 발전소에서 불량 패널을 찾아내는 임무는 A.I.에 의존하는 것이 훨씬 효율적일 수 있다.

셋째, 대규모 다중 드론을 활용한 서비스들이 등장하게 되었다. NASA의 경우 소형 UAS(Unmanned Aircraft System) 교통관리시스템인 UTM(UAS Traffic Management) 운영 컨셉을 정의하고 개발 중이다[2]. UTM의 주 목표는 150미터 이하의 저고도 공역에서 대규모 소형 UAS의 안전한 운영을 보장하는 시스템을 개발하는 것이다. 산업계에서도 Amazon, Boeing, GE, Google 등이 대규모 드론 운영을 위한 프라이빗 UTM을 개발 중이다[3,4].

상기와 같은 사용자 요구를 해결하기 위하여 광범위하고 다양한 연구가 필요하나, 실제 드론을 활용한 기술 개발에는 명확한 한계가 있다. 첫째, 드론은 위험하다. 따라서 불완전한 시험용 비행 A.I.를 탑재한 드론을 비행시키는 것은 안전사고 유발 등 위험요소가 매우 크다. 둘째, 비용 부담이 높다. 예를 들어, 대규모 고성능 드론의 군집 비행 기술 개발을 위해서는 막대한 금액의 드론 구매가 선행되어야 하며, 드론 파손 시의 금전적 손실 또한 무시할만한 수준이 아니다. 셋째, 개발 속도가 느리다. 특히, 기계학습의 기반이 되는 빅데이터를 수집하는 데 매우 긴 시간이 소요된다.

위와 같은 한계를 극복하기 위하여 드론 시뮬레이터를 활용한 연구 개발이 대안으로 고려되고 있다. 본 고에서는 국내외 드론 시뮬레이터 기술 동향을 살펴보고, 그들의 특징 및 장단점을 비교한다. 또한 ETRI에서 개발 중인 드론 시뮬레이터 *Octopus*를 소개함으로써 향후 드론 관련 연구개발

을 가속화할 수 있는 플랫폼으로서의 *Octopus* 역할을 제안한다.

II. 드론 시뮬레이션 기술 동향

1. 이용 분야별 기술 동향

드론 시뮬레이터는 응용 분야에 따라 필요로 하는 기능 및 성능 요구사항이 다르다. 이 절에서는 응용 분야별 드론 시뮬레이터 기술 동향에 대해 소개한다.

가. 사용자 훈련용

드론 시뮬레이터는 사용자 훈련 목적으로 활발히 개발되고 있다. 훈련용 드론 시뮬레이터는 시뮬레이션 서버에 실제 드론 조종기를 연결할 수 있도록 하여 사용자가 드론 조종을 연습할 수 있도록 돕는다. 이 응용 분야는 국내외 드론 기업들을 중심으로 마케팅 목적으로 개발되고 있다. 바이로봇[5] 등 많은 국내 드론 중소기업들은 자사의 드론 비행 훈련을 위해 드론 시뮬레이터를 자체 개발하여 사용자들에게 무료로 제공하고 있다. 세계 1위 드론 제작 기업인 DJI에서도 드론 시뮬레이터(DJI simulator)[6]를 제작하여 제공 중이다. DJI 시뮬레이터의 경우 고사양 게임 제작에 사용되는 Unreal engine을 기반으로 제작되어 시각적으로 사실적이다. 이뿐만 아니라 사용자 입장의 시각, 지상 효과, 오브젝트와의 충돌, 비행시간에 따른 배터리 소모 등 실제 드론 비행 환경과 유사하게 만들기 위해 고도화된 물리 모델을 적용하여 사용자가 자사의 드론에 더욱 익숙해지도록 만들고 있다.

사용자 훈련용 시뮬레이터는 단순한 일반인 비행 연습용 수준을 넘어 임무 수행 훈련용으로 발전하고 있다. 앞서 언급한 DJI 시뮬레이터의 경우 단순한 비행 연습 기능뿐만 아니라 드론을 이용한 전

력선(Powerline) 점검 훈련 기능도 제공한다. 시뮬레이션상 드론에 IR(Infra-Red) 카메라를 모사/장착하고 전력선 이상 상황(전력선 온도 분포 적용을 통해 판단) 유무에 따라 사용자가 드론의 IR 카메라를 통해 전력선 점검 기능을 훈련할 수 있도록 하고 있다. DRL sim3.0[7]은 드론 레이싱 선수 훈련용 시뮬레이터로, 실제 드론 레이싱 환경과 유사하게 가상공간(경기장 등)을 구축하였고 유체 역학, 프로펠러 동역학 등 시뮬레이터에 적용되는 동역학을 고도화하여 드론 레이싱 선수들이 손쉽게 연습할 수 있는 환경을 제공한다.

