

다중로봇 임무모의 및 강화학습을 위한 전투급 시뮬레이터 연구

배정호¹⁾ · 이영일¹⁾ · 김도현¹⁾ · 김희수²⁾ · 김명영²⁾ · 김명준²⁾ · 김희영^{*1)}

¹⁾ 국방과학연구소 국방AI센터

²⁾ 리얼타임비주얼(주) 기술연구소

Battle Simulator for Multi-Robot Mission Simulation and Reinforcement Learning

Jungho Bae¹⁾ · Youngil Lee¹⁾ · Dohyun Kim¹⁾ · Heesoo Kim²⁾ · Myoungyoung Kim²⁾ ·
Myungjun Kim²⁾ · Heeyoung Kim^{*1)}

¹⁾ Defense AI Center, Agency for Defense Development, Korea

²⁾ R&D Center, REALTIMEVISUAL Inc., Korea

(Received 27 February 2024 / Revised 29 August 2024 / Accepted 30 August 2024)

Abstract

As AI technology advances, interest in performing multi-robot autonomous missions for manned-unmanned teaming (MUM-T) is increasing. In order to develop autonomous mission performance technology for multiple robots, simulation technology that reflects the characteristics of real robots and can flexibly apply various missions is needed. Additionally, in order to solve complex non-linear tasks, an API must be provided to apply multi-robot reinforcement learning technology, which is currently under active research. In this study, we propose the campaign model to flexibly simulate the missions of multiple robots. We then discuss the results of developing a simulation environment that can be edited and run and provides a reinforcement learning API including acceleration performance. The proposed simulated control module and simulated environment were verified using an enemy infiltration scenario, and parallel processing performance for efficient reinforcement learning was confirmed through experiments.

Key Words : Multi-Robot(다중로봇), Simulation(시뮬레이션), Reinforcement Learning(강화학습)

1. 서론

AI 기술이 발전함에 따라 유무인 협업(MUM-T)을

위한 다중로봇 자율 임무 수행에 관한 관심이 높아지고 있다. NATO 과학기술기구(STO) 협력 지원 사무국(CSO) 정보 시스템 기술(IST) 패널은 미국, 독일, 벨기에 등 7개국 합동으로 수행한 ‘무인 지상 시스템 및 C2 간 상호 운용성을 위한 가능성 개념 시연’ 프로젝트인 IST-149를 통해서 NATO 표준인 RAS-G IOP^[1]

* Corresponding author, E-mail: heeyokim@gmail.com

Copyright © The Korea Institute of Military Science and Technology

기반 UGV 및 OCU(Operator Control Unit)와 산업계에서 널리 쓰이는 ROS^[2] 기반 UGV 및 OCU들의 조합으로 실제 환경에서 기지주변 정찰 임무를 원활하게 수행하는 시연을 보여주었다^[3].

특히, 기존 규칙기반의 한계를 극복하고 복잡한 비선형 임무를 해결하기 위하여 심층강화학습 기법을 이용한 다중로봇 강화학습(MARL) 모델도 활발히 연구되고 있다. 구글의 AlphaStar^[4]와 OpenAI의 OpenAI Five^[5]가 각각 200:200 유닛 대결인 스타크래프트 2와 5:5 대결인 Dota 2에서 세계 최정상을 차지함으로써 다중로봇을 제어하는 AI의 효용성을 검증하였다. 또한 군집 경로 탐색, CSLAM, 협업공중교전 등 다양한 분야에서 다중로봇 강화학습 기술의 효용성이 확인되고 있다.

다중로봇의 자율임무 수행 기술을 개발하기 위해서는 적합한 시뮬레이션 기술이 필요하다. 첫 번째로, 자율임무의 모델링 및 실행 기술이 필요하다. 개발하고자 하는 AI 모델의 특성에 따라서 다양한 지역의 다양한 절차의 시나리오가 필요할 수 있다. 대부분의 시뮬레이터가 특정 시나리오를 예제로 제공하고 있지만, 기존의 맵 등의 설정에서 상황을 편집할 수 있는 등의 임무 수정의 용이성은 고려하지 않고 있다. 두 번째로 실제 로봇에 적용하는 Sim-to-Real을 고려하여 충실도 높은 물리엔진이 적용되어야 한다. 바닥 마찰력이나 바람 등의 외력을 고려한 로봇의 동역학을 고려하지 않는다면 실로봇 적용시에 기대한 성능을 얻기 어려울 수 있다. 마지막으로 심층강화학습을 효과적으로 적용하기 위한 기술들도 적용되어야 한다. 이때, 학습의 효율을 높이기 위하여 병렬 실행을 고려할 수 있다.

본 연구에서는 다중로봇의 임무를 유연성있게 모델링하기 위한 캠페인 모델을 제안하고, 이를 활용한 고효율 강화학습 수행을 위한 시뮬레이션 기법을 제안한다. 제안한 기법들을 검증하기 위하여 고품질도 물리엔진을 지원하는 다중로봇 임무 모의환경과 캠페인 모델 작성 및 시뮬레이션 실행 관리를 위한 시뮬레이션 환경을 개발하였다. 또한 효과적으로 다중로봇 강화학습을 수행하기 위한 인터페이스를 제공하고 학습 효율을 높이기 위한 병렬 실행 기법을 적용하였다.

