

20CTA
P-C152
021-02

보안 과제(), 일반 과제(√) / 공개(√), 비공개()발간등록번호()

국토교통기술촉진 연구사업 최종보고서

R&D / 20CTAP-C152021-02

다
중
실
시
간
인
공
지
능
임
무
계
획
에
이
전
트
기
반
개
발
초
자
율
무
인
기
최
종
보
고
서

다중 인공지능 에이전트 기반 초자율 무인기 실시간 임무계획 알고리즘 개발 최종보고서

2021

2021. 03. 19.

주관연구기관 / 건국대학교

국
토
교
통
부

국토교통과학기술진흥원

국토교통부
국토교통과학기술진흥원

제 출 문

국토교통과학기술진흥원장 귀하

‘다중 인공지능 에이전트 기반 초자율 무인기 실시간 임무계획 알고리즘 개발’(연구개발 기간 : 2019. 04.01 ~ 2020.12.31) 과제의 최종보고서 10부를 제출합니다.

2021. 03. 19.

주관연구기관명 : 건국대학교 산학협력단 (대표자) 송창선 (인)



주관연구기관책임자: 이재우

국토교통부소관 연구개발사업 운영규정 제29조에 따라 최종보고서 열람에 동의합니다.

보고서 요약서

보고서 요약서

과제 고유 번호	20CTAP-C152 021-02	해당 단계 연구 기간	2019.04.15. - 2020.12.31	단계구분	1/1
연구사업명	중사업명	국토교통기술촉진연구사업			
	세부사업명	국토교통기술촉진연구사업			
연구과제명	대과제명	다중 인공지능 에이전트 기반 초자율 무인기 실시간 임무계획 알고리즘 개발			
	세부과제명	다중 인공지능 에이전트 기반 초자율 무인기 실시간 임무계획 알고리즘 개발			
연구책임자	이재우	해당단계 참여연구원 수	총: 12 명 내부: 12 명 외부: 0 명	해당단계 연구비	정부: 370,000천원 민간: 천원 정부외: 천원 상대국: 천원 계: 370,000천원
		총 연구기간 참여연구원 수	총: 12 명 내부: 12 명 외부: 0 명	총 연구비	정부: 370,000천원 민간: 천원 정부외: 천원 상대국: 천원 계: 370,000천원
연구기관명 및 소속 부서명	건국대학교 산학협력단			참여기업명	
국제공동연구	상대국명:			상대국 연구기관명:	
위탁연구	연구기관명:			연구책임자:	

※ 국내·외의 기술개발 현황은 연구개발계획서에 기재한 내용으로 같음

연구개발성과의 보안등급 및 사유	일 반
-------------------------	-----

9대 성과 등록·기탁번호

구분	논문	특허	보고서 원문	연구시설 · 장비	기술요약 정보	소프트 웨어	화합물	생명자원		신품종	
								생명 정보	생물 자원	정보	실물
등록·기 탁 번호											

국가과학기술종합정보시스템에 등록된 연구시설·장비 현황

구입기관	연구시설 · 장비명	규격 (모델명)	수량	구입연월일	구입가격 (천원)	구입처 (전화)	비고 (설치장소)	NTIS 등록번호
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

요약 : 다음 장의 요약문 참조

보고서 면수

134

요약문

연구의 목적 및 내용	인공지능을 이용한 경로 계획 및 장애물 회피 기술, 인공지능 상황판단을 통한 임무 재설계 기술 및 고 정밀 시뮬레이션을 이용한 인공지능 훈련 데이터 획득 기술 개발을 통해 자율화 레벨(ATRA) 6 이상의 초자율 무인기를 위한 다중 에이전트 개발 및 결합 원천 기술을 확보하며, 최종적으로 자율화레벨 9~10 수준의 무인기/드론 개발에 대한 장기적 연구 비전을 가지고 유기적이고 융합적으로 수행함				
연구개발성과	본 연구개발 과제 수행을 통해, 4차 산업혁명의 신기술로 인식되고 있는 다중 인공지능 기술의 원천 기술로써 인간이 조종하던 기계 시스템인 무인기/드론 비행체에 다중 인공지능을 도입하기 위한 다음과 같은 초자율 무인기/드론 다중 에이전트 개발 및 결합 핵심 기술을 확보하였음 (1) 장애물 인식 및 회피 / 실시간 경로 계획 및 최적 경로 생성 인공지능 알고리즘 • DQN을 이용한 장애물 식별, 시스템 모니터링 인공지능과 결합하여 무인기의 기동성 및 안전 거리를 고려한 회피 인공지능 에이전트 기술 연구 • 실시간 계산 및 정적/동적 주변 환경에 따른 실시간 경로 계획 및 최적 경로 생성 인공지능 에이전트 기술 연구 (2) 시스템 모니터링 / 임무 재설계 알고리즘 기술 • 센서 오작동 및 임무 지속성 가능 여부 판단 • 센서 융합을 통한 무인기 시스템 상태 모니터링 인공지능 • 다중 인공지능 에이전트 기반 임무 지속성 판단 및 임무 변경/취소 판단 결정 인공지능 에이전트 개발 (3) 다중 인공지능 에이전트 결합 및 초자율 무인기 시뮬레이션을 통한 검증 • 무인기 비행 제어 연구를 통한 인공지능 에이전트 및 시뮬레이션 결합연구 • 다중 인공지능 에이전트 기반 초자율 무인기 시뮬레이션 기술 연구				
연구개발성과의 활용계획 (기대효과)	(1) 활용계획 • 높은 수준의 자율 지능 무인기 기술을 기업에 이전하여 지능형 무인기 개발에 활용 • 지능형 의사결정 알고리즘, 지능 제어계 및 지능임무계획 원천기술을 획득하여 민,군용 드론/자율주행 자동차/로봇 등 첨단산업 분야에 적용 • 미래 지능형 무인기 기술을 국내에서 개발함으로써 급증하는 무인기 시장을 선점하며, 향후 항공 선진국들과의 협력개발에 주도적으로 참여 (2) 기대효과 • 다수의 석·박사과정 학생들이 연구에 참여하게 됨으로써 미래를 위한 지능형 무인기 전문 인력양성에 기여함. • 본 연구를 통하여 양성된 전문 인력들의 세계적인 연구 성과 창출을 통하여 지능형 무인기 관련 신산업을 창출함 • 지능형 무인기 분야 국내 연구를 선도할 수 있도록 학문적 연구 토대 마련 • 학계와 산업체의 협력으로 시너지 효과를 창출하여 향후 지능형 무인기 분야를 선도하는 전문 연구 네트워크 구축 • 협업이 중요한 향후 지능무인기 연구 네트워크 자원 양성 및 국제협력 시 질적 제고 가능				
국문핵심어 (5개 이내)	실시간 임무계획	다중 인공지능 에이전트	초자율 무인기	시뮬레이션	드론
영문핵심어 (5개 이내)	Real Time Mission planning	Multi Agent Artificial Intelligence	Autonomous UAV	Simulation	Drone

〈 목 차 〉

1. 연구개발과제의 개요	5
2. 연구수행내용 및 성과	26
3. 목표 달성도 및 관련 분야 기여도	116
4. 연구개발성과의 활용 계획 등	124
붙임. 참고 문헌	129
<별첨> 주관연구기관의 자체평가 의견서	

1. 연구개발과제의 개요

가. 연구 개발의 필요성

1) 연구 개발의 흐름



그림 1 연구 개발 개요

2) 연구 개발의 필요성

- ▶ 현재 미국에서는 NAS-NextGen의 초석을 마련하고 무인기가 안전하게 저고도 공역(Class G)의 운항을 가능하게 하는 UTM (Unmanned Aerial System Traffic Management)의 연구를 진행중임.

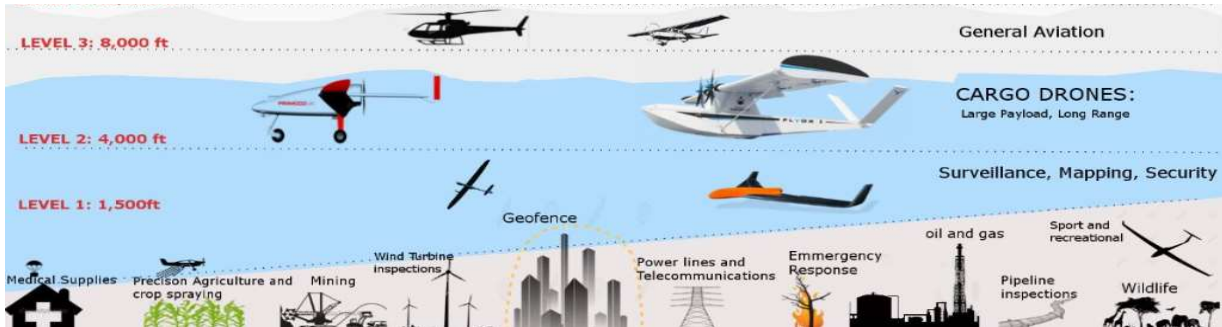


그림 2 UTM을 활용한 각 공역별 무인기 운항 개념

- ▶ 이에 따라 많은 무인비행체가 현재 유인항공기와 같이 공역 내에서 안전하게 비행하기 위해서는 조종사가 지속적으로 무인기를 제어하는 형태의 운항 개념이 아닌 무인기 스스로 인지 및 판단하여 비행 하는 **‘초자율’ 형태의 운항 개념**이 필요.
- ▶ 기존 개발 중인 저고도 무인비행장치 교통관리 체계(UTM)의 개념은 각 무인비행장치의 정보를 전송 받아 유인항공기와 유사한 관리 체계로써 무인비행장치-관리체계와 지속적인 관리를 수행함.
- ▶ 본 연구에서는 무인비행장치 **‘초자율’ 형태의 운항이 가능한 연구로써 관리자와 통신을 최소화하며 이를 통하여 효율적인 운항체계 구축**이 가능할 것으로 예상됨.

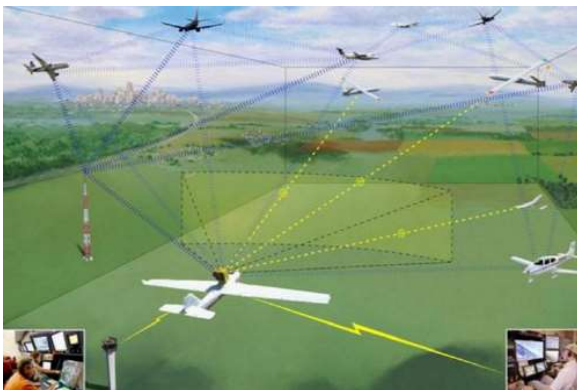


그림 3 UAV 장애물 인지 및 회피 시스템



그림 4 Class G 공역의 무인기의 운항

- ▶ UTM의 주관 연구 기관인 미항공우주연구소(NASA)는 연구 방향 중 무인 비행체에 탑재 되어 운항을 돕는 UAV Support Services (USS)의 세부 항목을 정의 하였음.
- ▶ 그중 상충 및 충돌 회피(De-Confliction and Collision Avoidance)의 기능 중 필요한 기능은 ‘비행 계획 작성(Flight Planning Data)’ 와 ‘비행 정보 (Flight Information Data)’ 그리고 ‘통행 의사결정(Air Traffic Control Technology)’로 구성 되어 있음.
- ▶ 현재 무인기의 안전한 운항을 하기 위해 기존 2D MAP 기반 경로 산출 기능과 장애물 인식 및 정보처리의 기술과 방법은 위 UTM의 세부적인 기능을 따르기 위해서는 한계가 있는 상황임.

- ▶ 이를 극복하기 위해서는 다중 인공지능 에이전트 융합 기술을 적용하여 많은 비행 상황을 무인 비행체가 능동하게 대처할수 있는 **무인기 자율화 6 이상의 단계의 ‘초자율’ 시스템이 필요**함.

Farid Kendoul은 다음과 같이 무인기 시스템 자율화 레벨 (ATRA: Autonomy and Technology Readiness Assessment)을 제시함.

레벨	0	1	2	3	4	5	6	7	8	9	10
자율화 단계	원격조종	자동 비행조종	외부 시스템 독립탐색	결합/사건 적응가능	장애물/사건감지, 경로계획	협력탐색/경로계획	동적 임무계획	협력적 임무계획	상황인식/감지	군중인지/그룹 의사결정	완전 자율

그림 5 무인 항공기의 자율화 단계

현재 국내 무인기 자율 알고리즘은 자율화 10단계 중 2~3 단계인 “자동비행조종/외부시스템 독립비행” 수준으로 **본 과제에서는 자율화 6단계 “실시간 자율임무계획 및 비행”이 가능한 수준으로 세계 최고 수준의 초자율 무인기를 다중 인공지능 에이전트를 탑재하여, 이를 시뮬레이션을 통해 검증**을 목표로 함.

- ▶ 높은 자율성을 가지는 무인기 개발을 위해서는 기계학습 기반의 지능 의사결정 알고리즘 설계 원천기술과 실시간 경로최적화/임무계획 기술, 실시간 장애물 감지 및 회피 기술이 강결합 되어야 함.
- ▶ 초자율 지능임무계획의 구현 및 검증을 위하여 본 연구실에서 보유한 임무설계기술, 의사결정 알고리즘 기술과 고충실도 비선형 6DoF 비행동역학 기술을 동원, 목표를 달성할 예정임.
- ▶ 실시간으로 경로를 생성하고 최적화하기 위한 알고리즘 결합 (ADA*와 MILP 기법을 결합)
- ▶ 동적 임무 계획 및 재계획을 위한 인공지능과 실시간 임무계획 기법의 융합
- ▶ 상황인지 및 시스템 모니터링을 위한 Data Capturing, Data Management, Data Analysis 기술 개발
- ▶ 초자율 무인기/드론에 적용 가능한 임무 재계획 및 의사결정 알고리즘 기술의 개발
- ▶ 본 연구를 통하여 다음과 같은 노력 및 연구가 필요함

● 인공지능경망 학습

- ✓ 인공지능경망의 성능을 높이기 위해서는 방대한 학습 데이터가 필요
- ✓ 일반적인 경우 외의 데이터를 확보하는 것은 매우 어려운 작업이며, 또한 방대한 데이터의 품질 확보 및 관리에도 주의가 필요.

● 항공공학과 컴퓨터공학의 학문 융합

- ✓ 임무계획기법과 지능학습 모델을 이용한 의사결정 모델은 각각 항공공학과 컴퓨터공학의 영역임.
- ✓ 항공공학 분야중 하나인 임무계획 및 이를 위한 의사결정 모델에 적합한 신경망 구조 및 성능 향상 기법 정립을 위해서는 수많은 시행착오를 거쳐야 함.
- ✓ 본 연구팀은 지식의 융합을 위해서 학부 컴퓨터공학 전공 본연구실의 박사과정 학생들을 보

유, 본 연구에 참여, 연구의 시너지를 높여서 초자율 UAV 기술 확보 및 달성이 가능하도록 함.

● **동적 무인기 임무환경 변화**

- ✓ 이륙 후, 항공기가 처한 물리환경은 시시각각 동적으로 변화하며 초자율 무인기의 고신뢰성을 보장하기 위해서는 이를 실시간으로 감지해야 함.
- ✓ 동적인 환경 변화는 항공기 외부의 요인과 내부의 요인이 있으며, 이는 임무수행에 영향을 미침.

● **정보 및 결과의 불확실성**

- ✓ 항공기는 외부 환경에 대한 정보를 전부 알고 있는 상태가 아니며, 사전 입력된 정보는 실시간으로 변화하는 환경을 반영하지 못함.
- ✓ 운항 중 변화하는 상황에 따라 실시간으로 대처가 이루어져야 함.
- ✓ 동일한 상황에서 동일한 행동을 취하여도 다른 결과를 초래할 수 있음.
- ✓ 결과의 불확실성이 높아질수록, 예측을 위해 많은 정보와 데이터분석 기법이 필요함.
- ✓ 임의의 행동은 임무에 악영향을 끼칠 수 있으며, 사전에 이에 대한 예측과 실제도 악영향을 받았을 때 회복할 수 있도록 데이터 분석과 인공지능 기반 임무계획이 필요함.

가) 초자율 무인기/드론 개발을 위한 원천 기술 개발

- ▶ 현재 무인 항공기의 자율화 단계는 2~3단계로서, 사전 입력된 정보에 따라 단일 비행체의 임무 수행 및 외부 시스템 독립 탐색/인지와 이에 대한 대처가 가능한 기본적인 수준임.
- ▶ 초자율 무인기는 현재 수준과 완전자율 무인기 사이의 개념으로서, 외부 지원 없이 단일 비행체가 예기치 못한 상황을 인지(Situation Awareness)하고 이에 대한 대처 및 능동적 임무계획(레벨 6) 능력이 요구됨.
- ▶ 본 연구를 통해 얻어진 결과는 다음과 같이 융합 및 확장하여 완전자율에 보다 근접할 수 있음.
- ✓ 센서 시스템 및 비행 조종 시스템과 결합, 사용자가 제시한 임무의 조건을 받아 능동적으로 조종 및 임무를 완수하는 에이전트 구성이 가능함.
- ✓ 다수의 항공기간의 네트워크 및 협업 시스템을 구축, 다중 무인 항공기가 협업을 통하여 사용자의 개입 없이 임무를 수행함.

나) 다중 인공지능 에이전트 기반 초자율 무인 이동체 기술

- ▶ 현재 많은 연구자들과 회사들은 인공 지능 기능 기술이 자율 이동체에서 가장 핵심이 되는 기술임을 인식하고 있음.
- ▶ 자율 이동체용 인공지능과 알고리즘은 현재 2~3단계에 이르는 무인 항공기 자율화 단계보다 더 나아가 초자율 군집 비행을 가능케 하는 핵심 원천 기술임.
- ▶ 자율 이동체를 위한 인공지능은 실시간 상황인지 및 동적 환경에 따라 스스로 결정을 내릴 수 있는 능력이라고 할 수 있음.
- ▶ 딥 러닝(Deep Learning)은 최근 자각, 임무 계획, 제어 분야에서 뛰어난 결과를 보여주고 있음.
- ▶ 수집된 다양한 데이터로부터 학습하는 딥 러닝은 여러 종류의 무인 자율 인공지능을 개발하는데 적합함.

- ▶ 또한 현재 무인 항공기(UAV)는 보안, 감시, 재해 구조에서 배달 및 창고 관리에 이르기 까지 여러 유형의 사업에 광범위하게 적용되고 있음.
- ▶ 하나의 인공지능 에이전트만으로 동적으로 변화하는 주변 환경을 파악하고 판단을 내리기가 어려움.
- ▶ 다음과 같은 하나의 기능을 수행 할 수 있는 인공지능 에이전트를 개발하고 에이전트들을 연결하여 종합적 상황인지와 최적 경로 파악 및 임무 재설계 등이 가능한 인공지능을 개발 할 수 있음.
- ✓ 시스템 모니터링 인공지능 에이전트
- ✓ 장애물 인식 및 회피 인공지능 에이전트
- ✓ 최적 경로 파악 인공지능 에이전트
- ✓ 종합 상황 파악 및 임무 재설계 알고리즘



그림 6 다중 에이전트 기반 초자율 무인기

나. 연구 개발 대상의 국내·외 현황

1) 자율 무인 이동체 기술의 국내·외 현황

가) 국외 자율 이동체 현황

- ▶ 자율 무인기를 위한 위치 및 자세 파악은 중국 DJI사의 D-RTK 기술로 개활지에서 수평 1cm 이내, 수직 2cm 이내의 정밀도를 보유하고 있음.



그림 7 DJI사 D-RTK 기술

- ▶ 최근 미국 DARPA의 지원으로 MIT의 CSAIL(Computer Science and Artificial

Intelligence Laboratory)에서는 NanoMap이라고 불리는 depth-sensing 기술을 이용한 맵을 생성하고 이를 바탕으로 사용자의 개입 없는 초자율 임무기반 무인기 연구가 다양하게 진행되고 있음 일부 선진국에서 시제품으로 출시되고 있는 상황임.



그림 8 NanoMap 기반 자율 무인기

- ▶ 사용자의 개입 없이 13대의 카메라를 이용하여 가상의 주변 지도를 만들어 장애물 회피 및 목표물을 추적/감시하는 자율 무인기가 출시되어 있음.

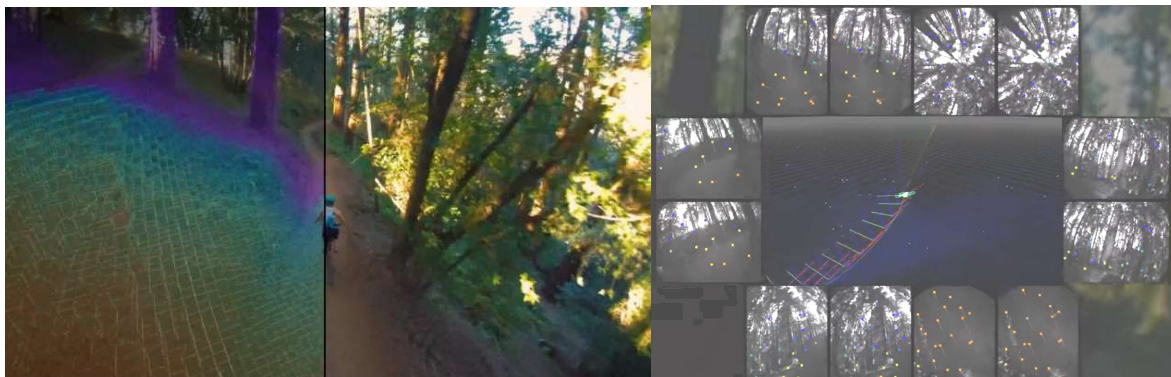


그림 9 가상의 지도(좌) 실제 카메라 화면(우)

그림 10 가상 지도 구축 과정



그림 11 자동차의 자율 주행 모드

- ▶ 자동차의 경우 카메라와 인공지능을 도입하여 상황인지 및 돌발 상황에 대처가 가능한 자율 주행 모드가 탑재된 제품이 출시 되고 있음.

나) 국내 자율 무인기 현황

- ▶ 국내 대부분의 무인 항공기는 무인기의 위치 및 자세 파악을 바탕으로 사전에 입력된 경로를 따라가는 수준으로 3차원 웨이 포인트 설정 후 사전 계산된 경로를 비행하는 수준임.

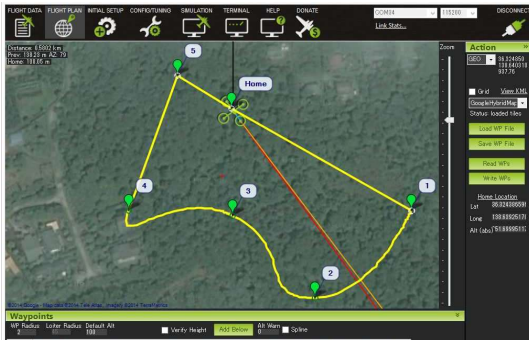


그림 12 웨이포인트 기반 자동 비행

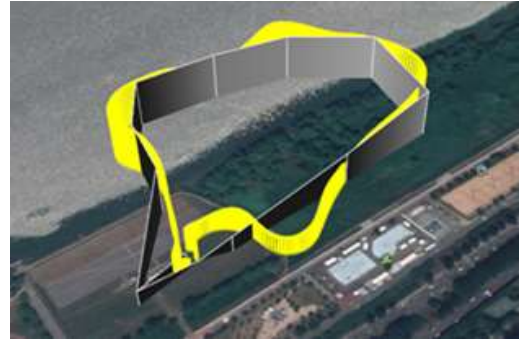


그림 13 웨이포인트 기반 자동 비행 궤적

- ▶ 현재 국내 무인기 자율 알고리즘은 자율화 10단계 중 2~3 단계인 “자동비행조종/외부시스템 독립비행” 수준임.

2) 자율 비행을 위한 핵심 기본 이론

(가) 국내외 자율 비행을 위한 핵심 인공지능 기본 이론 현황

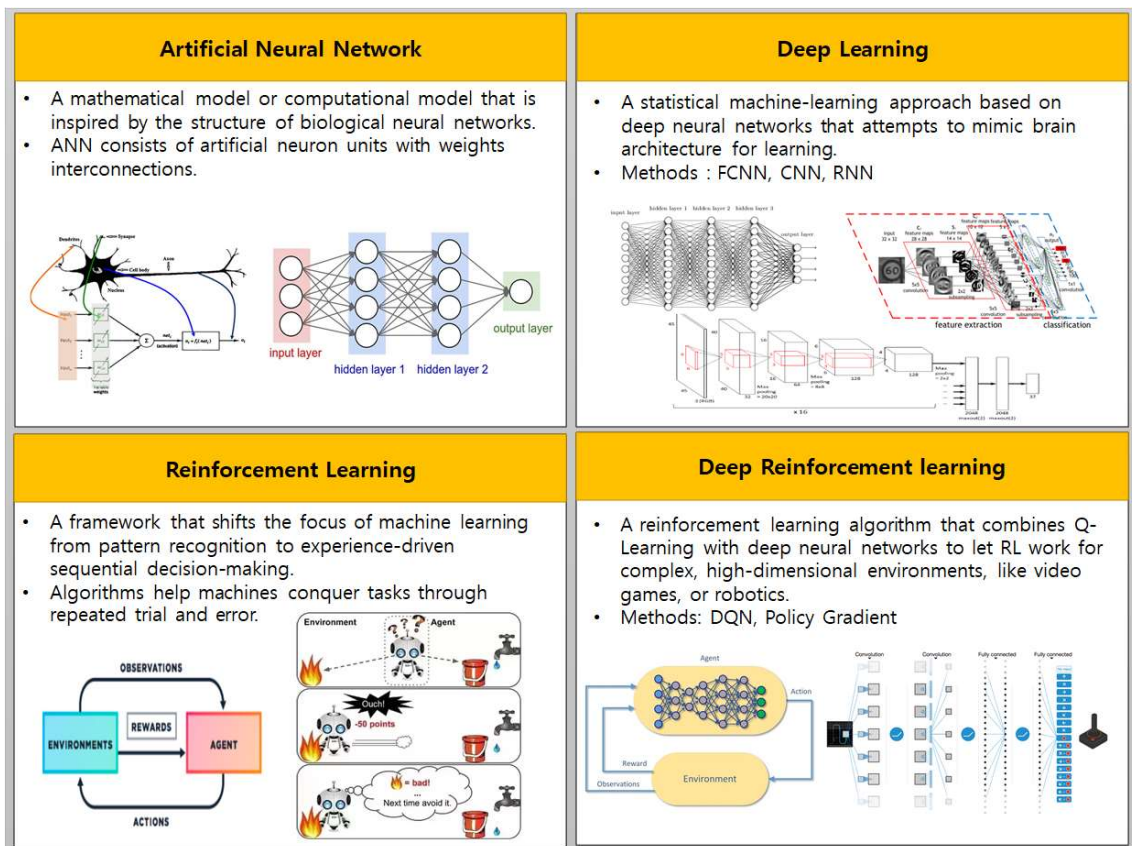


그림 14. 국내외 인공지능 연구 현황

◎ Artificial Neural Network (ANN)

- ▶ 기본적으로 생물학적 신경망의 구조와 기능에 기초한 계산 모델임.

- ▶ ANN의 구조는 정보 흐름에 영향을 받음.
- ▶ 신경망의 변화는 입력과 출력을 기반으로 함.
- ▶ ANN은 비선형 통계 데이터 및 입력과 출력의 복잡한 관계를 다룰 수 있다.
- ▶ 결과적으로 복잡한 패턴을 발견하고, ANN또한 인공지능망의 일종이다.

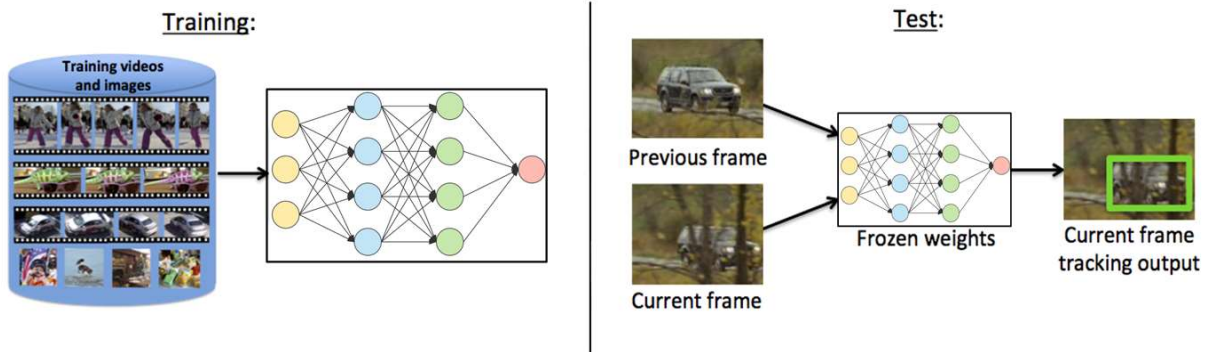


그림 15. Network Learn generic Object Tracking[4]

◎ Deep Learning (DL)

- ▶ 딥러닝의 의사 결정 데이터 처리와 패턴 생성과정은 인간 두뇌의 작동 방법을 모방함.
- ▶ 딥러닝은 머신러닝의 일종으로써, 구조화되지 않았거나 분류되지 않은 데이터를 사용자 개입 없이 자율 학습이 가능함.
- ▶ Deep Neural Learning 또는 Deep Neural Network 라고도 함.

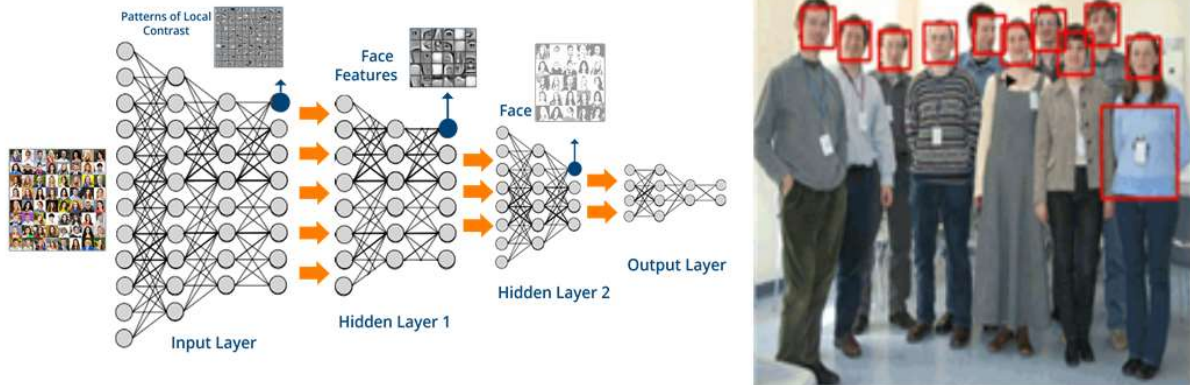


그림 16. CNN을 이용한 안면 인식

◎ Reinforcement learning

- ▶ 기존의 머신 러닝이 보통 패턴을 찾는 것에 중점을 두는 것에 반해 강화 학습은 의사 결정에 중점을 두었으며 AI가 실제 환경에서 학습하고 실행하는 영역에서 더 나은 결과를 보이도록 해줌.
- ▶ 경험 기반 의사 결정의 프레임워크로 수십 년 동안 사용되어 왔지만, 표현과 스케일링 문제로 실제로는 큰 성공을 얻지 못함.
- ▶ 구글 딥마인드에서 개발한 컴퓨터 프로그램 알파고의 최근 성과인 인간 바둑 챔피언과 5게임 경기에서 승리했던 것은 대부분 이 강화 학습에 기인한 것임.

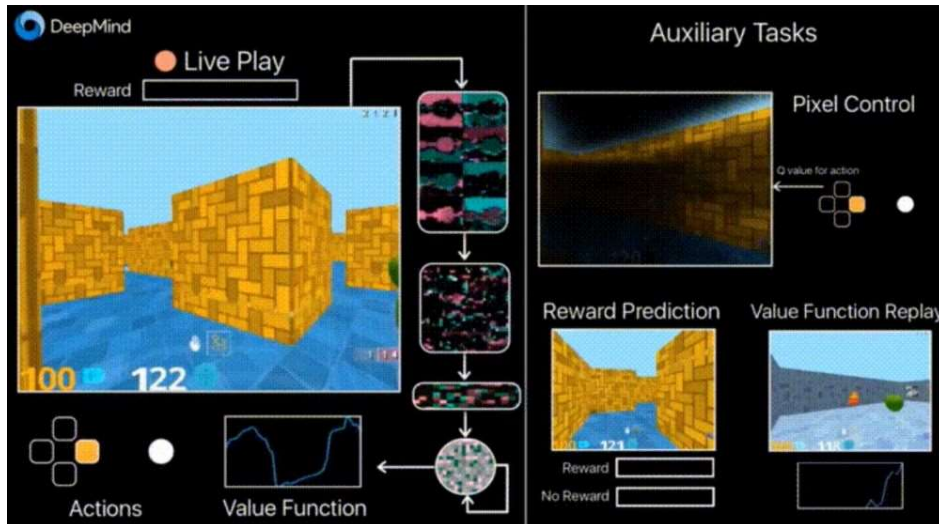


그림 17. Google 딥마인드 학습과 테스트

◎ Deep Reinforcement Learning

- ▶ 관찰, 행동 및 보상을 통해 환경과 상호 작용하여 학습하는 유형의 신경망임.
- ▶ Deep Q Learning과 Policy Gradients의 두 가지 알고리즘이 있음.
- ▶ DQN
 - ✓ 비디오 게임이나 로봇과 같은 복잡하고 고차원적인 환경에서 강화 학습이 작동하도록 하기 위해 Q-Learning과 Deep Neural Networks를 결합한 강화 학습 알고리즘임.
 - ✓ DQN은 Atari나 바둑과 같은 게임 전략 학습에 사용되는데 인간 바둑 챔피언과의 게임에서 승리한 알파고가 사용한 학습 방법임.
- ▶ Policy Gradients
 - ✓ 강화 학습의 목표는 에이전트가 최적의 보상을 얻는 최적의 행동 전략을 찾는 것임.
 - ✓ Policy Gradient 방법의 목표는 정책(policy)을 직접 모델링하고 최적화하는 것임.

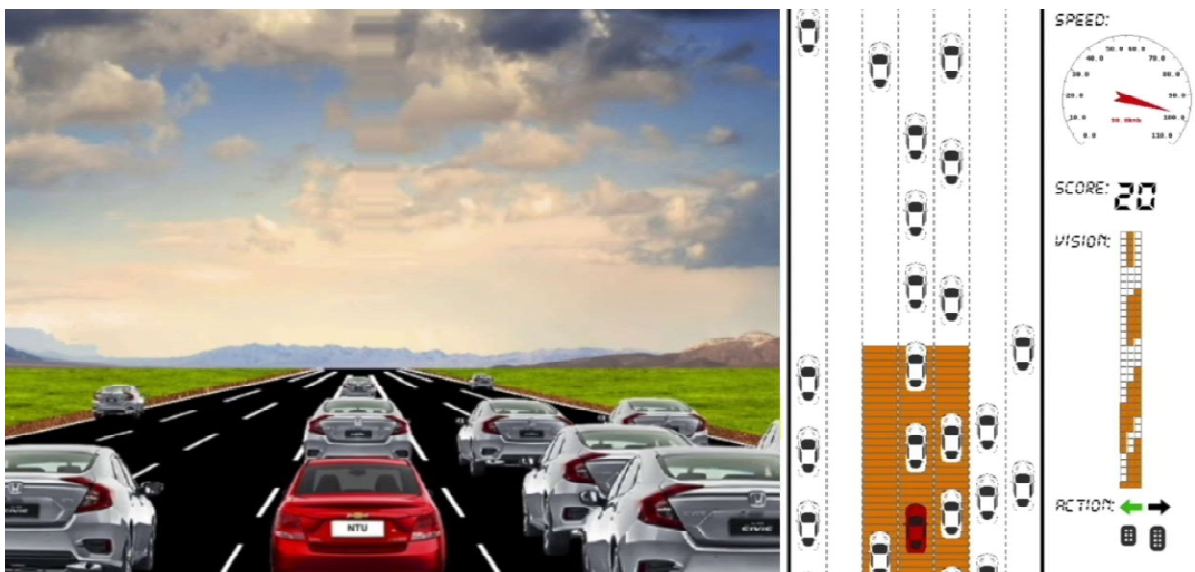


그림 18. DQN을 이용한 자율 주행자동차 개념

3) 경로 계획 및 장애물 회피 기술 현황

- ▶ 경로 계획 알고리즘은 크게 다음과 같은 **두 가지 형태로 분류** 가능함.
 - ✓ 하나는 **최적의 경로를 찾는 것**으로, 여기서 최적 경로란 출발지점으로부터 도착지점까지의 경로의 길이가 가장 짧아 그 경로를 따라 움직였을 때 가장 적은 시간이 걸리는 경로를 의미함.
 - ✓ 다른 하나는 **빠른 계산 속도에 초점**을 맞춘 알고리즘으로 주로 **지역 경로 계획(Local Path Planning)에 적합**한 알고리즘임.
- ▶ 기존의 알고리즘들은 오랜 시간 개발되어 왔기 때문에 포화상태이며, 현재는 완전히 새로운 개념의 알고리즘을 개발하기 보다는 기존의 **두 종류의 알고리즘을 결합하여 전체적인 경로 계획의 성능을 개선하는 연구가 진행**되고 있음.
- ▶ 많은 연구진들이 그래프 기반 또는 샘플링 기반 경로 계획 프로세스를 기계학습(Machine Learning) 알고리즘과 결합한 **하이브리드 계획 방법을 개발**하여 네비게이션 문제에 대한 실시간 임무계획 또는 경로 계획 방법을 제안함.
- ▶ 하이브리드 경로 계획 방법은 경로 계획기를 활용하여 **장애물이 없는 경로를 보다 신속하게 계획하여 전역 경로를 도출**할 수 있으며, **강화 학습(Reinforce Learning)**을 이용하여 UAV가 장애물을 인지하고 회피하는 과정에 대한 학습을 수행하여 **지역적인 경로 계획을 함께 수행할 수 있음**.
- ▶ 다음의 표 9는 기존에 연구된 자율 주행 및 경로 계획에 대한 대표적인 연구 내용들을 정리하였음.

표 9 자율 주행 이동체를 위한 강화학습 기법을 적용한 경로계획 알고리즘 연구

no	연구자	적용 알고리즘	소요 학습시간	경로계획	실험 횟수 /성공률	문제특성
1	Park al. [48]	PRM + DQN	장시간(2일 가량)	지상 로봇의 장애물 회피를 고려한 전역/지역 경로계획(주행 테스트)	-	동적 환경(실험실 규모의 소형맵)
2	Kato et al. [49]	Topological graph + DDQN	장시간(160 시간)	장애물 회피를 고려한 전역/지역 경로계획(시뮬레이션)	19/58%	동적 실내 환경(큰 규모)
3	A. Faust et al. [50]	PRM + AutoRL	장시간(12 일)	드론을 이용한 배송 임무 수행을 위한 전역 경로 계획(비행체 대상)	Run on 3 trails only	복잡한 정적 실내 환경(30×10m의 소규모)
4	A. Faust et al. [51]	PRM + DDPG	빠름(1일 이하)	실내 환경에서 전역 경로 계획 및 장애물 회피 (비행체 대상, 시뮬레이션)	100/54 %	부분적으로 미확인의 대규모 환경(134 × 93m)
5	Chiang et al [52]	AutoRL + RRT	장시간(7일)	장애물 인지에 훈련된 RL를 이용한 경로 계획 (지상 주행 로봇)	10/70 - 80 %	정적 실내 환경(22.7×18.0m의 소형 맵에서 훈련, 46.1 × 49.5m 맵에서 테스트)
6	Khalil et al.[53]	TSP + DQN	장시간	경로점 최적화(Waypoint optimization)	-	정적 소형 맵
7	Francis et al. [54]	PRM + RL	빠름(1일 이하)	소음과 다수의 장애물이 존재하는 영역에서의 전역/지역 경로 계획(지상 로봇 주행테스트)	20 /82 %	동적 환경(실내, 183×66m)

- ▶ 박정준 등의 2007년 연구(참고문헌 53)에서는 정적환경에서는 좋은 성능을 발휘하지만 동적 환경에서는 성능이 저하되는 확률적 로드맵 경로 계획 (PRM: Probabilistic Road Map)과 강화 학습을 결합한 새로운 효율적인 실시간 하이브리드 경로 계획 방법을 제시하였음.
- ✓ 해당 연구는 **불확실한 동적 환경에서의 PRM 전역 경로 계획기의 적용성 향상**을 위한 연구를 진행하였음
- ✓ 단, 동적 환경 내에서의 비행체와 같은 3차원 환경이 아닌 2차원의 **지상 로봇 경로 계획**에 초점을 맞춤.

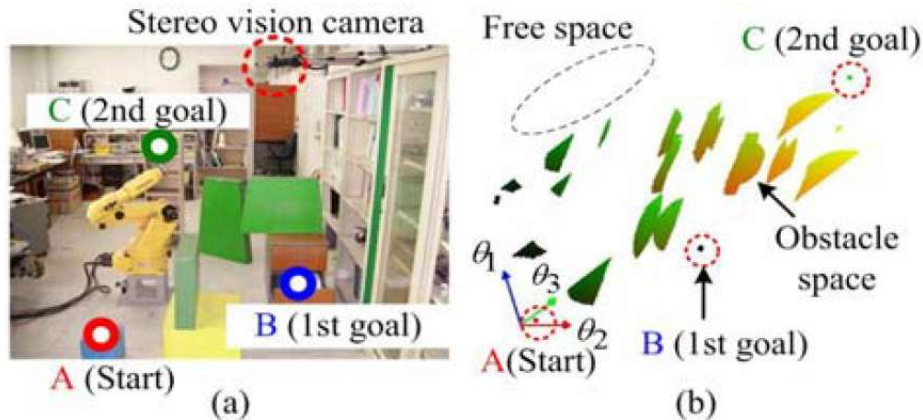


그림 19 Constructed environment for first experiment: (a) its workspace and (b) its configuration space.(박정준, 2007)

- ▶ Kato 등이 발표한 2017년의 연구(참고문헌 52)에서는 사람들로 혼잡한 환경에서 인간 교통 상황에 대한 적응형 로봇 네비게이션을 목표로 하는 시스템을 제안함.
- ✓ 해당 연구에서는 복잡한 동적 환경 상에서 운행체 속도에 대한 제한을 고려한 경로 계획을 수행하였음.
- ✓ 지역(Local) 네비게이션을 위해서는 심층강화학습 기법을, 전역(Global) 네비게이션으로는 전역 위상 지도(Topological Map)의 두가지 기법을 결합하는 방식을 적용.
- ▶ Faust 등이 수행한 2017년의 연구는 **드론을 이용한 공중 배달 임무 계획**을 위해 **PRM과 강화 학습을 결합**한 하이브리드 경로 계획 방법을 제안하였음.
- ✓ 해당 연구에서 강화 학습 에이전트는 가치 함수(Value Function)의 근사화와 훈련 영역을 넘어선 일반화를 위해 특별히 고안된 특징 벡터를 사용하는 장애물이 없는 환경에서 학습을 시작함.
- ✓ 상태(State)를 입력으로 사용하며, 이산 행동(Discrete Action), 제어 명령을 출력으로 하는 강화학습 에이전트를 고려하여 샘플링 기반 행동 계획기가 충돌 없는 경로를 계획하는 정적인 장애물이 존재하는 환경에서의 임무로 확장함.
- ✓ 본 연구는 가정용 드론이 30m × 10m × 5m 크기의 카페 내에서 카운터와 각 테이블 간을 이동한다는 상정하에 시뮬레이션을 통한 벤치마킹의 수행과 실제 검증을 위한 시험을 아래 그림에서 볼 수 있는 2개의 테스트 환경에서 수행하였음.