나. 기술 검증용

실제 드론을 활용한 실험 도중 드론 추락 등 위험 상황이 발생할 수 있다. 따라서 실제 비행 전 사전검증을 진행하고자 시뮬레이터에 대한 개발이 진행되고 있다. 특히 드론에 사용되는 FC(Flight Computer)를 검증하기 위한 연구가 활발히 진행 중이다. FC는 드론 비행에 필수적인 항법/비행제어 알고리즘, 조종기/GCS(Ground Control System)와의 통신 인터페이스 등을 제공하는 비행용 컴퓨터이다. 시뮬레이터를 통해 FC를 사전 검증하는 방법은 크게 SITL(Software-In-The-Loop)과 HITL(Hardware-In-The-Loop)로 나눌 수 있다. SITL은 FC를 소프트웨어적으로 모사하여 시뮬레이션상에 적용시키는 방법이다. 하드웨어 없이도 소프트웨어 결함을 확인할 수 있다는 장점이 있으나 시간 동기화, 하드웨어 버그 등의 문제는 발견하지 못한다는 단점이 있다. HITL은 FC를 서버에 직접 연결하여 시뮬레이션을 통해 실제 드론의 데이터를 획득하도록 하는 방법이다. 이 방식은 하드웨어와 시뮬레이터상의 인터페이스를 구현하기 힘들다는 단점이 있지만 FC 하드웨어 자체 결함을 확인할 수 있다는 장점이 있다. 현재 개발되는 대부분의

오픈소스 드론 시뮬레이터는 오픈소스 FC 펌웨어인 PX4를 HITL/SITL 기법을 통해 실험할 수 있도록 지원한다.

드론 검증용 시뮬레이터는 FC뿐만 아니라 다양한 연구용 알고리즘 동작을 확인하기 위해 발전하고 있다. 오픈소스 드론 시뮬레이터를 기준으로 알고리즘 성능 확인용 시뮬레이터들이 개발 중이며, 특히 ROS의 Gazebo[8] 기반 드론 시뮬레이터가 가장 활발히 사용되고 있다. Gazebo는 ROS에서 기본으로 제공하는 3차원 물리엔진으로, 이를 통해 로봇 및 드론 알고리즘의 성능을 검증할 수 있다. 또한 PX4를 개발한 Dronecode에서는 다양한 드론 모델과 정교한 센서 모델, ROS와 PX4 인터페이스를 구현해서 배포하고 있기 때문에 많은 드론 연구자들이 이를 사용하고 있다.

다. A.I. 학습용

최근 A.I.에 대한 연구가 활발히 일어나며 RL(Reinforcement Learning)을 드론에 적용하기 위한 시뮬레이션 기술 연구가 활발히 진행 중이다. RL은 시행착오를 반복하며 배우는 기법이기 때문에 대규모 데이터셋을 요구하는 기존 A.I. 기법들과 달리 실험할 수 있는 환경을 필요로 한다. 실패 시 추락으로 인해 큰 비용이 발생하는 드론 실험 특성상 실제 드론으로 RL을 실험하는 것은 무리가 있어 시뮬레이션을 통한 학습을 진행한다. 최근 드론 연구자들은 RL을 통해 카메라 기반 자율 비행 기법을 연구하고 있기 때문에 시뮬레이터는 시각적으로 실제와 유사한 환경을 제공해야 한다.