본 논문의 구성은 다음과 같다. 2장에서 로봇 시뮬레이션 관련 연구들을 소개하고, 3장에서 개발한 시뮬레이터에 대한 아키텍처와 기능들에 대하여 논의한다. 4장에서 해당 기능들에 대한 효용성과 성능을 실험적 결과를 토대로 살펴보고, 5장에서 결론을 기술한다.

2. 관련 연구

PettingZoo^[6]는 MARL 알고리즘을 개발을 위한 다양한 시뮬레이션 환경을 제공한다. 단일 에이전트의 성능 평가를 위한 대표적인 벤치마크 환경인 OpenAI Gymnasium^[7] Atari의 다중로봇 버전인 Multiplayer-Atari를 비롯하여, 격자세상에서 1,000대 이상을 모의할 수 있는 MAgent, 연속공간을 모델링할 수 있는 MPE 등 약 50개의 환경을 제공한다. 이들은 모두 2D 환경으로 새롭게 개발한 MARL 알고리즘의 성능을 확인하기 위한 용도로는 사용될 수 있지만 충실도가 낮아 Sim-to-Real에는 부적합 할 수 있다. 또한 실제 임무 시나리오와는 다소 차이가 나고 기 개발되어있는 시나리오를 수정하기 어렵다는 한계가 있다.

GRF(Google Research Football)^[8], SMAC(Starcraft Multi-Agent Challenge)^[9] 등 특정 도메인에 특화된 2D MARL 시뮬레이터들도 공개되어 있다. 이들은 도메인의 특성에 따라 다양한 초기 구성을 편집할 수 있는 기능을 제공한다. 하지만 도메인이 특정되어 있거나 시간적 절차에 따른 시나리오를 모의하는 기능은 제공하지 않는다.

Gazebo^[10], UnrealEngine, Unity, MuJoCo^[11] 등의 3D 시뮬레이션 엔진을 이용한 고품질도의 강화학습 시뮬레이터도 활발히 개발되고 있다. Microsoft의 AirSim^[12], NVIDIA Isaac Gym^[13] 등은 단일 로봇을 위한 강화학습 시뮬레이션 도구로 널리 사용되고 있다. Unity에서 제공하는 3D MARL 환경인 MLAGents는 고품질도의 환경을 제공하며 병렬 기능을 제공한다. 스네이크 게임 등 17개 이상의 다양한 기본환경을 제공하지만 시나리오를 편집하기 위해서는 직접 C# 기반으로 개발해야 하고, HILS를 고려한 인터페이스를 제공하지 않는다.

3. 아키텍처

본 연구에서 제안하는 시뮬레이터의 목표 기능 및 성능은 재사용성 높은 시나리오 모델 검증 및 실행, 가상환경 및 다중로봇 제어 명령 검증, 다중로봇 강화학습 인터페이스 제공에 있다. Fig. 1은 시뮬레이터 아키텍처의 개념도를 보여준다. 모의통제 모듈을 통해서 캠페인 모델을 작성할 수 있으며, 이를 이용하여 시뮬레이션을 실행하는 기능을 제공한다. 다중로

다중로봇 임무모의 및 강화학습을 위한 전투급 시뮬레이터 연구

로봇 임무 실행 모의 환경은 아군/적군/장애물 등의 아티팩트를 모의하는 기능을 제공한다. 강화학습 API는 강화학습을 위한 시나리오 선택, 임무 시작 및

로봇 상태 정보 수신 임무 명령 전송 등의 API를 제공한다. 다음 각 절에서 각 모듈의 세부 기능을 설명한다.