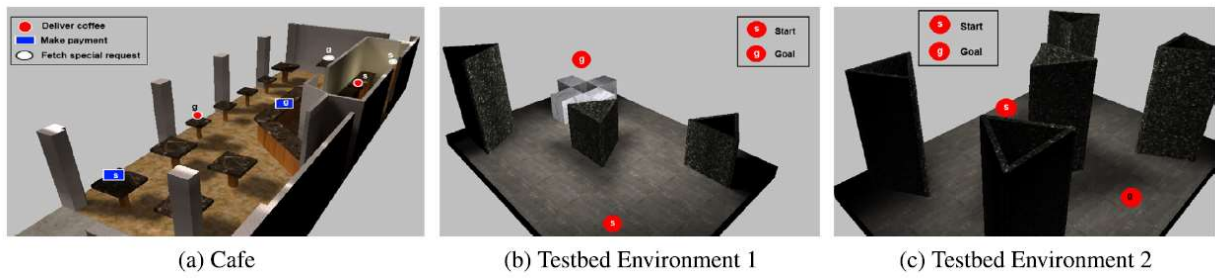


그림 20 Benchmark environments studied in simulation(a) and experimentally(b-c)(Faust et al, 2017)

- ▶ Faust 등의 2018년 연구에서는 2017년의 연구를 PRM 방법의 지역 경로 계획기로 DDPG(Deep Deterministic Policy Gradient)를 사용하는 PRM-RL 실내 계획 방법으로 확장함.
- ✓ 센서를 통해 맵을 생성하며, 2017년의 선행 연구와 동일한 방법으로 속도를 제어함.
- ✓ 에이전트의 보상(Reward)과 네트워크 구조를 모델의 성능을 향상하는 방향으로 조정하는 AutoRL을 통해 훈련함.

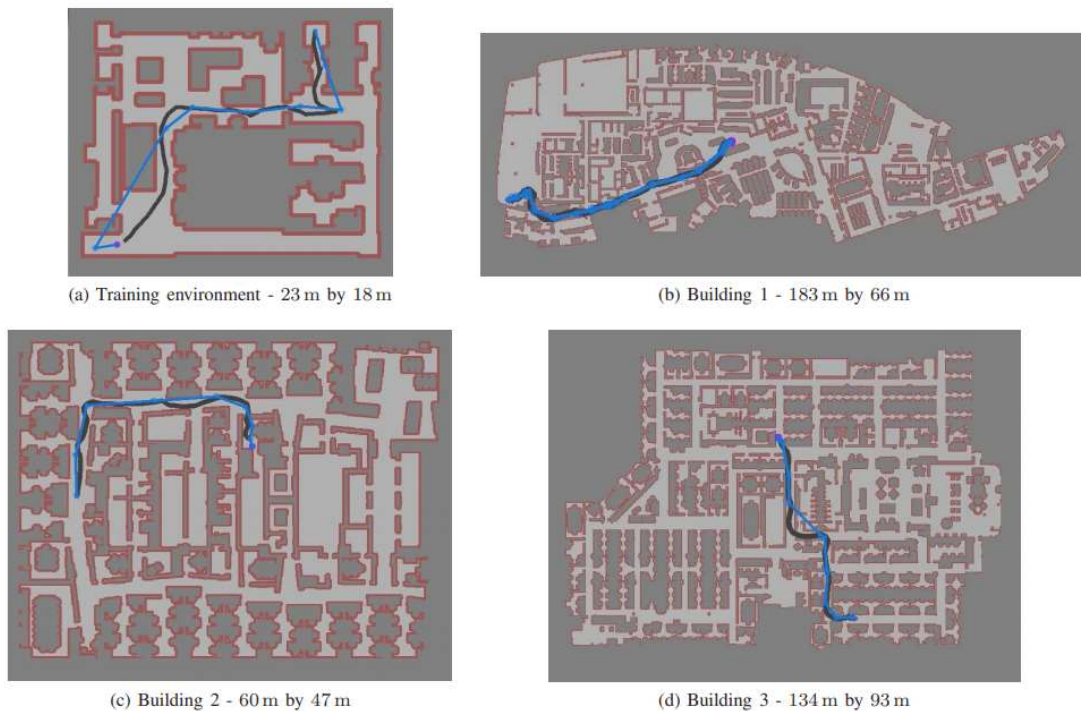
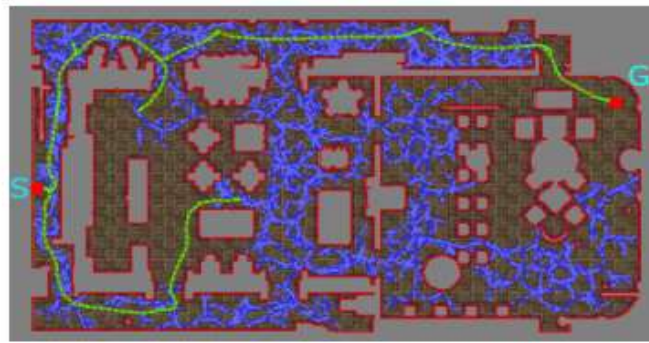


그림 21 The environment used for the indoor navigation tasks in researches by Faust et al. (2018) and Francis et al. (2020).

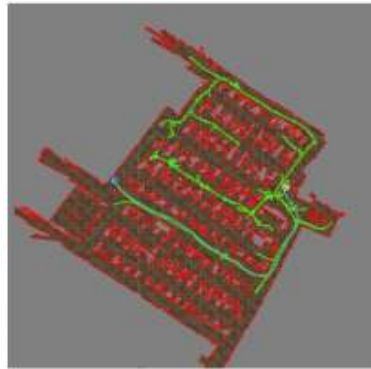
- ▶ Chiang 등의 2019년 연구에서는 RRT(Randomized Tree)와 강화학습을 통합한 효율적인 Kinodynamic Motion Planner를 제안함.
- ✓ 해당 연구는 **지상의 2차원 환경에서 주행하는 로봇의 장거리 경로 계획 문제에 대해 효율적인 해결책을 도출**하는 것을 목적으로 하였음.



(a) RL-RRT and SST in Map 1 (46.1 x 49.5 m)



(b) The Fetch robot



(c) Trajectory execution of Fetch in Map 2 (46.1 x 49.5 m)

그림 22 RL-RRT map and the robot in a research by Chiang et al. (2020).

- ▶ Khalil 등이 2019년에 수행한 연구는 그래프를 기반으로 조합 최적화(Combinatorial Optimization) 문제를 해결하기 위해 임베딩 네트워크(Embedding Network) 학습에 메타 휴리스틱을 적용함.
 - ✓ Minimum Vertex Cover, Maximum Cut, Travelling Salesman Problem(TSP)와 같은 문제에 대한 학습된 탐욕 정책 행동(Greedy Policy Behavior)을 사용함.
 - ✓ 네트워크 내의 솔루션 탐색은 포인터 네트워크 기반 신경 조합 최적화(Neural Combinatorial Optimization)의 경우보다 효율적임.
- ▶ Francis 등이 2020년에 수행한 연구에서는 실내 환경에서 지상 로봇의 장거리 네비게이션을 위한 PRM과 강화 학습(RL)을 조합한 하이브리드 경로 계획 방법을 제안함.
 - ✓ PRM-RL 방법에서 강화 학습 에이전트는 장거리 환경 구조에 독립적인 시스템 역학 및 센서 노이즈를 통합하여 로컬 지점 간 임무를 학습함.
- ▶ 여러 연구자들의 상기의 연구 내용으로부터 알 수 있듯이 **컴퓨터 시뮬레이션 또는 실제 주행 및 비행시험을 통한 전역/지역 경로 계획에 대한 연구가 수행**되고 있으나 **그 대부분은 2차원의 지상 주행체를 대상으로** 하고 있음.
- ▶ 항공운송 드론과 같은 비행체에 대한 경로 계획도 수행되고 있으나 **실내와 같은 작고 정적 환경에서의 운용**을 목적으로 이에 대한 경로를 계획에 초점을 맞춘 것이 대부분 임.

다. 연구 개발의 목표 및 내용

1) 연구 개발의 최종 목표 및 성과물

구분	내용
최종 목표	초자율 무인기를 위한 다중 인공지능 에이전트 기반 실시간 임무계획 알고리즘 개발
세부 목표	- 시스템 모니터링 인공지능 에이전트 알고리즘 개발 - 장애물 인식 및 회피 인공지능 에이전트 알고리즘 개발 - 최적 경로 파악 인공지능 에이전트 알고리즘 개발 - 에이전트 융합 및 임무 재설계 알고리즘 개발 - 인공지능을 도입한 초 자율 무인기 시뮬레이션 기술

가) 최종성과물의 기술성숙도 설정 근거

구분		연구착수	연구종료	단계별 설명	
기초연구 단계	TRL 1	■	☐	TRL 1	이론적 연구단계
	TRL 2	☐	☐	TRL 2	기술적 개념정립 단계
실험 단계	TRL 3	☐	■	TRL 3	실험적 증명 단계
	TRL 4	☐	☐	TRL 4	실험실 성능평가 단계
시작품 단계	TRL 5	☐	☐	TRL 5	유사환경 시험 단계
	TRL 6	☐	☐	TRL 6	유사환경 기술시험 단계
실용화 단계	TRL 7	☐	☐	TRL 7	실제환경 시작품 데모 단계
	TRL 8	☐	☐	TRL 8	시제품 설치 및 성능시험 단계
사업화 단계	TRL 9	☐	☐	TRL 9	현장적용/사업화 단계

최종 성과물	기술성숙도(TRL)		설정 근거
	연구시작시	연구종료시	
초자율 무인기 시뮬레이션	TRL 1	TRL 3	-(연구시작시) 보유 기술 본 연구진은 국내 무인 비행체의 자율화 관련 기술을 선행 조사 및 연구하였으며 이를 바탕으로 자율화 단계 ATRA 6 단계 이상의 무인 비행체 기술 수준을 TRL 1로 선정하였음. • 동적 임무 계획 ▷ 현재 상용화된 무인기의 임무 계획은 조종사가 지상에서 수동으로 경로를 입력해주는 방식 사용함. ▷ 이러한 단점을 극복하고자 무인 비행체의 GSC를 개발하는 업체에서 DEM과 같은 수치 표고 데이터가 탑재된 3D GSC를 개발하여 자동으로 경로를 찾는 소프트웨어를 개발중임. ▷ 그러나 경로에 따른 비행 리스크를 실시간으로 계산하여 무인기 임무에 반영하는 기술은 다양한 환경변수 & 비행 변수 필요 함. ▷ 이를 실시간으로 계산하기 위해서는 무인기에 탑재된 임베디드 컴퓨터에서 계산하여 산출하는 방식이 아닌 인공지능의 개념 적용이 필요 함. ▷ 기존 무인기의 동적 임무 계획중 장애물 탐지 및 회피 기능의 경우 CV (Computer Vision) 중점으로 연구 방향이 맞춰졌으나 느린 계산시간 높은 하드웨어의 장벽으로 인해 인공지능 개념을 활용한 접근방식으로 변경 됨. ▷ 현재 많은 무인기 업체 및 이에 관한 연구소가 DL (Deep Learning) / ML (Machine Learning) 과 같은 인공지능 개념을 무인기에 도입하고자 노력 중. -(연구종료시) 최종 목표 인공지능 에이전트 개발 및 융합하여 다중 인공지능 에이전트 기반 초자율 무인기를 컴퓨터 시뮬레이션을 통하여 검증 예정임에 따라 TRL 3로 선정 하였음. • 시스템 모니터링 인공지능 에이전트 개발 ▷ 본인의 상태를 파악 할 수 있는 인공지능 에이전트 • 장애물 인식 및 회피 인공지능 에이전트 개발 ▷ 딥 러닝(Deep Learning)을 통하여 장애물을 인식하고 안전거리 확보를 위한 인공지능 에이전트 개발 • 최적 경로 파악 인공지능 에이전트 개발 ▷ 동적 환경 변화에 적응하여 실시간 최적 경로를 계산 할 수 있는 인공지능 에이전트 개발 • 종합 상황 파악 및 임무 재설계 알고리즘 개발 ▷ 본인의 상태 / 장애물 인식 / 최적경로 에이전트로부터 얻은 정보를 이용하여 종합 상황 파악 및 임무 재설계 인공지능 에이전트 • 개발된 인공지능을 탑재한 초자율 무인기 시뮬레이션

2) 연차별 연구 개발 목표 및 내용

가) 1차 연도

(1) 개발 목표

- 주관연구기관 : 건국대학교

- 시뮬레이션을 위한 무인비행체의 비행 제어 연구
- 임무계획 기법 연구 및 AI 이론 연구
- 장애물 인식 및 회피 인공지능 에이전트 개발
 - ▶ 상황 인식을 위한 CNN 기반 인공지능 기술 연구
 - ▶ 회피를 위한 RNN (Recurrent Neural Network) 기술 연구
- 최적 경로 파악 인공지능 에이전트 개발
 - ▶ Anytime Dynamic A-star (ADA*)
 - ▶ Mixed Integer Linear Programming (MILP)

(2) 개발 내용 및 범위

- 주관연구기관 : 건국대학교

(가) 시뮬레이션을 위한 무인비행체의 비행 제어 연구

- ▶ 아래 그림과 같이 Multicopter의 제어는 각각의 모터의 추력 변화로부터 얻어짐

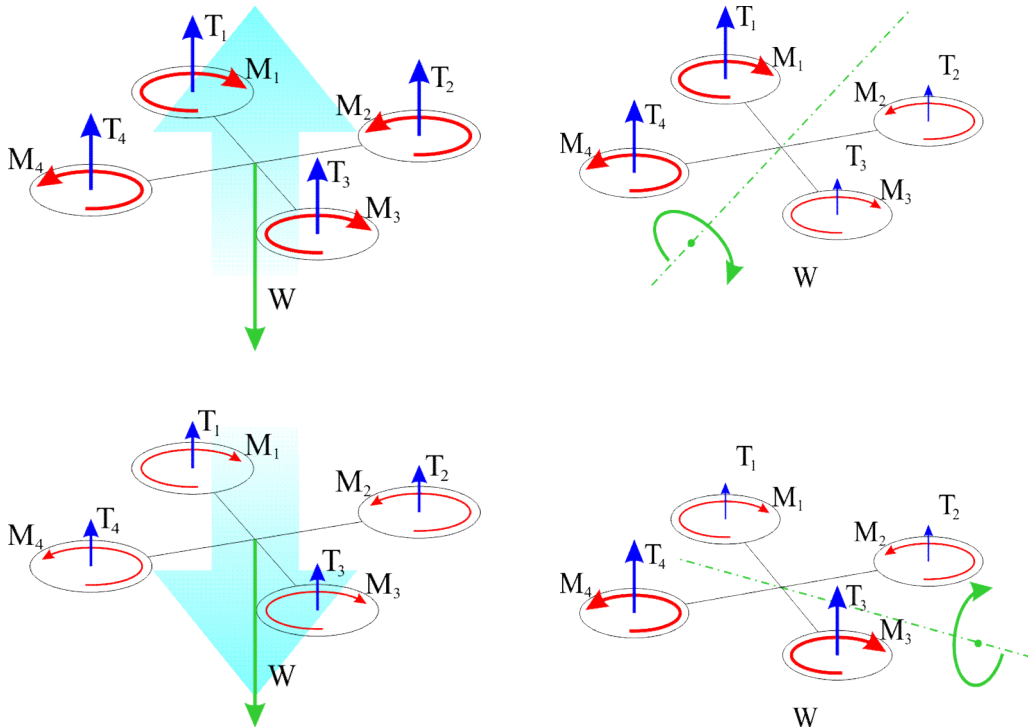


그림 24 멀티 콥터의 기본적인 움직임

- ▶ 각각의 모션들은 아래 표와 같은 회전수 증감으로 얻어질 수 있음

표 1 각각의 모션들을 얻기 위한 회전수 증감

Force / Moment	Ω_1	Ω_2	Ω_3	Ω_4
Roll Moment		-		+
Pitch Moment	+		-	
Yaw Moment	+	-	+	-
Vertical Thrust	+	+	+	+

◎ Simple PID Controller:

- ▶ 아래 그림은 Multicopter의 간단한 PID 제어기의 구조를 나타냄

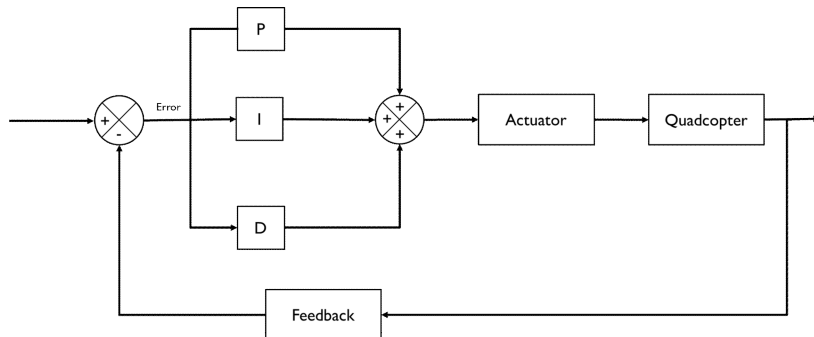


그림 25 PID 제어기의 구조

- ▶ 각각의 gain 값들은 다음 표와 같은 효과를 가지고 있음

표 2 Response of the different parameters of the PID gain

Controller	Rise Time	Overshoot	Setting Time	Steady-state error	Stability
K_p	Decreases	Increases	Small change	Decreases	Degrade
K_i	Decreases	Increases	Increases	Eliminate	Degrade
K_d	Small change	Decreases	Decreases	No Effect in Theory	Improve if small

(나) 임무 계획 및 인공지능 이론 연구를 통한 최적 경로 파악 인공지능 에이전트 개발

◎ Anytime Dynamic A-star (ADA*)

- ▶ 실시간 임무 계획을 위해서 비행체는 반드시 주어진 가용 시간 내에 최적의 솔루션을 찾아야 함
- ▶ Anytime 알고리즘은 제한된 시간동안 최고의 계획을 찾는 알고리즘으로, 환경요소에 대한 정보가 누락되거나 변하는 경우에도 경로 계획이 가능
- ✓ 하지만 Anytime 알고리즘은 문제가 복잡해질 경우, 주어진 가용 시간 내에 최적의 결론을

찾지 못할 수도 있음

- ✓ 본 알고리즘은 D*, D* Lite와 같이 이전에 쓰이던 효율적인 알고리즘과 ARA*와 같은 복잡한 알고리즘을 환경이 변화하는 동적 환경에서도 사용할 수 있도록 확장

◎ Mixed Integer Linear Programming (MILP)

- ▶ 최적화된 솔루션을 도출할 때, 의사 결정 변수 중 일부를 정수형으로 사용 가능
- ▶ 정수형 변수를 사용할 경우, 최적화 문제를 적용할 수 있는 범위가 넓어짐
- ▶ 하지만, 정수 변수가 매우 많을 경우(수백 개 이상) 연산량이 늘어나 최적화된 결론을 내지 못함
- ▶ 일반적으로 선형적인 경로에 대하여 사용하지만, 경로를 분할/선형화 하여 곡선 경로 최적화에도 사용 가능
- ▶ 이진(Binary) 결정 변수를 사용하면, 장애물의 “왼쪽” 또는 “오른쪽” 경로를 선택할지, 또는 계획 문제에 포함될 대상에 비행체를 개별적으로 포함할지를 선택할 수 있음
- ▶ 경로의 길이와 피해야할 장애물의 수가 증가하면 계산 시간이 빠르게 증가하므로, 실시간으로 수행하기 어려움
- ▶ MILP와 ADA*를 함께 사용하면, 장애물이 있는 동적 상황에 대해서도 적용 가능

◎ improved Anytime Dynamic A* (iADA*)

- ▶ 이 알고리즘은 동적이고 복잡한 환경에서 안정적이고 효율적인 경로를 제공하는 ADA* 알고리즘의 확장 버전임.
- ▶ 결과값으로 시작 상태와 최종 상태 사이의 최소 비용 경로 또한 제공 가능.

(다) 장애물 인식 및 회피 인공지능 에이전트 개발

◎ 상황 인식을 위한 CNN (Convolutional Neural Network)기반 인공지능 기술 연구

- ▶ CNN은 사진을 기반으로 얼굴 인식, 물체 인식, 교통 신호 인식 등 이미지 기반의 물체 및 상황을 인식하는데 효과적임
- ▶ 따라서 ConvNets은 오늘날 대부분의 기계 학습 연구자에게 중요함.

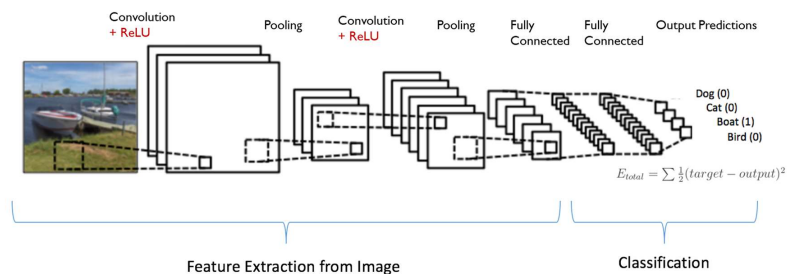


그림 26 CNN의 개념도

- ▶ 시스템에서 새로운 입력을 추가하고 보상 기능을 업데이트할 때 새 네트워크를 재설계, 재교육하지 않고 UAV가 장애물 회피를 수행할 수 있도록 UAV용 DQN 에이전트를 학습시킴.
- ▶ 입력신호는 AirSim에서 지원하는 위치 정보와 심층 카메라 데이터로 분리되고 난 다음에 학습된 상태로 주어짐.

- ▶ 장애물 회피 인공지능 에이전트를 훈련시키는 과정의 속도는 이익 값(Advantage Values)에 의해 결정되며 이 이익 값은 Hyper Parameter 및 최적경로 파악 인공지능에 의해 결정됨.
- ▶ 최종 값을 얻은 후에는 최대 이익 값에 해당하는 행동을 선택해 최종 결정을 내릴 수 있음.

나) 2차 연도

(1) 개발 목표

- 주관연구기관 : 건국대학교
- 장애물 인식 및 회피 인공지능 에이전트 개발 (계속)
- 시스템 모니터링 인공지능 에이전트 개발
- 임무 재설계 알고리즘 개발
- 초자율 무인기 시뮬레이션 통한 검증

(2) 개발 내용 및 범위

- 주관연구기관 : 건국대학교

(가) 장애물 인식 및 회피 인공지능 에이전트 학습 (계속)

- ▶ 미션 플래너에서 작업 선택, 스케줄링, 리스크 평가 및 상위 의사 결정을 수행할 특정 에이전트를 정의함.
- ▶ 비행체의 상태 매개변수를 고려하는 네트워크를 설계하고 DQN을 사용하는 장애물 회피 네트워크와 결합함.
- ▶ 미션 플래너 에이전트는 장애물 회피 네트워크, 시스템 모니터링, 경로 플래너에서 얻은 결과에 기반해 행동을 선택함.
- ▶ 산출된 결과는 시뮬레이터에 보내져 시각화됨.

(나) 시스템 모니터링 인공지능 에이전트 개발

- ▶ 무인기 스스로의 시스템을 인지하고 임무에 적합한지를 판단하는 에이전트임.
- ▶ 파손 정도, 주변 온도, 배터리 잔량, 배터리 소모율, 잔여 비행가능 시간등을 모니터링 하고 임무 지속 여부를 판단함
- ▶ 이를 위하여 다양한 센서 기술과 임무 적합성 판단 에이전트 개발이 필요함

(다) 임무 재설계 알고리즘 개발

- ▶ 한 번의 의사 결정을 내리는 것은 쉬운 일이지만, 실시간으로 임무를 계획한다는 것은 임무가 끝날 때까지의 복수의 행동에 대하여 실시하여야 하므로 보다 높은 난이도를 가짐
- ▶ 환경의 불확실성과 동적인 변화는 한 번의 의사결정 조차도 어렵게 만듦
- ▶ 이러한 문제를 해결하기 위하여 동적인 환경변화 상에서 의사결정을 내리기 위한 프레임워크 구조를 구성, 의사 결정이 일정한 형식을 갖도록 함.



그림 27 실시간 임무계획 알고리즘 개요

- ▶ 실외 환경
- ▶ 건물, 길, 다리 등

(2) 다중 인공지능 에이전트 평가 기준



그림 30 Neighborhood environment with 1000 x 1000 map size

◎ 강건성

- ▶ 1000x1000 사이즈의 주변환경에서의 200번의 실험 케이스에서 2% 미만의 실패율

◎ 계산 시간

- ▶ 장애물 회피 반응을 포함한 비행경로 계산등을 포함한 최종 의사결정 속도 < 7초
- ▶ 장애물 회피 반응 속도 < 1/30 초
 - ✓ 1000x1000 사이즈의 주변환경에서의 200번의 실험 케이스에서 정적, 동적 장애물에 대한 실시간성 확인
- ▶ UAV 초기 비행 경로 계산 < 120 초
 - ✓ 1000x1000 사이즈의 주변환경에서의 200번의 실험 케이스에서 시간 측정
- ▶ 실시간 새로운 최적 경로 생성 < 5 초

나) 정량적 연구성과 목표

구분		1차년도 (2019)	2차년도 (2020)	계
국외논문 게재	SCI	2	2	4
	비SCI	-	-	-
	계	2	2	4
국내논문 게재	SCI	1	1	2
	비SCI	3	4	7
	계	4	5	9
특허출원	국내	1	2	3
	국외	-	-	-
	계	1	2	3
특허등록	국내	-	1	1
	국외	-	-	-
	계	-	1	1
인력양성	석사	3	3	6
	박사	1	1	2
	계	4	4	8
등록·기탁 대상 성과물	소프트웨어	1	1	2

2. 연구수행내용 및 성과

가. 연구 내용

1) 연구 개발 추진체계 및 추진계획

가) 연구 개발 추진 계획

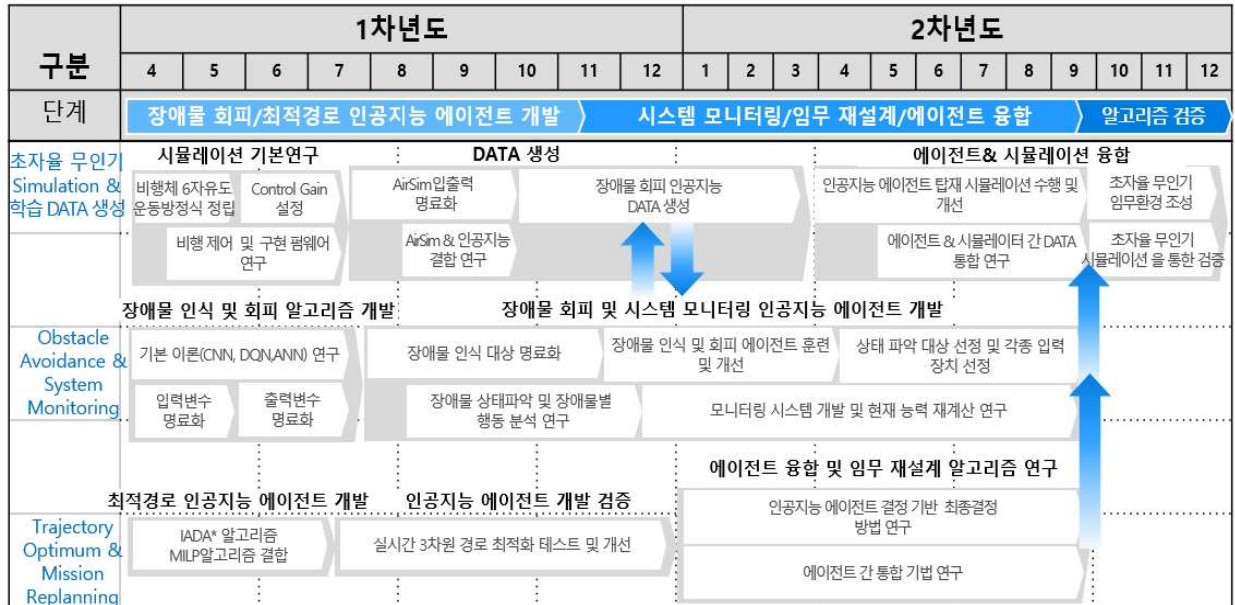
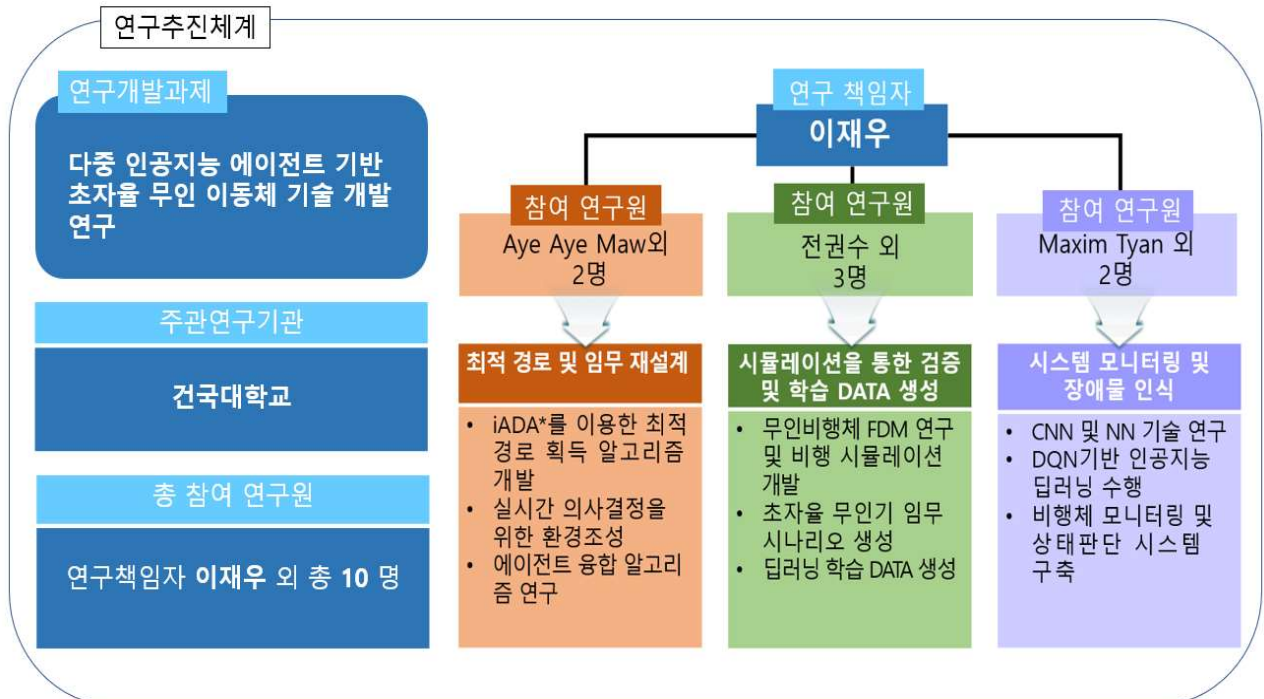


그림 31 연차별 연구 개발 추진 계획표

나) 연구 개발 추진 체계



다) 연구개발 추진 일정

TASK	2차년도																							
	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12			
시뮬레이션 기본연구																								
Multicopter 기반 FDM 구축 연구																								
비행제어 연구																								
HILS 구현을 위한 비행제어 펌웨어 연구(PX4 or APM)																								
Control Gain 값 설정 및 최적화																								
AirSim의 입력값/출력값 명료화																								
AirSim과 인공지능 결합을 위한 OpenAIGym 연구																								
장애물 회피 인공지능을 위한 데이터 생성																								
개발된 에이전트와 시뮬레이터 통합 방법 연구																								
개발된 에이전트 탑재 시뮬레이션 수행 및 개선																								
장애물 인식 및 회피 인공지능 에이전트 개발																								
기본 이론 연구 (CNN, DQN, ANN etc)																								
인공지능 에이전트 입력변수 명료화																								
인공지능 에이전트 출력변수 명료화																								
장애물 인식 대상 명료화																								
장애물 상태 파악 및 장애물별 행동 분석 연구																								
장애물 인식 및 회피 에이전트 훈련 및 개선																								
경로 최적화 인공지능 에이전트 개발																								
기본 이론 연구 (ADA*, MILP)																								
기본 이론 개선 및 적용 (i-ADA*)																								
개선 이론 실시간성 개선 및 검증																								
실시간 3차원 경로 최적화 테스트 및 개선																								
시스템 모니터링 에이전트 개발																								
상태 파악 대상 선정 및 확인																								
모니터링을 위한 각종 입력 장치 선정																								
파손 정도 모니터링 시스템 개발																								
시스템 오류로 인한 비행체의 현재 능력 재계산 연구																								
임무 재설계 알고리즘 개발																								
에이전트들의 결정에 기반한 임무 재설계 방법 연구																								
임무 재설계 알고리즘 개선 및 검증																								
인공지능 에이전트 통합 및 검증																								
에이전트 통합 기법 연구																								
다른 출력을 가지는 에이전트 통합 연구																								
에이전트 결합 및 개선																								

그림 33 추진 일정

- ▶ 다중 에이전트 기반 시스템을 이용한 실시간 임무계획 기술을 정의하기 위해 다음과 같은 목표를 제시함.
- ◎ **전역 경로 계획 및 재 계획 에이전트**
 - ▶ 본 에이전트의 주 역할은 장애물이 존재하지 않는 최단 거리의 경로를 계획하는 것임.
 - ▶ 에이전트를 통해 계획된 경로는 거리 및 비행시간을 포함함.
- ◎ **장애물 탐지 및 회피 에이전트(학습 기반 지역 경로 계획)**
 - ▶ 본 에이전트는 실시간으로 장애물을 회피해 전역 경로 계획을 보완하기 위한 지역 경로 계획으로써 작용함.
 - ▶ 다가오는 장애물을 탐지하고 회피해 최종 임무 목표지점에 도달할 수 있도록 전역 경로 계획 에이전트가 경로를 재계획할 수 있는 안전한 위치로 이동하도록 함.
- ◎ **시스템 모니터링 에이전트**
 - ▶ 본 에이전트는 실시간으로 비행체의 상태를 모니터링함.
 - ▶ 시스템을 관찰하여 비행체에 오류 또는 고장이 발생한 경우 긴급 의사 결정을 수행함.

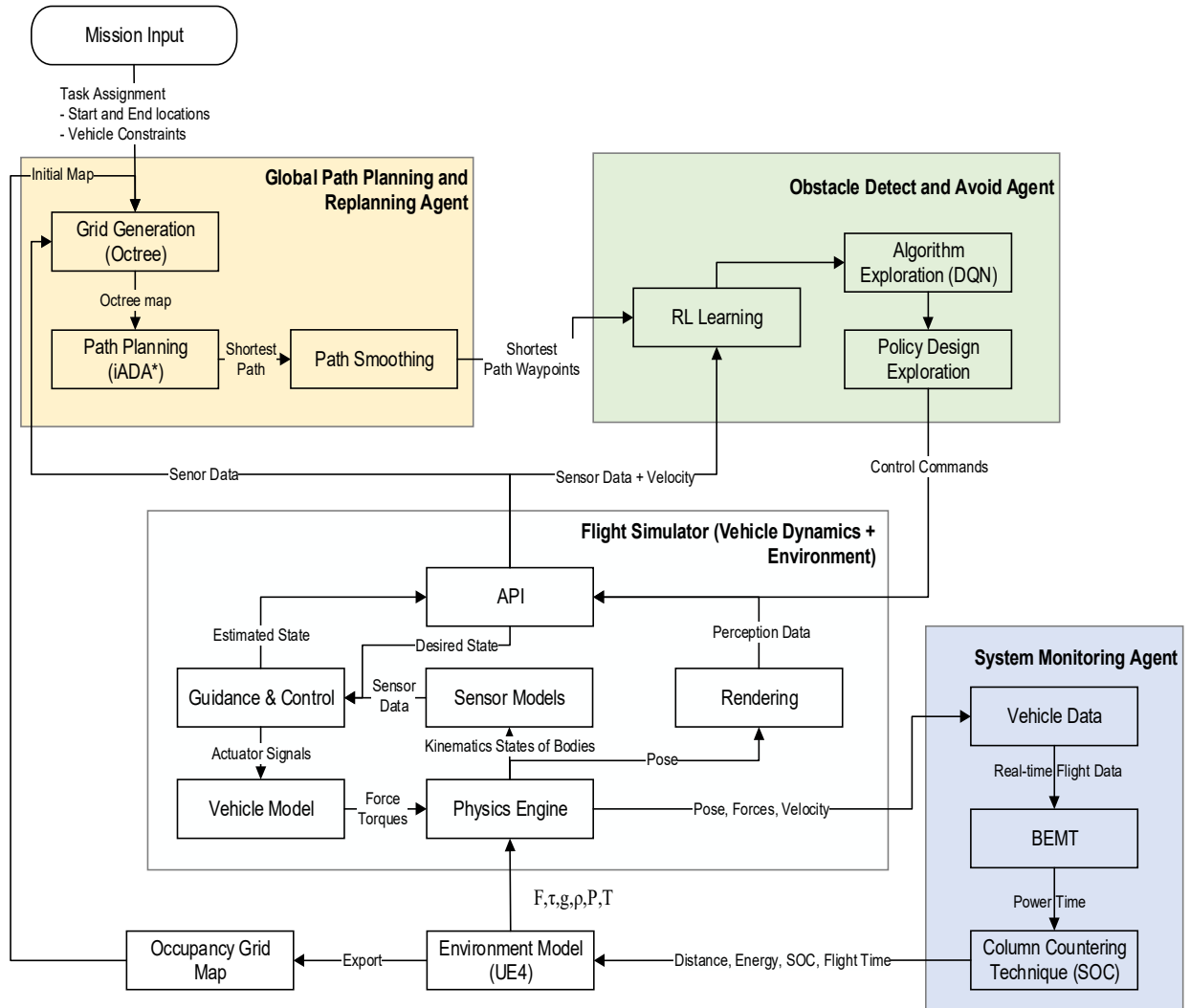


그림 34 다중 에이전트 기반 실시간 임무계획 시스템 아키텍처

2) 1차년도 연구 개발 내용

가) 시뮬레이션 기본연구

(1) 멀티콥터 기반 FDM 구축 연구

- ▶ 고정익 FDM과 유사하게, 멀티콥터 FDM은 CLME (선 운동량 방정식 보존), CAME (각운동량 방정식 보존) 및 중력 방정식과 같은 운동 방정식으로 구성.
- ▶ 데이터베이스에는 공기 역학적 힘과 모멘트, 중량 및 균형에 관련된 데이터를 포함되어 있으며 이는 멀티콥터 동역학 모델 방정식의 입력으로 적용.
- ▶ FDM에 대한 제어 입력은 로터 RPM으로, 액추에이터 동역학 방정식을 통해 필요한 토크와 스로틀을 계산하는 데 사용.
- ▶ FDM 출력은 선형 위치(linear position) 및 각도 위치(angular position)와 같은 항공기의 상태로 비행 경로의 생성 및 멀티콥터 컨트롤러 모듈의 응답 피드백으로 사용 됨.

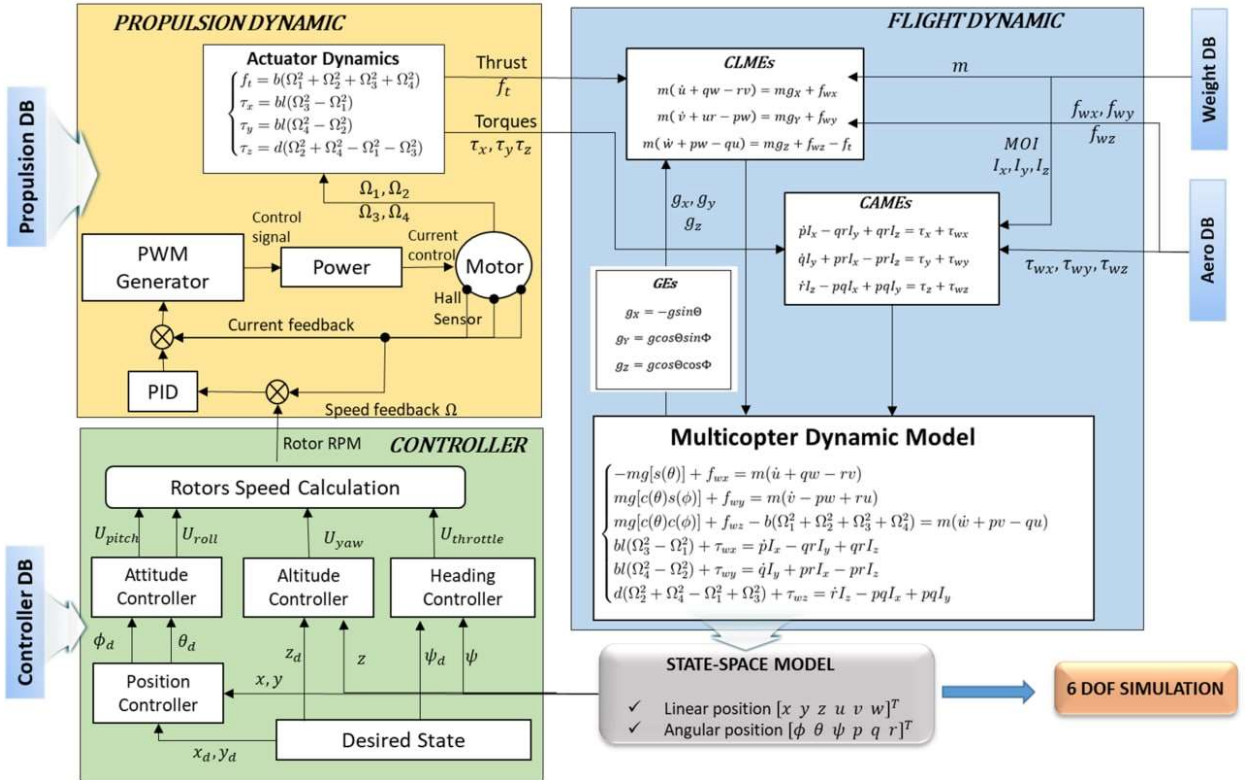


그림 35 The general flowchart of multicopter flight dynamics mode

(2) 비행제어를 위한 공기역학에 대한 수학적 모델링

- ▶ 본 연구에서 멀티콥터 공기역학에 대한 수학적 모델인 운동 방정식과 멀티콥터 비행 역학 모델에 통합된 방정식을 제시하였음.
- ▶ 아래 그림에서 쿼드콥터 로터 속도 방향 및 표기법의 예를 제시.