AirSim[9]은 고사양 게임 엔진인 Unreal engine을 기반으로 하여 시각적으로 현실적일 뿐만 아니라 카메라, 깊이 지각 카메라, Lidar 등 다양한 센서를 제공하는 오픈소스 드론 시뮬레이터이다. 이 시뮬레이터는 다른 오픈소스 드론 시뮬레이터에 비해

고도화된 센서 모델, ROS 인터페이스 등을 제공하고 있어 많은 연구자들이 AirSim을 활용하여 RL 기반 드론 자율 비행 기법을 연구하고 있다. 특히 AirSim에는 RL 연구자들을 위해 OpenAI Gym[10] 환경이 추가되었다. OpenAI Gym은 다양한 시뮬레이터에 대해 A.I. 성능을 벤치마킹하는 사이트로, 연구자들은 OpenAI Gym용으로 AirSim 기반 시뮬레이터를 제작하여 드론 자율 비행 성능을 벤치마킹하고 있다.

사우디아라비아의 KAUST(King Abdullah University of Science and Technology)에서는 시각적으로 현실적인 오픈소스 시뮬레이터인 UE4sim[11]을 개발하였다. 이 시뮬레이터는 AirSim과 마찬가지로 Unreal engine을 기반으로 하여 시각적으로 현실적이며, GUI를 제공하여 사용자 편의를 고려하였다. 그리고 사용자 맞춤형 시뮬레이션 환경 제작 시스템을 제공한다는 특징이 있다. 개발자는 이 시뮬레이터를 이용하여 드론의 자율 비행 및 DNN(Deep Neural Network) 기반 물체 추적 기술을 실험하였다. 이 밖에도 다양한 A.I. 학습용 드론 시뮬레이터들이 제작되고 있다.

라. CPS용

현재 드론 시뮬레이터를 CPS(Cyber Physical Systems)로 확장하려는 움직임이 일어나고 있다. CPS는 실제 공간에서의 물리적 요소들이 가상공간에 그대로 투영되어 상호 간 모니터링 및 제어가 가능한 시스템을 의미한다. MIT에서는 AlphaPilot이라는 자율 주행 드론 레이싱 챌린지에 사용하기 위한 드론 CPS 시스템인 FlightGoggles[12]를 개발하였다. FlightGoggles는 시각적으로 현실적인 시뮬레이션 환경을 제공하고 있다. 실제 드론의 움직임을 모션 캡처 시스템을 통해 획득하고 시뮬레이션 환경상의 같은 위치에 가상 드론을 투영한다. 가상

드론은 카메라를 장착하고 있어서 가상 환경으로부터 얻은 이미지를 기반으로 실제 드론을 제어하는 알고리즘을 동작시킬 수 있다. CPS를 구성할 경우 시뮬레이터만을 사용했을 때에 비해 두 가지 큰 이점이 있다. 첫째, 바람의 영향 등 시뮬레이션으로 완벽하게 구현하기 힘든 드론의 동역학적 요소를 표현할 수 있고 펌웨어, 센서 등 하드웨어적 요소를 모두 고려하여 드론의 움직임을 나타낼 수 있다. 둘째, 실제 공간에서는 충돌로 인한 추락으로 인해 RL을 통한 학습이 힘든 반면에, CPS를 구성할 경우 가상 공간상의 오브젝트와 충돌을 하더라도 추락하지 않기 때문에 실제 드론(및 이와 동기화된 가상 드론)을 통해서 RL을 진행할 수 있게 된다. FlightGoggles는 드론을 시뮬레이션상에 투영할 뿐만 아니라 VR(Virtual Reality) 장비를 활용하여 사용자가 실제 드론 레이싱 경기장에 있는 것처럼 느낄 수 있도록 만들었다.

마. 데이터셋 제작용

최근 컴퓨터 비전 및 A.I. 기술의 발전으로 벤치마킹용 또는 학습용 데이터셋 합성을 위한 시뮬레이션 기술 연구가 활발히 진행 중이다. 실제 드론 및 로봇을 통해 데이터셋을 제작하기 위해서는 데이터 획득용 센서뿐만 아니라 위치 참값 측정, 이미지상의 물체 라벨링 등 부수적인 작업에 많은 시간 및 비용이 소모되었다. 그러나 최근 Unreal engine, Unity 등 게임엔진의 발전으로 시뮬레이터 상에서 실제와 매우 유사한 이미지를 얻는 것이 가능해지면서 시뮬레이터상에서 데이터셋을 합성하고자 하는 움직임이 일어나고 있다. 시뮬레이터를 통해 데이터셋을 합성할 경우 방대한 양의 데이터를 저비용으로 신속히 획득할 수 있으며 드론의 위치, 이미지상의 물체 라벨링 등에 대한 참값을 쉽게 구할 수 있고, 움직이는 물체가 존재하는 환경

등 사용자가 정의한 환경으로 데이터셋을 구축할 수 있다는 이점이 존재한다.