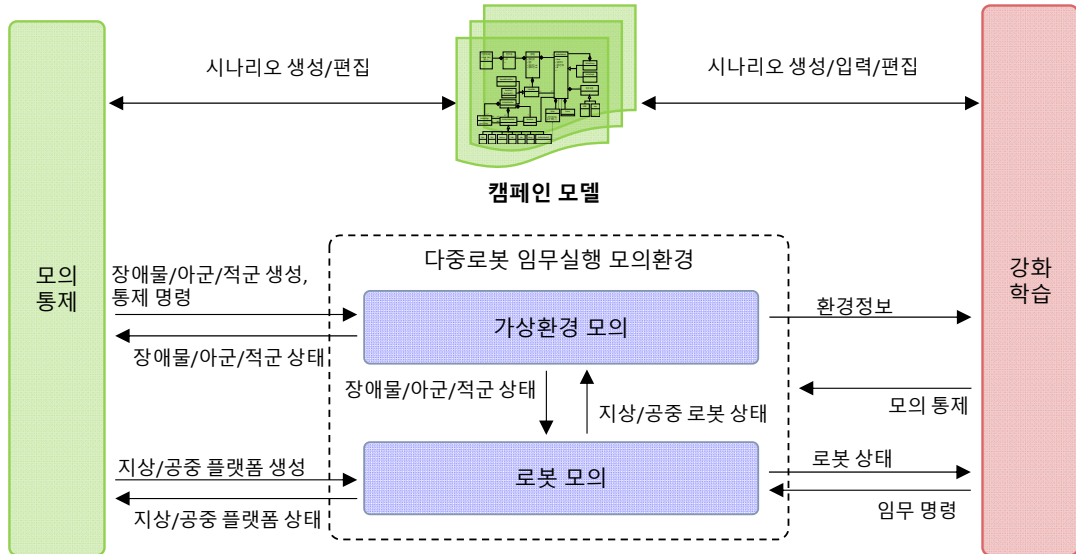


Fig. 1. Simulator architecture overview

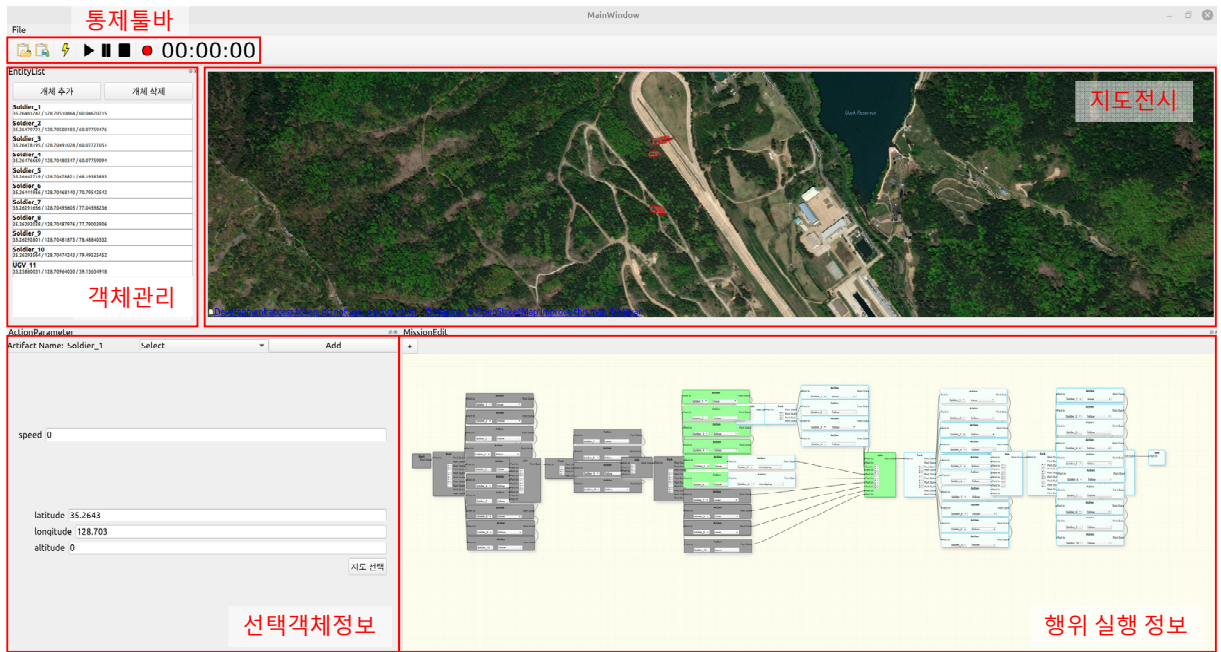


Fig. 2. Simulation control GUI

3.1 자율임무 모델링 및 실행

모의 통제 모듈은 캠페인 모델을 관리하고 모의 실행 및 정지 기능 등을 제공한다. Fig. 2는 모의 통제 모듈의 사용자 인터페이스를 보여준다. 통제툴바를 통해서 임무를 명세하는 캠페인 모델을 불러오거나 저장할 수 있고, 모의를 시작하거나 일시정지, 정지 또는 녹화를 진행할 수 있다. 객체 관리 영역에는 현재 모의에 참여하는 객체들에 대한 정보가 리스트 형식으로 전시되며 모의를 시작하기 전 임무를 관리하기 위하여 객체를 추가하거나 삭제 할 수 있다. 지도전시 영역에서는 2D 이미지로 지도가 전시되며 모의 중 객체들의 현재 위치를 실시간 전시한다. 선택객체정보 영역에서는 선택한 객체의 위치, 속도 등의 기본정보를 전시하고, 행위 실행정보 영역에서는 NPC(Non Playable Character)의 기 계획된 행위를 전시하며, 모의가 시작되기 전에는 수정이 가능하고 시작된 후에는 현재 진행상황을 색깔을 달리하여 표현하는 기능을 제공한다.

본 연구에서는 시뮬레이션 가시화 도구 등에 독립적으로 임무를 관리하기 위하여 재사용성 높은 캠페인 모델 제안한다. 캠페인 모델의 구성은 Fig. 3과 같다. 모의 통제 모듈은 캠페인 모델을 json 포맷으로 관리한다.