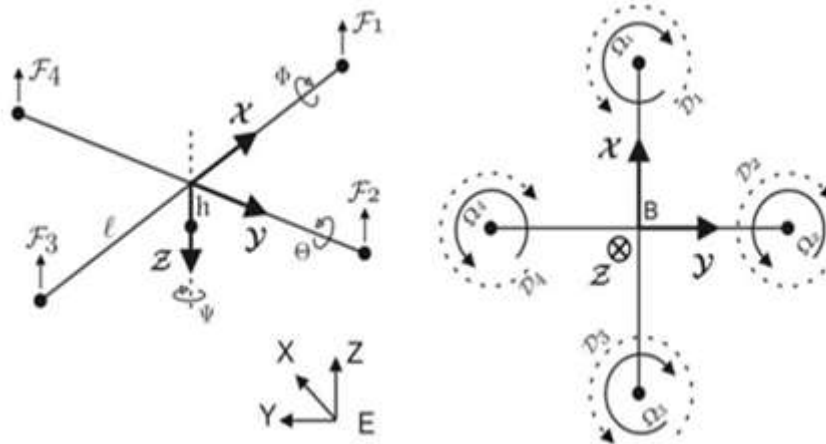


그림 36 Quadrotor coordinate system

- ✓ $[x, y, z, \phi, \theta, \psi]^T$ 는 지구 기준 좌표계(earth frame)에 대한 선형 및 각도 위치
- ✓ $[u, v, w, p, q, r]^T$ 는 물체 좌표계(body frame)에 대한 선형 및 각도 위치

(가) 트림 조건 및 선형/안정성 해석

- ◎ 쿼드 로터의 움직임에 대한 기본 개념은 다음과 같으며 다음 그림에 도시.

- ▶ 수평 프레임 상에서의 이동 : 플랫폼을 기울임
- ▶ 수직 프레임 상에서의 이동 : 모터의 전체 추력을 변경

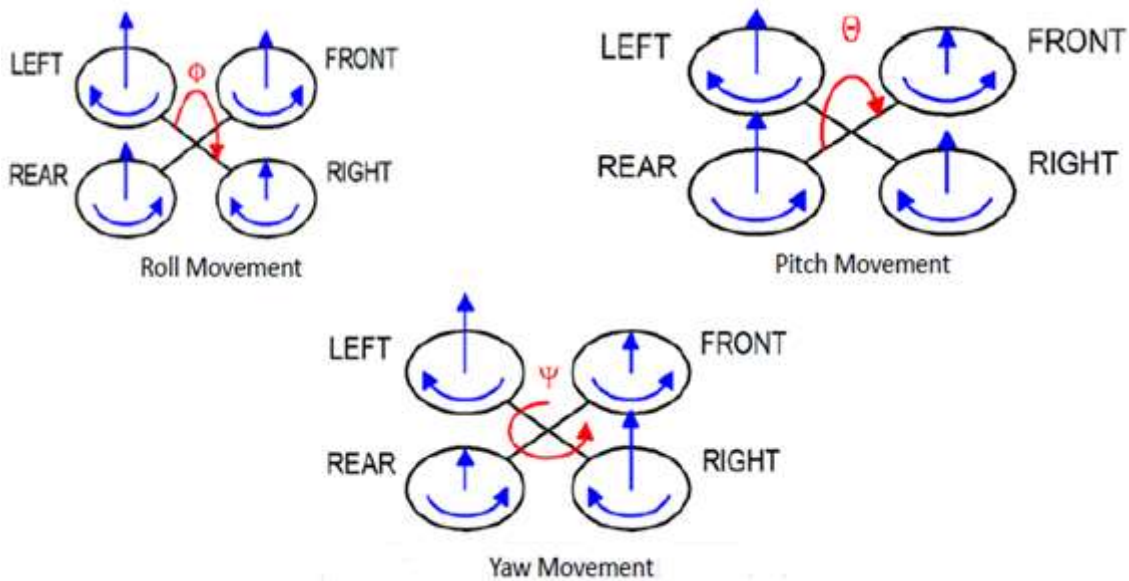


그림 37 Basic flight movement of a quadrotor

- ◎ 로터 속도에 따라 제어력과 모멘트가 발생하며 다음의 표에서는 로터 속도를 변경하여 양의 제어력과 모멘트를 생성하는 방법을 보여주고 있음.
- ▶ (+) 기호는 해당 로터의 속도를 증가시키면 양력/모멘트가 생성됨을 의미

표 16 Variation of rotor speeds to generate control forces and moments

Force / Moment	Ω_1	Ω_2	Ω_3	Ω_4
Roll Moment		-		+
Pitch Moment	+		-	
Yaw Moment	+	-	+	-
Vertical Thrust	+	+	+	+

(3) 멀티콥터 제어기 구조(architecture)

- ▶ 다음 그림은 멀티콥터의 제어기 구조를 보여주고 있는데 실제로 수동 또는 자동 제어의 임무 제어는 선형 위치 및 방위각으로 원하는 궤적을 참조하거나 원하는 상태를 정의함.
- ▶ 일반적으로 위치(position), 자세(attitude), 방향(heading) 및 고도(altitude)에 대한 4개의 제어기 모듈이 있음.
- ▶ 이러한 각 제어기의 기능을 다음의 표에서 도시하였음.

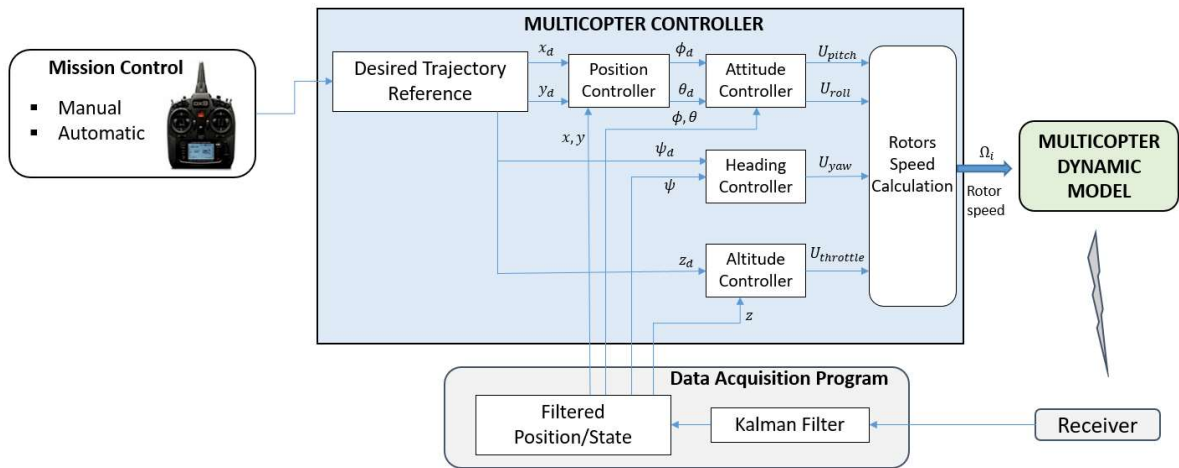


그림 38 Multicopter controller architecture

표 17 Multicopter controller modules and function

구성요소	제어기 기능	
	입력	출력
위치 제어기	목표 선형 위치와 그 피드백	목표 각도 위치
자세 제어기	목표 각도 위치	피치(Pitch)와 롤(Roll) 신호
고도 제어기	목표 고도와 피드백	스로틀 신호
방향 제어기	목표 방향과 피드백	요우(Yaw) 신호
로터 속도 계산	피치, 롤, 요우와 스로틀 신호	로터 속도의 분당회전수(RPM)
데이터 획득 프로그램	센서에서 실시간 신호	피드백 신호

(4) 단순 PID 제어기

▶ 다음 그림에서 쿼드콥터용 단순 PID 제어기의 구조를 도시하였음.

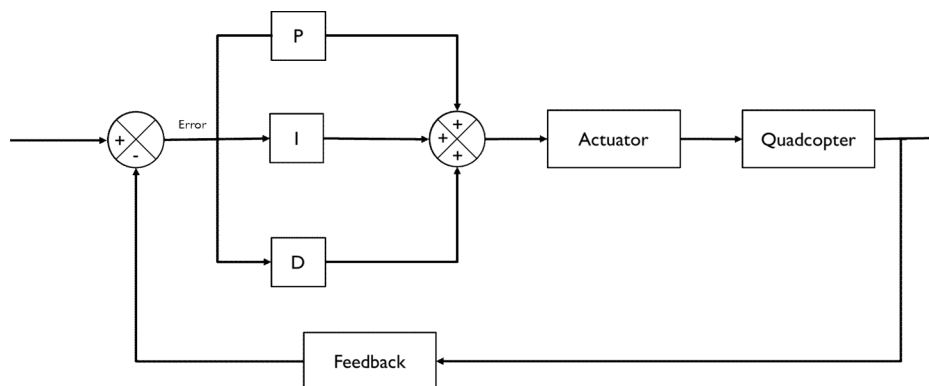


그림 39 Structure of PID controller

- ▶ PID 제어기의 일반적인 전달 함수는 다음과 같음.

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

✓ 여기서, K_p 는 비례 이득(gain), K_i 는 적분 이득, K_d 는 미분 이득을 의미.

- ▶ 다른 파라미터의 응답에 대한 제어기 이득(gain)의 영향은 아래 표에서 볼 수 있음.

표 18 Response of the different parameters of the PID gain

제어기	상승시간 (rise time)	오버슈트 (overshoot)	정착시간 (setting time)	정상상태 오차 (steady state error)	안정성 (stability)
K_p	Decreases	Increases	Small change	Decreases	Degrade
K_i	Decreases	Increases	Increases	Eliminate	Degrade
K_d	Small change	Decreases	Decreases	No effect in theory	Improve if is small

(5) SITL(Software-In-The-Loop) 시뮬레이션

- ▶ SITL은 FC를 소프트웨어적으로 모사하여 시뮬레이션 상에 적용하는 방법임.
- ▶ 하드웨어 없이도 소프트웨어 결함을 확인할 수 있다는 장점이 있으나 시간 동기화, 하드웨어 버그 등의 문제는 발견하지 못한다는 단점이 있음.

◎ Ardupilot 환경은 Linux와 Windows 모두에서 기본적으로 실행 가능하도록 개발하였음.

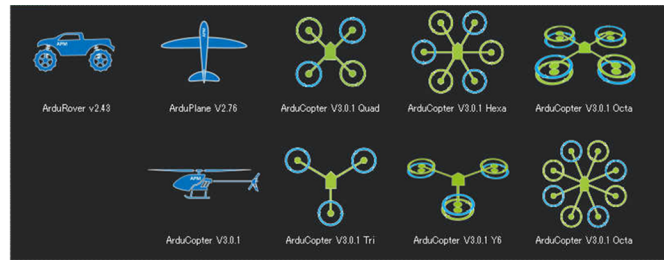
- ▶ 하드웨어로는 주변 장치 센서, 제어기 및 출력 장치가 있음.
- ▶ 펌웨어는 제어기에서 실행되는 코드로 Copter, Plane, Rover, Sub, 또는 안테나 트래커와 같이 차량 및 임무에 맞는 펌웨어를 선택하는 것이 가능.
- ▶ 소프트웨어는 컨트롤러에 대한 인터페이스로써 GCS (Ground Control Station)라고도하는 이 소프트웨어는 PC 또는 모바일 장치에서 구동 가능함.

◎ 다른 특징들

- ▶ Acro, Stabilize, 로이터(Loiter), Alt-hold, 출발지로 귀환(Return To Launch), 착륙(Land), Follow Me, GeoFence 등 모든 운행체 유형에 맞는 다양한 명령 모드를 제공함.
- ▶ GPS 위치, 배터리 상태 및 기타 실시간 정보를 포함하여 GCS와 제어기 간의 실시간 양방향 통신.
- ▶ 포괄적인 사후 분석을 위한 전체 데이터 로깅



Hardware



Firmware

그림 40 Hardware and Software/Firmware of Ardupilot

나) 장애물 인식 및 회피 인공지능 에이전트 개발

(1) Artificial Neural Network (ANN)

- ▶ ANN은 기본적으로 생물학적 신경망의 구조와 기능을 기반으로 한 계산 모델로 ANN의 구조는 정보 흐름의 영향을 받음.
- ▶ 인간의 뇌에서 뉴런들이 신호, 자극 등을 받고 그 자극이 임계값을 넘어서면 결과신호를 전달하는 과정에서 착안 됨.
- ▶ 신경망은 다수의 입력 데이터를 받는 입력층, 데이터의 출력을 담당하는 출력층, 입력층과 출력층 사이에 존재하는 은닉층이 존재함.
- ▶ 신경망의 성능은 은닉층의 개수와 노드의 개수를 어떻게 구성하느냐에 따라 달라지며, 은닉층에서는 활성화 함수(activation function)를 사용해 최적의 weight와 bias를 찾아내야함.
- ▶ 신경망 변경은 입력 및 출력을 기반으로 함.
- ▶ 기본적으로 ANN은 비선형 통계 데이터로 간주할 수 있으며 복잡한 관계는 입력과 출력 사이를 정의함.
- ▶ 결과로써 다른 패턴을 발견하고 또한 신경망으로써 ANN을 호출함.
- ▶ 출력값을 결정하는 활성화 함수의 종류에 따라 gradient가 0에 수렴해 weight를 구할 수 없는 경우, 또는 부분적인 오류를 최저 오류로 인식하여 학습을 종료하는 등의 문제로 인해 학습과정에서 파라미터의 최적값을 찾기 어려움.
- ▶ 과적합 문제가 발생하여 파라미터의 최적값을 찾지 못하고 그에 따라 적합한 출력을 얻을 수 없는 경우가 존재함.
- ▶ 은닉층의 수가 증가하면 학습의 정확도 또한 증가하나, 그만큼 연산량이 기하급수적으로 늘어 학습시간이 오래 걸리는 문제가 발생함.

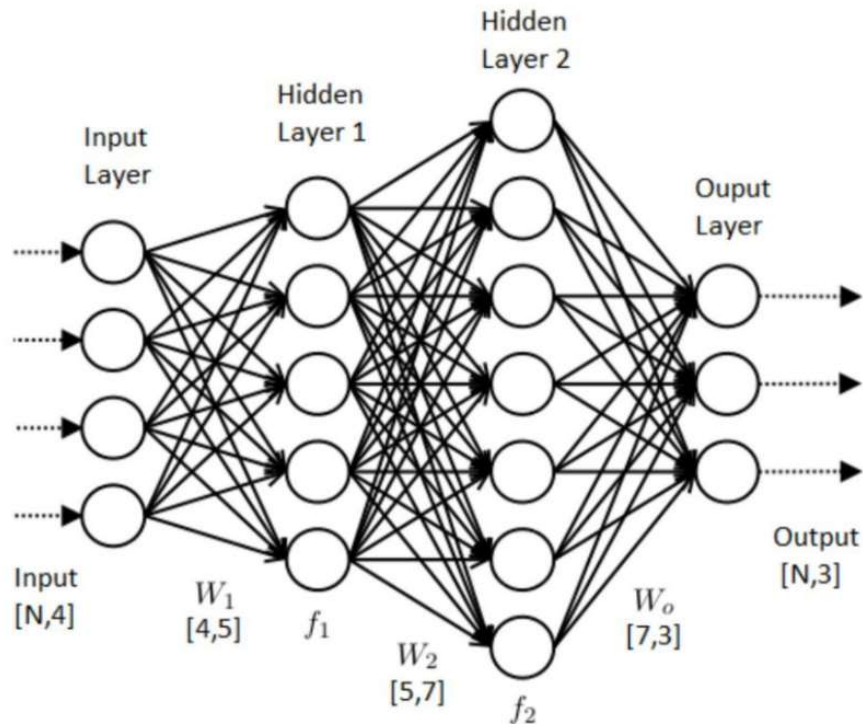


그림 41 An example of ANN

(2) Convolutional Neural Network (CNN)

- ▶ ANN 기법의 단점이 그래픽카드의 발전, 하드웨어의 성능 향상, 사전 훈련 등으로 해소되며 모델 내 은닉층을 늘려 학습의 결과를 향상시키는 방법이 등장함.
- ▶ CNN은 은닉층을 2개 이상 지닌 학습 방법인 DNN(Deep Neural Network)을 응용한 알고리즘임.
- ▶ 기존의 방식이 데이터에서 지식을 추출해 학습을 진행했다면, CNN은 데이터의 특징을 추출하여 특징들의 패턴을 파악하는 구조임.
- ▶ CNN 알고리즘은 convolution과정과 pooling과정의 반복으로 구성됨.
- ▶ Convolutional Neural Networks (ConvNets 또는 CNN)는 이미지 인식 및 분류와 같은 영역에서 매우 효과적인 것으로 입증된 신경망의 범주임.
- ▶ ConvNets을 이용하여 로봇 및 자율 주행 자동차의 파워 비전 외에 얼굴, 물체 및 교통 표지판을 식별하는 것까지 성공하였음.
- ▶ 따라서 ConvNet은 오늘날 대부분의 머신러닝 전문가에게 중요한 도구라고 할 수 있음.

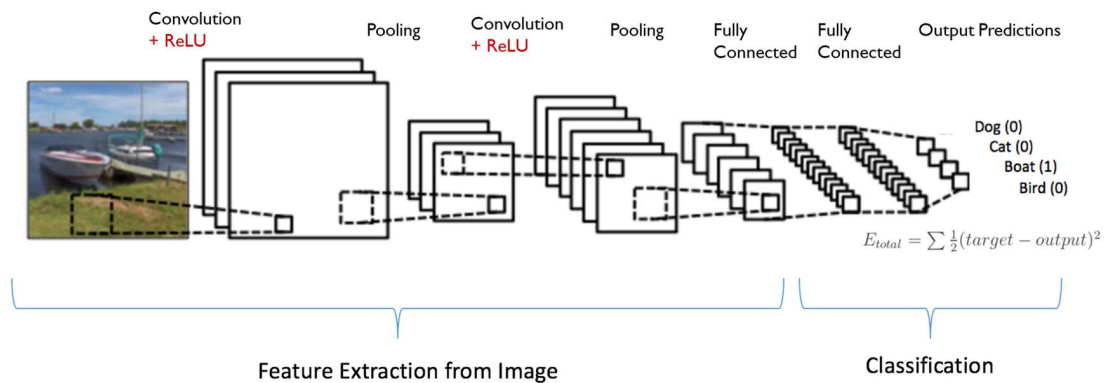


그림 42 A simple convNet source

(3) 심층 학습(Deep Learning, DL, 딥 러닝)

- ▶ 심층 학습(Deep Learning)은 데이터 처리 및 의사 결정에 사용할 패턴을 생성하기 위해 인간 두뇌의 활동을 모방하는 인공지능 기능이며 통상 ‘딥 러닝’이라고 부름.
- ▶ 딥 러닝은 인공지능 (AI)에서 기계 학습의 하위 집합으로, 구조화되지 않거나 레이블이 지정되지 않은 데이터에서 감독 되지 않은 학습이 가능한 네트워크를 갖추고 있음.
- ▶ 딥 러닝은 머신러닝의 한 분야로서 연속된 층에서 점진적으로 의미 있는 표현을 배우는데 강점이 있으며, 데이터로부터 표현을 학습하는 새로운 방식임.
- ▶ 최근 딥 러닝 모델은 표현 학습을 위해 수십 개, 수백 개의 연속된 층을 지님.
- ▶ 이 기법은 또한 심층 신경 학습(Deep Neural Learning) 또는 심층 신경 네트워크라(Deep Neural Network)고도 함.

(4) 강화 학습(Reinforcement Learning, RL)

- ▶ 전통적인 기계 학습(Machine Learning)은 주로 패턴 마이닝에 중점을 둔 반면, 강화 학습은 의사 결정에 초점을 맞추어 AI가 현실 세계에서 행동을 배우고 실행하는 영역으로 더 깊이 발전하도록 도와줌.
- ▶ 특히 복잡한 로봇제어 문제 해결과 복합적인 기술 학습, 제약사항이 존재하는 경우에 최선의 선택과 같은 문제들에 많이 사용됨.
- ▶ 경험 중심의 순차적 의사 결정을 위한 프레임워크로서 수십 년 동안 존재해 왔지만, 이 방법은 주로 표현 및 확장 문제로 인해 실제로 큰 성공을 거두지 못해왔음.
- ▶ 구글 딥 마인드가 개발한 컴퓨터 프로그램인 알파고 (AlphaGo)는 바둑에서 인간 챔피언인 이세돌 9단을 5번 중 4번 승리한 컴퓨터 프로그램으로 이 프로그램의 능력은 강화 학습에 기반한 것임.

(5) 심층 강화 학습(Deep Reinforcement Learning, DRL)

- ▶ 강화 학습과 심화 학습의 결합으로 에이전트는 환경에서 특정 행동에 따른 반응, 보상, 그리고 불이익을 통해 학습하며, 신경망 학습을 통해 행동-가치 함수를 추정 가능함.
- ▶ Q-Learning과 심층 신경망을 결합한 강화 학습 알고리즘을 통해 복잡하고 고차원의 환경에서 작용하며 에이전트는 환경과 상호작용하며 어느 단계에서도 각 상태에 가장 적합한 행동을

찾아냄.

- ✓ 에이전트는 환경과의 상호작용에서 누적된 보상을 극대화할 수 있는 최적의 정책을 찾기 위해 상태 공간의 탐색과 활용의 균형을 맞춘다.
- ✓ 같은 맥락에서 에이전트는 모든 시간 단계에 대한 인식, 취한 행동 및 보상으로 다음 행동 또는 정책을 수정하며 전체 상태 공간에 걸쳐 최적화 프로세스를 구성함.

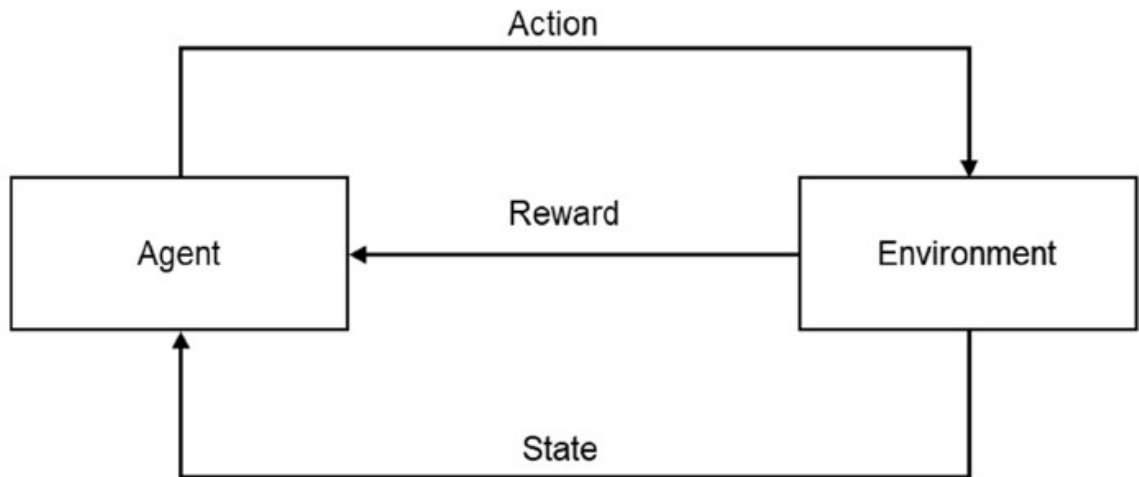


그림 43 Schematic process of Reinforcement Learning

- ▶ 심층 강화 학습기법으로는 다음과 같은 방법들이 있음.

◎ Deep Q Learning(DQN)

- ▶ Q-Learning 기반 강화학습은 매 상태-행동 pair에 대한 Q 값을 근사한 후, 이를 기반으로 어떤 상태에서 어떤 행동을 취할지 결정하는 방법임.
- ▶ 탐색을 위해 대개 ϵ -탐욕 정책을 많이 이용하며, 이는 특정 상태에서 확률 ϵ 로 무작위의 행동을, $(1-\epsilon)$ 의 확률로 제일 높은 Q 값을 가지는 행동을 선택하는 방법임.
- ▶ Q-Learning과 심층 신경망을 결합하여 비디오 게임 또는 로봇 공학과 같은 복잡한 고차원 환경에서 강화 학습을 수행할 수 있도록 하는 알고리즘.
- ▶ DQN은 유명한 AlphaGo를 포함한 여러 프로그램에서 게임 전략을 학습하기 위해 사용되어 옴.

◎ 정책 경사(Policy Gradient)

- ▶ 강화 학습의 목표는 에이전트가 최적의 보상을 얻을 수 있는 최적의 행동 전략을 찾는 것임.
- ▶ 정책 경사 기법은 정책을 직접 모델링하고 최적화하는 것을 목표로 하며 정책 기반 강화 학습의 대표 예로 PPO(Proximal Policy Optimization)이 있음.
- ▶ 정책이란 관측값을 입력으로 받아서 행동 값을 출력하는 함수로 매개변수 벡터 θ 로 정의됨.
- ▶ 심층 강화학습에서는 이 함수를 심층 신경망을 이용하여 근사하며, 이때 매개변수 벡터 θ 는 신경망의 가중치와 편향 값임.
- ▶ 정책 경사 기반 강화 학습은 θ 를 구하기 위해 경사 상승법을 이용하는 방법으로, 특정 θ 에

대한 목적 함수의 gradient를 구한 후 이 경사가 상승하는 방향으로 일정 거리만큼 θ 를 업데이트 하는 과정을 경사가 수렴할 때까지 혹은 최대 타임 스텝만큼 반복하는 것임.

- ▶ 목적 함수는 정책 π_θ 에 따라 행동 시 얻게 되는 누적 보상의 기대치이며, 다음과 같은 식으로 표현됨.

$$J(\theta) = E_{\pi_\theta} [r(s, a)]$$

- ▶ 목적 함수의 경사는 정책 경사 정리에 의해 다음과 같이 정의됨.

$$\nabla J(\theta) = E_{\pi_\theta} [Q_\pi(s, a) \nabla_\theta \ln \pi_\theta(a|s)]$$

- ▶ 정책 경사는 목적 함수를 최대화하는 방향으로 정책의 매개변수를 다음과 같이 업데이트함.

$$\theta \leftarrow \theta + \alpha \nabla J(\theta)$$

- ▶ 정책 기반 강화학습의 단점 중 한 가지는 정책 경사를 통해 매개변수를 점진적으로 갱신하는 것이, 이 매개변수로 이루어진 정책을 급격하게 변화시킬 수 있다는 것임.
- ▶ 이는 학습의 불안정성을 야기하며 학습 속도 및 성능 저하의 원인이 됨.
- ▶ TRPO(Trust Region Policy Optimization)은 급격한 정책 갱신을 방지하기 위해서 정책 갱신 전후 KL(Kullback-Leibler) divergence를 일정 수준 이하로 제약하는 조건을 추가함.
- ▶ TRPO는 DQN으로는 해결하지 못했던 연속적인 행동공간을 갖는 로봇제어 문제를 성공적으로 해결하였으나 제약조건을 풀기 위해서 많은 계산이 필요할 뿐아니라 다양한 신경망 구조와 호환되지 않는 단점이 존재함.
- ▶ PPO는 TRPO의 성능은 유지하면서도 계산량을 감소시켜 성능, 계산 효율성, 구현 용이성등의 측면에서 매우 뛰어남.

◎ 내적 보상(Intrinsic Reward)을 이용한 강화 학습

- ▶ 강화 학습은 시행착오를 통해서 환경을 탐험하며 학습하는 방법으로 매 행동마다 보상이 주어지는 경우에는 학습이 잘 이루어질 수 있지만, 보상이 드물게 혹은 지연되어 주어지는 환경에서는 정책 학습이 성공적으로 이루어지기 어려움.
- ▶ 내적 보상이란 환경으로부터 주어지는 보상이 아닌 에이전트 스스로 생성하는 보상을 의미함.
- ▶ 사람의 내적 동기를 통한 학습 방법을 모방한 것으로 대표적인 내적 보상 함수로 예측오차가 있음.
- ▶ 예측 오차는 에이전트가 예측한 다음 상태와 실제 다음 상태 간 차이로 정의되며, 에이전트는 다음 상태를 예측하기 위해서 대개 하나 이상의 예측 모델을 정의하고, 이를 정책과 함께 학습함.
- ▶ 에이전트가 환경을 탐색하며 예측모델의 학습이 이루어지기 때문에 에이전트에게 익숙한 상태에서는 이 예측 오차가 낮아지게되고, 낯선 상태에서는 이 예측 오차가 높아지게 됨.
- ▶ 즉, 예측 오차는 에이전트의 탐험을 장려하는 효과가 있으며 에이전트가 죽거나 실패해 초기 상태로 돌아가는 것을 억제하는 효과가 있음.
- ▶ 내적 보상함수는 에이전트가 매 행동을 수행할 때마다 보상을 얻을 수 있도록 설계되기 때문에 환경으로부터 드문 혹은 지연된 보상이 주어지는 환경에서도 좋은 학습성능을 낼 수 있으며, 매 환경 내지는 매 태스크 마다 사람이 조밀한 보상함수를 직접 설계해야 하는 필요성을 낮출 수 있기 때문에 강화학습 적용 분야를 넓히는데 큰 도움이 됨.

- ▶ 에이전트가 달성해야 하는 태스크의 방향성과 내적 보상의 방향성 일치여부, 학습이 진행됨에 따라서 동일한 상태에서 동일한 행동을 수행했을 때 얻는 보상의 비정상성, 그리고 환경마다 달라지는 보상의 스케일 문제 등을 고려해 주어야 함.

(6) DQN을 이용한 장애물 회피 네트워크 개발

- ▶ 자율 주행 비행체에서는 안전이 가장 중요하므로 DQN 방법을 사용하는 장애물 회피 네트워크는 장애물 회피 메커니즘을 입력으로 사용하여 무인비행체가 출력으로 장애물과 충돌하지 않도록 하는 동작을 생성하게 함.
- **입력**
 - ▶ 무인비행체의 카메라 이미지 예측 메커니즘은 충돌 시간 또는 예상 최소 거리와 같은 측정값을 생성함.
 - ▶ 이러한 조치를 기반으로 로컬 경로 재계획을 수행하기 위해 장애물 회피 메커니즘이 활성화됨.
- **동작 생성**
 - ▶ 자율 주행 비행체의 장애물 회피 시에 고려해야 할 또 다른 사항 중 하나는 비행체가 가능한 동작을 생성하는 것임.
 - ▶ 비행체의 행동을 예측하기 위해, 장애물이 도달 가능한 위치 세트의 확률적 모델을 생성하고 이러한 도달 가능한 세트를 확률 분포와 연관시킬 수 있음.

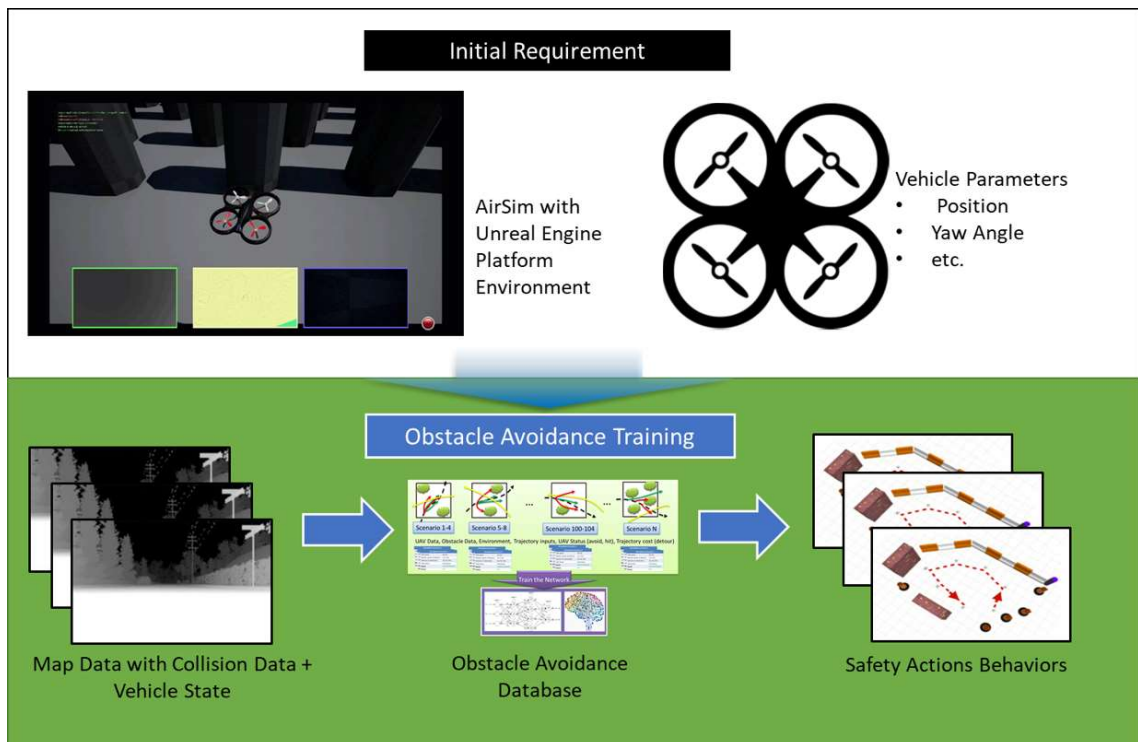


그림 44 Unreal Environment of AirGym-v42 with working on AirSim Simulator

(가) AirSim Simulator

- ▶ Microsoft AirSim은 테스트 및 AI 교육을 위한 시뮬레이션을 개발하고 지원하기 위한 실시

간 플랫폼을 제공하는 언리얼 엔진(Unreal Engine, 이하 UE4) 용 오픈 소스 플러그인이며 드론, 차량 및 다양한 이동체에 사용 가능함.

- ▶ UE4를 통해 AirSim은 게임 산업이 만든 렌더링 기술의 꾸준한 발전을 활용함.
- ▶ UE4는 물리 기반 재료, 광도 조명, 평면 반사, 광선 추적 거리 필드 그림자, 조명 반투명 등과 같은 최첨단 그래픽 기능을 제공함. 물리적 및 시각적으로 시뮬레이션 할 수 있도록 PX4와 같은 널리 사용되는 비행 컨트롤러를 사용하여 하드웨어 인 루프를 지원함.
- ▶ 즉, AirSim은 시뮬레이션 된 쿼드콥터 (Parrot AR Drone 2.0)를 게임 엔진 내부에 놓을 수 있음.
- ▶ AirSim은 AI 연구가 자율 주행 비행체를 위한 심층 학습, 컴퓨터 비전 및 강화 학습 알고리즘을 실험 할 수 있는 가장 대중적인 플랫폼 중 하나임.
- ▶ AirSim은 API를 공개하여 플랫폼과 독립적인 방식으로 데이터를 검색하고 비행체를 제어함.

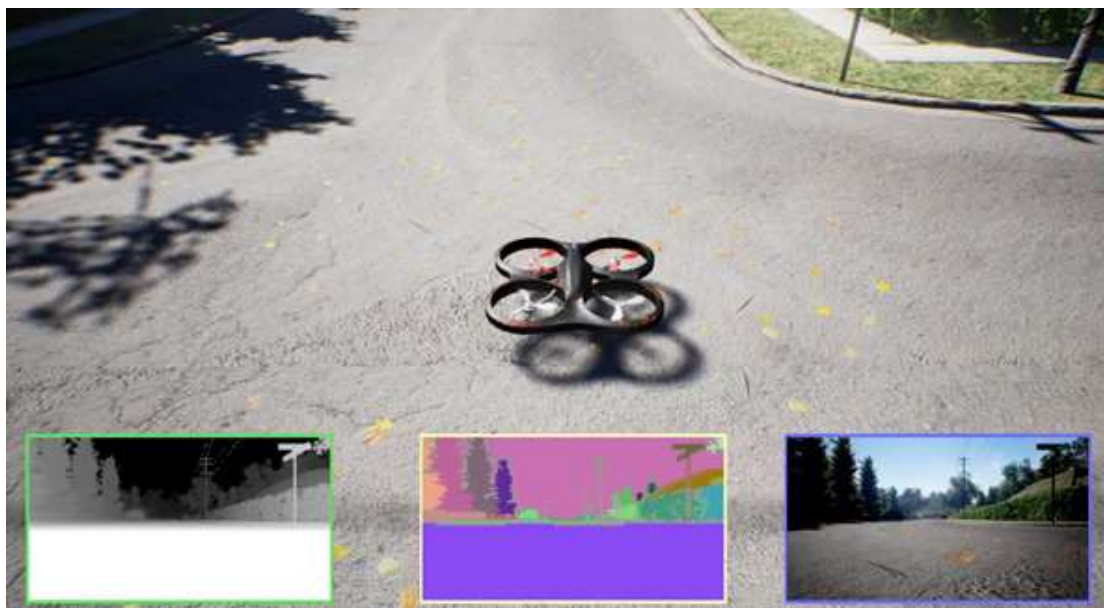


그림 45 Unreal Environment of AirGym-v42 with working on AirSim Simulator

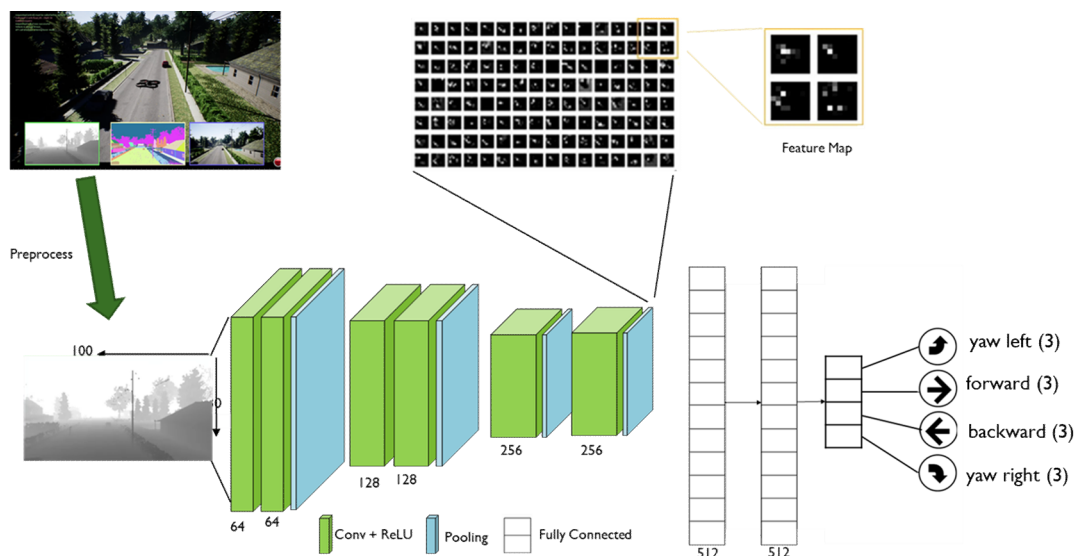


그림 46 Current Network Architecture of DQN

표 19 DQN Network Specification

List	Param#
No. of Convolutional Layers	6
No. of Flatten	1
No. of Fully Connected Networks	3
No. of Training steps	10,000
No. of Actions	3
Optimizer	Adam
Activation Function	ReLU , Linear
Learning Rate	0.000025
Metrics	mse
Library	Tensorflow Keras rl

- ▶ 네트워크 입력은 30×100 상태의 표현임.
- ▶ 6개의 컨볼루션(convolution)과 3 개의 완전히 연결된 레이어(fully connected layer)가 포함되어 있음.
- ▶ 이 네트워크의 동작 공간은 12개의 개별 작업으로 구성됨.
- ▶ 1, 3 및 5m/s 속도로 앞뒤로 이동할 수 있도록 6가지 동작이 할당됨.
- ▶ 고정 각도 15, 30 및 45도에 대해 좌우로 돌리기 위해 6가지 동작을 할당함.
- ▶ 네트워크의 보상 기능은 다음에서 확인할 수 있으며 거리는 유클리드 거리를 사용하여 계산함.

```

if collided == True:
    done = True
    reward = - 100.0
elif distance_to_goal_now < 2:
    done = True
    reward = 100.0
else
    done = False
    •
    reward = (distance_to_goal_before - distance_to_goal_now)

```

그림 47 Reward function of DQN

(나) DQN를 이용한 테스트 수행

- ▶ 테스트 및 훈련은 16Gb RAM 및 Nvidia GTX 1050 Ti GPU가 장착된 Intel i7-7700 CPU에서 환경으로 수행.
- ▶ 다음 그림에서 10,000 훈련 단계에서 실행되는 보상 가치에 대한 DQN 및 DDQN 결과의 비교를 확인 가능.
- ▶ 이 시점에서 DQN과 DDQN의 차이점은 적은 수의 훈련 단계와 환경 복잡성으로 인해 거의

구분하기 어려움.

- ▶ 네트워크 훈련 및 무인기의 장애물 회피에 대한 두 가지 접근 방식을 비교하기 위해 100 개 이상의 테스트 사례를 시뮬레이션을 수행하였음.
- ▶ 실패한 비행에 대한 성공적으로 임무를 완료(충돌 없이 목표 위치에 도달)한 비율을 이용하여 알고리즘 효율성을 평가하였음.

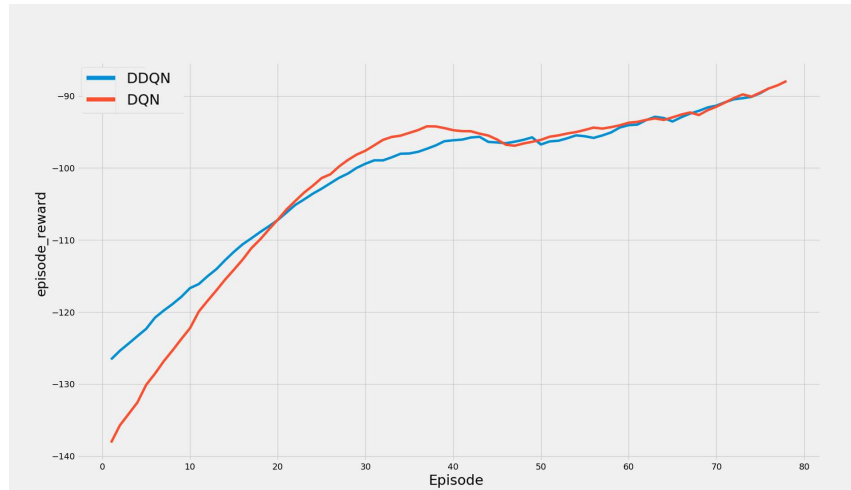


그림 48 Reward result Comparison DQN and DDQN

- ▶ 다음 그림은 250,000 개의 훈련 단계에서 실행되는 보상 값에 대한 DQN 및 reference DQN (AirSim page의 DQN 모델) 결과를 비교한 것임.
- ▶ 장애물을 피하기 위해 **각 단계에 대한 응답 시간은 평균 약 0.8초가 소요**되었으며 현 모델에 대한 개선은 지속적인 연구를 통해 수행할 예정임.