MIT LIDS는 시뮬레이터 환경에서 합성한 UAV(Unmanned Aerial Vehicle)의 물체 인식용 데이터셋인 Blackbird[13]를 발표하였다. 이 데이터셋은 스테레오 카메라 이미지, IMU(Inertial Measurement Unit) 및 모터 스피드 센서 데이터, 모션 캡처 시스템으로 측정한 드론 위치 참값을 포함하고 있다. 가정집, 박물관 등 실제 드론 운용이 어려운 5가지 환경에서 17가지의 경로로 데이터를 획득하였다. 이 데이터셋은 구글의 FlightGoggles 시뮬레이터를 이용하여 생성되었기 때문에 시각적으로 현실적이며, 드론의 물리적 데이터도 다른 합성 데이터셋에 비해 현실적이라는 장점이 있다.

2. 오픈소스 드론 시뮬레이터

앞서 다양한 용도로 활용되는 드론 시뮬레이터들의 기술 동향에 대해 소개하였다. 현재 대부분의 드론 시뮬레이터들은 오픈소스를 기반으로 하여 제작되고 있으며, 이 절에서는 대표적인 오픈소스 시뮬레이터들에 대해 기술하고자 한다.

가. JMavSim

JMavSim[14]은 PX4의 통신 인터페이스인 MAVLink를 지원하여 SITL/HITL을 모두 지원한다. Java3D를 기반으로 그래픽을 구현하여 계산량이 매우 낮기 때문에 다중 드론 시뮬레이션이 가능하고 UDP 프로토콜을 통신에 사용하고 있으며, 코딩이 간단해 개발이 편하다는 장점이 있다. 하지만 다른 오픈소스 시뮬레이터에 비해 시각적으로 현실적이지 못한데다가 추가적인 센서 장착이나 시뮬레이션 환경 내 장애물 추가가 매우 힘들어 연구용으로 확장이 힘들다는 단점이 존재한다.

나. Gazebo 기반 시뮬레이터

Gazebo는 ROS의 대표적인 3차원 로봇 시뮬레이터이다. ROS는 Gazebo상에서 사용 가능한 다양한 로봇 기본 부품 및 센서들을 제공하고 URDF(Unified Robot Description Format)를 통해 로봇의 각 부품을 실제 조립하듯 연결하여 로봇을 구현하는 기능을 제공하고 있다. Gazebo 기반의 시뮬레이터를 통해 획득한 데이터를 가시화할 수 있는 툴인 RViz를 제공하고 있기 때문에 많은 연구자들의 관심을 받고 있다. Gazebo를 기반으로 제작된 드론 시뮬레이터는 대표적으로 Hecto_quadrotor, Dronecode 제공 시뮬레이터, RotorS가 있다.

1) Hecto_quadrotor

Hecto_quadrotor[15]는 ROS 초창기에 개발된 드론 시뮬레이터로서 SLAM(Simultaneous Localization And Mapping) 실험을 주 목적으로 한다. 기본적인 실내외 환경, Lidar 센서 모델, 쿼드콥터 제어 인터페이스를 제공하기 때문에 드론 Lidar 기반 SLAM 및 자율비행 기술의 검증이 가능하다. 하지만 쿼드콥터 드론 모델만을 지원하는데다가, 이 또한 센서 무게 등을 무시한 실존하지 않는 쿼드콥터이기 때문에 실제 드론의 움직임과는 차이가 있다. 그리고 SITL/HITL 등 FC를 검증하기 위한 기능들 또한 제공하고 있지 않다.

2) Dronecode 제공 시뮬레이터

PX4, MAVLink, QGroundControl 등 UAV 관련 오픈소스 프로젝트를 진행하는 비영리 단체인 Dronecode에서도 PX4 검증을 위한 Gazebo 기반의 시뮬레이터를 통해 다양한 형태의 드론 모델 및 센서들을 제공하고 있다. 기본적인 쿼드콥터 모델뿐만 아니라 고정익, VTOL(Vertical Take Off and Landing), Tailsitter VTOL, 차량 등 PX4가 지원하는

다양한 형태의 이동체에 대한 모델을 지원하고 있다. 또한 IMU 등 항법센서와 카메라, Lidar를 지원한다. URDF를 통해 모델링을 하기 때문에 드론의 물리적 성질도 실제와 유사하게 구현된다. PX4를 이용한 SITL/HITL을 모두 지원하고, ROS의 MAVROS 패키지를 이용해 쉽게 FC와 통신이 가능하다.