캠페인 모델은 기본정보로 캠페인 명, 설명, 성공

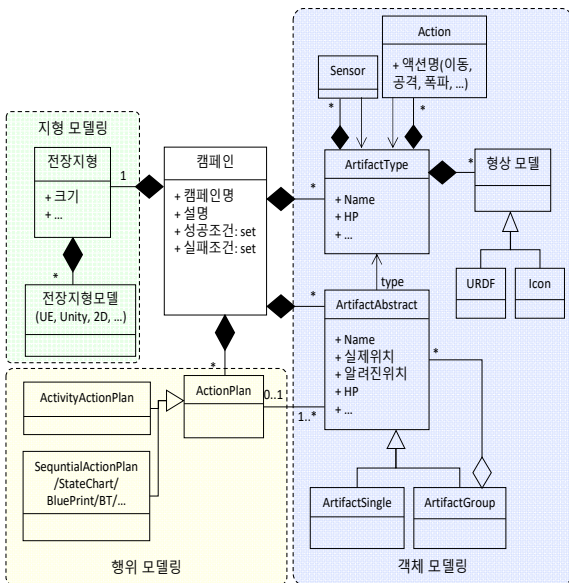


Fig. 3. Campaign meta-model

또는 실패조건을 기술할 수 있으며, 크게 지형 모델링 정보, 객체 모델링 정보, 행위 모델링 정보를 가진다. 지형 모델링 정보는 지형이름, 버전, 크기, 위치 등의 기본 지형정보를 가지며 FBX, MAP(UE), Scene(Unity), 2D 이미지 등의 다양한 포맷으로 지형 모델을 따로 관리한다.

객체 모델링은 크게 객체 타입(ArtifactType)과 추상 객체(ArtifactAbstract)로 구성된다. 객체 타입에서 종류, 체원, 보유 센서, 수행가능한 액션, URDF 등의 형상 모델을 정의하고, 이를 이용하여 추상객체를 통해서 캠페인에서 임무를 수행하는 객체를 구체화하고 그룹화할 수 있다.

행위 모델링에서는 각 객체의 행위를 계획함으로써 임무의 흐름 설계할 수 있다. 행위계획은 UML^[14] 액티비티 다이어그램, 상태머신 다이어그램, 시퀀스, 블루프린트, 행위트리 등 다양한 행위 모델링 기법으로 기술이 가능하도록 설계되어 있다. 본 연구에서는 액티비티 다이어그램으로 행위계획을 설계하고 이를 실행하는 스케줄러를 구현하였다.

3.2 가상환경 및 다중로봇 모의

본 연구에서 사용한 시뮬레이션 환경은 언리얼 엔진(UE) 5.1 기반으로 개발하였다. 지형으로는 국과연 창원시험장, 실내 실험실 등 다수의 지형을 개발하여 적용중에 있으며, 눈, 비, 안개 등의 날씨 변화 및 주/야간 시간 변화를 지원한다. 눈, 비, 바람 등 로봇 제어와 관련된 환경요소들은 로봇의 동역학 엔진과 연계하여 영향을 주도록 설계하였다.

제어 가능한 로봇으로 차륜형, 궤도형 UGV와 쿼드콥터 드론을 지원하며 적군으로는 로봇들 외에 병사도 선택 가능하다. Fig. 4는 적군병사, UGV, 드론, 지형 등 시뮬레이션 예제를 보여준다. 로봇의 형상은 URDF로 관리하며 모의 성능을 고려하여 기본 플랫폼, 구동기, 센서 등의 수준으로 구분하였다.

UGV의 물리엔진은 UE5의 카오스 비히클을 적용하였고, 드론의 물리엔진은 쿼터니언 연산 기반 6자유도 비선형 운동방정식 모델을 이용하는 jMAVSim^[15]의 330급 쿼드콥터 비행 동역학을 토대로 개발하였다. 드론의 상세 파라미터는 실 기체 궤적과 비교하여 튜닝하였다. Fig. 5는 실기체와의 궤적을 비교하여 보여준다. 주황색 선은 명령 값이고, 노란색과 파란색은 각각 모의결과와 실기체의 이동 궤적을 나타낸다. ROS2 인터페이스는 rcIUE를 적용하였다.



Fig. 4. A simulation example

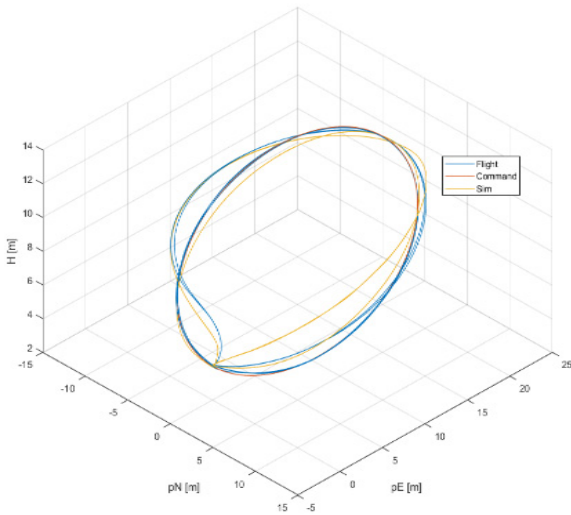


Fig. 5. Drone physics model validation with real machine

3.3 다중로봇 강화학습 인터페이스

본 연구에서 개발한 모의환경은 강화학습을 위한 API로 PettingZoo의 Parallel API를 지원한다. Parallel API는 모든 에이전트의 관측값(observation)을 받아서 같은 시간이 모든 에이전트의 명령(action)을 실행하는 구조로, RLLib^[16], SMAC 등 많은 다중 에이전트 강화학습 환경에서 적용하는 방식이다.