그림 49 Result comparison between DQN and Ref_DQN (DQN model from Microsoft AirSim Page) training on 250,000 steps

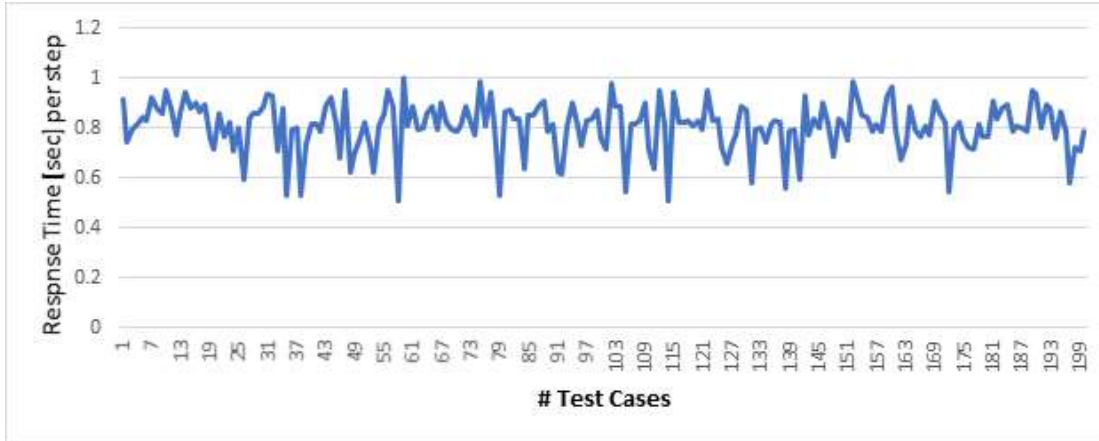


그림 50 Tested DQN network result run on 200 test cases

다) 경로 계획과 최적화를 위한 AI 에이전트 개발

(1) 경로 계획기(Path planner)

- ▶ 무인 이동체는 자율 운항을 전제로 하며, 특히 무인 비행체는 3차원 공중에서 운항.
- ▶ 무인 비행체가 주로 비행하는 구역은 일반적인 항공기가 순항하는 고도에 비해 저고도로 산, 건물, 지형지물과 같은 정적 장애물과 함께 새와 같은 비행생물, 다른 유·무인 비행체의 존재, 국부적인 기상 변화와 같은 동적 장애 요인에 대해 실시간으로 대처할 수 있어야 함.
- ▶ 동적 환경에서 자율적이고 반응이 빠른 비행체의 경로를 계획하는 것은 복잡한 문제이며 특히 차량이 전 기동 능력을 사용 해야 하는 경우에는 더욱 그러함.
- ▶ 여기서는 결정론적이고 완전한 알고리즘을 사용하는 것이 하나의 방법으로 가능한 모든 경로를 검색하고 목적 함수를 사용하여 최적의 경로를 식별해야 함.
- ▶ 그러나 이러한 작업에는 엄청난 양의 계산 자원을 필요로 하며, 따라서 실시간으로 경로를 탐색하는 것이 불가능할 수 있음.
- ▶ 이 계산의 복잡성을 피하고 효과적인 실시간 경로 계획을 제공하기 위해 확률론에 기반한 경로 계획기를 사용하였음.

(2) 경로 계획을 위한 탐색 알고리즘

(가) Anytime heuristic search algorithm

- ▶ 최적의 경로를 찾는 것보다 연산 시간이 더 중요한 경로 계획 문제의 경우 Anytime planning 알고리즘이 적절함.
- ▶ Anytime search 알고리즘은 초기에 높은 수준의 차선의 최적 경로를 빠르게 계산할 수 있고, 시간에 여유가 있다면 계산한 경로를 발전시킬 수 있음.
- ▶ Anytime 알고리즘은 A* heuristic 알고리즘을 발전시킨 것으로 초기 경로가 생성된 뒤에도 계속 연산을 진행하여 더 나은 경로를 찾음.

(나) Anytime Repairing A* algorithm(ARA*)

- ▶ Anytime Repairing A*(ARA*)은 일관성의 개념을 사용하여 이전 검색의 비용이 새로운 인플

레이션 계수 값에 대해 유효하지 않은 경우를 고려해 각 검색 중 수행되는 처리를 제한함.

- ▶ ARA*는 Weighted A*(WA*)알고리즘을 기반으로 하며 이는 주어진 초기 인플레이션 계수 값을 이용하여 초기 경로를 계산하고 인플레이션 계수 값을 줄여나가며 더 나은 경로를 구하기 위한 계산을 반복함.
- ▶ 탐색을 진행하면 상태가 확장되고 주변 상태와 관련된 경계 비용이 변화해 일관성이 없어지면 확장될 상태 큐에 재삽입되지 않음.
- ▶ 현재 탐색 과정이 종료되면 일관성이 없는 리스트의 상태가 다음 탐색을 위해 사용되는 비용인 새로운 우선순위 큐에 삽입됨.
- ▶ 알고리즘은 앞서 설명한 과정을 인플레이션 계수가 다른 탐색의 인플레이션 계수와 동일해지거나 목표 상태까지 확장될 때까지 반복함.

Algorithm 2: ARA*

```

1 Function Key(s):
2   return  $g(s) + \epsilon * h(s)$ 
3 Function ImprovePath():
4   while  $Key(s_{goal}) > \min_{s \in OPEN} (Key(s))$  do
5     s = OPEN.Pop()
6     CLOSED.append(s)
7     forall  $s' \in Successor(s)$  do
8       if  $g(s) + cost(s, s') < g(s')$  then
9          $g(s') = g(s) + cost(s, s')$ 
10        if  $s' \notin CLOSED$  then
11          OPEN.insert( $s', Key(s')$ )
12        else
13          INCONS.insert( $s'$ )
14        end
15      end
16    end
17  end
18 Function Main():
19  Set all nodes  $g(s) = \infty$ 
20  OPEN = CLOSED = INCONS =  $\emptyset$ 
21   $g(s_{start}) = 0$ 
22  OPEN.insert( $s_{start}, Key(s_{start})$ )
23  ImprovePath()
24  Choose initial sub-optimal bound  $\epsilon$ 
25  Publish  $\epsilon$ 
26  while  $\epsilon > 1$  do
27    Decrease  $\epsilon$ 
28    Move nodes in INCONS to OPEN
29    Update all priority values in OPEN according to Key(s)
30    CLOSED =  $\emptyset$ 
31    ImprovePath()
32    Publish  $\epsilon$ 
33  end

```

그림 51 ARA* pseudo code

(다) D* Lite algorithm

- ▶ Dynamic A*(D*) 알고리즘을 기반으로 구현된 새로운 재계획 방법으로 incremental heuristic 탐색 알고리즘의 하나임.
- ▶ D* Lite의 실험특성은 D* 알고리즘보다 효율적인 결과를 보이며, D* Lite 알고리즘은 로봇 행동 계획 분야에서 최단 경로 계획 및 재계획을 A* 알고리즘보다 높은 성능으로 수행 가능함.

- ▶ D* Lite 알고리즘 연산 과정의 각 노드마다 두 개의 비용 추정치가 존재하는데, $g(s)$ 는 목적 함수를 의미하며 $rhs(s)$ 는 목적함수 보다 한 단계 앞선 함수를 의미함.
- ▶ 최단 경로를 계산하기 위해서 알고리즘은 $g(s)$ 와 $rhs(s)$ 의 일관성을 항상 체크함.

```

Key(s)
1  return[ $\min(g(s), rhs(s) + h(s_{start}, s)), \min(g(s) + rhs(s))$ ]
UpdateState(s)

1  If  $s$  was not visted before
2   $g(s) = \infty$ 
3  If  $(s \neq s_{goal}), rhs(s) = \min_{s' \in Succ(s)} (g(s') + c(s, s'))$ 
4  If  $s \in OPEN$ , remove  $s$  form  $OPEN$ 
5  if  $(g(s) \neq rhs(s))$  insert  $s$  into  $OPEN$  with key  $(s)$ 
ComputeShortestPath()
1  while ( $\min_{s \in OPEN} (keys(s_{start} \text{ OR } rhs(s_{start}) \neq g(s_{start}))$ 
2      remove state  $s$  with minimum key from  $OPEN$ 
3      if  $g(s) > rhs(s)$ 
4           $g(s) = rhs(s)$ 
5          for all  $s \in Pred(s)$  UpdateStare( $s'$ )
Main()
1   $g(s_{start}) = rhs(s_{start}) = \infty; g(s_{goal}) = \infty$ 
2   $rhs(s_{goal}) = 0; OPEN = \emptyset$ 
3  insert  $s_{goal}$  into  $OPEN$  with key  $(s_{goal})$ 
4  forever
5      ComputeShortestPath()
6      wait for changes in edge costs
7      for all directed edges  $(u, v)$  with changed costs
8          Update the edge cost  $c(u, v)$ 
9      UpdateState  $u$ :

```

그림 52 D*Lite pseudo code

(라) Anytime Dynamic A* (ADA*)

- ▶ Anytime Dynamic A*(ADA*)은 주어진 시간 내에 최상의 계획탐색을 목표로 함.
- ▶ 복잡한 경로 계획 문제에서는 무인 비행체가 이용 가능한 시간 내에 최적의 솔루션을 얻지 못할 수도 있음.
- ▶ 무인 비행체는 사용 가능한 계산시간 내에 생성할 수 있는 최상의 솔루션을 만족해야 함.
- ▶ 미지의, 또는 정보가 부분적으로 존재하거나 환경이 변화하는 상황에서도 효율적으로 경로를 계획할 수 있음.
- ▶ 환경이 변경되거나 알 수 없는 장애물에 막히게 되면 알고리즘은 다른 경로를 효율적으로 재 계획을 수행함.
- ▶ 동적 환경 (예 : D* 및 D* Lite) 및 복잡한 계획 문제 ARA*에 대처하기 위한 탈출에 효율적인 알고리즘으로부터 확장됨.

<pre> Main() 1 $g(s_{start}) = rhs(s_{start}) = \infty$; $g(s_{goal}) = rhs(s_{goal}) = \infty$; $\epsilon = \epsilon_0$; 5 $OPEN = CLOSED = INCONS = \emptyset$; 6 insert s_{goal} into $OPEN$ with $key(s_{goal})$; 7 ComputeOrImprovePath(s); 8 publish ϵ – current suboptimal solution; 9 forever 13 if changes in edge costs are detected 14 for all directed edge (u, v) with changed edge costs 17 for all directed edges (u, v) with changed edge costs 19 Update the edge cost $c(u, v)$; if significant edge cost changes were observed increase ϵ or replan from scratch; 25 else if $\epsilon > 1$ 26 decrease ϵ 28 move state s from $INCONS$ into $OPEN$; 29 update the priorities for all $s \in OPEN$ according to $key(s)$ 30 $CLOSED = \emptyset$ 31 ComputeOrImprovePath(s); 32 return publish ϵ – current suboptimal solution; 33 if $\epsilon = 1$ 34 wait for changes in edge cost </pre>	<pre> Key(s) 1 If $(g(s) > rhs(s))$ 2 return $[rhs(s) + \epsilon \cdot h(s_{start}, s); rhs(s)]$; 3 else 4 return $[g(s) + h(s_{start}, s); g(s)]$; UpdateState(s) 1 If s was not visited before 2 $g(s) = \infty$ 3 if $(s \neq s_{goal})$ $rhs(s) = \min_{s' \in Succ(s)} (c(s, s') + g(s'))$; 4 if $(s \in OPEN)$ remove s from $OPEN$; 5 if $(g(s) \neq rhs(s))$ 6 if $(s \notin CLOSED)$ 7 insert s in $OPEN$ with $Key(s)$; 9 else 10 inset s into $INCONS$; ComputeOrImprovePath() 1 while $(\min_{s \in OPEN} (key(s)) < key(s_{start}) OR rhs(s_{start}) \neq g(s_{start}))$ 2 remove state s with the minimum key from $OPEN$; 10 if $g(s) > rhs(s)$ 11 $g(s) = rhs(s)$; 13 $CLOSED = CLOSED \cup \{s\}$; 14 for all $s' \in Pred(s)$ UpdateState(s'); 15 else 16 $g(s) = \infty$; 17 for all $s' \in Pred(s) \cup \{s\}$ UpdateState(s'); </pre>
--	--

그림 53 ADA* pseudo Code

- ▶ ADA*는 재계획 및 anytime planning 알고리즘 두 가지 모두에서 장점을 가짐.
- ▶ Anytime search 측면에서 알고리즘은 ARA*를 실행할 때 상태의 재확장을 제한하는데 즉, 가능한 솔루션에 대한 검색이 조금 더 빠름.
- ▶ 재계획 측면에서 그래프의 상의 변화가 거의 또는 전혀 없는 소규모 환경에서는 상태를 확장하는 것이 더 많은 노력을 요함.
- ▶ A*는 ADA*보다 효율적일 수 있으나 그래프가 커지고 변화가 증가함에 따라 ADA*의 장점인 재계획 기능을 확인 가능하며 선호됨.
- ▶ 강건성 측면에서 ADA*는 U자형 장애물에 대한 제약을 고려하지 않았으므로 다음 검색에서 알고리즘이 U자형 장애물에서 다시 순회할 가능성이 더 높음.

(마) Mixed Integer Linear Programming (MILP)

- ▶ 결정해야 할 변수 중 일부가 최적해에서 정수 형태로만 제한되는 경우로 정수형 변수를 사용하면 유용한 최적화 문제의 범위를 크게 확장 가능함.
- ▶ MILP를 사용하면 경로 최적화에 non-convex 구속 조건과 이산 결정을 포함시킬 수 있음.
- ▶ 이진 결정 변수를 통해 장애물의 "왼쪽" 또는 "오른쪽"을 통과할지 또는 계획 문제에 포함시킬 대상에 비행체를 이산적으로 할당할지를 선택할 수 있음.
- ▶ MILP를 사용하여 목표에 고정된 계획 지평으로 전체 경로를 설계하는 것은 경로의 길이와 피해야 할 장애물의 수에 따라 필요한 계산 노력이 빠르게 증가하기 때문에 실시간으로 수행하기가 매우 어려움.
- ▶ ADA*와 함께 MILP를 사용하면 장애물이 있는 동적인 환경에서 작동하는 다수의 지상 및 비행하는 이동체가 있는 시나리오에서 실시간 경로 최적화를 적용할 수 있음.

(3) improved Anytime Dynamic A*(iADA*)

- ▶ 본 연구에서는 효율적인 경로 탐색을 위해 **기존의 오리지널 ADA* 알고리즘을 개선한 iADA*를 제안**하였음.
- ▶ 새로운 알고리즘의 주요 개선점은 인플레이션 계수가 감소한 후의 상태 재확장임.
- ▶ 환경에 영향을 주는 변경 사항이 있는 경우, 알고리즘은 상태 목록을 다시 정렬하는 검색 방법을 사용하여 검색 전체에서 경로 계산 빈도를 줄이며 이 단계를 원래의 알고리즘에 비해 훨씬 빠르게 수행함.
- ▶ 또한 가상의 벽을 생성하여 비행체가 U자형 장애물과 같은 복잡한 장애물을 가로질러 이동하는 것을 방지하는 새로운 구속 조건을 도입하였고 필요한 경우 비행체가 동일한 격자에서 다시 움직이지 않게 함.
- ▶ 구속 조건은 특히 장애물이 서로 마주 보는 경우 경로가 두 장애물 사이를 가로지르지 않도록 보호함.
- ▶ 원래 ADA*에 비해 새로운 방법은 비용 계산 방법, 검색 시간 및 목표에 도달하는 총 시간의 계산 방식 등을 새롭게 개선하였음.
- ▶ 본 연구에서 제안한 개선된 알고리즘은 지정되지 않은 지형 환경에 존재하는 목표 지향 탐색을 위한 빠른 재계획 방법을 제공할 수 있도록 설계되었음.
- ▶ 경로 재계획 알고리즘은 다음과 같은 단계를 포함함.
 - ✓ 시작 상태에서 목표 상태까지의 최적 경로를 탐색하기 위해 상기 목표 상태를 출발하여 상기 시작 상태에 도착하는 초기 경로를 생성하는 단계
 - 특정 상태와 인접하는 적어도 하나의 인접 상태를 결정하는 단계
 - ✓ 상기 초기 경로를 따라 상기 시작 상태에서 상기 목표 상태까지의 경로 비용을 산출하는 단계
 - 적어도 하나의 인접 상태에 대해 시작 상태에서 해당 인접 상태까지의 제1 경로 비용과 해당 인접 상태에서 목표 상태까지의 제2 경로 비용 간의 합산을 통해 인접 경로 비용을 산출하는 단계
 - 특정 상태에서 인접 경로 비용이 최소인 인접 상태로 이동하는 이동 경로를 포함하도록 초기 경로를 갱신하는 단계
 - ✓ 상기 경로 비용의 변화유무를 검출하여 인플레이션 계수를 갱신하는 단계
 - 경로 비용의 변화가 설정된 기준을 초과하는 경우 인플레이션 계수를 단위 증가량만큼 증가시키는 단계
 - 경로 비용의 변화가 설정된 기준을 초과하지 않는 경우 인플레이션 계수를 단위 감소량만큼 감소시키는 단계
 - ✓ 갱신된 상기 인플레이션 계수를 기초로 상기 초기 경로를 갱신하여 재계획 경로를 생성하는 과정에서 상기 인플레이션 계수가 감소된 경우 재정렬된 상태 목록을 기초로 재계획 경로를 생성하는 단계
 - 현재 상태가 이전의 이동 경로 상에 존재하는 경우 현재 상태의 상태정보를 임시 장애물로 변경하여 재계획 경로의 생성과정에서 탐색 되는 것을 방지하는 단계
 - 인플레이션 계수가 증가된 경우, 임시 장애물에 해당하는 상태들의 상태정보를 초기화하는 단계
 - 주어진 시작 상태, 목표 상태 및 인플레이션 계수를 갖는 고정된 환경에서 현재 경로를 계

산하거나 개선하기 위한 탐색 과정을 반복적으로 수행하는 단계

- 제1 대기열의 첫 번째 키값이 주어진 시작 상태의 키값보다 작을 때까지 탐색 과정을 반복하는 단계

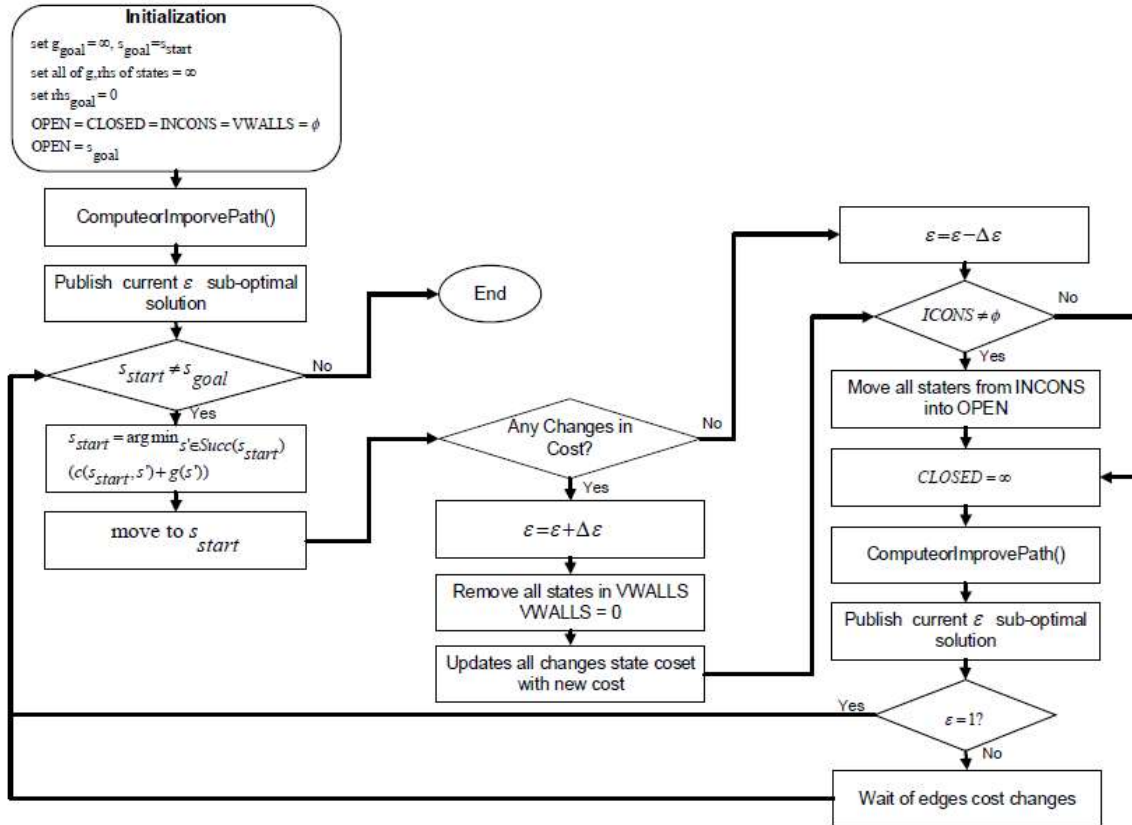
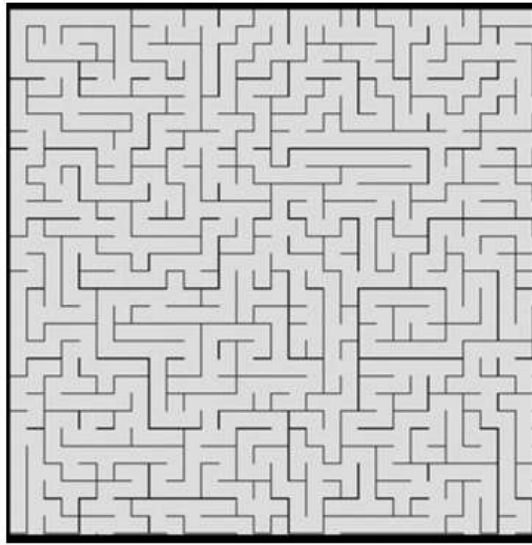


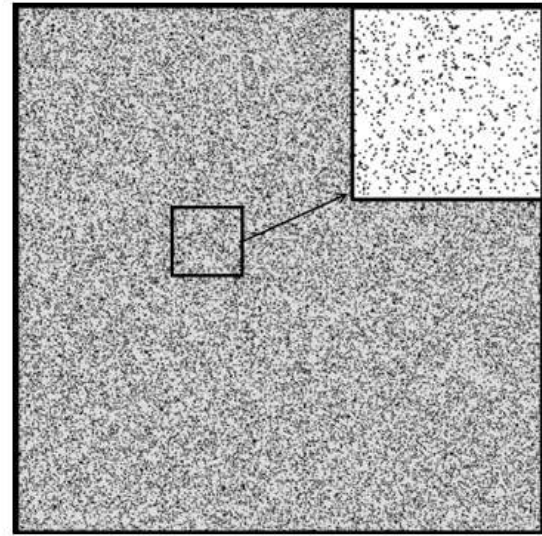
그림 54 iADA* flowchart

(가) iADA*를 이용한 실시간 경로 재계획 및 최적화의 개선 및 테스트 내용

- ▶ 테스트 결과는 D*, D* Extra Lite, MPGAA* 및 ADA *의 4가지 알고리즘을 통해 얻어진 결과와 비교하였음.
- ▶ 테스트는 16GB RAM이 장착 된 3.60GHz @ Intel® Core™ i7-7700 CPU 하드웨어를 이용하여 Ubuntu 16.04 운영 체제 상에서 수행하였음.
- ▶ 모든 알고리즘은 2D 격자 기반 경로 계획을 위한 큐 및 도메인 특화 함수(domain-specific function)에 대한 동일한 적용 내에서 C ++로 구현되었음.
- ▶ 다음 그림에서 보여주고 있는 벤치마크 문제 및 맵 세트들은 2012년의 Sturtevant의 연구결과를 참고함.
- ✓ maze16 : 16 개의 맵 셀의 회랑 너비를 가진 미로
- ✓ random15 : 맵 셀의 15%가 랜덤으로 배치된 장애물
- ▶ 랜덤 타입 맵 세트의 경우 관찰된 장애물은 경로에 크게 영향을 미치지 않음
- ▶ 단, 미로 타입 맵 세트는 사소한 변경이 있더라도 멀리 우회하는 상황이 발생할 수 있는데 그 이유는 이러한 종류의 맵에는 많은 막다른 골목이 포함되어 있기 때문임.

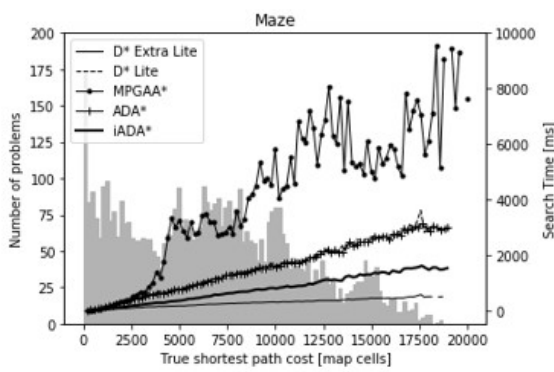


(a)

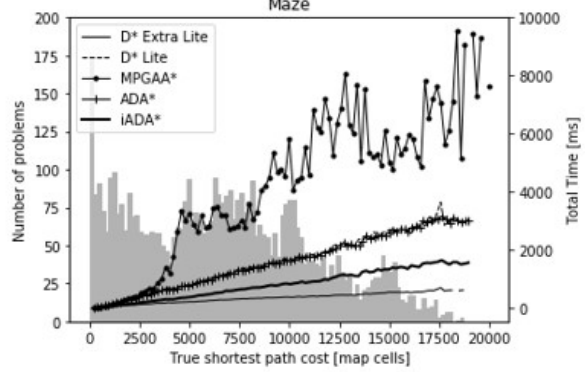


(b)

그림 55 Sample maps used from the map sets for testing (a) maze map (b) random15 map



(a)



(b)

그림 56 Running times for planning in mazes type map set: (a) Search time and (b) Total time ; in the background the histogram of problems plotted in gray.

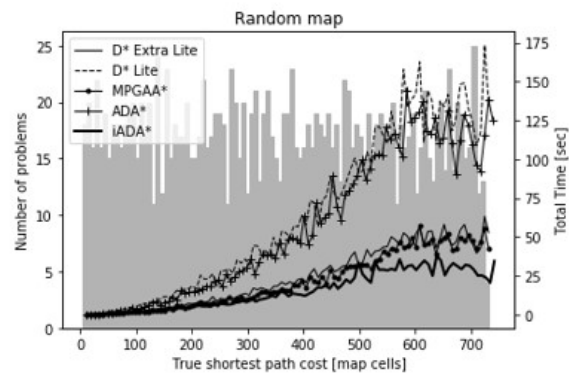
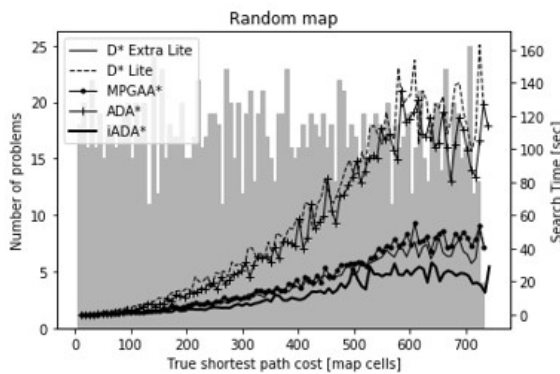


그림 57 Running times for planning in random15 type map set: (a) Search time and (b) Total time ; in the background the histogram of problems plotted in gray.

표 20 Average experimental results for planning of the algorithms on maze and random type map sets

Algorithm	Maze			Random		
	Search Time [ms]	Total Time [ms]	Path Cost	Search Time [ms]	Total Time [ms]	Path Cost
D*Extra Lite	212.69	253.78	6761.572	17.88	23.59	362.399
D*Lite	1137.03	1152.51	6761.572	58.02	62.98	362.376
MPGAA*	2986.64	2988.86	6737.989	20.32	20.98	361.891
ADA*	1132.67	1148.59	6902.349	51.15	56.1	370.125
iADA*	561.91	571.56	6894.372	11.67	15.16	369.061

- ▶ 본 결과를 통해 iADA* (improved Anytime Dynamic A*)의 적용은 기존의 경로 계획 알고리즘을 적용할 때에 비해 그 성능이 전반적으로 향상되었음을 확인할 수 있음.
- ▶ 획득한 데이터에서 iADA*는 무인 비행체를 위한 최적의 경로와 빠른 계산 결과를 제공함.
- ▶ 두 가지 사례 연구는 미로 및 무작위 형태의 512×512 크기의 맵에서 수행하였으며 각 맵 유형에 대해 총 5000개의 실행을 통한 테스트를 수행하였음.
- ▶ iADA*는 미로 형태의 맵에서는 D* Extra Lite 이외의 다른 알고리즘들에 비해 우월한 성능을 보임.
- ▶ 랜덤 형태의 맵에서는 iADA*는 **다른 모든 알고리즘보다 평균 3배 더 빠른 속도**를 나타내었음.
- ▶ 오리지널 ADA*와 비교할 때 본 연구를 통해 개발된 **iADA* 알고리즘은 랜덤 형태 맵에서 속도가 약 3.7배, 미로 형태 맵에서는 2.0 배 더 빠른 것**으로 나타남.
- ▶ 흥미롭게도 iADA*의 경로 비용은 모든 형태의 지도에서 ADA*에서 소요되는 경로 비용보다 낮은 것으로 나타남.
- ▶ iADA* 알고리즘은 특히 동적 장애물이 많은 대규모의 맵에서 강점을 보이는 것을 확인할 수 있었음.
- ▶ 본 알고리즘을 이용해 UAV가 불특정 다수의 정적/동적 장애물이 존재하는 공간에서 이를 인지하고 회피하기 위한 최적의 경로를 실시간으로 만들어 낼 수 있음.
- ▶ iADA* 알고리즘을 통해 비용계산 방법, 검색 시간 및 목표까지 도달하는 총 시간에 대한 계산 방식을 개선하고, 지정되지 않은 지형 환경에 존재하는 목표 탐색을 위해 신속하고 효율적인 경로 재계획을 수행할 수 있음.

(나) iADA* 알고리즘의 성능 검증

● 평균 실행 시간(Mean Execution Time, t_m)

- ▶ 알고리즘의 평균 실행 시간은 ms 단위로 표현됨.
- ▶ 본 알고리즘은 최적의 경로를 효율적으로 구성해야 하므로 최소의 시간에 이를 달성하는 것이 중요함.

◎ 실행 시간의 상대 표준 편차(Relative Standard Deviation, RSD) : RT

- ▶ 다음은 알고리즘 실행 시간에 대한 상대 표준 편차로 이의 수학적 표현은 다음과 같음.

$$RT = \frac{\sqrt{\frac{1}{(N \cdot K)} \cdot \sum_{t=1}^{N \cdot K} (t_i - t_m)^2}}{t_m}$$

- ▶ 여기서, t_i 는 실험 중 특정 방법을 이용하여 실시한 i 번째 실행 시간을 의미함.

◎ 평균경로 거리(Mean path distance, D_m)

- ▶ D_m 은 셀에서 생성 된 경로의 평균 거리를 의미함
- ▶ 평균 경로 거리가 경로 상의 이동 시간과 직접적으로 관련이 있는데 이는 상황에서의 탐색에서 최대의 관심사이므로 매우 중요하다.
- ▶ D_m 은 각 경로의 거리 D 의 평균으로 계산되며 이때 경로 D 는 다음과 같이 도출함.

$$D = \sum_{i=0}^{PathSize-1} Dist(Path_i, Path_{i+1})$$

- ▶ 여기서, $Path_i$ 는 각 경로를 구성하는 i 번째 부 목적지, $PathSize$ 는 경로상에서의 부 목적지의 개수를, D 는 각 경로의 거리를 의미한다.
- ▶ $Dist(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$ 는 p 와 q 사이의 유클리드 거리(Euclidean distance)임.

◎ 성공률(Success Rate, SR)과 실패율(Failure Rate, FR)

- ▶ 다음으로 가장 중요한 측정 지표는 각 알고리즘에 대한 성공률로 즉 a) 환경적 조건, b) 비행체와 목표의 위치에 관계없이 경로를 도출할 수 있는 능력을 의미.
- ▶ 경로 계획 알고리즘은 3차원 상에서 a) 실행시간의 초과, b) 비좁은 통로 또는 지나갈 수 있는 영역이 없을 때와 같은 주변 환경의 제약조건에 의해 경로의 도출에 실패할 수 있음.
- ▶ 따라서 이런 조건에서 특정 환경에 대한 알고리즘의 성공률은 다음과 같이 계산함.

$$SR = \frac{S}{N \cdot K} \cdot 100\%$$

- ▶ 여기서, N 은 알고리즘이 경로를 생성해야만 하는 연속적인 목표의 개수를, K 는 요소의 ID를, S 는 경로 생성의 성공 개수를 의미함.
- ▶ 앞서 정의한 성공률에 대해 실패율은 다음과 같이 정의 됨.

$$FR = 100\% - SR = \left(1 - \frac{S}{N \cdot K}\right) \cdot 100\%$$

- ▶ 따라서 본 연구의 성과지표로 제시한 실패율은 성공률과 상기의 관계를 가진다는 것을 고려하면서 성과 지표를 확인할 필요가 있음.

(4) 최적 경로 생성 알고리즘 iADA*와 DQN 기반 장애물 인식 회피 기술의 통합

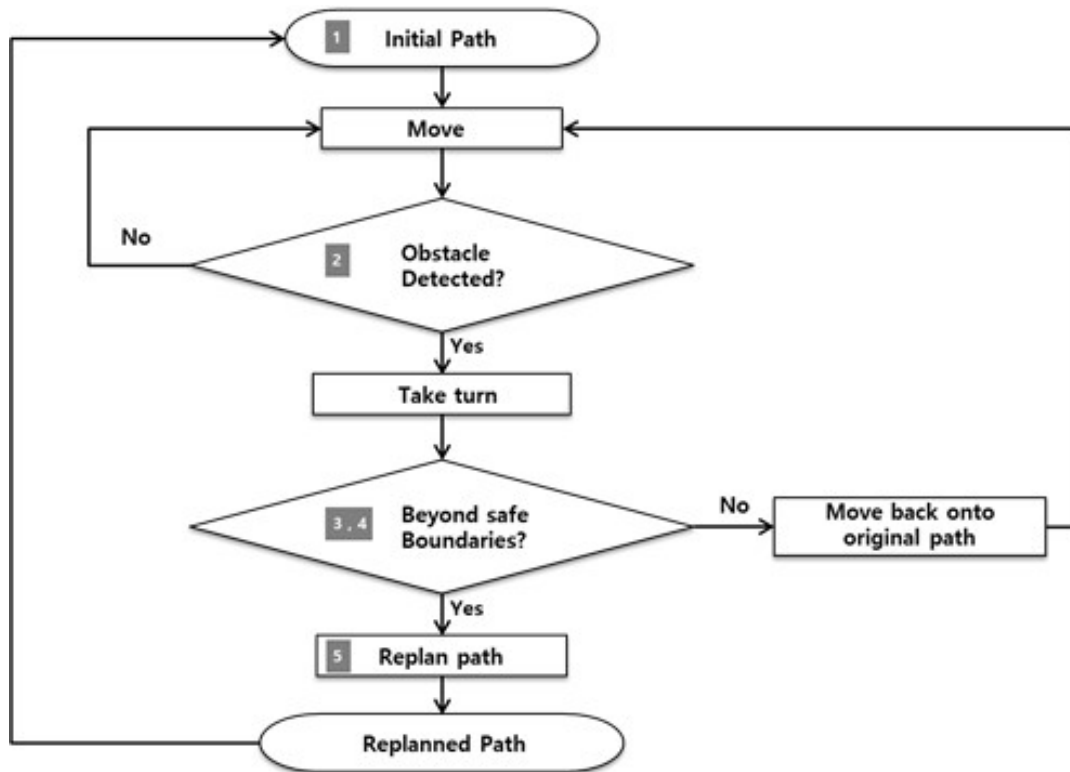
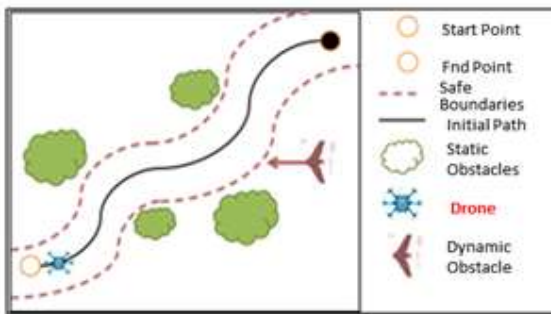
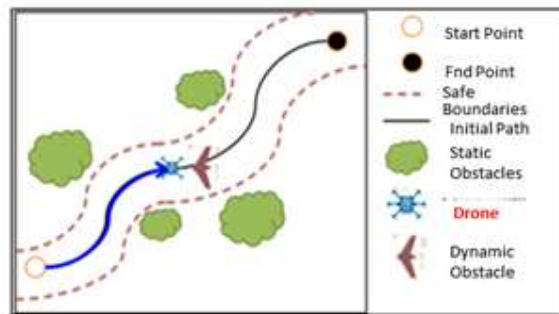


그림 58 Data flow chart between integration of iADA* and DQN

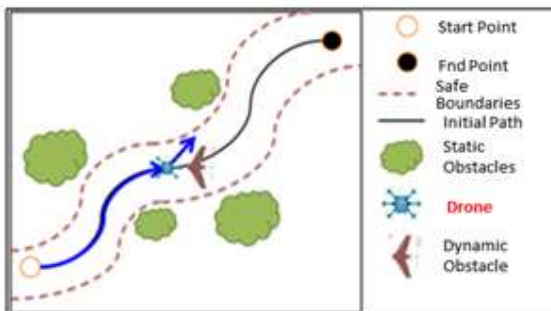
- ▶ 그림 54에서는 iADA*와 DQN을 통합하여 운용할 때의 데이터 흐름을 보여주고 있음
- ▶ DQN의 경우, iADA*에 의해 생성된 경로/궤적을 입력 매개 변수로 간주함.
- ▶ 또한 경로 생성기가 경로를 다시 계획하거나 DQN 에이전트가 생성한 행동 동작을 단지 따라야 하는지 확인하기 위한 안전 경계 여백(Safe Boundary Margin)이 포함되어 있으며 이 행동은 항상 그림 54상의 Step 1 초기 경로로 돌아가야 함.
- ▶ 그림 55에서는 iADA*와 DQN의 역할 및 각 알고리즘에 존재하는 작업의 차이를 단계별로 시나리오로 도식적으로 기술하였음.



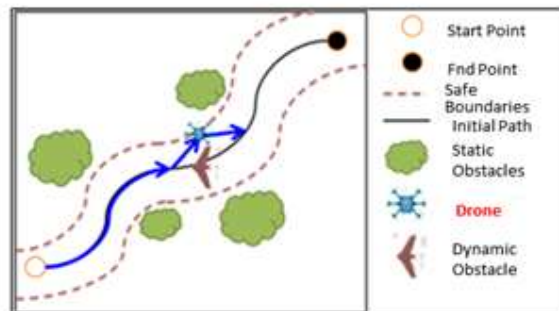
Step-1 : Initial path planned by path planning agent (iADA*) with safety boundaries margin.



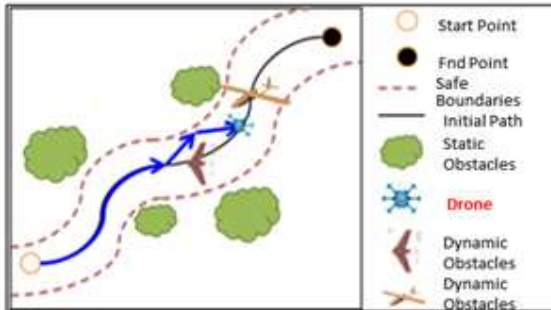
Step-2: Dynamic obstacle detected by DQN Agent on the way of the UAV



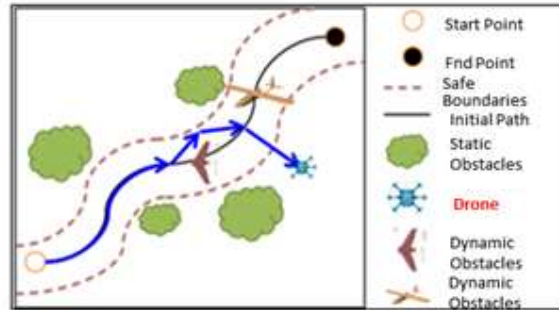
Step-3 : (i) Obstacle avoided maneuver within safe boundaries by DQN agent



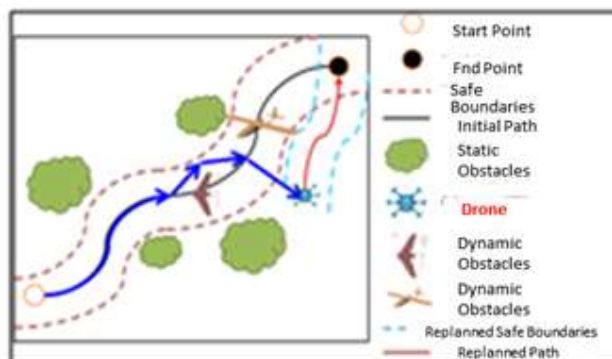
(ii) Drone must move back onto the original planned path



(iii) Another obstacle detected on the way: DQN



Step-4: Obstacle avoided maneuver by DQN gave beyond safe boundaries



Step-5 : Path replanning by iADA*

그림 59 The scenario description about the role and differences between iADA* and DQN algorithms

(5) 3차원 모델을 위한 시뮬레이션

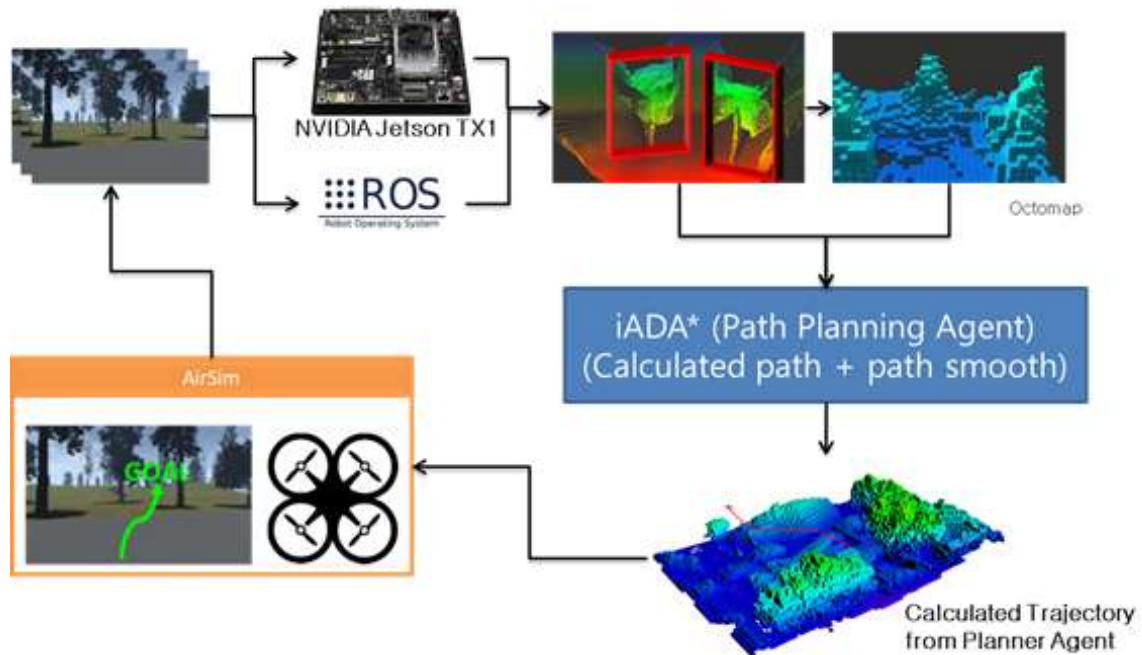


그림 60 OctoMap generation for the path planner to generate the path

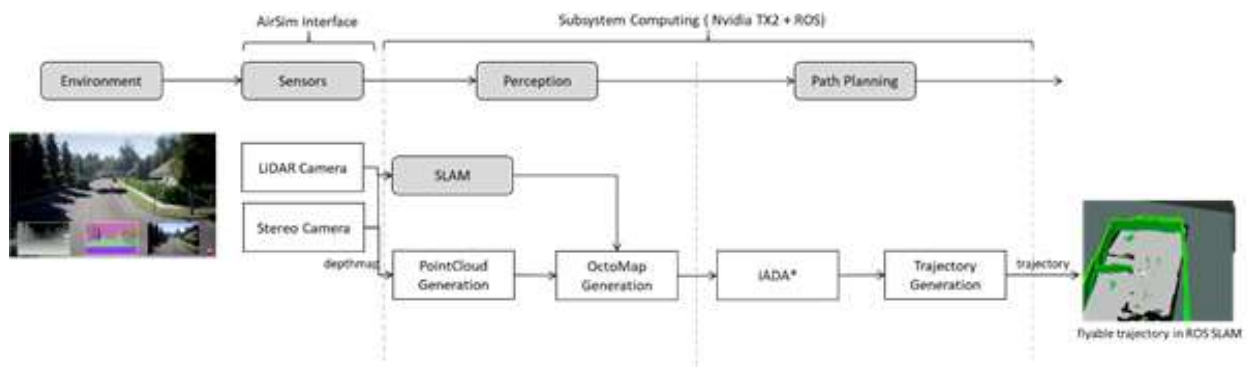


그림 61 Detail steps and processes how we going to generate the trajectory from iADA* in the 3D environment.