3) RotorS

RotorS[16]는 ETH ASL에서 발표한 드론 시뮬레이터이다. RotorS는 AscTec Firefly 헥사콥터를 고정밀 모델링하여 적용시켜 놓았다는 특징을 가지고 있다. 또한 상위 응용 분야 연구를 위해 센서 모델을 보다 정교하게 만들었고, 3차원 맵핑 및 SLAM에 사용되는 Octree를 사용할 수 있도록 구현해놓았다. 하지만 PX4 SITL/HITL 기능을 제공하지 않는다.

다. AirSim

AirSim은 Microsoft에서 개발 중인 Unreal engine 기반의 오픈소스 드론 시뮬레이터이다. SITL/HITL 등 드론 요소 기술 검증뿐만 아니라 A.I. 연구 목적으로 꾸준히 개선되고 있다. Unreal engine을 기반으로 하였기 때문에 시각적으로 사실적인 환경을 제공함을 특징으로 한다. 또한 태양 위치 설정뿐만 아니라 비, 낙엽 등 날씨 및 계절 설정을 할 수 있도록 하여 시각적 요소를 고도화하였다. 그리고 타 드론 오픈소스 시뮬레이터에 비해 다양하고 고도화된 센서 모델링을 사용한다. 드론 비행에 필수적인 IMU, 기압계, GPS 등 항법 센서들의 노이즈 모델링을 사실적으로 구현하였고, 카메라, Lidar 등의 센서도 노이즈 및 바이어스 설정을 가능하도록 한다. 특히 카메라 센서의 경우 일반 RGB

획득 기능뿐만 아니라 깊이 지각 이미지, 평면 방향 정보 이미지, 물체 분류 이미지, IR 이미지 등의 획득이 가능하며, 이미지의 노이즈, 블러, 조도 등 알고리즘 성능에 영향을 주는 카메라 파라미터들을 설정할 수 있도록 구현하였다. 그리고 드론뿐만 아니라 자동차 모델을 적용할 수 있고, 자동차 모델에 드론에 적용되는 센서들을 적용하여 데이터를 획득할 수도 있다.

사용자 편의를 위해 다양한 인터페이스를 제공한다. PX4를 SITL/HITL을 통해 적용할 수 있는 인터페이스뿐만 아니라 RPC(Remote Process Call) 기반의 API를 제공하여 C++, Python 환경에서 시뮬레이션 제어가 가능하다. 또한 IP 기반으로 별도의 컴퓨터에서 접속하여 동작할 수도 있다. ROS wrapper을 제공하기 때문에 ROS를 통해서도 시뮬레이터 조종 및 데이터 획득이 가능하다. 기본적으로 Windows에서 동작하지만, Linux상에서도 unstable 버전의 동작이 가능하고 Ubuntu상 구동용 docker를 따로 제공하고 있다.

하지만 AirSim에도 몇 가지 단점이 존재한다. 우선, 고성능 게임엔진을 사용하기 때문에 많은 계산량을 필요로 하고, 따라서 대규모 다중 드론의 운용이 힘들다. 일례로, 드론을 통해 이미지 및 Lidar 데이터를 획득하는 경우 렌더링의 FPS(Frame Per Second) 손실이 크지 않은 수준으로 실시간 운용할 수 있는 드론 대수는 고성능 컴퓨터상에서도 다섯 대 이하에 불과하다. 그리고 기본적인 뉴턴 오일러 방정식을 적용한 자체 물리 엔진을 사용하기 때문에 드론을 사용자 임의로 설정하는 것이 힘들고 드론 간 충돌, 타 물체와의 상호작용이 힘들다. 이와 같은 단점들을 해결하기 위해 ETRI에서는 AirSim을 기반으로 하는 *Octopus* 시뮬레이터를 개발 중이다.