학습환경은 파이썬으로 개발할 수 있도록 제공한다. 파이썬 기반의 학습환경과 언리얼엔진 기반의 모의환

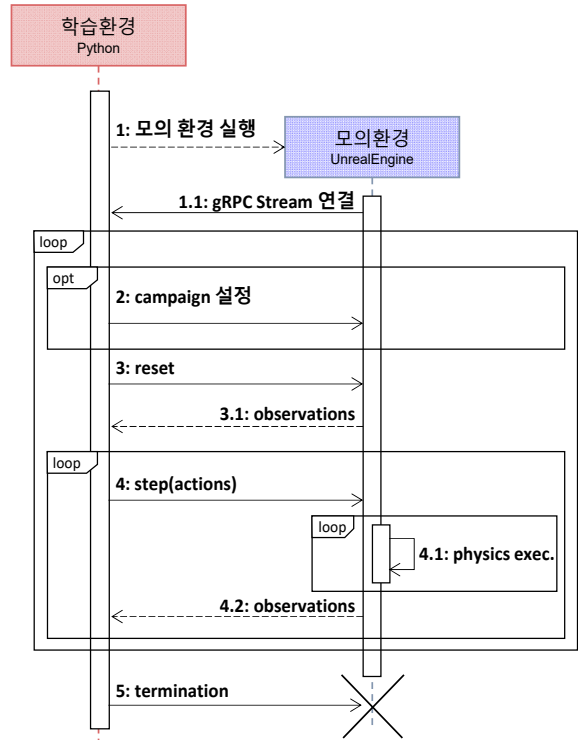


Fig. 6. Reinforcement learning sequence

경은 gRPC^[17]를 통하여 통신하도록 설계하였다. Fig. 6은 학습을 진행하는 일련의 과정을 보여준다. 학습환경에서 학습을 시작하면 모의환경을 실행하여 gRPC 연결을 시도한다. 통신이 연결되면 학습환경에서 캠페인 시나리오를 전송하여 따라서 아군, 적군, 지형/지물 등의 아티팩트들 정보를 설정한다. 이후 *reset*과 *step* (actions) API를 통하여 강화학습을 진행한다. Fig. 7은 강화학습 진행 예제를 보여준다.

본 연구에서 제안하는 모의환경은 블로킹과 언블로킹의 두 가지 스텝의 처리 로직을 지원한다. 블로킹 방식은 스텝을 통하여 명령을 전송하면 한 스텝을 진행하고 관측값을 계산 후 다음 스텝이 실행될 때까지 멈추는 방식이다. Atari, SMAC 등 다양한 강화학습 환경에서 이 방식을 사용하고 있다. 이 방식은 입력한 명령을 지연없이 실행하기 때문에 학습의 수렴이 쉽고, 스텝 후 가능한 빠르게 모의를 진행하기 때문에 컴퓨팅 파워를 효율적으로 사용할 수 있다는 장점이 있다. 하지만 실제 환경은 명령의 수신 여부에 상관없이 상황이 진행되는 언블로킹 방식이다. 즉 학습모델이 관측값 수신 후 다음 명령을 추론하는 과정에서 시간



Fig. 7. Reinforcement learning progress example

지연 이 발생하기 때문에 블로킹 방식으로 학습된 모델이 의도한바대로 진행될 것이란 보장이 없다. 이 두 방식의 모의는 서로 장단점을 가지고 있으며 sim-to-real을 위해서는 모두 필요할 수 있다.

일반적으로 강화학습은 일정수준의 샘플의 수가 모이면 GPU를 사용하여 모델을 업데이트하기 때문에 모든 학습데이터가 미리 준비되어 있는 덩어리와 달리 샘플을 획득하는 속도가 학습 속도에 큰 영향을 미친다. 본 모의 환경은 샘플 획득 속도를 높이기 위하여 병렬화 기법을 제공한다. 병렬화 기법은 동시에 둘 이상의 에피소드를 동시에 진행하는 방식으로 본 연구에서는 두 가지 형태로 지원하며, 첫 번째는 단일 환경에서 복제된 다수의 맵을 동시에 탑재하는 복제 방식이고, 두 번째는 언리얼 환경을 다수 실행하는 병렬 컴퓨팅 방식이다. 첫 번째 방식은 컴퓨팅 파워가 좋은 서버급에서 실행할 때 적용하기 용이하고 두 번째 방식은 다수의 컴퓨터로 분산 실행할 수 있다는 장점이 있다. 이 두 방식은 서로 독립적으로 동시에 적용 가능하다.

3.4 시뮬레이터 비교

본 절에서는 기존의 시뮬레이터와 제안한 시뮬레이터를 비교한다. 비교 항목은 다음과 같다.

- 임무편집 도구(Mission Editing Tool): 임무 생성 및 수정을 위한 도구를 지원하는 가를 평가한다. 넓은 공간에서 수행하는 복잡한 임무를 효율적으로 정의하기 위해서는 적군/장애물의 배치 및 행위를 특정 프로그래밍 언어를 사용하는 것이 아니라 시각화하여 편집할 수 있는 기능을 제공하여야 한다.
- 고 충실도(High Fidelity): Sim-to-Real 적용을 고려하여 고충실도 물리엔진을 사용하였는가를 평가한다.
- 다중로봇 강화학습 지원(MARL): 다중로봇 강화학습을 지원하는가를 평가한다. 다중로봇 강화학습을 지원하기 위해서는 시뮬레이터가 다중로봇 제어를 지원하고 다중로봇 강화학습을 위한 인터페이스를 제공하여야 한다.
- 병렬실행(Parallel): 둘 이상의 환경을 동시에 실행할 수 있는가를 평가한다.