◎ Step 1. 3차원 환경을 위한 환경 맵(Environment map) 작성

- ▶ 기존의 2차원 환경이 아닌 3차원 환경에서 경로 계획기를 이용하여 경로를 생성하기 위해서는 우선 시스템에서 환경 맵의 작성 수행

◎ Step 2. 3차원 환경에서의 환경 맵 작성을 위한 깊이 정보(depth information) 이용

- ▶ 3차원 환경에서의 환경 맵 작성을 위해 일반적으로 UAV에 장착된 환경 데이터 센싱을 위한 스테레오 카메라 센서에서 깊이 정보(depth information)를 이용해야함.
- ▶ 본 과제에서 시뮬레이션을 위해 사용하는 AirSim에서는 이를 포함한 다양한 센서의 종류를 지원하고 있으므로 AirSim 프로그램을 통해서도 충분히 실제와 유사하게 모사할 수 있음.

◎ Step 3. SLAM 기능을 사용한 OctoMap 생성

- ▶ 다음 단계로는 ROS (Robotic Operating System)의 SLAM (Simultaneous Localization

and Mapping) 기능을 사용하여 OctoMap으로 불리는 경로 계획을 위한 점유 격자(occupancy grid)를 생성

- ▶ 본 연구에서는 업데이트 가능하고 유연하며 컴팩트한 3차원 맵을 제공하기 위해 OctoMap 방식을 점유 맵(Occupancy map)생성기로 사용.
- ▶ OctMap을 생성 한 뒤, iADA* 경로 생성기는 충돌이 없는 경로를 생성하며 이 경로 정보는 장애물 회피 에이전트에 입력 데이터로 제공됨.

◎ Step 4. iADA*를 이용한 경로 생성

- ▶ 본 연구에서 적용한 iADA*는 경로를 계산하는 경로 계획기(Path planner)로 경로를 생성할 수 있는 점유 격자 정보가 있다면 2D와 3D와 관계없이 어떤 환경이든 경로 생성이 가능함.
- ▶ iADA* 알고리즘에서 사용되는 lookahead function, $v(s)$ 는 가장 가까운 경로를 추정하기 위해 유클리드 거리에 기반을 두므로 3D 환경에서 이용하기 위해 확장이 용이함.

$$v(s) = \begin{cases} 0 & \text{if } s = s_{goal} \\ \min_{s' \in Succ(s)} (c(s, s') + g(s')) & \text{otherwise} \end{cases}$$

- ▶ 여기서, $Succ(s) \in S$ 는 후속 상태(successor)를 의미하며 또한 s로부터 단지 한 단계 떨어져 위치해 있으므로 인접상태(adjacent state)라고도 부를 수 있음

◎ Step 5. 장애물 회피 에이전트의 훈련

- ▶ 장애물 회피 에이전트(DQN)는 시각 데이터(Vision data)와 UAV의 현재 위치 및 속도 그리고 iADA *의 초기 경로 정보를 기반으로 훈련을 실시함

◎ Step 6. iADA*기반 경로 계획기와 DQN 기반 장애물 회피 에이전트의 통합

- ▶ 장애물 회피 에이전트 (DQN)는 역시 이미 3D 환경에서 작동함을 확인했으므로 이 두 가지 에이전트를 통합하는 것을 최종적으로 수행할 계획임.

◎ 정리하면 iADA*의 경우, 2D와 3D와 관계없이 경로를 생성 가능하므로 이에 대한 **알고리즘적인 수정은 필요하지 않음.**

◎ 다만 현재의 프로그램은 2D 균일 격자(uniform grid)를 기반으로 하므로 3D 맵 정보를 저장하도록 **프로그램 구현 시 확장을 수행.**

3) 2차년도 연구 개발 내용

가) 장애물 인식 및 회피 인공지능 에이전트 개발(2차년도 계속)

- ▶ 1차년도에 수행한 심층 강화 학습기반 장애물 회피 네트워크를 기반으로 지역 경로 계획 수행이 가능한 장애물 탐지 및 회피 인공지능 에이전트 개발을 수행함.

(1) 강화 학습기반 인공지능 에이전트 연구

◎ Q-Learning

- ▶ Q-Learning은 1989년 Watkins에 의해 개발된 가장 유명하고 기본적인 강화학습 알고리즘임.
- ▶ Bellman 방정식을 기반으로 Q-value 표인 $Q(s,a)$ 값을 업데이트해 MDP의 최적의 정책을 찾는 것이 가능함.

$$Q(s,a) = r + \gamma(\max_{a'}(Q(s',a')))$$

- ✓ r : 즉각적인 보상(reward)
- ✓ $\gamma(\max_{a'}(Q(s',a')))$: successor state(후속 상태)에서의 감쇄된 최대 Q-value, $\gamma \in [0,1]$
- ✓ s' : successor state(후속 상태)
- ✓ a' : s' 에서 최대 Q-value를 얻었을 때의 행동(action)

◎ Deep Q-Network(DQN)

- ▶ Bellman 방정식과 같은 선형 방법을 사용하여 RL 알고리즘의 문제를 해결하는 것은 경우에 따라 효율적이고 간단하지만 행동-가치 함수(action-value function)를 반복적으로 업데이트 하는 상황에는 적용되지 않음.
- ▶ Google DeepMind와 같은 기업은 데이터 셋 D로 표현되는 경험 리플레이 메모리(experience replay memory)를 사용하는 Q-Learning 알고리즘을 CNN을 활용하여 학습시키는 DQN이라는 딥 러닝 모델을 제안함.
- ▶ 각 단계에서 에이전트의 경험을 데이터 셋 D에 저장할 수 있으며, 업데이트를 수행하는 동안 경험의 미니배치(mini-batch)가 무작위로 샘플링됨.
- ▶ Atari 게임의 경우 에이전트는 상태가 $s_t = x_1, a_1, x_2, a_2, \dots, a_{t-1}, x_t$ 로 나타나는 에뮬레이터(emulator)와 상호작용 해야 하며, 이 상태는 Q-네트워크라고 알려진 신경망 입력으로 변환 됨.
- ▶ 에이전트의 출력은 유효한 각 작업에 대해 Q 값을 예측하는 것이며 Q-네트워크의 학습을 위해 사용되는 손실함수(loss function)는 다음과 같이 나타남.

$$L_i(\theta_i) = (r + \gamma(\max_{a'}(Q(s',a';\theta_i^-)) - Q(s,a;\theta_i))^2$$

- ✓ $r + \gamma(\max_{a'}(Q(s',a';\theta_i^-)))$: 반복 i의 목표값.
- ✓ θ_i^- : 이전 반복의 가중치. 확률적 경사 하강법(stochastic gradient descent)을 활용해 손실함수를 최적화할 때 고정되는 값.
- ✓ 확률적 경사 하강법은 기울기를 계산하기 위한 배치(batch)의 크기를 1로 하며, 이때 선택되는 배치를 반복에서 무작위로 선택하는 방법임.
- ▶ 다음은 gradient를 계산하는 방정식을 나타냄.

$$\Delta_{\theta_i} L_i(\theta_i) = (r + \gamma(\max_{a'}(Q(s', a'; \theta_{i-1})) - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i)$$

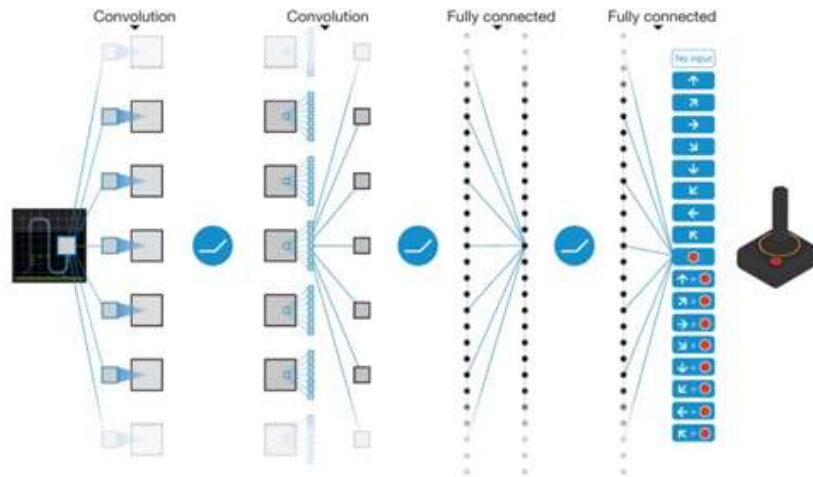


그림 62 The network architecture of DeepMinds Atari game using DQN

- ▶ DQN의 기술적 특징은 크게 다음과 같음.
 - ✓ 이미지 인식을 위한 합성곱 신경망 이용.
 - ✓ 샘플 간 상관 관계를 없애고, 샘플 효율성을 높이기 위한 경험 리플레이 도입.
 - ✓ 학습의 안정성을 위해서 에이전트의 행동을 결정하는 online Q-network와 target Q 값 계산에 사용되는 target Q-network 분리.

◎ Double Deep Q-Networks(DDQN)

- ▶ DQN은 과대평가로 인해 비현실적으로 높은 행동 가치를 학습하는 경우 존재한다는 단점이 있음.
- ▶ DDQN은 타겟의 행동 선택과 행동 평가를 분리하여 과대평가를 줄이기 위해 도입됨.
- ▶ Google DeepMind에 의해 제안된 DDQN은 두 개의 신경망으로 이루어진 Q-value 추정기(estimator)를 제안했으며 각 신경망은 서로 다른 생태에 의해 업데이트됨.
- ▶ DDQN은 타겟 네트워크로 알려진 target Q-value를 추정하기 위한 Q-네트워크를 가짐.
- ▶ 또 다른 Q-네트워크는 예측 Q-값(predicted Q-value)을 추정하기 위해 사용되며 로컬 네트워크 또는 예측 네트워크라고 불림.
- ▶ 타겟 네트워크는 로컬 네트워크와 동일한 아키텍처를 가지며 타겟 네트워크를 업데이트 하기 위해 동일한 매개변수가 초기화됨.
- ▶ DQN과 동일하게 타겟 네트워크를 업데이트 하나, DDQN의 타겟 네트워크의 타겟 Q-값은 다음과 같은 식으로 표현됨.

$$y_i = r + \gamma Q(s', \max_{a'}(Q(s', a; \theta); \theta_i^-))$$

- ✓ 행동은 각 단계에서 가중치 θ_i 에 의해 탐욕적으로 선택되며, Q-값은 타겟 네트워크 θ_i^- 에 의한 행동에서 할당됨.
- ▶ DDQN은 과대평가의 영향을 감소시키며, 이는 타겟 네트워크를 다소 고정된 상태로 유지하기 때문에 일반적인 DQN에 비해 성능을 향상시키고 보다 안정적인 학습이 가능함.

● 우선적 경험 리플레이(Prioritized Experience Replay)

- ▶ DQN은 경험 리플레이로부터 경험을 균일하게 추출하여 학습함.
- ▶ DDQN의 우선적 경험 리플레이는 무작위 샘플링된 경험에서 리플레이 메모리에 저장된 각 경험의 우선순위에 따라 경험을 선택하는 방법을 변경함.
- ▶ 리플레이 메모리에서 각 경험의 우선순위 값은 시간차(TD) 오류를 이용해 계산됨.

$$\delta_i = r + \gamma Q(s', \max_{a'} (Q(s', a'; \theta); \theta_i^-)) - Q(s, a)$$

- ▶ 여기서 δ_i 는 타겟 값 $y_i = r + \gamma Q(s', \max_{a'} (Q(s', a'; \theta); \theta_i^-))$ 와 Q의 차이로 현 상태에서 다음 δ_i 가 가장 큰 다음 상태로의 전환은 메모리에서 반복되며 이를 순수 탐욕 우선순위라고 함.
- ▶ 다양성의 부족을 피하기 위해 완전히 탐욕스러운 우선순위와 균일한 무작위 샘플링 사이를 보간하는 확률적 우선순위를 개발하여 샘플링하며 샘플링 전환확률 i 는 다음과 같이 정의됨.

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

- ✓ α 는 확률적 우선순위화에 사용되는 $[0,1]$ 사이의 값이며, 순수 탐욕 우선순위는 사용되지 않고 $\alpha=0$ 일 경우 모든 경험이 무작위로 균일하게 샘플링됨.

(2) 학습기반 지역 계획 및 장애물 회피 에이전트 개발

(가) 지역 경로 계획 및 장애물 회피 문제

- ▶ UAV는 센서 데이터의 환경 정보를 추출하고 운용 명령을 생성하여 동적 환경에서 충돌이 없는 경로를 계획해야 함.
- ▶ 드론은 **경험을 통해 학습하고 학습경험으로부터 네트워크의 파라미터를 업데이트해 수학적 모델을 생성**함. 문제는 다음과 같은 공식으로 표현이 가능함.

$$(v_t, \omega_t) = f(O_t)$$

✓ 이때, v_t, ω_t, O_t 는 선형 속도, 각속도 및 각 시간 단계에서 센서의 관측값을 의미함.

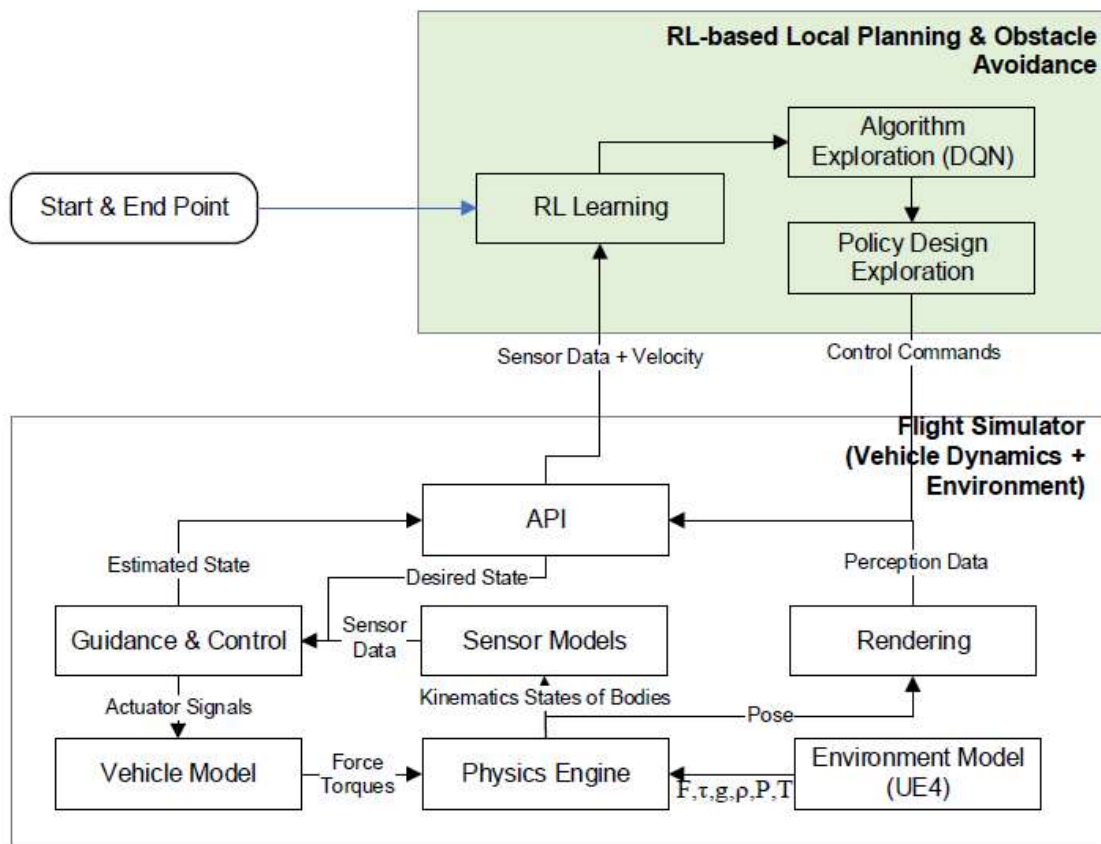


그림 63 Process for local planning, obstacle avoidance training, and learning

(나) 강화학습 기반 지역 계획 및 장애물 회피

- ▶ 신경망은 심층강화학습 문제에서 강화학습 과정을 개선하기 위한 action-value function(행동-가치 함수)를 추정하기 위한 비선형 함수 추정기로 활용됨.
- ▶ 장애물 회피 문제는 마르코프 결정 과정(MDP)으로 정의되며 튜플 (S, A, P, R, γ) 은 상태 공간 $S(s \in S)$ 에서 가능한 모든 상태 s 를 의미하는 MDP를 표현하는 데 사용됨.
- ▶ 행동공간 A 에는 가능한 행동 $a(a \in A)$, 상태 전이 모델 M , 보상함수 R 및 감가율 γ 가 포함됨.
- ▶ 두 개의 은닉층을 포함한 신경망이 가중치가 θ 인 action-value function $Q(s, a; \theta)$ 를 추정

하기 위한 비선형 함수 근사값으로 사용됨.

- ▶ 전이 과정(transition)과 같은 방대한 상호작용 데이터는 신경망 훈련을 위해 수집되어야 하며, 본 연구에서는 AirSim을 활용해 데이터를 수집하고 신경망을 훈련함.
- ▶ 각 시간 단계 t 에서 비행체는 현재 상태 s_t 를 입력으로 신경망에 전송하며, 행동 값을 출력으로 제공함. 비행체는 ϵ -탐욕 정책(ϵ -greedy policy)과 같은 정책에 따라 행동을 선택하며 ϵ 값은 감쇠율 β 에 따라 1에서부터 감소함.

$$\epsilon_{k+1} = \beta_{\epsilon k}$$

✓ k : epoch

✓ ϵ : ϵ -greedy policy에 사용되는 특정 상태의 확률로 0.05에 도달하면 감소를 멈추고 고정.

- ▶ **선택된 행동 a_t 를 수행한 뒤에 비행체는 새로운 상태 s_{t+1} 가 되며 보상 r_{t+1} 을 받으며 네트워크 업데이트를 위해 모든 전이 과정이 기록됨.**

(다) 강화 학습 구현 세부 사항

- ▶ 네트워크는 30×100 크기(pixels x pixels)의 상태표현, 현재 위치, 그리고 속도를 입력을 사용.
- ▶ 첫 번째 층은 8×8 (pixels x pixels)의 32개 커널을 스트라이드 값을 3으로 해 컨볼루션을 진행.
- ▶ 두 번째 층은 4×4 (pixels x pixels)의 64개 커널을 스트라이드 값을 2로 해 컨볼루션을 진행.
- ▶ 세 번째 층은 3×3 (pixels x pixels)의 64개 커널을 스트라이드 값을 1로 해 컨볼루션을 진행.
- ▶ 네 번째 층의 출력은 위치벡터 및 속도벡터와 연결되며 2개의 완전 연결층(fully connected layer)와 연결됨. 본 실험에는 8개의 행동공간이 존재함.

● 행동공간

- ▶ 에이전트의 행동공간은 장애물을 피하는 8개의 불연속적인 행동으로 구성됨.
- ▶ 행동은 1, 2, 3, 4 m/s로 전진하거나, 1 m/s로 후진, 오른쪽 또는 왼쪽으로 30도 회전하거나 2 m/s의 속도로 상승 또는 하강하는 것을 포함함.
- ▶ 각 시간단계에서 정책은 관찰공간을 입력으로, 8가지 행동 중 하나를 출력으로 선택함.
- ▶ 높은 수준의 행동은 AirSim에서 제공하는 비행 제어를 사용하여 낮은 수준의 비행 명령으로 mapping.

● 상태공간

- ▶ 입력층에 사용되는 입력 상태는 다음 2가지로 구성됨.
 - ✓ Depth image 데이터
 - 상태는 에이전트가 실시간 장애물을 피하기 위한 정보를 제공 해야함.
 - 장애물을 회피하기 위해 에이전트는 전면 카메라로부터 depth image를 수신하고 iADA* 경로 계획 에이전트가 제공한 계획 경로 및 웨이포인트로 돌아와 드론의 현재 위치와 상태

를 수신함.

- 수신한 depth image는 256×144 (pixels x pixels) 크기의 그레이 스케일로 매핑된 카메라에서 보이는 장애물 표면과의 거리를 포함함.
- 본 연구에서는 계산 및 메모리 요구사항을 줄이기 위해 입력 차원을 줄이는 기본 이미지 처리 단계를 적용함.
- 일반적으로 드론은 이미지의 중간 부분을 사용하기 고정된 평면에서 이동하며, 전처리 된 depth image의 크기는 100×30 (pixels x pixels)임.

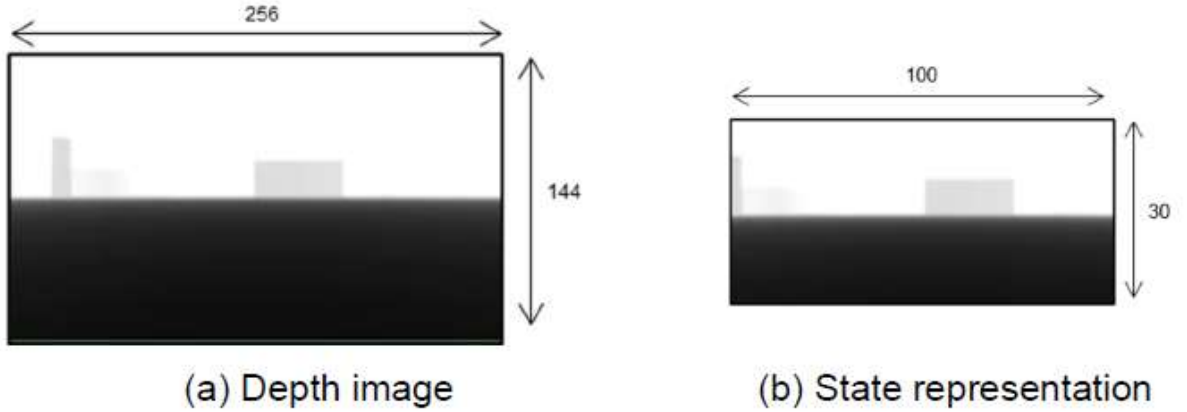


그림 64 State representation of depth image and preprocessed image

✓ 상대적인 목표지점과 현재 속도

- 비행체의 상대적인 전역 위치는 현재 위치에 대한 극좌표(거리 및 목표지점)로 목표를 나타내는 3D 벡터이며, 거리 $S_{distance}$ 와 목적지까지의 방향은 다음과 같은 식을 통해 계산됨.

$$S_{distance} = \| \rho \| = \| P_{current} - P_{goal} \|$$

- ▶ UAV가 각 목적지(Waypoint)에 도달하면 도착 보상(reward)이 제공되며, 본 연구에서는 이를 전체 맵 크기의 1.0%인 $S_{distance} < 0.5$ (m)를 달성하면 목적지에 도착한 것으로 간주함.
- ▶ 벽, 다른 UAV, 자동차와 같은 장애물과 충돌할 때 collision reward가 주어지며, 목적지에 접근하는 행동에 대해서는 positive reward, 목적지에서 멀어지는 행동에 대해서는 negative reward가 주어짐.

$$r_t = \begin{cases} 50 & arrived \text{ if } (S_{distance} < 0.5m) \\ -50 & collision \\ (S_{distance(t-1)} - S_{distance(t)}) * c_r & else \end{cases}$$

✓ $S_{distance(t)}$: 시간 “t”에서의 $S_{distance}$.

✓ c_r : 하이퍼 파라미터 효과를 약화하기 위해 사용하는 변수.

- ▶ 드론이 정적 또는 동적 장애물과 부딪히면 에이전트는 -50의 보상을 받으며 초기위치 [0,0,0]에서 새로운 episode가 시작됨.
- ▶ 네트워크의 가중치를 업데이트하기 위해서 손실함수(Lost Function)에서 미니배치 경사 하강법(Mini Batch Gradient Descent)을 선택함.

✓ 미니배치(mini batch)는 학습의 반복에서 함께 실행되는 전체 배치 중에서 무작위로 선택한 소규모 부분집합을 의미함.

✓ 미니배치 경사 하강법은 전체 배치 반복과 확률적 경사 하강법(stochastic gradient descent)의 절충안으로, 확률적 경사 하강법의 노이즈를 줄이면서 전체 배치에 대한 경사 하강법보다 효율적임.

▶ 미니배치 경사 하강법의 손실함수는 다음과 같음.

$$L(\theta) = \frac{1}{2n} \sum (y_i - Q(s_i, a_i; \theta))^2$$

✓ y_i : action-value function의 현재 목표 출력.

✓ n : 미니배치의 크기.

✓ 과대평가(overestimation)를 방지하기 위해서 에이전트는 타겟 y_i 를 다음과 같은 식을 이용해 업데이트함.

$$y_i = r_{i+1} = \gamma Q(s_{i+1}, a; \theta^-)$$

✓ r_{i+1} : a_i 행동 후의 즉각적인 보상.

✓ γ : 감가율(discount factor).

▶ 다른 신경망 $Q(s, a; \theta^-)$ 는 무작위 가중치 θ^- 로 초기화됨.

▶ 네트워크는 최대 값이 Q 인 동작을 선택하고 타겟 네트워크에서 선택된 행동 a 의 값은 신경망 Q 에서 결정되지만 a 의 값은 타겟 네트워크에서 결정됨.

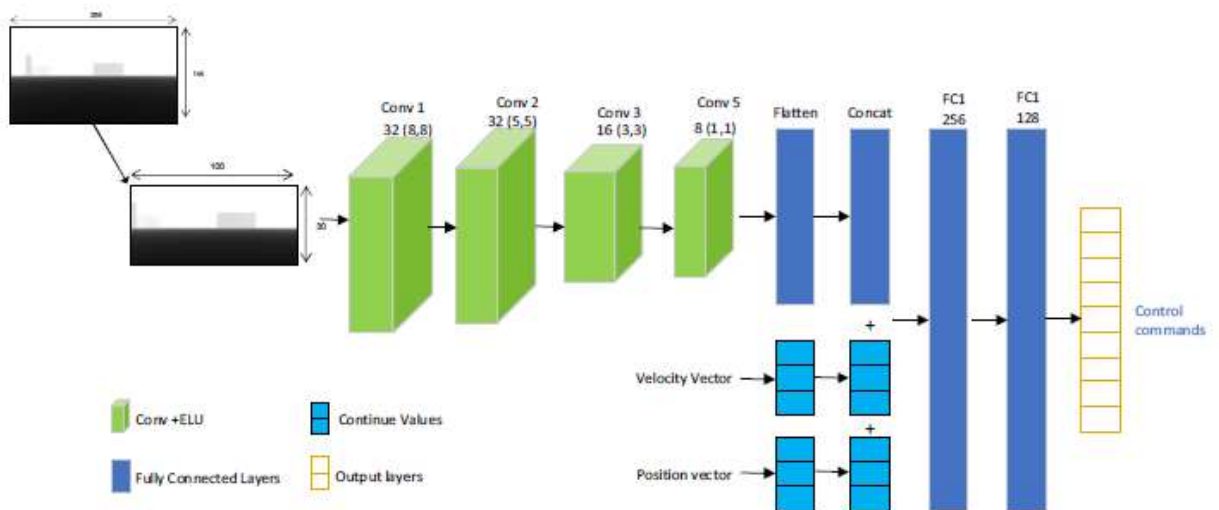


그림 65 The network architecture for the DQN and DDPG agents

▶ 위의 그림은 DQN 및 DDPG 에이전트의 네트워크 아키텍처를 나타냄.

▶ 두 에이전트는 depth image, 속도 및 계획된 경로 위치를 입력으로 간주하며, 본 장애물 회피 에이전트는 비행체의 목표지점을 고려하지 않고 장애물을 회피하도록 학습됨.

▶ DRL 에이전트의 주요 목표는 동적 장애물을 감지하고 회피하며, 경로 계획 에이전트가 생성한 경로로 다시 돌아오는 것임.

(3) 장애물 회피 시험 결과 및 분석

(가) Open AI Gym

- ▶ 강화학습은 환경에 있는 에이전트를 가정하며, 여기서 환경은 부분적으로 관찰 가능한 마르코프 결정과정(POMDP)로 공식화됨.
- ▶ OpenAI Gym은 에이전트의 경험이 일련의 에피소드로 나누어지는 강화학습의 에피소드 설정에 중점을 둠.
- ▶ 각 에피소드에서 에이전트의 초기 상태는 무작위로 샘플되며 환경이 최종상태에 도달할 때까지 상호작용이 이루어짐.
- ▶ 에피소드 강화학습의 목표는 에피소드 당 총 보상에 대한 기대치를 극대화하고 최대한 많은 에피소드에서 높은 수준의 성과를 달성하는 것임.
- ▶ OpenAI Gym은 에이전트 클래스를 포함하지 않으며, 에이전트가 사용해야 하는 인터페이스를 지정하지 않고 설명용으로만 포함함.



그림 66 Images of some environment which are part of OpenAI Gym

(나) 시험환경 설정

- ▶ 본 실험에서는 UE4 환경하에서 AirSim UE4 플러그인을 사용하여 정적 장애물을 포함한 환경을 구성함.
- ▶ 본 환경은 벽을 포함한 정적 장애물과 드론을 포함한 동적 장애물로 구성됨.
- ▶ 동적 장애물의 속도는 사용자가 정의한 최대 최소 속도 범위 내에 존재하며 환경은 동적 장애물의 속도를 범위 내에서 임의로 선택함.

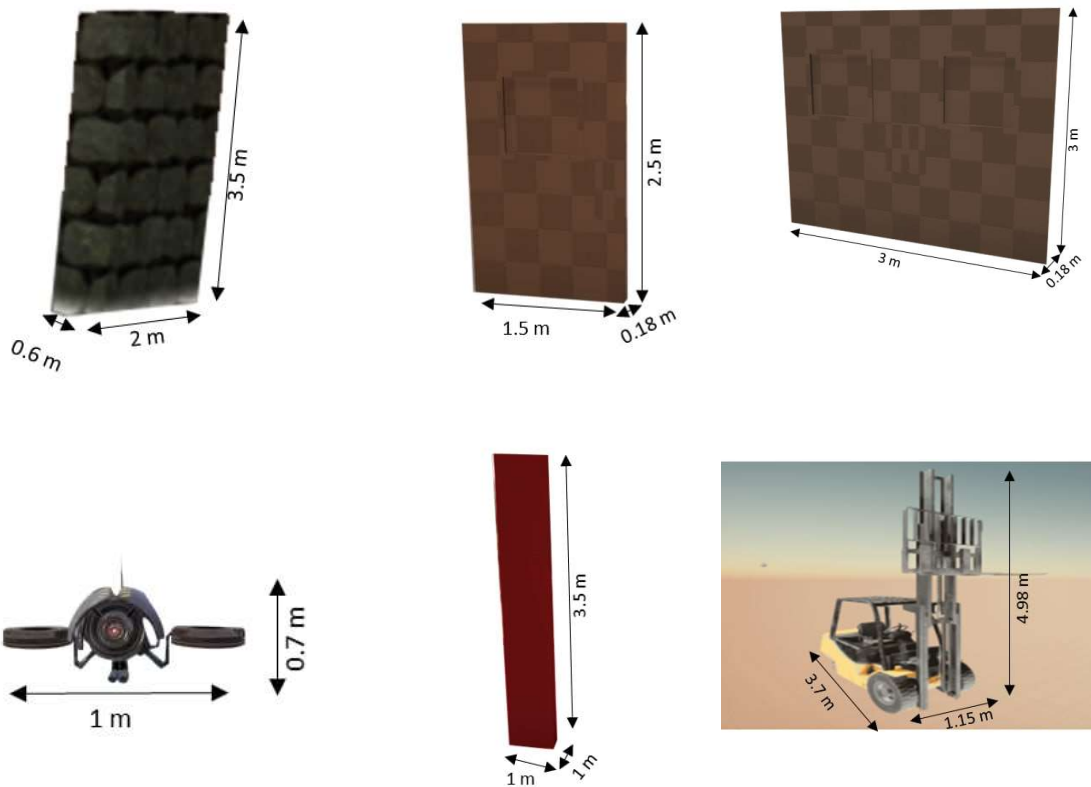


그림 67 Objects considered as obstacles in the environment

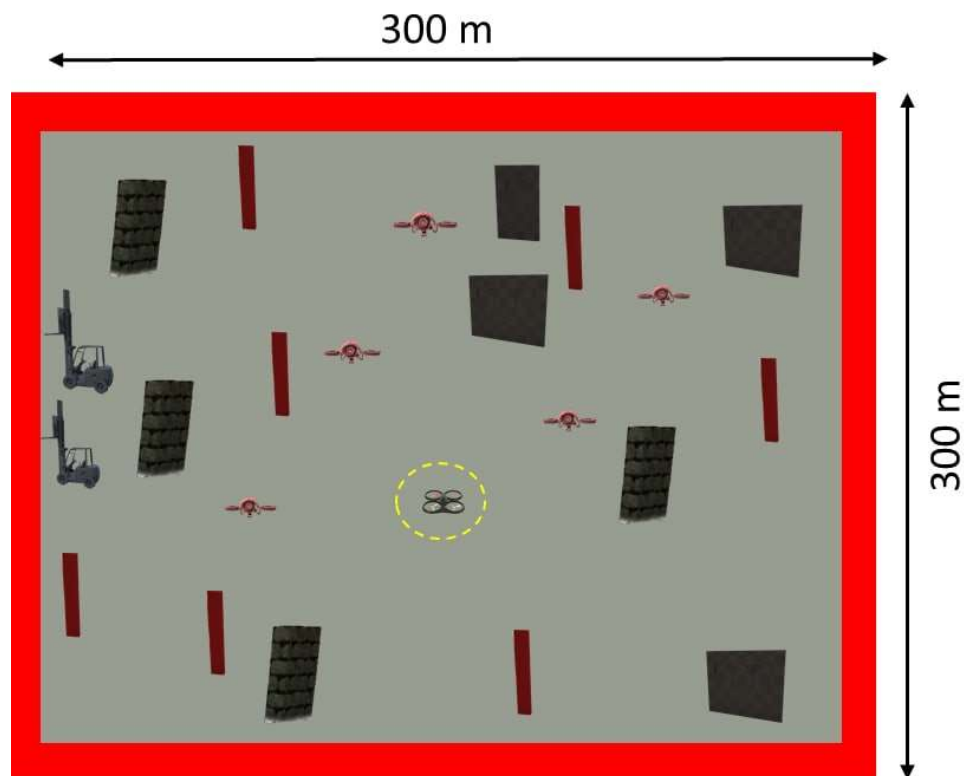


그림 74 A Top view of dynamic environment

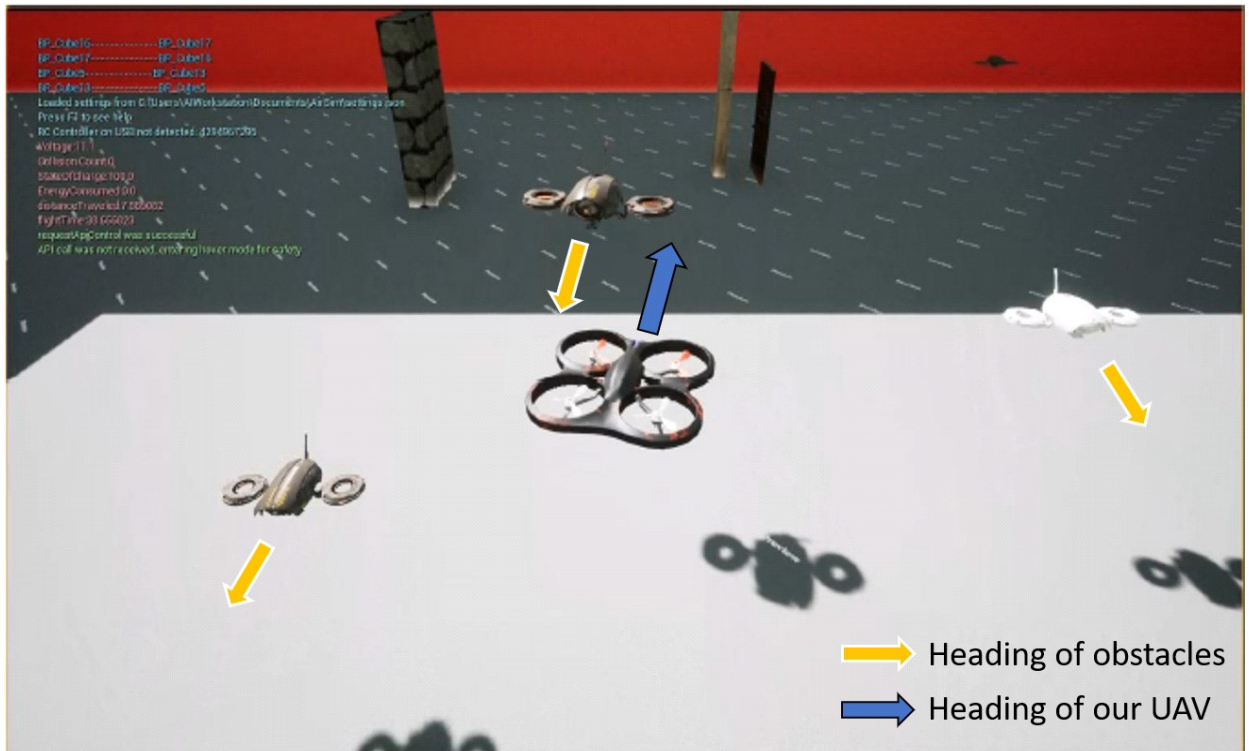


그림 75 3D view of dynamic environment

(다) 파라미터 설정 및 구현

- ▶ AirSim에서 DQN 알고리즘을 구현하기 위해 Keras-rl이 OpenAI Gym 에이전트로 사용됨.
 ✓ Keras-rl은 파이썬에서 심층 강화 학습 방법을 구현하였으며 C++로 작성된 심층 학습 모듈인 Keras와 통합됨.
- ▶ AirGym과 AirSim, OpenAI Gym, Keras-rl, tensorflow(GPU에서 수행을 위함)를 사용한 전체 에이전트 및 환경의 상호작용은 다음 그림과 같음.

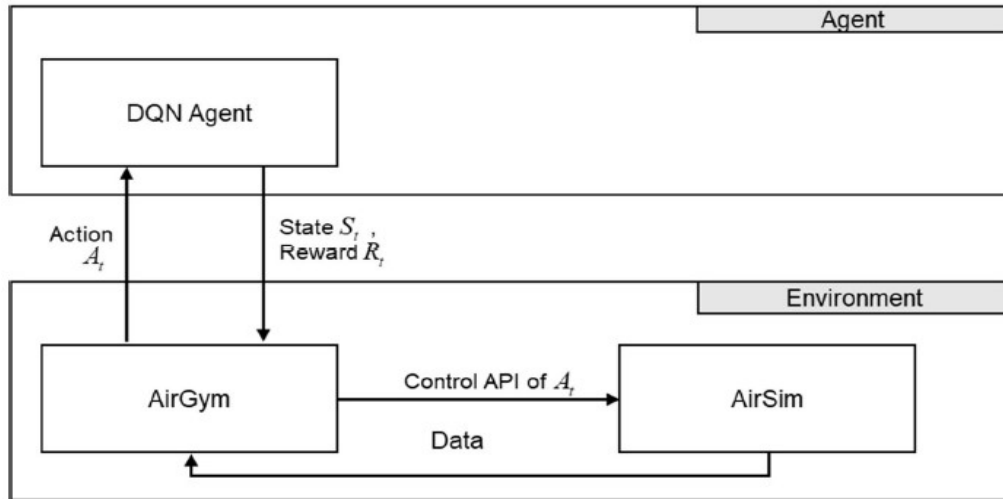


그림 64 Flow diagram for the interaction between agent and environment using AirGym system

- ▶ DQN 알고리즘의 기본 설정으로 전이 튜플(transition tuple)을 저장할 리플레이 버퍼의 크기는 200,000으로 설정되었으며, 경험 리플레이 버퍼를 사용하면 에이전트가 이전 정책의 경험에서 학습할 수 있고 기울기 단계에서 연관된 샘플을 피할 수 있음.
- ▶ 0.01로 고정된 타겟 계수 γ 이 있는 업데이트 된 타겟 네트워크가 추가되었으며, 학습과정의 안정성을 높이기 위해 사용됨.
- ▶ DQN 에이전트 네트워크는 DQN 알고리즘의 학습률에 권장되는 값인 $\alpha=0.0025$ 로 설정된 Adam 최적화 기법을 사용하여 훈련됨.
- ▶ 환경 탐색을 위해서 ϵ -탐욕 정책이 사용되었으며 이는 Q 값으로부터 확률 ϵ 에 따라 행동이 무작위로 선택됨을 의미함.