표 1 오픈소스 시뮬레이터 특성 비교

	JMavSim	Gazebo	AirSim
시각적 현실도	낮음	보통	높음
계산부하량	낮음	보통	높음
드론 모델의 다양성	쿼드콥터	다양함	쿼드콥터, 차량
동역학적 모델 정확성	낮음	높음	보통
센서 모델 정확성	낮음	보통	높음
인터페이스	PX4	ROS, PX4	다양함
PX4 지원 여부	가능	가능	가능
추천 사용 분야	군집 운용 검증	알고리즘 검증	AI 학습용

라. 오픈소스 시뮬레이터 특성 비교

오픈소스 시뮬레이터들은 서로 다른 특성을 지니고 있어 각자 적합한 응용 분야가 다르다. 현실감 있는 가상공간 구성이나 정밀한 물리모델 적용이 필요 없이, GCS상에서 단순히 대규모 다중 드론 운용 적합성만을 테스트할 목적이라면 연산량이 가장 적은 JMavSim을 선택하는 게 바람직하다. 물체 운송 등 카메라 기반이 아닌 알고리즘을 검증하기 위해서는 Gazebo 기반 시뮬레이터를 사용하는 것이 적합하고, 카메라 기반 알고리즘 혹은 A.I. 학습용으로는 AirSim의 사용이 권장된다. 오픈소스 시뮬레이터들의 특성을 정리하면 표 1과 같다.

III. Octopus

실제 드론을 활용한 드론 연구 과정에 직면하게 되는 여러 현실적 제약을 해결하기 위하여 드론 시뮬레이터의 개발이 고려되고 있으며, ETRI는 마이크로소프트사의 오픈소스 프로젝트인 AirSim을 활

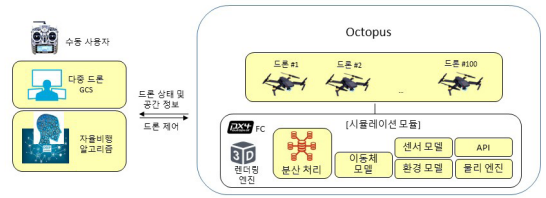


그림 1 Octopus 구조

용하여 2018년도부터 고정밀/고확장 드론 시뮬레이터인 Octopus를 개발해오고 있다. 그림 1은 Octopus의 구조를 보여주며, 본 장에서는 그 중 타 드론 시뮬레이터와 근본적인 차이를 지닌 물리엔진과 분산처리 기능에 대하여 설명한다.

1. 물리엔진

AirSim 전용 드론 물리엔진인 FastPhysicsEngine은 연산량이 적다는 장점에도 불구하고 실제 물리적 현상의 반영이 어렵다는 문제가 존재한다. 첫째, Unreal engine 내부의 다른 객체와 물리적 상호작용(예, 바람의 영향 반영 등)이 근본적으로 지원되지 않는다. 또한 드론의 물리적 특성 반영이 불가능하다. 예를 들어, 드론 우하단에 특정 센서를 장착할 경우 질량 증가와 무게 중심점 변경이 일어나게 되나, FastPhysicsEngine은 이러한 현상을 고려하지 못한다. 이와 같은 문제를 해결하기 위하여 Octopus는 Nvidia사의 물리엔진인 PhysX[17]를 활용하는 드론 물리엔진을 개발하였다.

그림 2는 PhysX기반 드론 물리엔진의 구조를 나타낸다. AirSim의 경우 드론 kinematics를 AirSim 라이브러리에서 모두 계산하며, Unreal engine에서는 렌더링만을 진행한다. 반면 Octopus는 제어 입력, 센서 정보들을 Unreal engine으로 전달하고, Unreal engine 내부에서 PhysX를 사용하여 드론 동역학적 요소를 계산하는 드론 클래스를 생성한다.