2장에서 소개한 시뮬레이터들을 대상으로 항목별로 정리한 내용은 Table 1에서 보여준다. AirSim과 Isaac은 고충실도 강화학습 시뮬레이션 환경으로 다양한 벤치마크를 제공하고 있다. 하지만 새로운 임무 학습을 위해서는 환경 요소들의 배치를 Unity/Unreal 상에서 직접 수정해야하며, 학습대상이 아닌 요소들의 행위는 각각 C# 등으로 코딩하여 정의해주어야 한다. SMAC과 PettingZoo는 다중로봇 강화학습 시뮬레이션 환경으로 다중로봇 알고리즘을 개발하기에는 적합할 수 있으나, 충실도가 낮아 Sim-to-Real을 위한 학습에는 한계가 있을 수 있다. GRF와 MLAGents는 고충실도 다중로봇 강화학습 환경을 제공하지만 임무가 제한적이다. 특히 GRF는 인원 수 등은 조정할 수 있지만 축구장 환경만을 제공하며, MLAGents는 물리엔진을 이용한 간단한 게임 환경은 제공하지만 시나리오를 수정하기 위해서는 텍스트 기반 프로그래밍 언어로 게임을 수정해야 한다. 본 연구에서 제안하는 고충실도 강화학습 시뮬레이션 환경은 학습 성능을 높일 수 있는 병렬실행을 지원하고 제안한 캠페인 모델을 중심으로 직관적인 임무 편집 및 실행 기능을 제공하여 다양한 자율임무를 학습할 수 있도록 지원한다.

Table 1. Simulator comparison

Simulator	Mission Editing Tool	High Fidelity	MARL	Parallel
AirSim		√		
Isaac gym		√	√	√
SMAC			√	√
PettingZoo			√	√
GRF		√	√	√
MLAgents		√	√	√
Ours	√	√	√	√

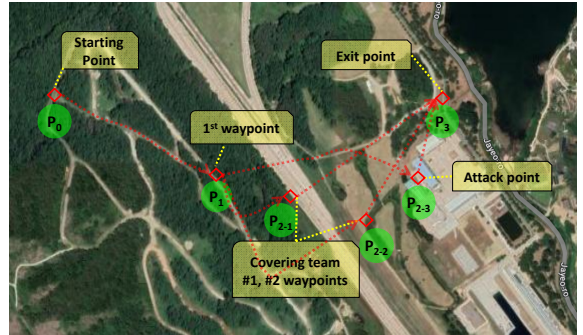


Fig. 8. Mission simulation experimental scenario

4. 실험결과

4.1 임무모의 실험 결과

본 연구에서 제안한 다중로봇 임무 실행 모의 통제 및 모의 환경을 검증하기 위하여 실험을 수행하였다. 실험의 목적은 액티비티 다이어그램 기반의 캠페인 모델로 적군의 행위를 모델링 할 수 있는지 확인하고, 모델링한 행위를 모의 통제 및 모의 환경을 통해서 모의를 진행할 수 있는 확인하는 것이다. 그리고 적군 행위 모의 중에 아군 로봇을 투입하여 ROS2 기반의 제어 명령을 아군 로봇이 잘 수행하는지 확인하고, 교전상황이 발생할 경우 교전 후에도 적군의 남은 행위를 잘 수행하는지 확인하는데 있다.

실험 시나리오는 Fig. 8과 같다. P₀ 지역에서 10명의 적군이 준비한다. 이후 P₁ 지역까지 이동 후, 2명, 2명 6명으로 나누어 P₂₋₁, P₂₋₂, P₂₋₃ 지역으로 이동한다. 이때, P₂₋₁, P₂₋₂ 지역으로 이동한 인원들은 감시 과업을 수행한다. P₂₋₃ 지역으로 이동한 인원 중 일부는 마인을 매설한다. 일련의 과업이 모두 끝나면 P₃ 지역으로 퇴각한다.

시나리오에 대한 ActionPlan 모델링 결과는 Fig. 9와 같다. 표준 액티비티 다이어그램 표현법에 따라서, 검은 점은 시작, 이중 검은 점은 종료, 둥근 네모는 액션, 긴 선은 동시에 액션들이 수행되는 fork와 합쳐지는 join을 나타낸다. 이 예의 행위모델에 표현되어 있는 아티팩트는 모두 ArtifactGroup이며, G₀₀은 10명의 모든 적병사 그룹, G₀₁는 병사1, 2, G₀₂는 병사