(라) 시뮬레이션 검증

- ▶ 강화학습 알고리즘은 알고리즘이 임무의 실패 여부를 의미하는 실패율을 기준으로 평가됨.
- ▶ 본 연구에서 장애물 회피 에이전트는 (나)에서 제시된 실험 환경에서 성공률과 보상 값을 기반으로 평가됨.
- ▶ 성과지표의 기준인 **실패율은 드론이 장애물과 충돌하여 임무를 완수하지 못한 횟수의 비율**로 이 숫자가 0%에 가까워질수록 에이전트가 좋은 성능을 보임.
- ▶ RL 알고리즘의 목표는 **보상 값을 극대화**하여 훈련과 테스트 모두에 대한 보상 값을 확인하는 것임.
- ▶ 본 에이전트에 대한 검증 결과는 라) 인공지능 에이전트 통합파트에서 제시함.

나) 시스템 모니터링 인공지능 에이전트 개발

- ▶ 에이전트는 수학적 함수를 이용하여 환경을 인지하고 어떤 행동을 취할지를 결정할 수 있는 능력을 가진 환경임.
- ▶ 시스템 모니터링 에이전트, 즉 시스템 모니터링 시스템은 **시스템의 결과 또는 이벤트를 측정하거나 시각화하는 과정**임.

- ▶ 시스템 모니터링은 시스템이 상당히 양호한 동작을 함을 보여주는 참조모델로 종종 사용됨.
- ▶ 기체 추적 모니터링 시스템은 다음과 같음.
 - ✓ 추적데이터는 GPS 위성에 의해 전송됨.
 - ✓ 기체의 실시간 정보를 관찰자에게 전달하기 위해 송신탑은 서버로 데이터를 전송함.
 - ✓ 이 과정을 통해 관찰자는 위험한 환경 또는 다른 기체를 피하거나 도착지점까지의 남은 거리 등의 정보를 제공함.
 - ✓ 사용자가 직접 타고 있는 경우, 또는 컨트롤러를 이용하는 경우 모두 시스템의 실시간 데이터 모니터링을 기반으로 기체를 제어할 수 있음.



그림 65 Vehicle tracking system

- ▶ 자율 비행체의 실시간 모니터링 기능은 안전한 운항을 위해 중요함.
- ▶ 시스템 모니터링 에이전트는 드론 비행속도, 비행모드 등의 환경 인지를 바탕으로 드론 전력 소모율과 드론의 위치 등을 결정함.
- ▶ 이동 거리, 기체의 위치, 이동 시간, 연료 소모량과 에너지 소모량 데이터는 자율 행 인공지능 모델의 안전시스템의 성능 향상을 위해 중요한 변수임.

(1) 방법론 제시

- ▶ UAV 시스템의 비행시간, 이동거리, 에너지 소모량, 배터리의 충전상태(SOC)는 시스템 모니터링을 통한 실시간 비행데이터 측정에 사용됨.
- ▶ 시스템 모니터링 에이전트의 전체 프레임워크는 다음 그림과 같음.

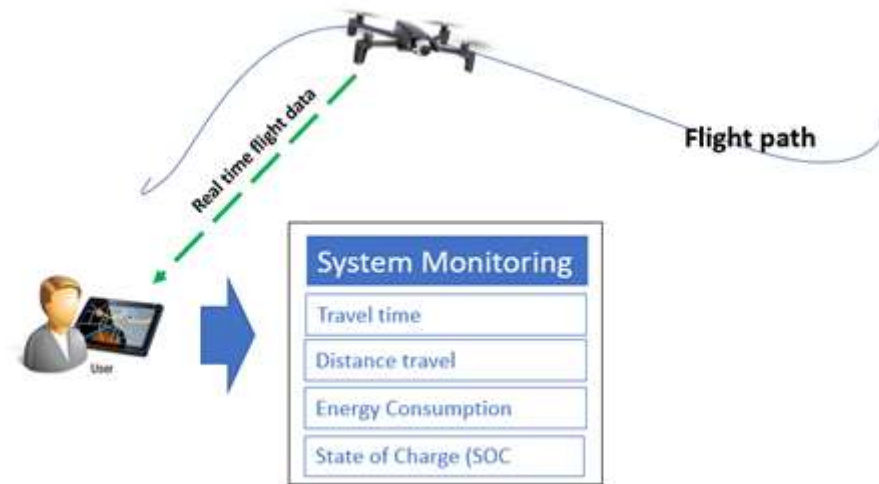


그림 66 Real time system monitoring of UAV

- ▶ 시스템 모니터링 에이전트에 사용되는 변수들의 추정방법은 다음 절에 기술됨.
- ▶ 실시간 비행 위치($\text{pose}(x,y,z)$)는 경로계획 모듈로부터 추정이 가능하며, 항력과 추력은 속도를 기반으로 한 현 위치를 이용하여 계산이 가능함.
- ▶ Anafi 드론의 필요 입력 형상 데이터는 BEMT(Blade Element Momentum Theory)를 이용한 드론의 power 소모량 계산에 사용되며, 배터리 모델은 현 비행 위치에서의 드론의 에너지와 배터리 충전상태를 추정하기 위해 사용됨.
- ▶ 배터리 충전상태 추정을 바탕으로 추정된 시스템 모니터링 에이전트의 결과 변수들은 AirSim의 GUI상에 나타나며, 더이상 남은 배터리가 없으면 드론이 더이상 비행을 할 수 없고, 재계획 또한 불가능하므로 임무 재계획 에이전트가 종료됨.
- ▶ 다음 그림은 시스템 모니터링 에이전트의 전체적인 방법론 구조를 나타냄.

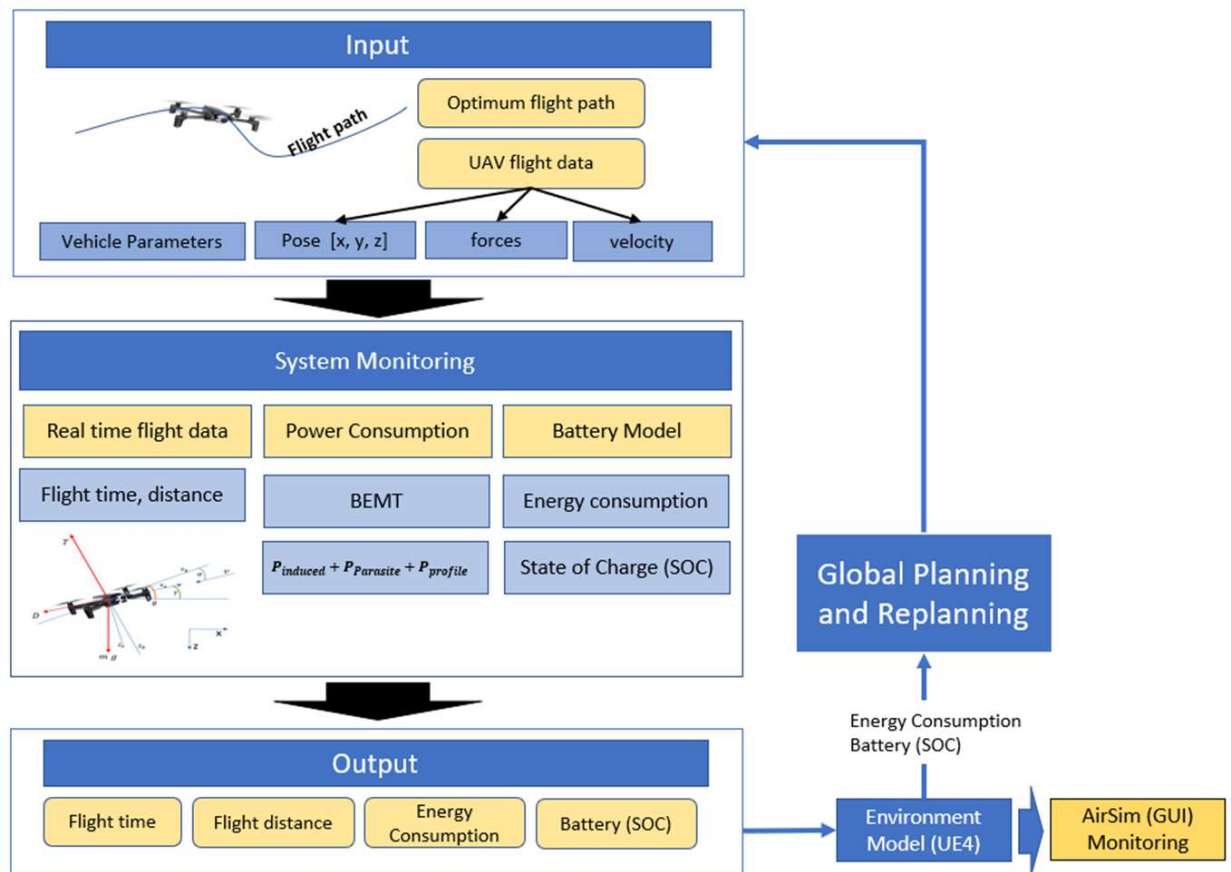


그림 67 System monitoring methodology

(2) 입력 변수

- ▶ 비행체의 형상 데이터와 최적 비행경로를 따라 비행할 때의 실시간 비행 데이터가 입력 변수로 요구됨.
- ▶ 실시간 비행데이터(현재 위치, 속도 및 힘)는 AirSim의 비행 경로 계획으로부터 얻어짐.
- ▶ 시스템 모니터링에 사용되는 UAV의 형상데이터는 본 연구에서 선택된 드론인 Parrot Anafi 드론의 데이터로, 다음 표와 그림에서 나타남.



그림 68 Parrot Anafi drone

표 21 Parrot Anafi drone configuration data

Parameters	Symbols	Values
Voltage	V	7.6 V
Capacity	C	2700mAh
Mass	m	320g
Number of rotors	N	4
Rotor diameters	d	0.07874m
Cross Sectional area	A_c	0.00435 m ²
Projected area	A_p	0.016348
Maximum vertical speed	$V_{z_{\max}}$	4m/s
Maximum horizontal speed	$V_{xy_{\max}}$	15m/s

(3) 시스템 모니터링 변수

(가) 이동 거리 (Distance traveled)

- ▶ UAV 이동 거리 추정은 시스템 모니터링의 중요한 변수임.
- ▶ 사용자 또는 제어기는 실시간으로 UAV가 얼마나 이동했는지를 알 수 있음.
- ▶ 이동을 추정한 후, UAV의 작은 이동량이 먼저 계산되며 UAV의 작은 이동량을 모두 합하여 총 이동을 추정 가능함.
- ✓ UAV의 이동을 계산하기 위해서 삼차원 공간에서의 유클리드 거리 공식을 이용함. 점 $A(x_1, y_1, z_1)$ 와 점 $B(x_2, y_2, z_2)$ 는 식을 위해 거리를 계산하기 위해 가정되는 매우 작은 값임.

$$Traveled\ distance = \sum \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

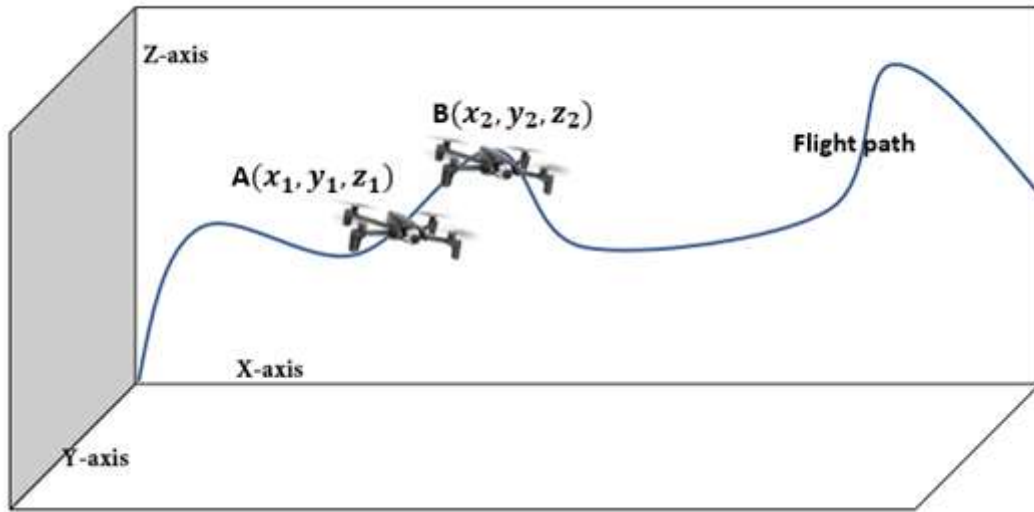


그림 69 3 Dimensional flight of UAV for distance estimation

(나) 에너지 소모량 (Energy consumption)

- ▶ 에너지 소모량은 비행 임무 운용을 달성하기 위한 드론 운용에 있어 중요한 제약조건임.
- ▶ UAV의 **에너지 소모량을 실시간으로 모니터링하여 제어기는 얼마나 많은 에너지를 사용하는지, 임무를 완수하기 위해 어느 정도 양의 에너지가 남아있는지를 확인** 할 수 있음.
- ▶ UAV의 남은 에너지가 비행 임무를 완수하기에 충분하지 않다면, 제어기 또는 사용자는 임무를 거부하고 완전한 비행 임무 완수를 위해 UAV의 배터리를 충전할 수 있음.
- ▶ 에너지 소모량 모델은 블랙박스(black box)와 화이트박스(white box) 두 종류가 존재함.
 - ✓ 블랙박스 접근방식은 기체모델에 대한 적은 지식과 많은 양의 실험 데이터, 드론의 에너지를 계산하기 위한 적은 변수를 요구함.
 - ✓ 화이트박스 접근방식은 직관적인 방식으로 모터 성능, 공력 특성, 배터리 시스템의 영향을 받음.
- ▶ 본 연구에서는 에너지 소모량, 전력 소모량 및 배터리 충전 상태(SOC)를 추정하기 위해 화이트박스 접근방식을 사용함.
- ▶ 화이트박스 접근방식은 UAV의 형상정보, 배터리, 비행동역학적 데이터를 필요로 하며 아래 그림과 같음.



그림 70 Energy consumption, power consumption and state of charge estimation method

- ▶ 필요전력계산은 드론의 전력을 추정하기 위한 또 다른 중요 변수임.
 - ✓ 드론의 필요 전력은 반드시 드론의 전력시스템(배터리)을 통해 제공 가능해야 함.
 - ✓ 필요 전력은 실시간 비행에서 기체의 비행동역학을 기반으로 계산 가능함.
- ▶ 멀티콥터 드론 비행 임무는 수직 이착륙, 수평 비행, 상승 비행, 그리고 호버링 상태로 나눌 수 있음.
- ▶ 아래 그림은 필요 전력 분석을 위한 전체적인 임무 프로파일(Mission profile)을 나타냄.



그림 71 General flight mission profile for multicopter drone

- ▶ 필요 힘은 수직 이착륙, 상승비행, 수평비행, 호버링의 비행 임무에 따라 달라지며 멀티콥터 드론, 회전익 및 배송 드론이 요구하는 필요 힘은 BEMT(Blade Element Momentum Theory)이론을 이용하여 연구함.
- ▶ BEMT(Blade Element Momentum Theory)이론을 적용하기 위해, 평형상태에서의 비행 동역학은 드론의 추력과 동력(Power)을 추정하는데 중요한 역할을 함.
- ▶ 다음 그림에 나타난 임무 형상의 상승 비행 구간에서의 드론의 평형상태는 필요 추력을 계산하기 위해 연구됨.

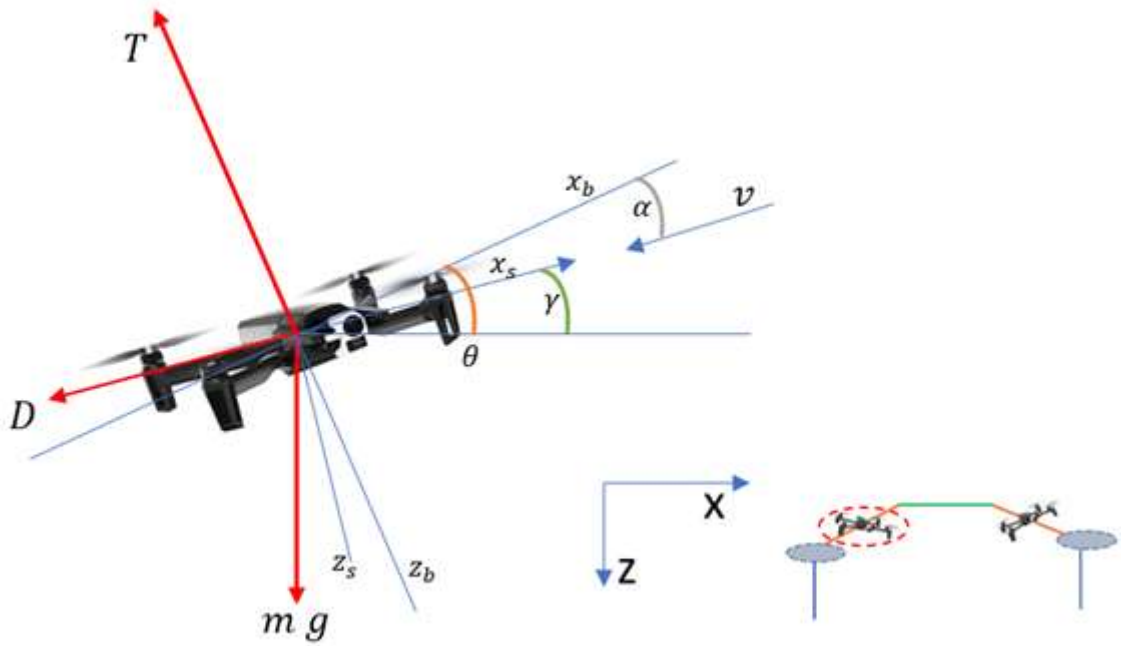


그림 72 Forces along the flight path of a drone at equilibrium condition

- ▶ 위 그림에 나타난 변수는 다음과 같음.

표 22 Parameters of equilibrium condition

파라미터	의미	파라미터	의미
T	Thrust	D	Drag
m	mass	g	gravity
v	velocity	θ	body pitch angle
α	angle of attack	γ	flight path angle
x_b, z_b	body axis	x_s, z_s	flight path axis

- ▶ 정상상태 비행에서, x_s, z_s 축에서 비행경로를 따라 존재하는 힘은 다음과 같음.

$$\sum F_{path} = T \sin(\alpha) - D - mg \sin(\gamma) = 0$$

- ▶ 같은 조건에서, x_s, z_s 축에 대해 수직인 힘은 다음과 같음.

$$\sum F_{perpendicular} = T \cos(\alpha) - mg \cos(\gamma) = 0$$

- ▶ 받음각이 매우 작다($\alpha \cong 0$)고 가정하고, 정상상태 비행의 힘 식의 양변을 제곱하면 비행경로에서 요구되는 추력은 다음과 같음.

$$T = \sqrt{m^2 g^2 + D_{body}^2 + 2 D_{body} m g \sin \gamma}$$

- ▶ 필요 추력은 비행 상태의 영향을 받으므로, 전진 비행과 수평 비행 조건을 우선 고려함.

● 수평 비행

- ▶ 수평 비행조건에서 비행경로 각도는 $\gamma \cong 0$ 임.
- ▶ 필요한 추력은 평형상태에서 얻을 수 있으므로, 필요 power가 우선 계산되어야 함.
- ▶ 수평 비행에서의 필요 power는 induced power, parasite power, 그리고 프로펠러의 profile power의 총합으로 계산됨.

$$P_{forward} = P_{induced} + P_{Parasite} + P_{Profile}$$

- ▶ Induced power는 공기는 아래로 밀어 추력을 만들어 내며 로터 디스크를 통과하는 수직인 공기 흐름에 의해 발생하고, 축 방향 운동량 이론을 이용해 계산할 수 있음.
- ▶ 아래 식은 Liu et al. 이 제안한 배송 드론의 induced power를 계산하는 간단한 식으로, T 는 전진비행을 위해 필요한 추력, ρ 는 비행 고도의 밀도, A_c 는 비행경로에 수직인 드론의 단면적을 나타냄. k_1 은 induced power의 계수이고, Liu et al.은 배송 드론의 경우 $k_1 \cong 0.88554$ 라고 제안함.

$$P_{induced} = \frac{k_1 T^{3/2}}{\sqrt{2\rho A_c}}$$

- ▶ Parasite power($P_{parasite}$)는 비행체와 바람 사이에 상대적인 변화 운동이 발생할 때, 동체 항력을 견디는 power이며, 이는 드론 항력의 영향을 크게 받음.

$$P_{parasite} = D_{body} v_{xy} = \frac{1}{2} \rho v_{xy}^2 A_c C_d v_{xy}$$

- ▶ 위 식에서 v_{xy} 는 수평 대기 속도, A_c 는 단면적, C_d 는 드론의 항력계수이고, Zhang J et al.은 소형 드론의 항력계수를 1.49라고 제안함. 본 연구에서도 동일한 항력계수를 사용하였으며, $C_d = 1.49$ 는 parasite power를 계산하는 데 사용됨.
- ▶ Profile power $P_{profile}$ 는 회전하는 프로펠러 블레이드에 의해 발생하는 회전 항력을 견디며, 이 power의 크기는 블레이드의 수, 로터의 코드, 로터의 반경과 같은 로터 블레이드 형상에 따라 달라짐.
- ▶ 소형 멀티콥터의 profile power는 Z.Liu et al.에 의해 연구되었으며, 식은 다음과 같음.

$$P_{profile} = \frac{k_3^{1.5} N c C_d \rho (\frac{d}{2})^4}{8} T^{3/2} + \frac{k_3^{0.5} N c C_d \rho (\frac{d}{2})^2}{8} (v_{xy} \cos r)^2 T^{1/2}$$

- ▶ 위의 profile power 식을 간단히 정리한 식은 다음과 같음.
- ▶ c_2 와 c_3 는 profile power 상수이며, 블레이드 형상에 따라 달라지는 값임. 그러나 소형 드론의 경우 Z.Liu et al.에서 제안된 $c_2 \cong 0.3177, c_3 \cong 0$ 을 사용함.

● 수직 비행

- ▶ 평형상태 운동방정식에서, 비행 경로의 각도 $\gamma = 90^\circ$ 일 때, 즉 수직 비행일 경우 필요한 추력은 $T = \sqrt{m^2 g^2 + D_{body}^2} + 2D_{body} m g \sin \gamma$ 와 같이 표현됨.

- ▶ 수직 비행에 필요한 power는 다음과 같이 나타남.

$$P_{climb} = P_{induced} + P_c$$

- ▶ Induced power, $P_{induced}$ 의 계산은 전진 비행의 경우와 동일하며, 수직 비행에서의 항력은 다음과 같음.

$$D_{body} = \frac{1}{2} \rho v_z^2 A_p C_d v_z$$

- ▶ 여기서 v_z 는 수직 비행의 속도, A_p 는 드론의 정사영 면적임.
- ▶ P_c 는 다음과 같은 식으로 추정 가능함.

$$P_c = T v_z$$

● 호버링(Hovering)

- ▶ 호버링은 드론이 전진 혹은 수직 방향으로 정지한 채 같은 고도를 유지하는 상태를 말함.
- ▶ 따라서 필요 추력은 드론의 무게와 동일함.

$$T_{hover} = mg$$

- ▶ 호버링 비행 시의 필요 추력은 다음과 같은 식을 이용해 추정 가능함.

$$P_{hover} = T_{hover} v_i = T_{hover} \sqrt{\frac{T_{hover}}{2\rho A}}$$

- ▶ 드론의 비행 임무별 power 추정을 완료하면, 비행경로에 따른 전력 소비량은 필요 전력 모델의 총합으로 추정됨.

$$P_{req} = P_{forward} + P_{climb} + P_{hover}$$

- ▶ 필요 힘은 기체 동역학을 이용해 추정 가능하며, 배터리에서 프로펠러로 전력을 제공하는 동안 손실이 발생함.
- ▶ 따라서 배터리에서 소모되는 전력은 필요 power를 배터리 동력 전달 효율로 나눠 계산할 수 있음. 배송 드론의 경우, 전달 효율은 70%로 $\eta = 0.7$ 로 제안됨.

$$P_{elec} = \frac{P_{req}}{\eta}$$

- ▶ 드론의 에너지 소모량은 전력과 이동시간을 곱해 추정이 가능함.

$$Energy\ Consumption = P_{elec} \times time$$

(다) 충전 상태(State of Charge: SOC)

- ▶ 실시간 배터리 용량 추정값은 제어기 또는 드론의 사용자가 알아야만 하는 중요한 변수임.
- ▶ 배터리의 용량을 측정하기 위한 파라미터의 하나로 배터리 충전 상태(SOC)가 사용됨.
- ▶ 본 연구에서는 K.S.Ravi Kumar et al. 에 의해 제시된 column counting 기법을 사용하는데, 총 열의 개수 즉 배터리의 충전량은 전력, 전압, 시간으로부터 추정할 수 있음.

$$Total\ columns = Charge = \int \frac{P_{elec}}{Voltage} dt$$

$$SOC = 1 - \frac{Capacity}{Total\ columns}$$

(4) 시스템 모니터링 에이전트 테스트 결과

(가) 시스템 모니터링 에이전트 검증

- ▶ 에너지 소모량 추정 모듈에 대한 검증은 참고문헌인 Z.Liu et al. 이 작성한 논문에서 선택한 드론인 IRS+multi-copter를 대상으로 수행함.
- ▶ 본 연구에서 선택한 드론인 Anafi 드론의 에너지 소모량과 SOC는 power를 최소한으로 소모하는 비행 속도를 이용해 찾은 최대 비행시간을 이용해 검증함.
- ▶ Z.Liu et al.의 논문은 실험적 데이터를 검증하기 위해 직사각형 형태의 비행경로의 수평비행을 가정했으며 그 경로는 다음과 같음.

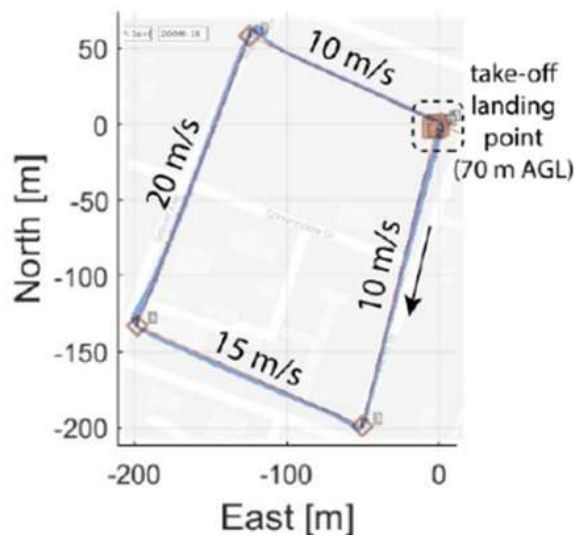


그림 73 Rectangular flight loop of IRS+ multi-copter

- ▶ 상단의 비행경로를 바탕으로 에너지 소모량은 앞서 제시된 방법론을 이용해 추정됨.
- ▶ 테스트 결과 오차율은 6.11%로, 본 연구에서 사용된 에너지 소모량 모델의 방법론이 적용될 수 있음.

표 23 Energy consumption validation for IRS drone

	Code	Reference model	Error percentages
Energy Consumption	13100 Joules	12300 Joules	6.1068 %

- ▶ Anafi drone을 대상으로 하는 검증과정은 데이터가 부족한 관계로 제조사에서 제공하는 최대 비행시간을 이용하여 배터리 충전 상태 추정 모델 검증을 수행함.
- ▶ 최대 비행시간을 구하기 위해서 최소 전력 소모량으로 비행이 가능한 속도를 추정함.
- ▶ 전력 소모를 최소로 하는 비행 속도는 7.05m/s로 계산되며, 그 때의 최소 전력 소모량은 82.32 watt로 계산됨.

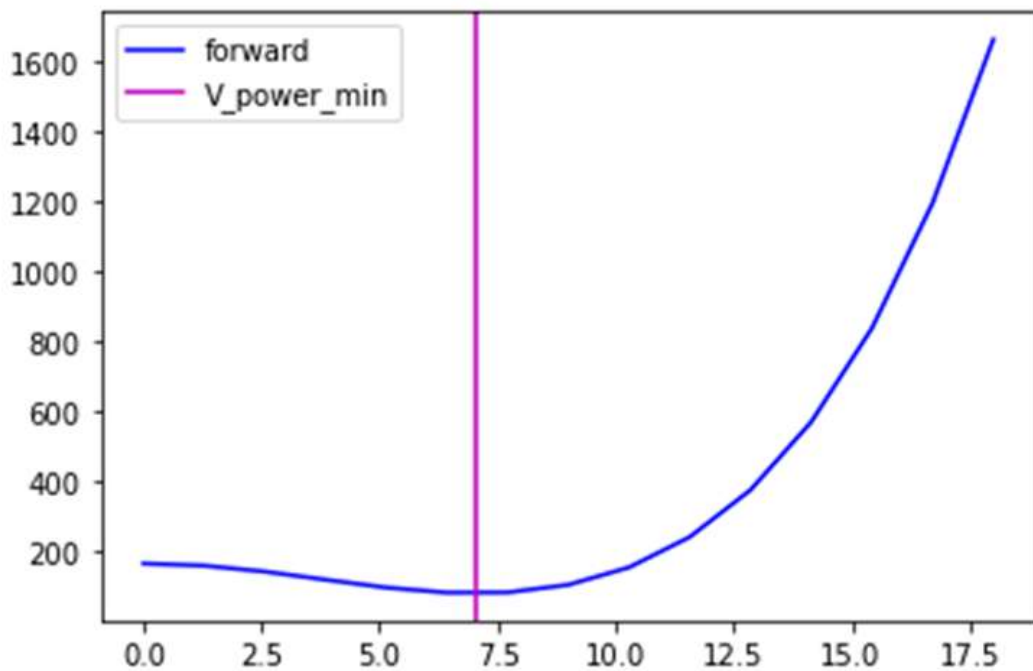


그림 74 Power consumption and minimum power velocity

- ▶ 비행 속도 7.05m/s는 AirSim 시뮬레이터의 입력변수로 사용되었으며 시스템 모니터링 에이전트를 이용해 추정한 SOC는 Parrot Anafi 드론의 최대 비행시간인 20분 동안 수집됨.
- ▶ 검증 과정에 AirSim이 제공하는 파이썬 기반 테스트 파일인 PythonClient가 사용되었으며 본 파일은 AirSim\PythonClient\multirotor\box.py 경로에 존재함.
- ▶ 검증 결과는 다음과 같음.

표 24 Flight time and SOC validation for Anafi drone

SOC (%)	Distance (m)	Flight Time (sec)	Maximum flight time error (%)
4.890	3089.203	1012.299	14.89
2.129	3146.610	1042.828	13.097
1.064	3168.445	1054.622	12.11
0.095	3188.458	1065.322	11.2

- ▶ Anafi 드론의 배터리 충전상태와 소모량에 대한 검증은 전력을 최소로 소모하는 비행 속도인 7.048819m/s에 대해 수행되었음.
- ▶ 최대 비행시간은 주어진 배터리 충전량을 이용해 드론이 얼마나 멀리 비행할 수 있는지를 의미함.
- ▶ SOC 추정 모델을 이용해 계산된 결과는 제조사가 제공하는 최대 비행시간인 20분에 미치지 못하는 결과를 보임.
- ▶ 오차율의 14.89%로 가장 크게 나타난 경우에는 SOC가 4.89%로 나타났으며, 최저 오차율인 11.2%를 보인 테스트 결과 SOC는 0%에 가깝게 나타남.
- ▶ SOC 추정 모델의 정확도를 높이기 위해 대기 속도와 전력 효율의 추정 정확도를 개선하고자 함.

(나) 시스템 모니터링 에이전트 개발 결과

- ▶ 시스템 모니터링 에이전트는 AirSim 시뮬레이터의 GUI 상에 모니터링 결과인 SOC, 비행 거리, 비행시간, 그리고 에너지 소모량을 보여줌,
- ▶ 시뮬레이터 상에 보여지는 모니터링 결과는 다음과 같음.

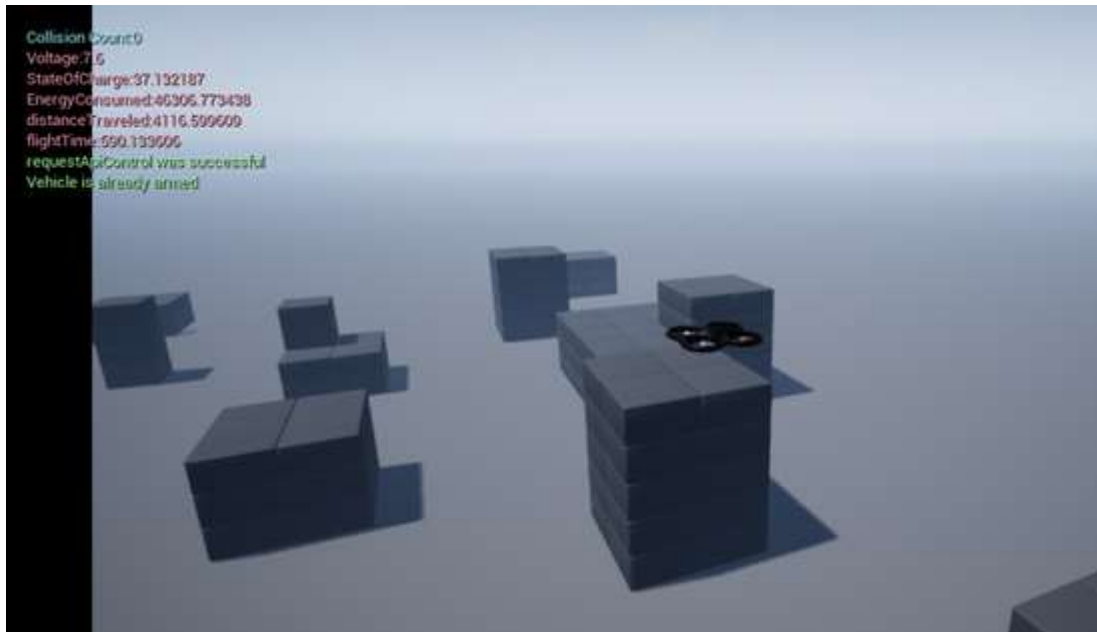


그림 75 System monitoring results of real time data

- ▶ 실시간 비행 데이터를 활용한 시스템 모니터링 에이전트의 모니터링 결과가 왼쪽 상단에 나타남.
- ▶ 모니터링 결과로 얻을 수 있는 SOC와 비행 거리를 이용하여 **AI 모델은 경로를 재계획 할지 또는 에너지 요구사항 부족 등의 이유로 비행 임무를 취소할지 결정할 수 있음.**

● 테스트 결과

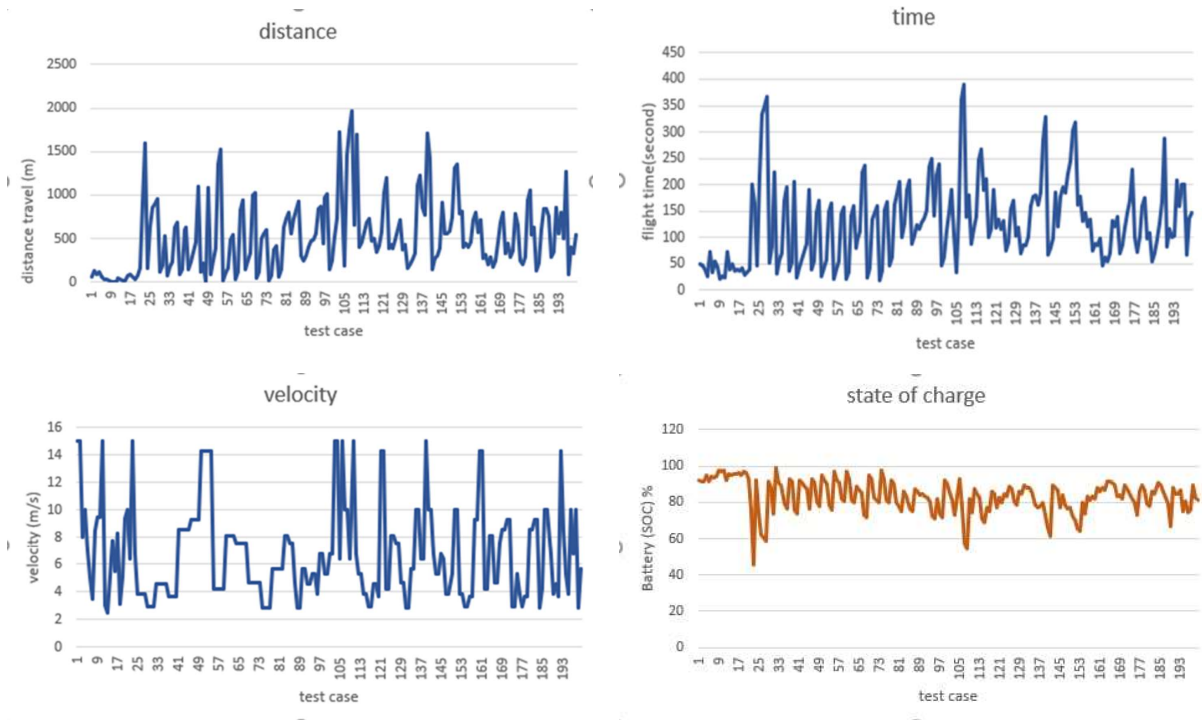


그림 76 System monitoring tested cases and battery(SOC) for success rate

표 25 Success and Failed case analysis for system monitoring agent

Test case	Success case	Failed case	Failure rate
200	200	0	0%

- ▶ 배터리 에너지 소모율 테스트의 성공률을 검증하기 위해서 다른 속도, 비행시간, 비행거리에 대해서 200번의 테스트 케이스를 수행함.
- ▶ 상단의 그림은 다른 거리, 비행시간, 속도에서 UAV 배터리 SOC 관찰 테스트 결과를 보여줌.
- ▶ 시스템 모니터링 에이전트는 배터리 SOC와 비행거리, 비행시간을 모니터링 해 임무 달성을 위한 임무 재설계여부를 결정함.
- ▶ 시스템 모니터링 에이전트 테스트의 실패율은 AirSim 시뮬레이터에서 실시간으로 비행하는 UAV의 배터리 SOC 모니터링을 통해 정의가 가능함.
- ▶ 실패 케이스는 배터리 SOC가 0에 가까워져 UAV가 임무 달성에 실패한 경우를 의미하며 전체 케이스 대비 실패 케이스의 비율을 실패율로 정의함.
- ▶ 실패율이 0%에 가까울수록 에이전트의 성능이 우수함을 보여줌.
- ▶ 200번의 테스트결과, 배터리 SOC는 UAV에 할당된 임무의 달성여부와 관계없이 실시간으로 관찰이 가능하며, UAV는 배터리의 방전 또는 고장 상황 없이 실패율 0%로 임무를 달성함.

다) 3차원 환경에서의 임무 재설계 알고리즘 개발

(1) 실시간 임무계획(Real-time Mission Planning : RMP) 알고리즘

- ▶ '임무계획(Mission planning)'은 UAV가 한 지점에서 다른 지점으로 장애물(지형지물, 건물, 새, 다른 비행체 등)을 회피하며 어떻게 비행할지 결정하는 과정을 의미함.
- ▶ 임무계획은 자율 운용 시스템의 가장 중요한 측면이며, 드론의 임무와 전체적인 임무 할당을 결정하는 접근방식이며, 실제 교통수단에 적용되어 복잡하고 넓게 연구되는 최적화 문제의 조합임.
 - ✓ 최근 자율 비행체는 지능형 감시와 같이 안전과 환경 문제가 두드러지는 다양한 영역에서 임무 경로 계획을 기반으로 사용됨.
 - ✓ 임무계획의 효율은 설계자가 시스템의 불확실성을 얼마나 알고 있는지, 전체임무완수를 얼마나 빨리할 수 있는지, 그에 따라 임무를 어떻게 할당하는지에 달려있음.
- ▶ 실시간 임무계획(Real-time mission planning)은 임무의 제약사항을 모두 만족해야 하며 실시간 환경에 적용이 가능해야함.
 - ✓ UAV는 초기 위치에서 출발하여 계획된 경로를 따라 임무를 완수해야 하며, 목표지점에 도착할 때까지 정적, 동적 환경을 실시간으로 마주할 수 있음.
 - ✓ 연료 소모율을 최소화하기 위한 최단경로와 같이 임무를 위한 목적 함수가 포함될 수 있음.
 - ✓ 목적 함수는 UAV가 다른 위험요소에 의해 위협받지 않고 임무를 완수했다는 것을 반드시 보증해야 하며, 따라서 극한 상황에서의 최단 거리 비행은 적절한 결정이 아님.
 - ✓ 임무 중 특정 타겟을 알아내는데 시간이 필요한 경우에도, 완수하는데 소요되는 시간을 최소화하는 것은 매우 중요함.
 - ✓ 가장 큰 임무의 제약사항은 임무 중 UAV를 잃을 수 있는 위험을 감소시키는 것임.
 - ✓ 군용 무인기의 경우 노출을 줄이고 적의 위험을 회피하는 것이 경로 계획기의 매우 중요한 요소이며, 이는 임무의 안전과 효과적인 성과에 영향을 미침. UAV는 군 임무 환경에서 적에게 탐지될 수 있으며 유해지역을 비행할 때 노출되어 격추될 수 있음.
- ▶ 임무계획은 크게 임무 스케줄링 및 계획과 경로 계획으로 나뉨.
 - ✓ 다음 그림은 시작점과 목표지점에서 특정 임무를 언제 그리고 어떻게 수행할지 임무를 할당하고 계획하는 과정을 보여줌.