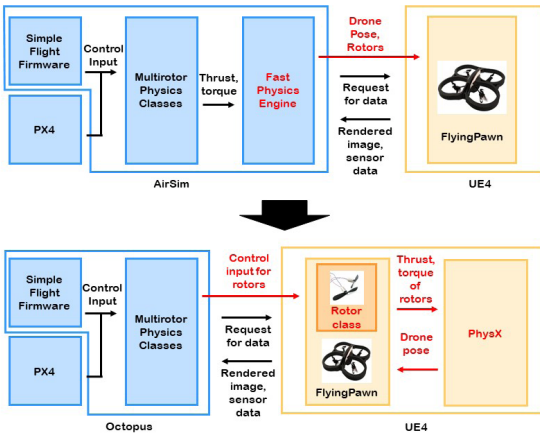


그림 2 PhysX 기반 드론 물리 엔진

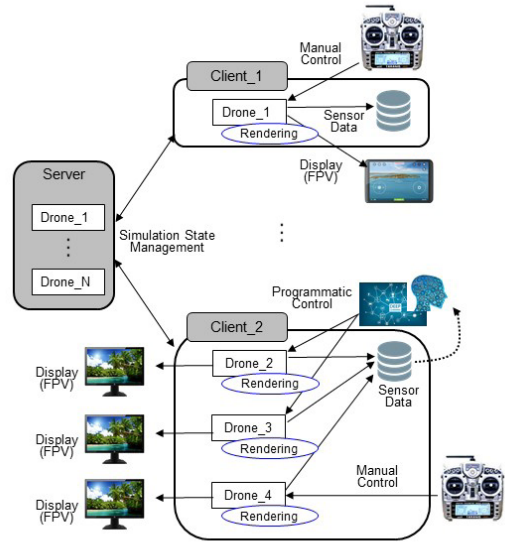


그림 4 분산 다중 드론 시뮬레이션 구조

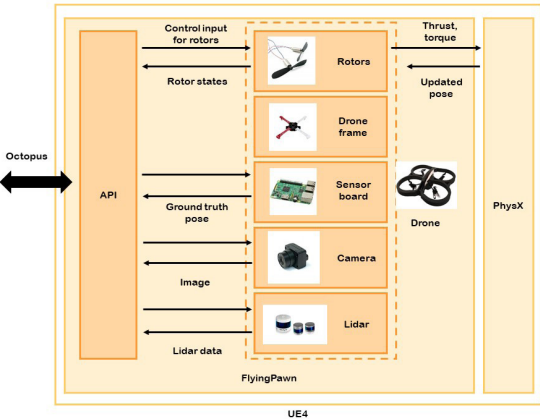


그림 3 드론 부품 탈부착 기능

PhysX 기반 드론 물리 엔진은 다음과 같은 장점을 가진다. 첫째, Unreal engine 내부 객체 간 물리적 상호작용이 가능해진다. 둘째, 드론 강체에 대한 부품 탈부착 시의 물리적 해석이 간단해진다. 이를 통해 그림 3과 같이 실제 드론을 조립하듯 부품별로 가상 드론의 조립이 가능하게 된다.

2. 분산 다중 드론 아키텍처

기존 드론 시뮬레이터들의 경우 한 대의 컴퓨터 상에서만 동작이 가능하다. 이 경우, 높은 현실감

을 위해 고화질 3D 렌더링이 구동되는 고성능 시뮬레이션 환경하에서는 다중 드론 시뮬레이션에 제약이 발생한다. 왜냐하면 한 대의 컴퓨터만으로는 대규모 다중 드론의 시뮬레이션이 요구하는 컴퓨팅 성능을 뒷받침할 수 없기 때문이다. 이와 같은 문제를 해결하기 위해 Octopus에는 분산 컴퓨팅 기술을 활용한 다중 드론 시뮬레이션 기능이 그림 4와 같이 탑재되었다.

분산 시뮬레이션 구조는 여러 대의 컴퓨터를 서버-클라이언트 구조로 연결한다. 분산 시뮬레이션 구조하에서 서버는 전체 시뮬레이션 상태(예, 드론의 위치)를 수집/관리하며, 클라이언트는 실질적인 시뮬레이션 태스크(드론 제어, 센서 데이터 아카이빙, 3D 렌더링 등)를 수행한다. AirSim의 경우 하나의 시뮬레이션 공간(Single Solid World)은 불변이며, 그 공간 내에 여러 대의 드론이 공존하는 구조이다. 하지만 Octopus의 경우 그림 5와 같이 시뮬레이션 공간(World)이 여러 복제공간(Replicated world)으로 나뉘게 되며, 이때 서버-클라이언트 간

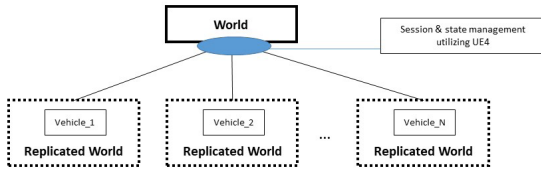


그림 5 복제공간 간 연결을 통한 분산 시뮬레이션

메시지 교환을 통해 서버가 시뮬레이션 상태정보를 취합/전파하게 되는 것이다.