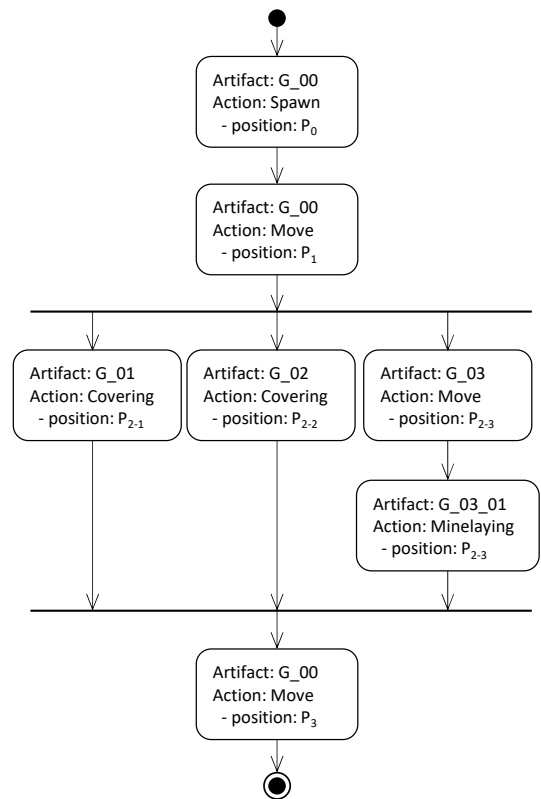


Fig. 9. An actionPlan example

3, 4, G₀₃은 병사 5~10, G_{03_01}은 병사 5, 6으로 구성된 그룹이다. 이 예에서 병사들이 수행 예정인 과업들은 Spawn, Move, Covering, Minelaying의 4가지이며, Spawn은 병사가 생성되는 과업이고, Move는 이동, Covering은 감시, Minelaying은 지뢰 매설 과업을 나타낸다. 모델링 결과를 통하여 해당 시나리오는 모두 표현됨을 확인하였다.

모의통제를 통하여 캠페인을 작성할 수 있고, json 포맷으로 저장된 캠페인을 정상적으로 읽어오는 것을 확인하였다. 돌발 상황이 없는 경우 캠페인이 기술된 행위모델에 따라서 퇴각 지역으로 이동까지 정상 진행되는 것을 확인하였다. 캠페인 진행 중 UGV 로봇을 투입하여 조이스틱을 통하여 ROS2 메시지 기반의 제어가 정상 동작함을 확인하고, P_{2,1} 지역에서 교전하였을 때, 이 지역의 인원이 행동 불능이 된 경우에도 남은 행위들이 종료까지 정상 동작함을 확인하였다.

4.2 강화학습 인터페이스 실험 결과

본 실험에서는 제안하는 시뮬레이터가 다중로봇 강화학습을 효율적으로 지원할 수 있는지를 검증하기 위하여 확장성(scalability)을 확인하였다. 이를 위해서 병렬환경의 수와 배속을 증가시키며 시뮬레이션 효율성을 측정하였다. 실험의 설정은 Table 2와 같다.

본 실험을 수행한 컴퓨터에서는 메모리의 한계로 한번에 최대로 운용가능한 로봇은 200대로 제한되었다. 또한 CPU/GPU의 한계로 동역학의 최대 속도는 150 Hz로 제한되었다. 이를 기반으로 실험 시나리오에서 로봇의 수는 5대로 고정된 상태에서 최대 병렬 생성가능한 환경의 수를 40개까지 증가시키며 실험하였다. 또한 로봇의 동역학의 주기를 30 Hz로 고정하여 최대 5배까지 가속을 하며 실험을 수행하였다. 학습에 사용한 컴퓨터의 사양은 CPU 13세대 인텔 코어 i9-13900H(14코어), RAM 64 GB DDR5, GPU NVIDIA GeForce RTX 4070 Laptop 8 GB이다. Fig. 10은 병렬 실행 환경 예를 보여준다.

실험의 평가지표인 시뮬레이션 효율성(SE)은 초당 획득한 평균 샘플의 수(#samples/sec)로 정의하였다. 이를 계산하는 수식은 수식 (1)과 같다.

Table 2. Simulation setting parameters

Parameter	Values
Robot dynamics period	30 Hz
Step period	10 Hz
#robots	5
speed	1x, 2x, 3x, 4x, 5x
#maps(envs)	1, 10, 20, 30, 40

$$SE = robots \times maps \times \delta \tag{1}$$

robots는 로봇의 수이고, maps는 병렬로 실행되는 환경의 수, δ는 초당 실제 액션을 수행한 주기로 100 스텝까지 측정된 평균값으로 측정하였다.

Fig. 11은 시뮬레이션 효율성 측정 실험 결과를 보여준다. 가로축은 소환된 배속이고, 세로축은 시뮬레이션 효율성이다. 실험 결과 5대의 로봇을 운용하는 실험에서도 초당 3,000개 이상의 샘플을 획득할 수 있음을 확인하였다. 이는 기본 시뮬레이션 도구를 사용했을 경우 10 Hz로 5대의 로봇을 운용하는 상황은 50개의 샘플만을 얻을 수 있음을 감안하면 약 60배 이상 효율이 향상되었음을 나타낸다.