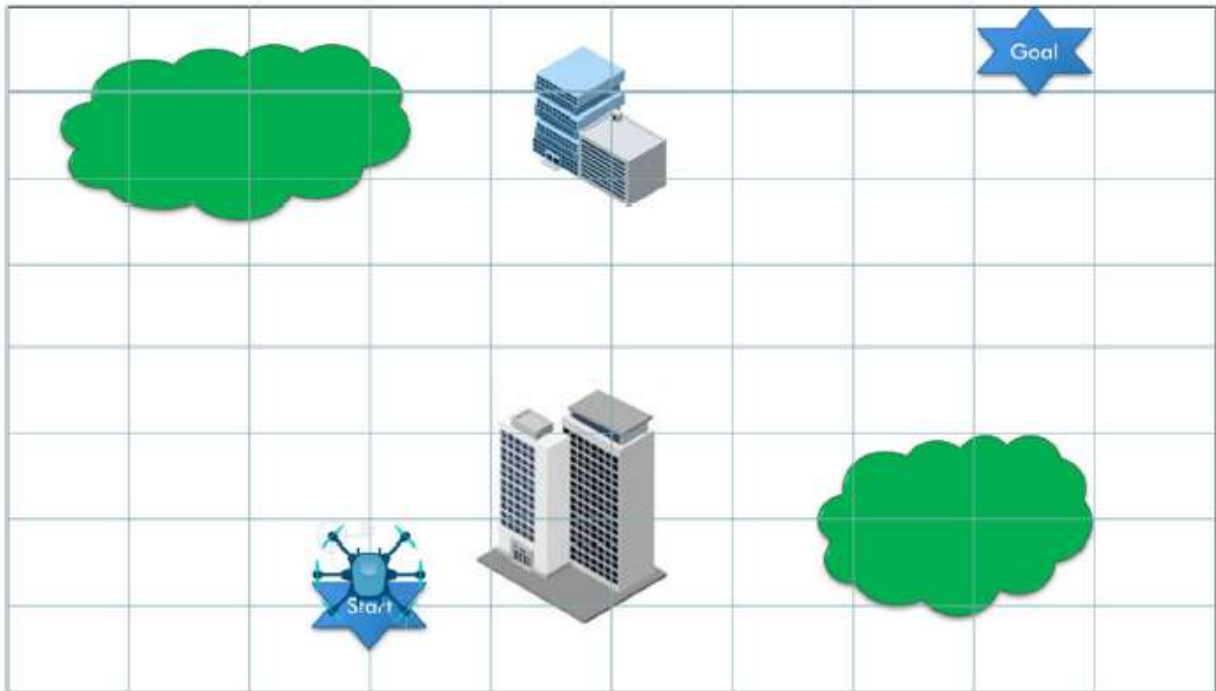
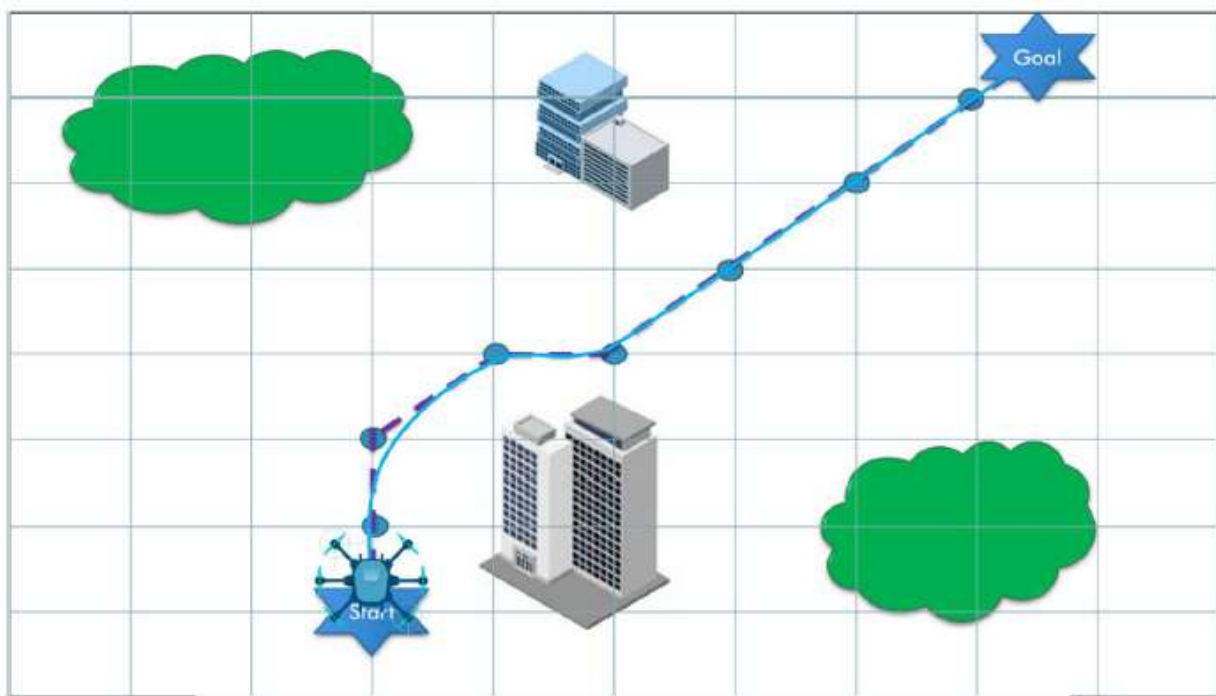
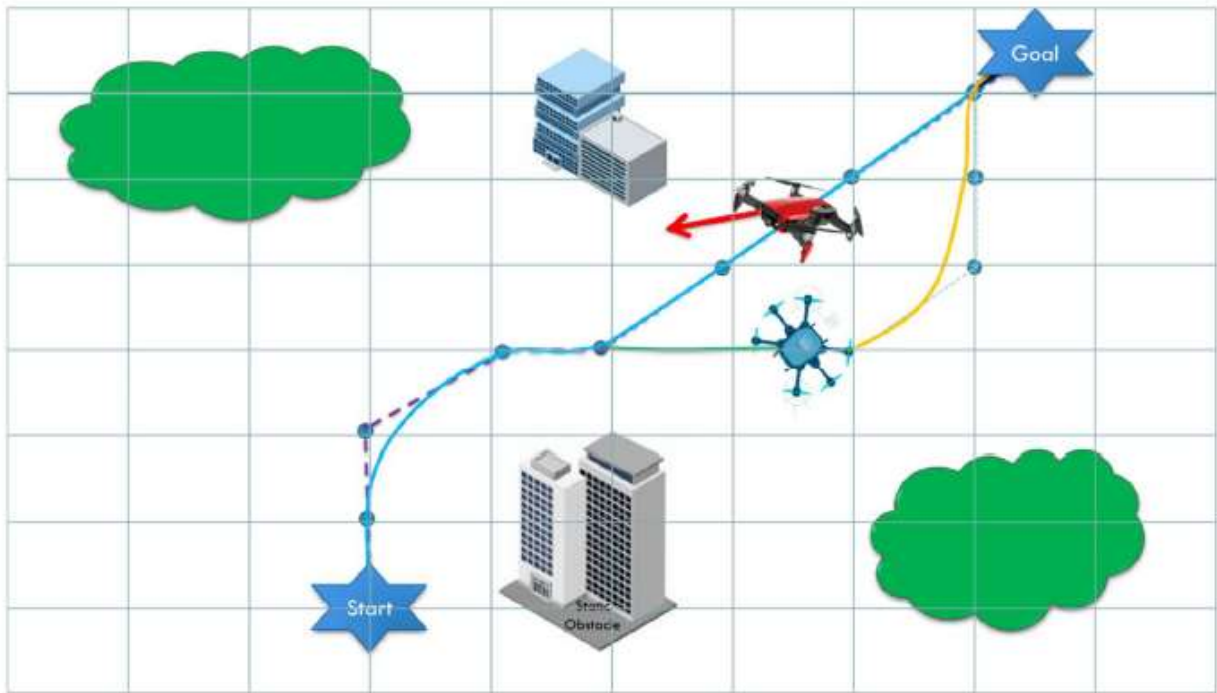


그림 81 Task planning for only one task destination for UAV to finish a mission



(a) Path planning to generate the optimal initial planned path before UAV fly to finish the mission



(b) Dynamic obstacle avoidance and replanning path during flight in the mission

그림 82 Path planning and replanning

(2) 전역 경로 계획 및 재계획

- ▶ 실시간 경로 계획, 재계획, 그리고 장애물 회피는 자율 비행체의 임무관리에 중요한 역할을 함.
 - ✓ 이하에 제안된 알고리즘은 정적인 장애물과 동적인 장애물을 모두 포함함.
 - ✓ 장애물 중 일부만이 미리 알려져 있고, 이 경우 고전적인 이중 아키텍처로 해결이 가능함.
- ▶ 전역 경로 계획 모듈은 완전히 알려진 장애물과 일부만 알려진 정적 장애물의 존재를 가정한 상황에서 임무를 완수하기 위해 시작지점과 목표지점 사이의 장애물이 없는 경로를 결정하고 생성함.
 - ✓ 본 과정은 UAV가 임무를 시작하기 이전에 수행됨.
 - ✓ UAV는 생성된 초기 경로를 따라 비행하며 이 과정에서 UAV의 센서가 새로운 장애물 또는 위협요소를 탐지할 수 있음.
- ▶ 지역 경로 계획 모듈은 장애물을 회피하고 전역 경로 계획을 위한 안전한 위치를 제공함.
- ▶ 전역 경로 계획 알고리즘은 지역 경로 계획 모듈이 제공한 위치에서부터 목표지점까지의 경로를 재계획함.
- ▶ 다음 그림은 전역 경로 계획 모듈이 정적 장애물과 동적 장애물을 회피하기 위해 조정되는 임무 시나리오를 나타냄.

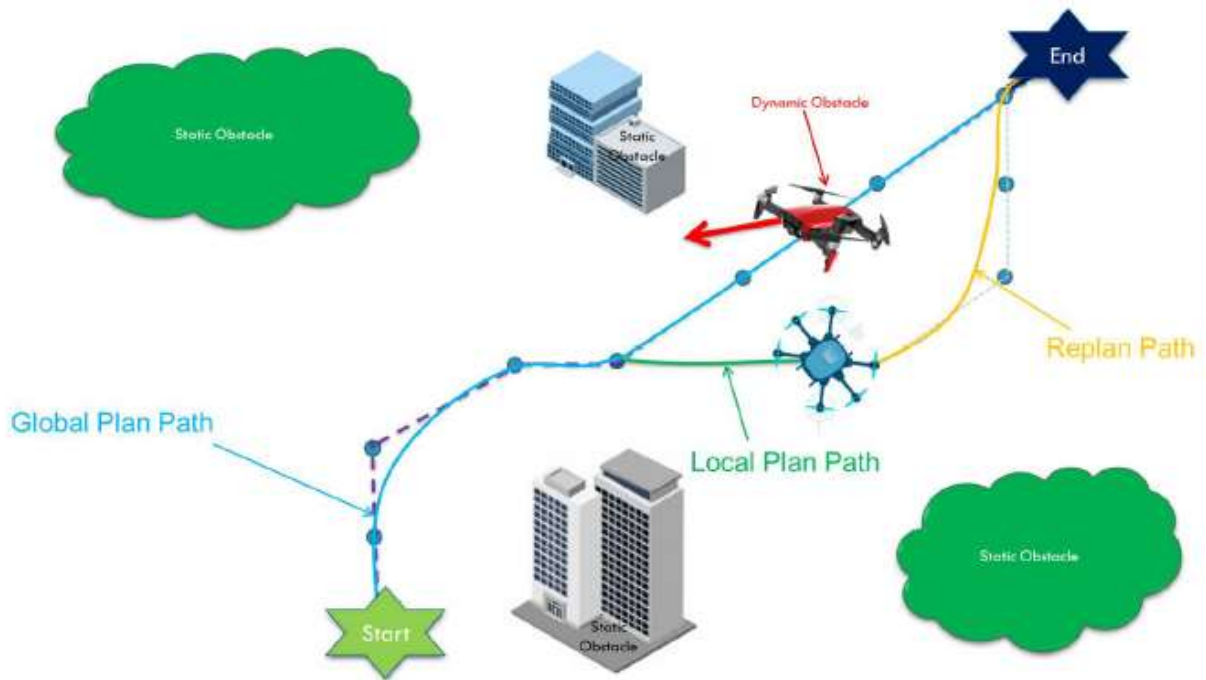


그림 83 UAV's Real-time mission with global and local plan path

- ▶ 두 경로 계획 아키텍처는 장애물, 다른 기체, 그리고 임무 제약사항을 고려하여 일련의 웨이 포인트로 구성된 전역 경로 솔루션과 지역 경로 솔루션을 제공한다.
- ▶ 본 연구는 다음의 두 주요 목표를 달성하기 위해 수행됨.
 - ✓ 전역 경로 계획 및 재계획 알고리즘 개발
 - 장애물이 존재하지 않는 최단경로의 계획.
 - 거리 및 비행시간이 계획된 경로의 계획.
 - ✓ 학습기반 지역 경로 계획 및 장애물 회피 알고리즘 개발
 - 실시간으로 장애물을 회피하여 전역 경로 계획을 보완할 수 있는 지역 경로 계획.
 - 새로 나타난 장애물을 회피하고 전역 경로 계획 모듈이 최종 목표지점에 도착할 수 있는 경로를 재계획 할 수 있는 안전한 위치 제공.
- ▶ 전역 경로 계획과 재계획은 단일 UAV 시스템에 대한 효과적인 경로 계획 알고리즘인 iADA*(improved Anytime Dynamic A*) 알고리즘을 활용함.
 - ✓ iADA* 알고리즘은 Anytime 경로 계획 알고리즘을 기반으로 향상된 경로 길이 연산 능력과 탐색을 활용한 경로 계산 빈도 감소 능력을 가져 상당히 빠른 경로 계획 알고리즘임.
 - ✓ 경로 재계획과 동시에 장애물 회피가 가능해 연산 시간이 감소 됨.
 - ✓ 모듈은 지형지도와 완전히 혹은 일부만 알고 있는 정적 장애물 데이터베이스를 활용해 임무

계획 단계에서 일련의 웨이포인트를 포함한 미리 계획된 경로를 제공함.

- ✓ 만약 장애물이 주어진 웨이포인트를 연결한 경로상에 존재한다면 최단 거리와 같은 구체적인 최적화 조건을 만족하는 장애물이 존재하지 않는 경로가 웨이포인트 행렬의 형태로 제공됨.
- ✓ 미리 계획된 경로를 따라 비행하여 임무를 수행하는 과정에서 **RMP 시스템은 현재의 센서 데이터와 위치, 속도정보를 지역 경로 계획 모듈과 장애물 회피 모듈에 제공함.**

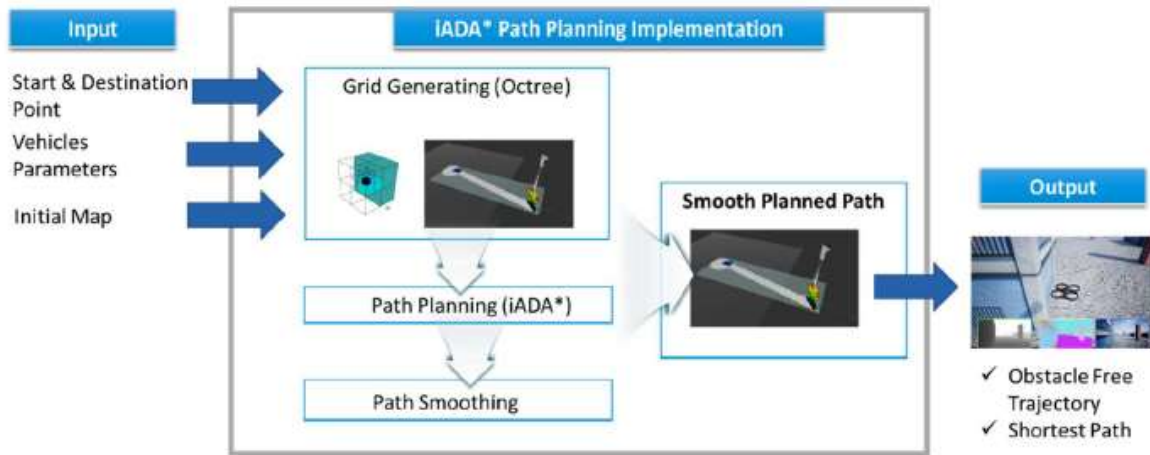


그림 84 Global path planning and replanning using iADA* algorithm

(가) 점유 격자 생성

- ▶ ROS(Robotic Operating System)과 통합이 가능한 오픈소스 라이브러리인 Octomap을 활용하여 **Octree 맵을 생성.**
- ✓ Octree는 데이터 구조의 일종으로 공간을 정육면체로 나누고, 나뉜 정육면체를 다시 1/8 크기의 정육면체로 나누는 과정을 여러번 반복하여 최소 크기의 정육면체 집합으로 만드는 방법임.
- ✓ 점유되었거나 그렇지 않고 개방된 작은 정육면체들로 구성된 정육면체는 단일 요소로 저장될 수 있고, 이렇게 구성된 맵은 상당히 파일 크기가 줄어들.
- ✓ Octomap은 공간을 점유되지 않은 공간, 점유된 공간, 그리고 미지의 공간의 3가지 상태로 분류함.

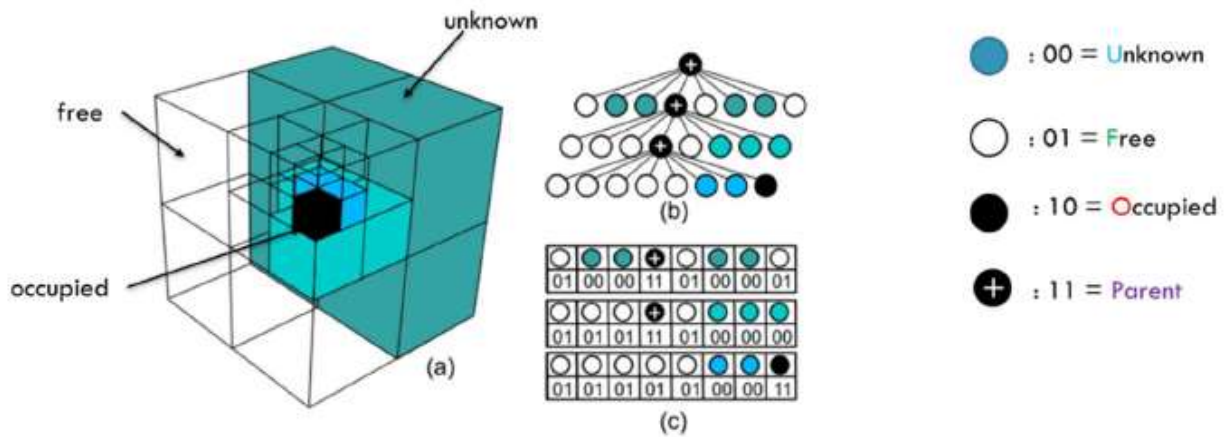


그림 85 Example of Octree storing free(shaded white), occupied cells(black), unknown cells(green blue) and parent not(black in+sign inside)



그림 86 Occupied voxels represented by different resolutions 0.08m, 0.64m, 1.28m respectively

(나) iADA* 알고리즘을 활용한 경로 재계획 테스트 구성

- ▶ 1차년도에 기존의 4가지 알고리즘과 비교하여 상당히 향상된 성능을 보이는 것을 검증한 iADA* 알고리즘을 활용하여 전역 경로 계획 및 재계획을 수행.
- ▶ 다음 그림은 단일 UAV의 임무 수행과정에서 iADA* 알고리즘의 전역 경로 계획 및 재계획 과정을 나타냄.

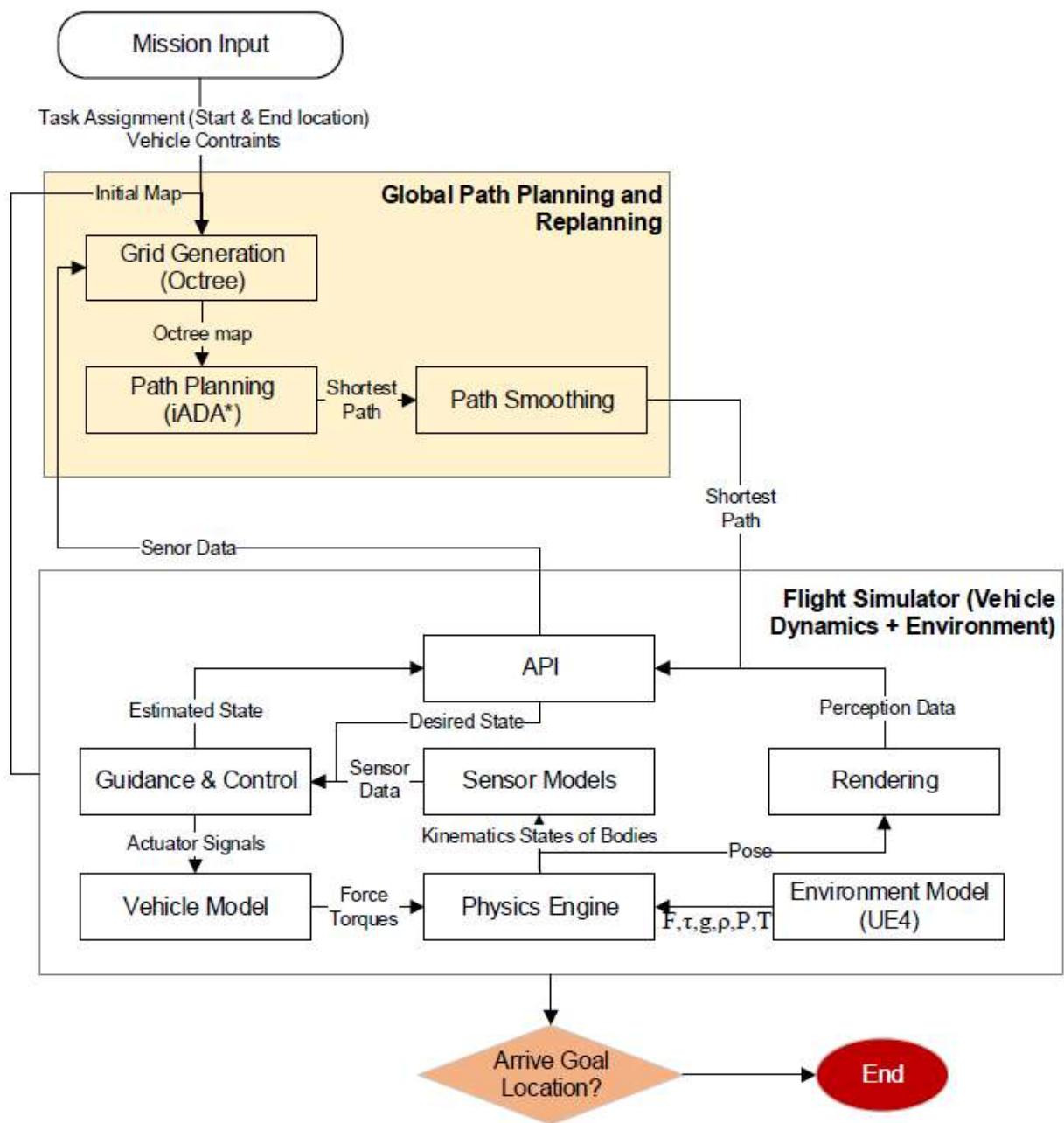


그림 87 Global path planning and replanning process

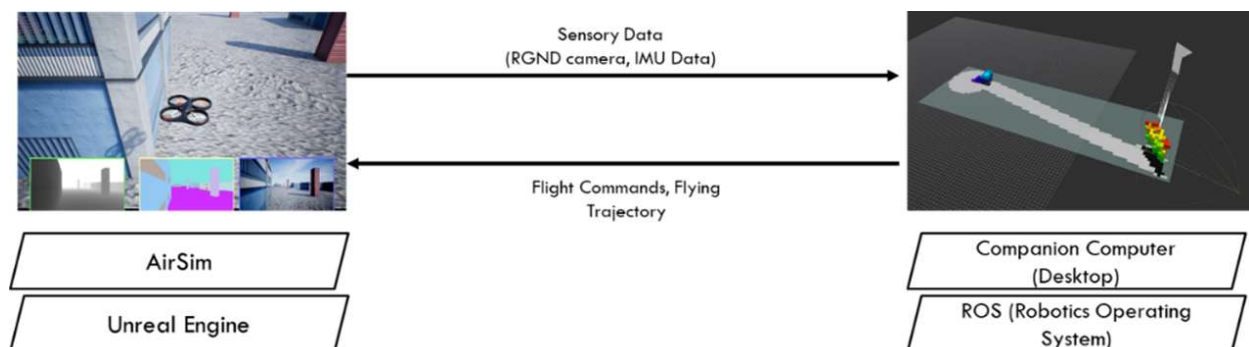


그림 88 Architecture of the SITL system between companion computer AirSim

- ▶ iADA* 경로 계획 알고리즘 테스트 결과는 환경에서 드론의 비행 임무의 완수 여부를 이용해 판단함.
- ▶ 테스트 케이스는 경로 계획을 위한 시작점과 도착점, 그리고 경로 계획 모듈의 초기 지도를 제공함.
- ▶ 에이전트는 octree 맵을 생성하고 iADA* 경로 계획기를 통해 초기 경로를 계산하며, UAV는 3미터 수직 상승 후 계획된 경로를 따라 비행함.
- ▶ 시스템이 장애물을 탐지하였을 경우, Octree 생성 알고리즘은 맵을 재생성하며 경로 계획기는 경로를 재계획함.
- ▶ 다음 그림은 테스트 환경 및 구조, 에이전트의 경로 계획기 예시를 나타냄.

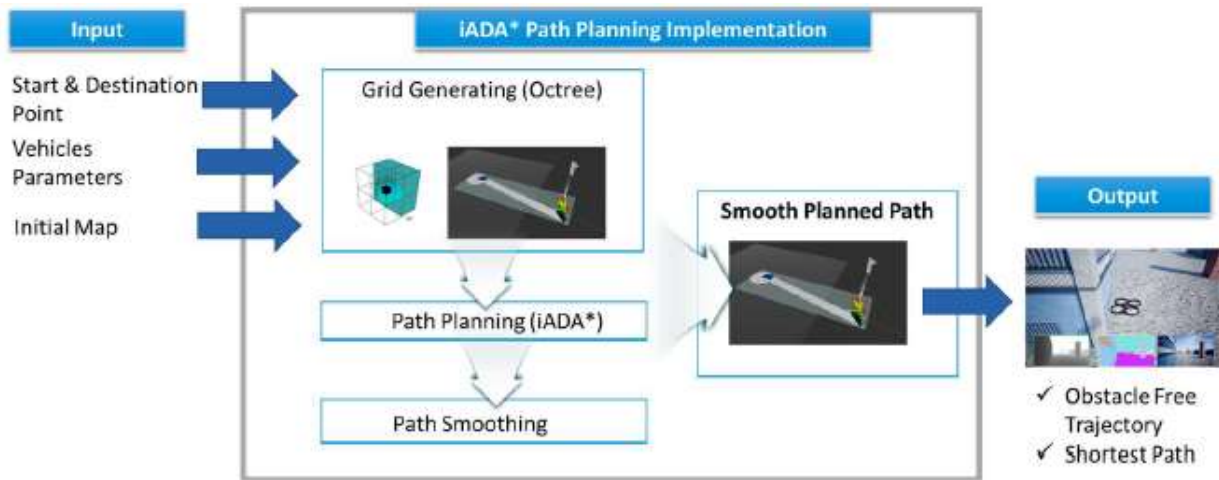


그림 89 Global path planning and replanning data structure

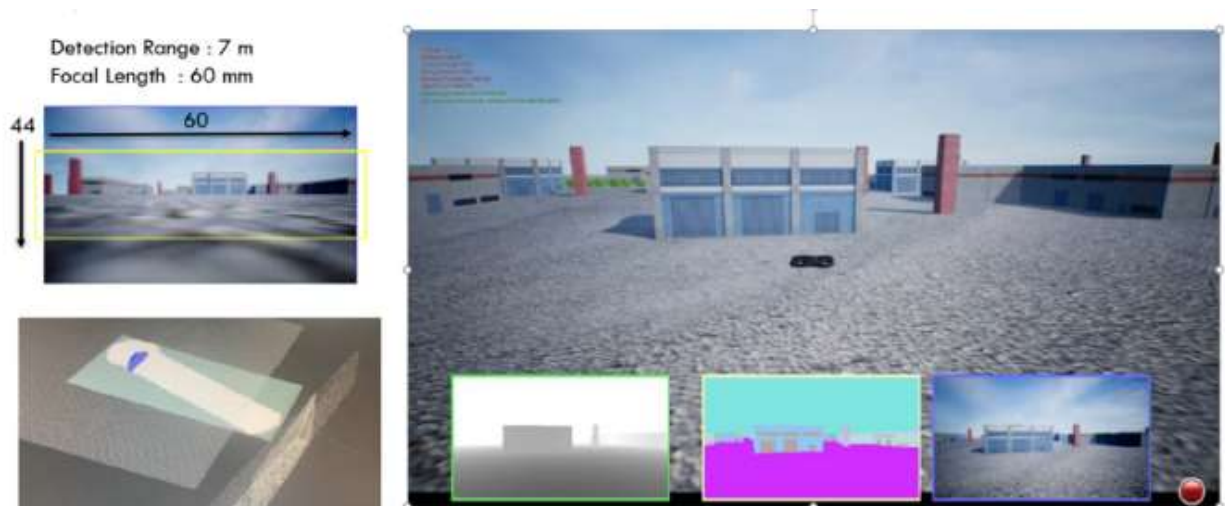


그림 90 iADA* path planning in the unknown environment demonstration

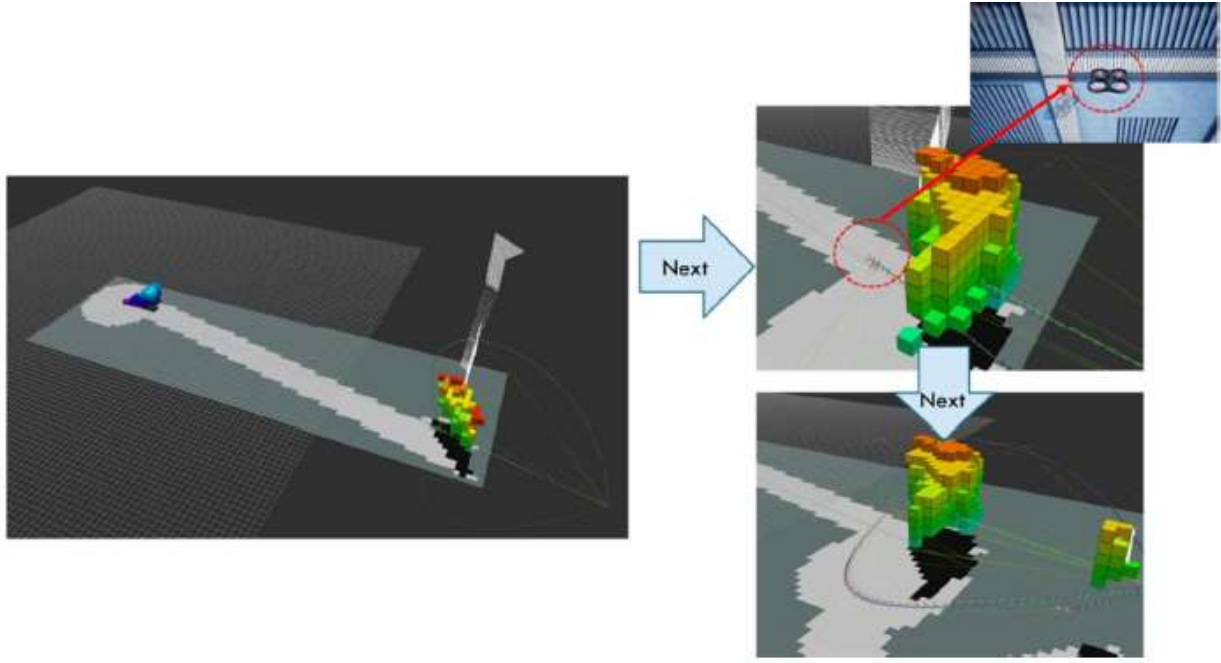


그림 91 One of the examples of iADA* path planner

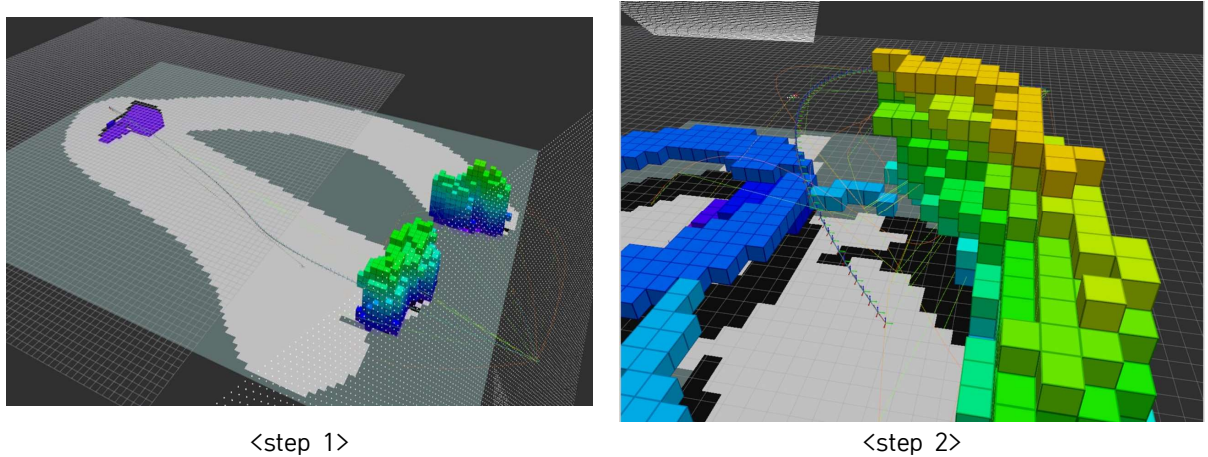


그림 92 Detail steps of building Octree map

(다) iADA* 경로 계획기의 성공률 분석

◎ 경로 계획기 성공률 분석 방법 및 기준

- ▶ 300m × 200m 크기의 정적 환경을 기반으로 iADA* 알고리즘을 이용하는 경로 계획기의 성능을 검증함.
- ▶ 도착지점이 무작위로 선정되는 200개의 테스트 케이스를 바탕으로 경로 계획기의 성능 테스트를 진행함.

표 26 Success and fail cases analysis for iADA* path planner

Test case	Static environment		
	Success case	Failed case	Failure rate
200	198	2	1%

- ▶ AirSim 시뮬레이터 상에 드론을 구현하는 과정에서 발생한 오류는 존재하지 않음.
- ▶ 테스트 결과, iADA* 경로 계획기는 1%의 낮은 실패율을 보이며 경로를 생성하고 목표 지점

까지 비행하는 임무를 수행함.

- ▶ 발생한 실패 케이스는 목표지점이 매 테스트 케이스마다 임의로 좌표로 선정됨에 따라 해당지점에 장애물이 위치한 경우 전역 경로 계획 모듈이 초기경로를 생성할 수 없기 때문에 분석됨.
- ▶하단의 그래프는 테스트 케이스의 초기 경로 계획 소요시간과 성공 케이스의 총 비행시간을 나타냄.
- ▶초기 경로 계획에 소요되는 시간은 평균적으로 0.001초 미만으로 매우 빠른 계산 결과를 보임.
- ▶그림 93의 초기 경로 계획 소요시간 및 성공 케이스의 총 비행시간 그래프와 그림 94에 나타나는 성공케이스의 총 비행거리 그래프는 임무 계획의 성능 및 우수함을 보여줌.

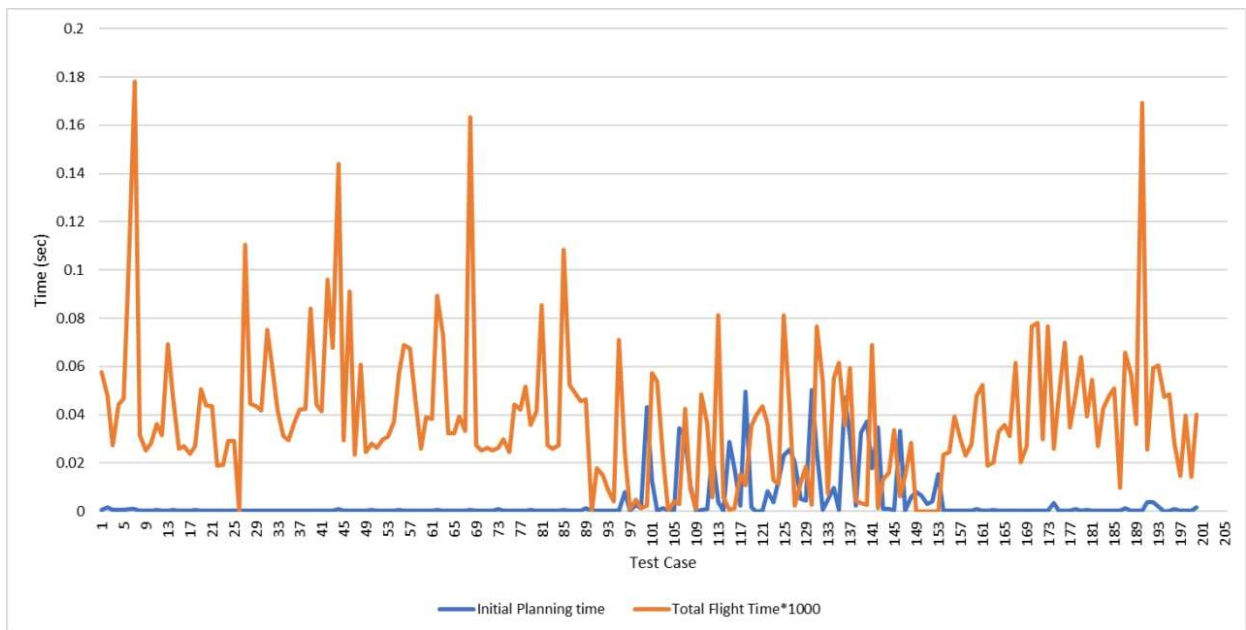


그림 95 Initial planning time and total flight time for success cases

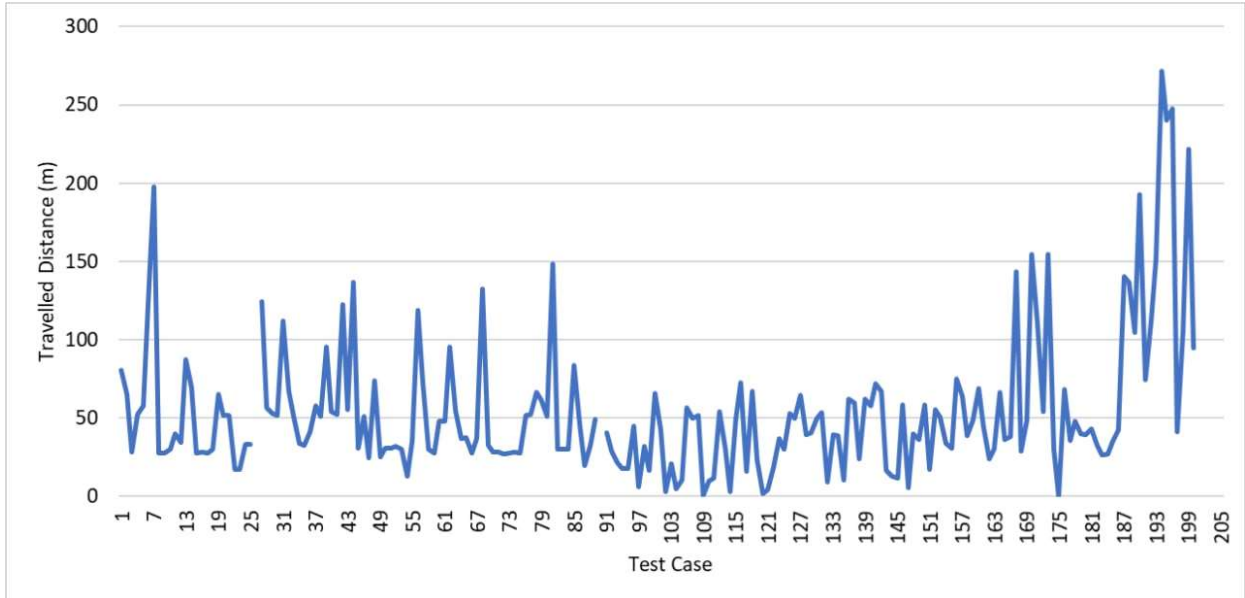


그림 96 Total traveled distance(meters) for success cases

라) 인공지능 에이전트 통합 및 검증

- ▶ 본 연구에서는 3차원 시뮬레이션 환경인 AirSim에서 경로 계획 및 재계획, 경로 평활화(path smoothing), 경로 생성을 포함한 실시간 임무계획을 수행하는 **소프트웨어 아키텍처와 상호작용하는 closed-loop 시뮬레이션 프레임워크를 제시함.**
- ▶ 제시된 프레임워크는 두가지 주요 closed-loop 시뮬레이션으로부터 참조되어 알 수 있음.
- ▶ Closed-loop 운용 시스템은 실제 드론 또는 시뮬레이션 상의 자율비행 UAV 또는 소형 UAV 시스템에게 중요한 요소임.
- ▶ 아래 그림은 off-the-shelf Intel Aero RTF MAV(Micro Aero Vehicle) 비행체 시스템의 3가지 주요 하위시스템을 보여줌.



그림 97 The three main subsystems of UAV

- ▶ 3가지 주요 하위시스템은 (1) 연산 시스템, (2) 액추에이터, (3) 센서를 포함하며, 본 기체 외의 다른 초소형비행체 또한 유사한 하위시스템 구조로 이루어짐.
- ✓ 연산 시스템은 병렬적으로 사용되는 동반 컴퓨터(Companion computer)와 비행 제어기를

포함함.

✓ 액추에이터는 모터와 프로펠러 장치를 포함함.

✓ 센서는 RGB 카메라를 포함함.

▶ 다음 그림은 현실 세계의 UAV의 closed-loop 데이터 플로우를 설명함.

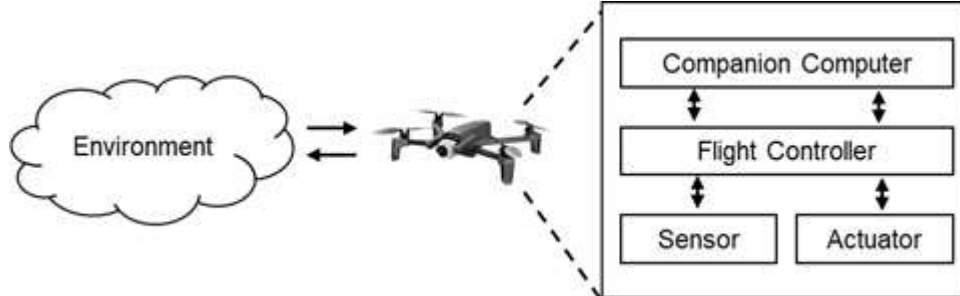


그림 98 Closed-loop data flow in a UAV in real-world

✓ UAV의 센서가 수집한 환경 정보 데이터가 연산 시스템으로 전달되고, 이것이 액추에이터의 작동으로 이어지며 다시 UAV가 비행하는 환경에 영향을 미침.