IV. 결론

실제 드론을 사용하는 연구 개발 과정 중에 직면하게 되는 다양한 한계를 극복하기 위하여 드론 시뮬레이터의 도입이 고려되고 있다. 그러나 기존 드론 시뮬레이터의 경우, 첫째 현실감이 부족하여 A.I. 연구에 활용하기 부적절하거나, 둘째 정밀하지 못하여 적용 가능한 시나리오에 제약이 있거나, 셋째 대규모 드론의 동시 시뮬레이션이 불가능하다는 단점이 존재한다. 이와 같은 문제를 해결하고자 ETRI에서는 고확장/고정밀 드론 시뮬레이터인 *Octopus*를 개발 중이다. *Octopus*는 향후 CPS 플랫폼으로 확장되어 실제 드론과 가상 드론의 상호작용을 필요로 하는 응용까지 지원할 예정이다.

용어해설

Kinematics 운동학으로 번역되기도 하며, 주로 움직이는 물체의 상대적 운동을 다룸

약어 정리

CPS	Cyber Physical Systems
DNN	Deep Neural Network
FC	Flight Computer
FPS	Frame Per Second
GCS	Ground Control System

HITL	Hardware-In-The-Loop
IMU	Inertial Measurement Unit
RL	Reinforcement Learning
ROS	Robot Operating System
RPC	Remote Process Call
SITL	Software-In-The-Loop
SLAM	Simultaneous Localization And Mapping
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
URDF	Unified Robot Description Format
UTM	UAS Traffic Management
VR	Virtual Reality
VTOL	Vertical Take Off and Landing

참고문헌

- [1] K. Wackwitz, "Industry Update: The Drone Market Environment Map 2019," June 25, 2019, <https://www.droneii.com/drone-market-environment>
- [2] P. Kopardekar et al., "Unmanned aircraft system traffic management (UTM) concept of operations," in Proc. AIAA Aviation Technol. Integr. Operation Conf., Washington, D.C., USA, June 2016, pp. 1-16.
- [3] H. Kesteloo, "Amazon, Boeing, GE and Google to develop private unmanned traffic management (UTM) system," DroneDJ, Mar. 12, 2016, <https://www.wsj.com/articles/amazon-google-others-are-developing-private-air-traffic-control-for-drones-1520622925>.
- [4] F. Lardinois, "Google and Amazon talk about managing drone traffic at CES," TC, Jan. 10, 2016, <https://techcrunch.com/2016/01/09/google-and-amazon-talk-about-managing-drone-traffic-at-ces/>
- [5] <http://ko.byrobot.co.kr/eng/software-drone-dronefighter-simulator-kor/>
- [6] <https://www.dji.com/kr/simulator>
- [7] <https://thedroneracingleague.com/drl-sim-3/>
- [8] <https://dev.px4.io/v1.9.0/en/simulation/gazebo.html>
- [9] S. Shah et al., "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," Field Service Robotics, vol. 5, Springer, Cham, 2018, pp. 621-635.
- [10] S. Krishnan et al., "Air learning: An ai research platform for

- algorithm-hardware benchmarking of autonomous aerial robots," arXiv preprint arXiv:1906.00421, 2019.
- [11] M. Mueller et al., "Ue4sim: A photo-realistic simulator for computer vision applications," 2017.
- [12] W. Guerra et al., "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," arXiv preprint arXiv: 1905.11377, 2019.
- [13] A. Antonini et al., "The blackbird dataset: A large-scale dataset for UAV perception in aggressive flight," arXiv preprint arXiv: 1810.01987, 2018.
- [14] Dronecode, "jMjMAVSim with SITL," <https://dev.px4.io/v1.9.0/en/simulation/jmavsim.html>
- [15] http://wiki.ros.org/hector_quadrotor
- [16] F. Furrer et al., "RotorS—A modular Gazebo MAV simulator framework," Robot Operating System (ROS), vol. 625, Springer, Cham, 2016. pp. 595–625.
- [17] <https://www.nvidia.com/ko-kr/drivers/physx/physx-9-19-0218-driver/>