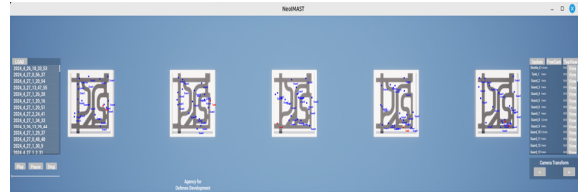


Fig. 10. Parallel execution example

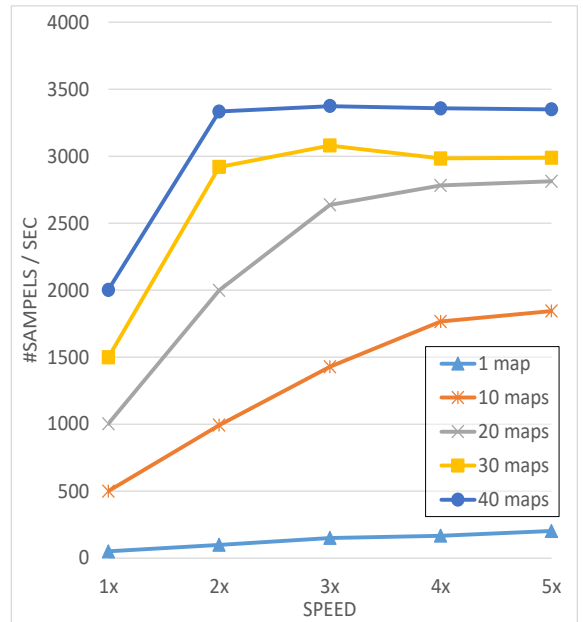


Fig. 11. A rollout experiment result

5. 결론

본 논문에서는 다중로봇 협업 임무 수행을 위한 전투급 시뮬레이터를 제안하였다. 주요 기능으로 동적으로 임무를 정의할 수 있는 캠페인 모델을 정의하고 이에 따라 적군을 모의할 수 있으며, 다중로봇 협업 강화학습을 위한 API를 제공한다. 제안한 기능을 검증하기 위하여 적군 침투 시나리오를 작성하여 전투급 시뮬레이션이 정상 동작함을 확인하였고, 30 Hz 동역학 주기 설정에서 병렬을 적용하여 동역학 결과의 변동없이 최대 60배 이상의 샘플링 속도 향상을 확인하였다. 향후 연구로 다양한 시나리오를 수행할 수 있는 다중로봇 강화학습 모델을 개발할 예정이다.

후 기

이 논문은 2024년 정부의 재원으로 수행된 연구임

References

- [1] P. Bounker, and A. Volk, "NATO LCG UGV ToE Interoperability Experiences," in Proc. AVT-241 Specialists' Meeting on Technological and Operational Problems Connected with UGV Application for Future Military Operations, 2015.
- [2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A.Y. Ng, "ROS: An Open-Source Robot Operating System," ICRA Workshop on Open Source Software, Vol. 3, p. 5, 2009.
- [3] K. Mathiassen, F. E. Schneider, P. Bounker, A. Tiderko, G.D. Cubber, M. Baksaas, J. Glówka, R. Kozik, T. Nussbaumer, J. Röning, J. Pellenz, and A. Volk, "Demonstrating Interoperability between Unmanned Ground Systems and Command and Control Systems," Int. J. Intelligent Defence Support Systems, Vol. 6, No. 2, pp. 100-129, 2021.
- [4] O. Vinyals et al, "Grandmaster Level in Starcraft II using Multi-Agent Reinforcement Learning," Nature, Vol. 575, No. 7782, pp. 350-354, 2019.
- [5] J. Raiman, S. Zhang, and F. Wolski, "Long-Term Planning and Situational Awareness in OpenAI Five," arXiv:1912.06721, 2019.
- [6] J. Terry et al, "PettingZoo: gym for Multi-Agent Reinforcement Learning," Advances in Neural Information Processing Systems, Vol. 34, pp. 15032-15043, 2021.
- [7] M. Towers et al, "Gymnasium," Zenodo, 2023.
- [8] K. Kurach, A. Raichuk, P. Stanczyk, and M. Zajac, "Google Research Football: A Novel Reinforcement Learning Environment," AAAI Conf. Artificial Intelligence, pp. 4501-4510, 2020.
- [9] M. Samvelyan et al, "The Starcraft Multi-Agent Challenge," Int'l Conf. Autonomous Agents and MultiAgent Systems, pp. 2186-2188, 2019.
- [10] N. Koenig, and A. Howard, "Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator," IEEE/RSJ Int'l Conf. Intelligent Robots and Systems, Vol. 3, pp. 2149-2154, 2004.
- [11] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A Physics Engine for Model-Based Control," IEEE/RSJ Int'l Conf. Intelligent Robots and Systems, pp. 5026-5033, 2012.
- [12] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," Springer Proc. Advanced Robotics, Vol. 5, pp. 621-635, 2018.
- [13] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "GPU-Accelerated Robotic Simulation for Distributed Reinforcement Learning," Conf. Robot Learning, pp. 270-282, 2018.
- [14] OMG, UML. <http://www.uml.org>
- [15] A. Babushkin, jMAVSIM. <https://pixhawk.org/dev/hil/jmavsim>
- [16] E. Liang, et al, "Ray RLLib: A Composable and Scalable Reinforcement Learning Library," arXiv: 1712.09381, 2017.
- [17] X. Wang, H. Zhao, and J. Zhu, "GRPC: A Communication Cooperation Mechanism in Distributed Systems," ACM SIGOPS Operating Systems Review, Vol. 27, No. 3, pp. 75-86, 1993.