✓ 본 과정은 **UAV의 센서가 환경 데이터를 감지하고 연산 시스템이 환경 정보를 해석하고 번역하여 의사결정과정에 활용, 그리고 액추에이터를 통해 비행하는 것을 반복하는 루프 과정을 포함함.**

▶ iADA* 알고리즘 기반의 임무계획 에이전트와 DQN과 DDPG의 심층 강화 학습기반의 장애물 회피 에이전트, 시스템 모니터링 에이전트를 통합하여 전체 시스템을 구성.

▶ 경로 계획 및 재계획 에이전트(PPRA)와 시스템 모니터링 에이전트(SMA)가 API(Application Processing Interfacing)을 통해 통합됨.

✓ 시스템 모니터링 에이전트는 비행체가 임무를 시작하는 순간부터 실시간으로 비행 데이터와 비행체의 정보를 수신함.

✓ 경로 계획 및 재계획 에이전트와 시스템 모니터링 에이전트는 경로 정보를 장애물 회피 에이전트(ODA)에게 커뮤니케이션 프로토콜을 통해 제공함.

✓ 경로 계획 및 재계획 에이전트로부터 시스템 모니터링 에이전트와 장애물 회피 에이전트는 프로토콜 커뮤니케이션과 API를 통해 호스트 컴퓨터로부터 센서 데이터를 수신함.

✓ 또한 본 에이전트는 현재 비행 중인 UAV에게 비행 명령을 전송함.

✓ 다음 그림은 에이전트들의 통합 플랫폼을 나타냄.

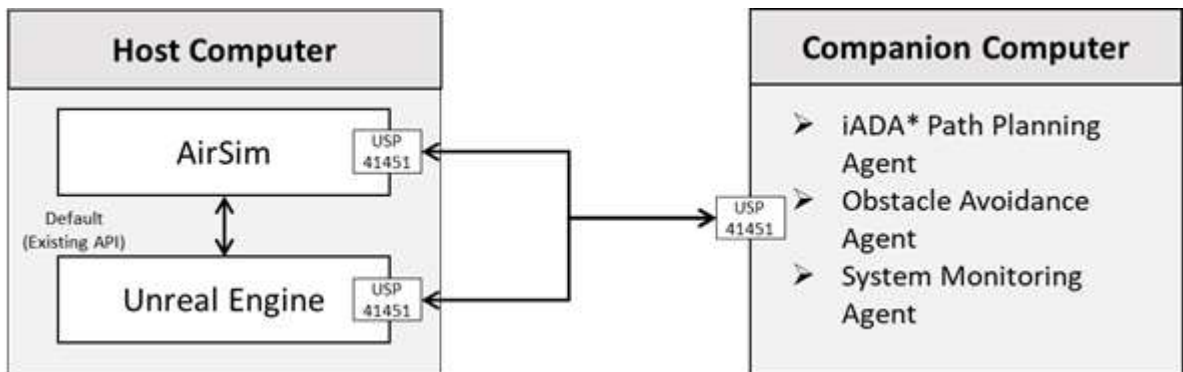


그림 99 Integration platform of different agents

- ▶ 본 검증 과정에서는 정적 장애물만이 존재하는 넓은 환경과 동적 장애물이 존재하는 중간 크기의 환경에서 검증이 진행됨.

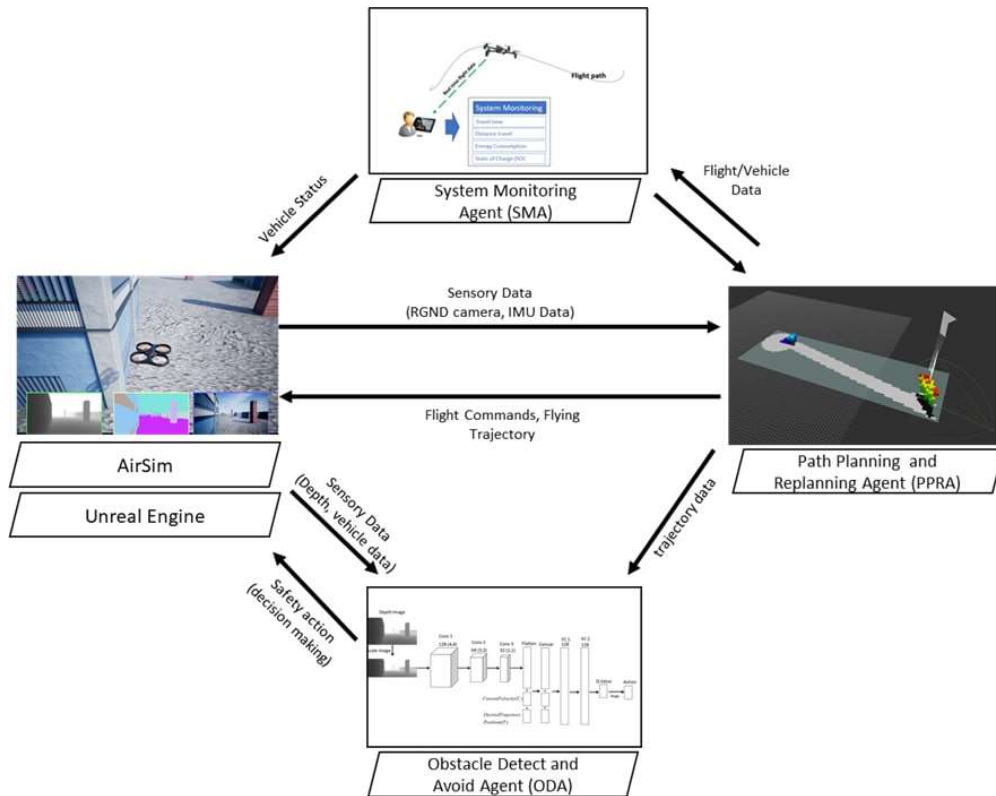


그림 100 Architecture of SITL closed-loop simulation for multi-agent based mission planning and replanning



그림 101 A Top view of static environment



그림 102 Screenshots of static environment

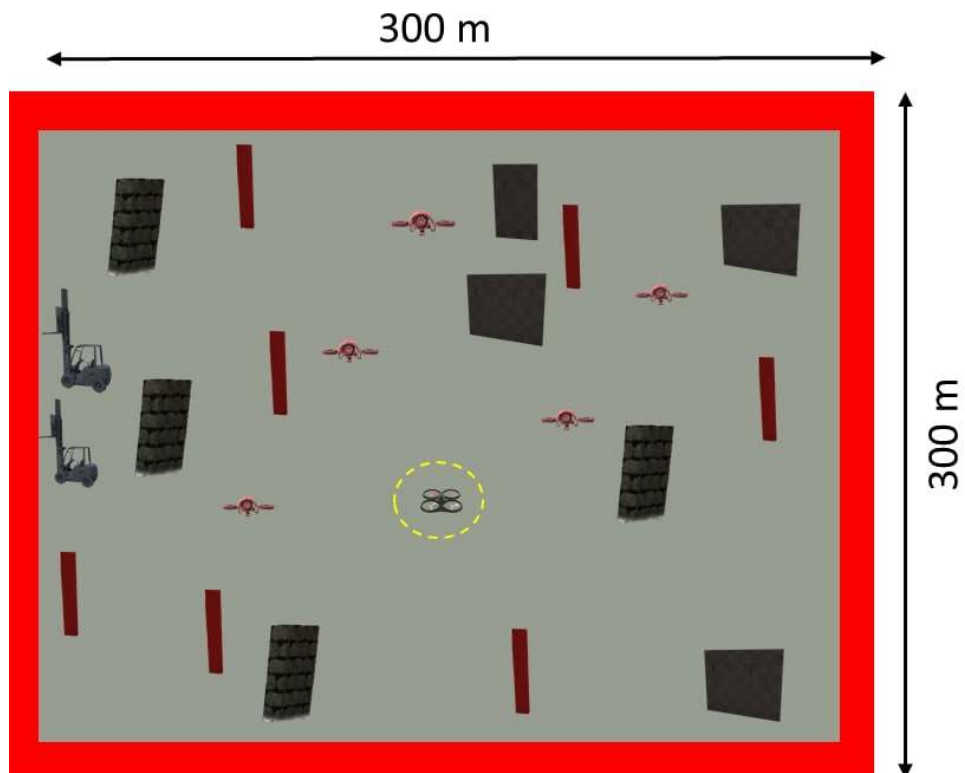


그림 103 A Top view of dynamic environment

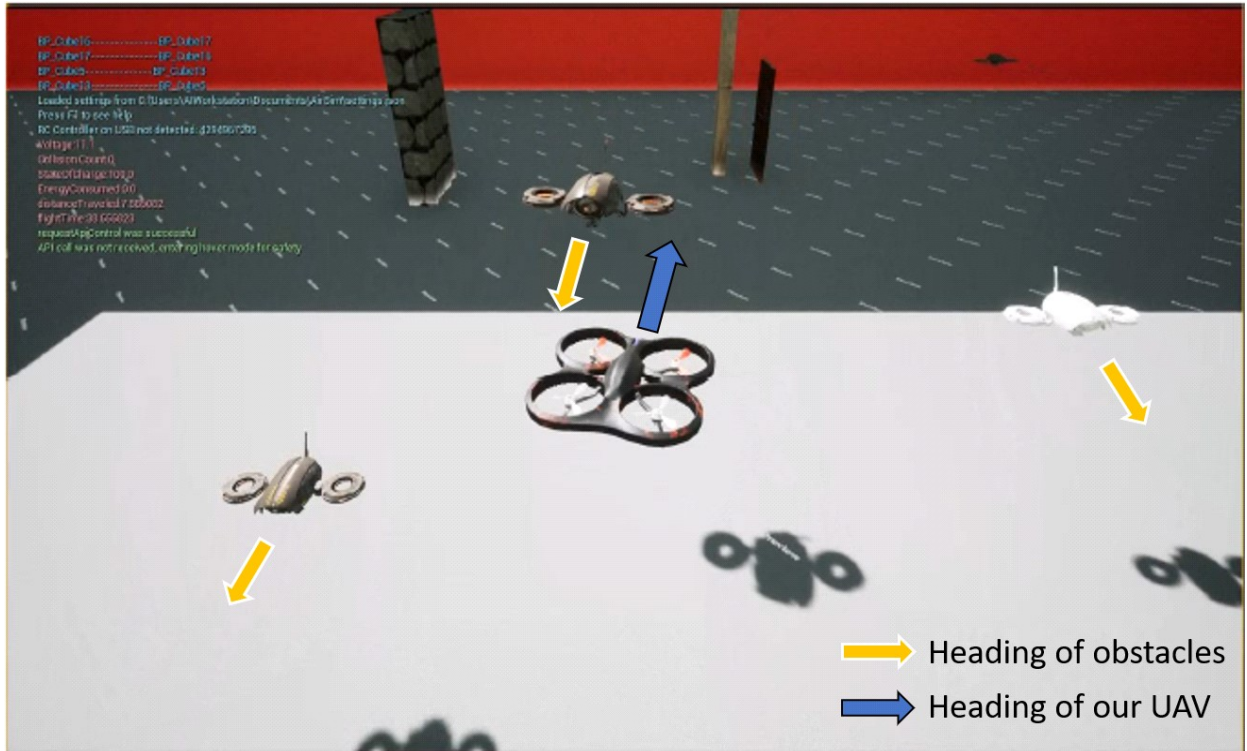


그림 104 Environment to test for dynamic obstacles

- ▶ 전체 시스템 결과는 성공과 실패 비율에 따라 평가됨.
- ▶ **테스트의 실패 경우는 드론이 장애물과의 충돌 등을 이유로 목표지점에 도달하지 못하는 상황을 의미함.**
- ▶ 동일한 DQN 구현 설정 하에 정적인 환경과 동적 장애물이 존재하는 환경에 대해서 평가됨.
 - ✓ 정적 환경은 300m × 200m 크기로 구성됨.
 - ✓ 동적 환경은 300m × 300m의 크기에 고도 제한이 없는 환경으로 구성되었으며, 고립 환경에서 복잡한 동적 장애물을 포함하여 동적 장애물 회피에 초점을 맞춤.
- ▶ 장애물은 5m/s에서 25m/s 범위 내에서 무작위의 속도로 방향 제한 없이 이동함.

(1) 실험 환경 설정

- ▶ UAV에게 무작위로 임무가 할당되며, iADA* 경로 계획기에 의해 계획된 경로를 따라 드론이 경로를 비행할 때 DQN 에이전트가 동적 장애물을 탐지함.
- ▶ 만약 경로를 따라 비행하는 드론의 DQN 에이전트가 장애물을 탐지하면 에이전트는 장애물을 피하기 위한 안전한 행동을 함.
- ▶ UAV가 안전한 곳에 도달한 뒤 iADA* 경로 계획기는 임무 목표지점에 도달하기 위한 경로를 재계획함.
- ▶ 이 장애물 탐지 및 회피 과정은 동적 장애물에 대해서 적용되며, 알려지지 않은 장애물의 경우 탐지하여 octree 맵이 재생성되며 경로계획 에이전트에 의해 경로가 재계획 됨.

(2) 실험 환경1 : 정적 환경

- ▶ 모든 실험 케이스에 대해서 시작점과 도착점은 무작위로 할당됨.

표 27 정적 환경에서의 다중 에이전트 테스트 결과

Test case	Failed case	Failure Rate
200	0	0%

- ▶ 본 연구에서 개발한 다중에이전트를 정적 환경에서 테스트한 결과 **실패 케이스가 존재하지 않는 성공률 100%(즉, 실패율 = 0%)**를 보임.
- ▶ 전역 경로 계획 및 재계획 알고리즘이 이미 환경을 알고 있어 모든 정적인 장애물에 대해 고려한 경로가 계획되었으므로 장애물 충돌로 인한 실패는 발생하지 않음.
- ▶ iADA* 경로 계획 에이전트와 DQN, DDPG 기반 장애물 탐지 및 회피 에이전트를 통합해 수행한 경우 실패한 에피소드가 발생하지 않았음.
- ▶ 이는 장애물 탐지 및 회피 에이전트가 수행하는 지역 경로 계획은 목표지점까지 최적 경로로 이동하지 않더라도 목표지점에 가까워지는 방향으로 이동하여 실패하지 않고 임무를 달성할 수 있기 때문임.

(3) 실험 환경2 : 동적 환경

- ▶ 가) 장애물 회피 에이전트 개발 - (3) 장애물 회피 시험 결과 및 분석 파트에서 제시한 동적 장애물 환경에서 테스트를 수행.
- ▶ 테스트 환경의 정적 및 동적 장애물 환경은 다음과 같음.

표 28 복잡한 테스트 환경의 장애물 수

총 장애물 개수	동적 장애물 수
22	11

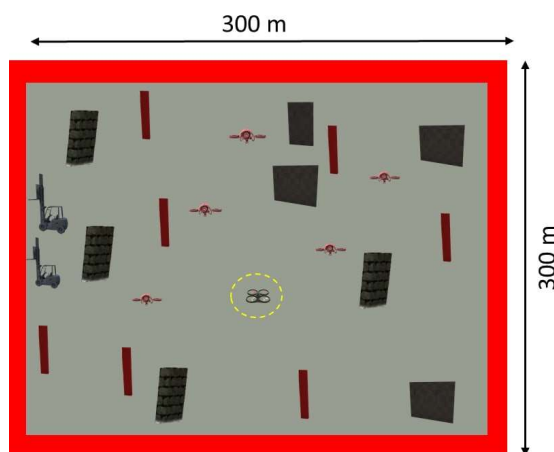


그림 105 A Top view of complex dynamic environment

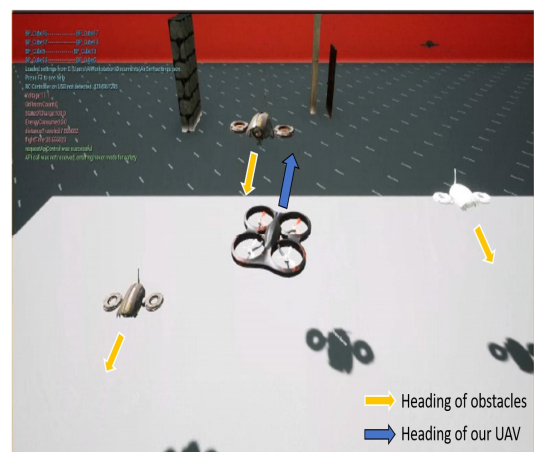


그림 106 Environment to test for dynamic obstacles

표 29 단순한 테스트 환경의 장애물 수

총 장애물 개수	동적 장애물 수
22	11

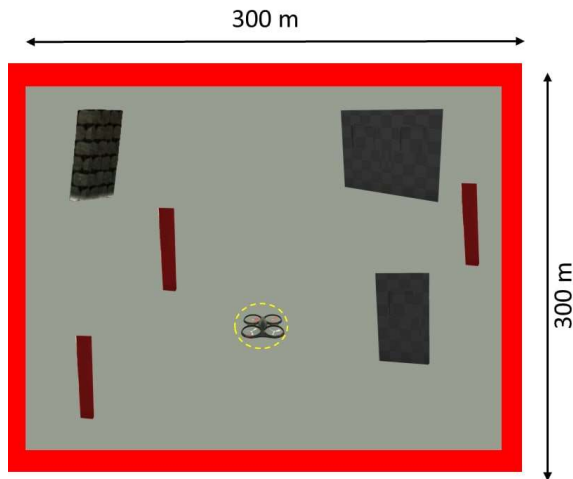


그림 107 A Top view of simple dynamic environment

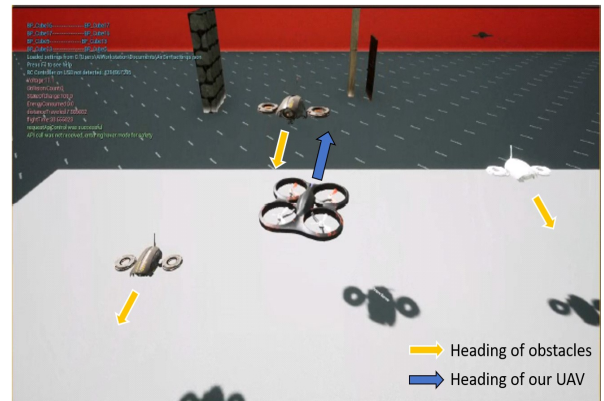


그림 108 Example for dynamic obstacles

표 32 복잡한 동적 환경에서의 다중 에이전트 테스트 결과

Test case	Failed case	Failure rate
200	3	1.5%

- ▶ 총 200번의 테스트 케이스를 수행한 결과, 3번의 실패 케이스가 존재해 98.5%의 성공률(즉, 1.5%의 실패율)을 달성함.
- ▶ 본 결과는 장애물 탐지 및 회피 에이전트의 목표 성공률인 98%(실패율 2% 이내)를 달성한 결과로 신뢰성 있는 에이전트를 개발함을 보여줌.

표 33 단순한 동적 환경에서의 다중 에이전트 테스트 결과

Test case	Failed case	Failure rate
200	1	0.5%

- ▶ 단순한 동적 환경에서는 실패 케이스가 1번 발생해 실패율은 0.5%로 나타남.

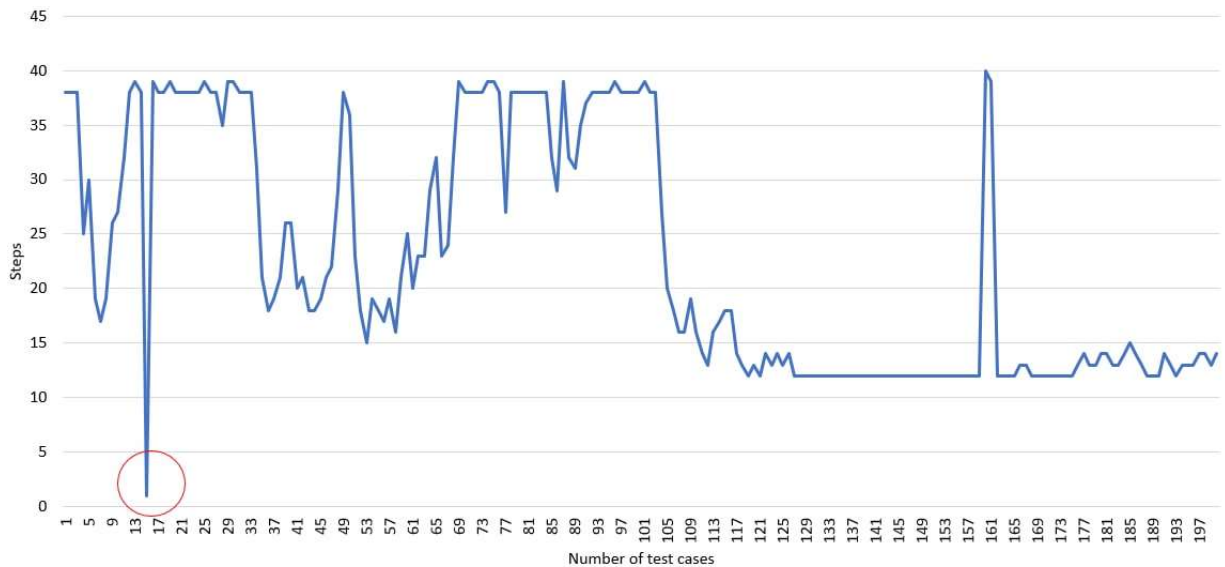


그림 109 Test result on complex environment

- ▶ 본 실패 경우는 출발지점에서 이륙 전 장애물이 다가와 발생한 경우로 에이전트의 성능이 문

제가 되는 것이 아니라 무작위로 위치하는 장애물로 인한 것임.

- ▶ 따라서 개발한 장애물 탐지 및 회피 에이전트의 성능은 우수함을 확인할 수 있음.

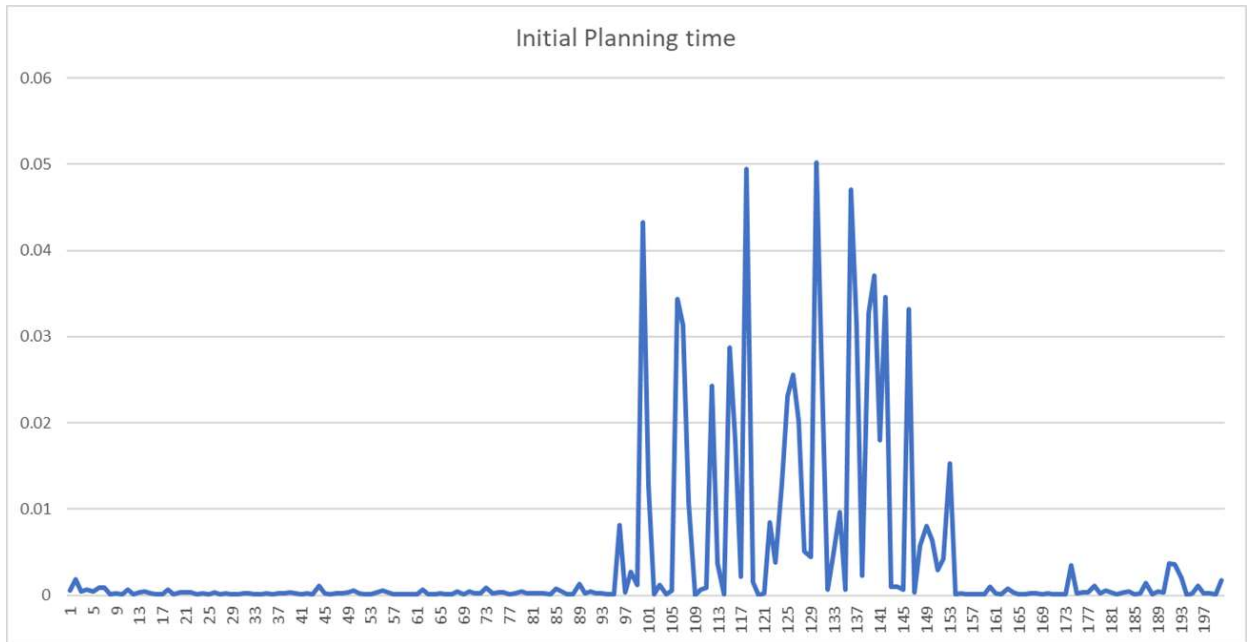


그림 110 초기 경로 계획 소요시간

- ▶ 상단 그림은 테스트 케이스 별 iADA* 경로 계획기의 초기 경로 계획 소요시간을 나타냄.
- ▶ 초기 경로 계획은 **200번의 테스트 케이스 평균 약 0.004058초가 소요됨.**
- ▶ 본 결과는 목표 소요시간인 120초보다 상당히 빠른 결과로 iADA* 경로계획기의 우수한 성능을 보여줌.
- ▶ 시작지점이 고정되어있고, 목표지점을 사용자가 원하는 지점으로 설정한 경우 iADA* 경로 계획기에 의한 실패 케이스는 발생하지 않았으므로, 실패율은 장애물 회피 에이전트의 성능에 의해 결정됨.

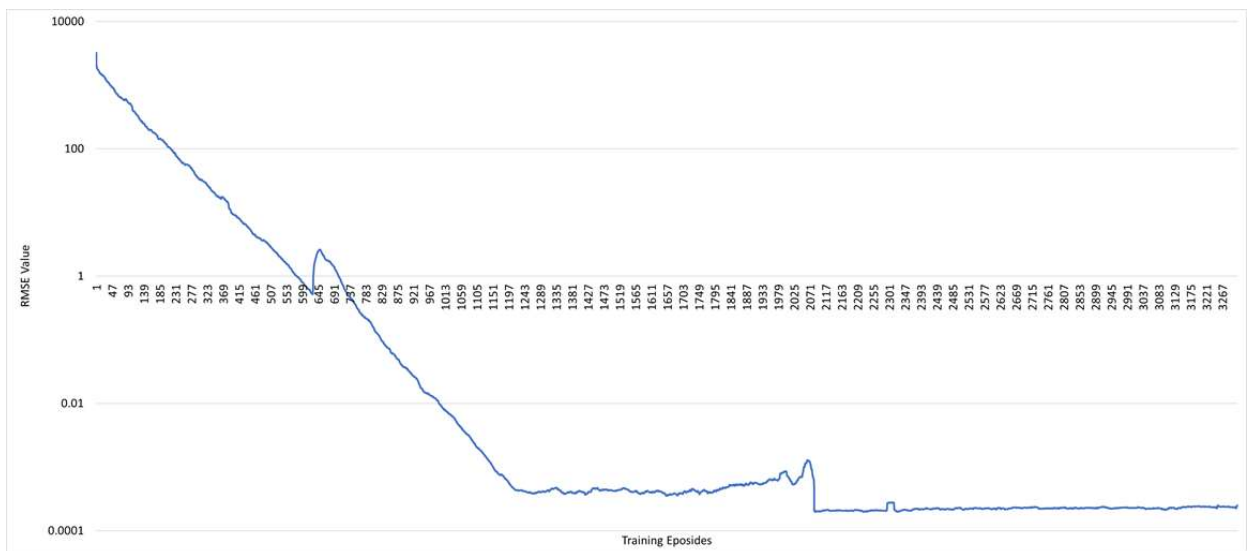


그림 111 복잡한 환경에서의 강화 학습 기반 장애물 회피 에이전트의 네트워크 정밀도

- ▶ 상단 그림은 장애물 탐지 및 회피 에이전트의 네트워크 정밀도를 나타낸다.
- ▶ 학습결과 **네트워크 정밀도인 RMSE는 약 0.0002의 0에 가까운 오차율로 본 강화학습 네트워크가 정확도가 높음**을 보여줌.
- ▶ 일반적으로 강화 학습 기반 에이전트의 학습 수행 시 RMSE가 0에 가까워지고 임무를 충분히 달성할 수 있는 수준으로 학습이 진행된다면 학습을 종료함.
- ▶ 이 경우, **동적 장애물이 많이 포함된 복잡한 환경을 가정하여 RMSE가 약 0.0002로 나타나, 학습 에피소드 수를 증가시킬 경우 네트워크 정밀도를 개선가능함.**
- ▶ 장애물 탐지 및 회피 에이전트가 장애물을 회피하지 못해 실패한 케이스는 동적 장애물이 UAV의 후방 또는 측면에서 다가올 경우, 카메라를 통한 장애물 인식이 어려워 발생한 것으로 분석됨.

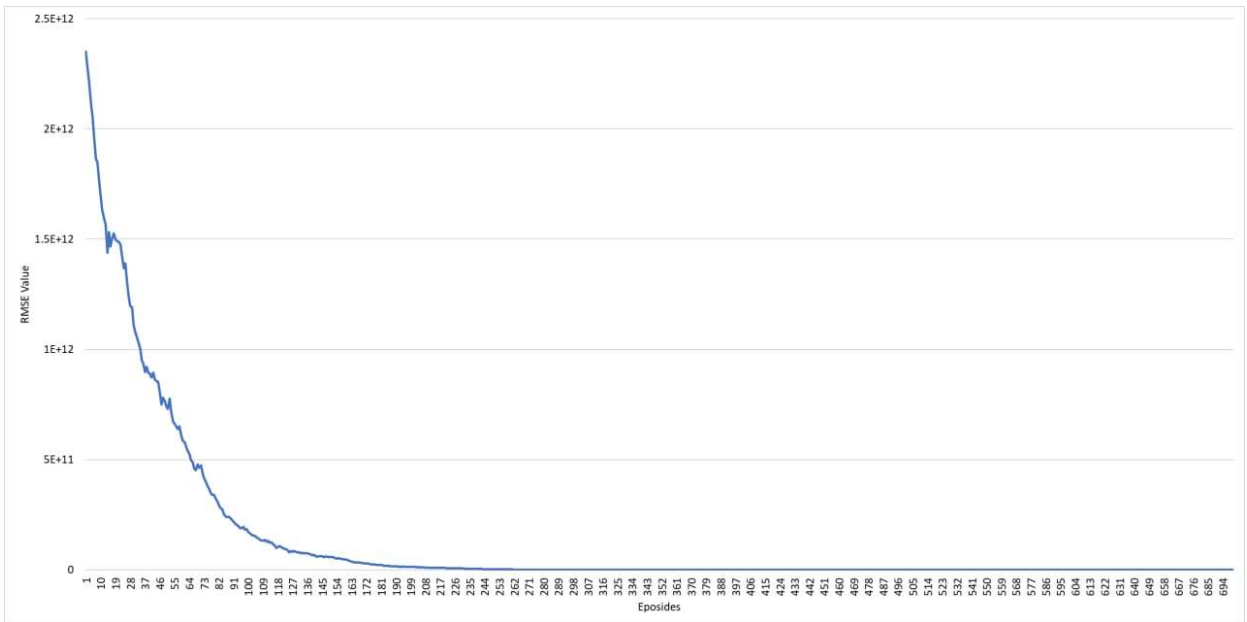


그림 112 단순한 환경에서의 강화학습기반 장애물 회피 에이전트의 네트워크 정밀도

- ▶ 학습 결과 **네트워크 정밀도는 약 0.00003으로 나타남.**
- ▶ 장애물의 수가 감소한 단순한 환경에서 진행한 테스트 결과로 본 연구에서 개발한 장애물 탐지 및 회피 에이전트의 네트워크 정밀도가 매우 높음을 확인할 수 있음.

(4) 다중에이전트 통합 결과 분석

- ▶ 본 연구는 UAV의 자율 운행 지능 수준을 향상하고, 사람의 조종으로부터 독립성을 높이기 위해 수행됨.
- ▶ 전역 경로 계획 및 재계획과 지역 경로 계획, 시스템 모니터링을 포함하여 자율 비행체를 위한 실시간 임무 계획 시스템을 개발함.
- ▶ 본 연구에서 개발한 에이전트는 모두 실시간 시뮬레이터인 AirSim 시뮬레이터에서 테스트를 수행함.

◎ 전역 경로 계획 및 재계획 에이전트

- ▶ 전역 경로 계획 및 재계획 알고리즘으로 새로운 애니타임 휴리스틱 알고리즘인 **iADA*(improved Anytime Dynamic A*)**를 개발하였고, 본 에이전트를 활용하여 **2차원뿐만**

아니라 3차원 환경에서도 실시간으로 최적 경로를 제시함.

- ▶ 무작위의 복잡한 환경에서 iADA*는 ADA*와 비교할 때 상당히 빠른 계산시간을 보였으며, 계산 비용또한 가장 낮은 것으로 나타남.
- ▶ 정적 장애물이 존재하는 3차원 환경에서의 테스트 결과, 실패율이 약 1% 정도로 매우 우수한 테스트 결과를 보임.

◎ 장애물 탐지 및 회피 에이전트를 활용한 지역 경로 계획

- ▶ 심층 강화 학습을 바탕으로 장애물 탐지 및 회피 에이전트를 개발함.
- ▶ 동적 장애물이 존재하는 3차원 환경에서 테스트를 수행한 결과, 98.5%의 매우 높은 성공률을 보임.

◎ 시스템 모니터링 에이전트

- ▶ 비행 중인 UAV로부터 실시간으로 데이터를 수집하여 고장, 오류 여부를 파악하고 임무 수행 가능여부를 판단함.
- ▶ 비행 거리, 비행시간, SOC 등을 실시간으로 모니터링하여 실시간 시뮬레이터인 AirSim 시뮬레이터 상에 구현함.

◎ SITL(Software-In-The-Loop) 프레임워크 제시

- ▶ 본 연구에서는 AirSim 시뮬레이터와 companion 컴퓨터를 기반으로 하는 실시간 임무 계획 및 재계획 에이전트를 개발하기 위해 SITL 프레임워크를 제시함.
- ▶ 본 프로세스는 실시간 시스템에서 데이터를 수집하고, 비행 상황 및 환경을 관찰함.
- ▶ 전체 프레임워크를 활용하여 다중에이전트 통합 시뮬레이션을 진행하여 높은 성공률을 보임을 입증함.

나. 연구 성과

연차	세부 연구 목표 (연구계획서의 목표)	비중 (%)	달성도 (%)	자체평가
1차년도	시뮬레이션 기본연구	10	100%	FDN 구축 연구, 공기역학적 모델링, 제어기 구조 연구등 컴퓨터 환경을 이용한 드론 시뮬레이션을 위한 기본 연구를 수행하였음.
	장애물 인식 및 회피 인공지능 에이전트 구현을 위한 기본 연구	20	100%	장애물 인식 및 회피를 위한 AI 에이전트를 개발에 적용될 ANN, CNN, RL, DRL과 같은 알고리즘에 대한 특징을 파악하고 이를 분석하여 본 연구의 목적에 부합하도록 구성하였음.
	DQN을 이용한 장애물 회피 네트워크 개발	10	100%	DQN을 이용한 에이전트를 개발하기 위해 입력과 출력 동작에 대해 파악하고 AirSim 시뮬레이터에 연결하여 알고리즘의 효율성에 대한 테스트를 일정대로 진행하였음.
	경로 계획을 위한 인공지능 에이전트 개발	10	100%	- 실시간 경로탐색을 위해 기존의 ADA* 알고리즘을 개선하여 iADA*를 이용한 인공지능 에이전트를 개발, 그에 대한 테스트를 수행하였음. - 그 결과로 기존 알고리즘 대비 2-3배의 속도 향상을 확인하였음.
2차년도	장애물 인식 및 회피 인공지능 에이전트 개발	10	100%	1차년도에 수행한 심층 강화학습 (DQN) 기반 장애물 회피 네트워크를 이용, 지역 경로 계획 수행이 가능한 장애물 탐지 및 회피 인공지능 에이전트 개발 - UE4 환경하에서 AirSim UE4 플러그인을 사용하여 벽을 포함한 정적 장애물과 드론을 포함한 동적 장애물로 환경을 만들고 테스트를 수행
	시스템 모니터링 인공지능 에이전트 개발	10	100%	무인비행체의 운항 중에 환 인지를 바탕으로 이동 거리, 기체의 위치, 이동 시간, 연료 소모량과 에너지 소모량 데이터 등의 현재 상태를 확인하기 위한 시스템 모니터링 인공지능 에이전트를 개발
	임무 재설계 알고리즘 개발	20	100%	- 무인비행체가 주어진 임무를 안전하고 효율적으로 수행할 수 있도록 출발지점에서 목적지점까지 자율비행을 위해 다음과 같이 전역/지역 경로 계획이 가능한 임무 재설계 알고리즘을 개발하였음 1) 전역 경로 계획 및 재계획 : 거리 및 비행시간이 계획된 최단 경로 계획

연차	세부 연구 목표 (연구계획서의 목표)	비중 (%)	달성도 (%)	자체평가
				2) 학습 기반 지역 경로 계획 및 장애물 회피 알고리즘 : 계획된 경로 상에서 출현하는 장애물을 회피하여 이미 세워진 전역 경로 계획을 보완
	인공지능 에이전트 통합 및 검증	10	100%	- 기 개발된 실시간 경로 계획 인공지능 에이전트, 정야물 인지 및 회피 인공지능 에이전트, 시스템 모니터링 인공지능 에이전트 등을 통합하여 3차원 시뮬레이션 환경인 AirSim에서 경로 계획 및 재계획, 경로 평활화(path smoothing), 경로 생성을 포함한 실시간 임무계획을 수행하는 소프트웨어 아키텍처와 상호작용하는 closed-loop 시뮬레이션 프레임워크를 개발하였음 - 통합된 시뮬레이션 프레임워크 상에 무인비행체, 정적/동적 장애물을 구현하고 테스트를 통한 검증을 수행
합계		100점	100%	

1) 성과 점검 결과

가) 다중에이전트 기반 최적 경로 생성 및 임무 재설계 인공지능 알고리즘

(1) 다중에이전트 기반 실시간 임무 재설계(1-1)

(가) 실시간 임무 경로 재설계(1-1-①)

● 측정 및 검증 방법

- ▶ 총 200번의 테스트를 진행 후 실시간 임무 경로 재설계에 소요되는 평균계산시간을 측정함.

● 테스트 환경

- ▶ AirSim 시뮬레이터 상에 구현한 300m x 200m 크기의 정적 장애물이 존재하는 환경에서 테스트를 수행함.
- ▶ 본 환경은 건물, 기둥 등의 장애물로 이루어져 있으며, 크기 또는 위치가 테스트 경우에 따라 바뀌지 않음.



그림 113 A Top view of static environment



그림 114 Screenshot of static environment

● 성과지표

- ▶ 총 테스트 케이스에 대해 에이전트의 평균 재계획시간을 성과지표로 함.
- ▶ 목표치 : 평균 재계획 계산 시간 < 5 sec
- ▶ 달성치 : 평균 재계획 계산 시간 = 0.000522sec

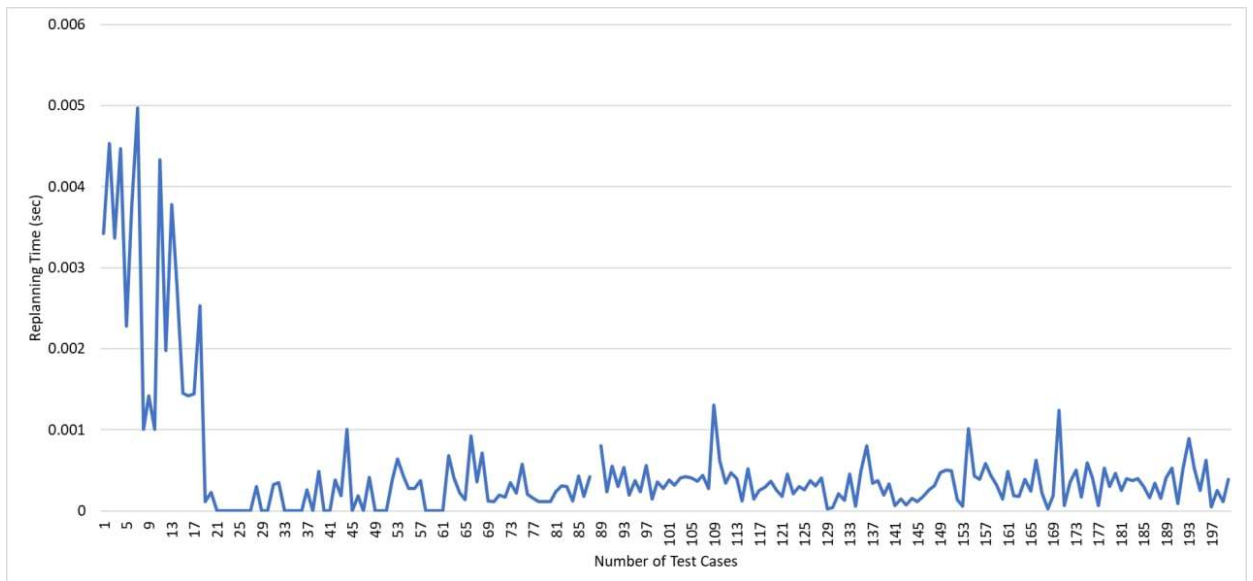


그림 115 Running times for planning in a static environment

- ▶ 상단 그림은 테스트 환경에서 수행한 200번의 테스트 케이스의 재계획 시간을 나타냄.
- ▶ **200번의 테스트 케이스 평균 약 0.000522초가 소요됨.**
- ▶ 본 결과는 목표 계산시간인 5초 이하보다 상당히 빠른 결과로 본 연구에서 개발한 전역 경로 계획 및 재계획 에이전트의 우수한 성능을 확인할 수 있음.

(나) 시스템 모니터링 에이전트 기반 임무 재설계(1-1-②)

● 측정 및 검증 방법

- ▶ 총 200번의 테스트를 진행 후 실패율을 계산함.

● 테스트 환경

- ▶ (가)에서 제시한 AirSim 시뮬레이터 상에서 구현한 300m x 200m 크기의 정적 장애물이 존

재하는 환경에서 테스트를 수행함.

● **성과지표**

- ▶ 임무 수행 중, 배터리의 SOC가 0에 근접하여 더이상 임무를 수행할 수 없는 상태를 실패 경우로 정의함.
- ▶ 이때, 총 테스트 케이스 대비 실패 케이스의 비율을 실패율로 정의함.
- ▶ 목표치 : 실패율 < 2%
- ▶ 달성치 : 실패율 = 0%

표 33 Success and Failed case analysis for system monitoring agent

Test case	Success case	Failed case	Failure rate
200	200	0	0%

- ▶ 표 33는 시스템 모니터링 에이전트 기반 임무 재설계 성능 테스트를 위해 200번의 테스트 케이스를 수행한 결과임.
- ▶ 200번의 테스트 결과, 배터리 SOC 및 비행 거리, 비행 시간, 비행 속도는 임무 달성여부와 관계없이 실시간으로 관찰이 가능하며 **UAV는 배터리의 방전 또는 고장 상황 없이 실패율 0%로 임무를 달성함.**
- ▶ 다음 그래프는 매 테스트 케이스 별 시스템 모니터링 에이전트를 통한 UAV 모니터링 결과를 보여줌.

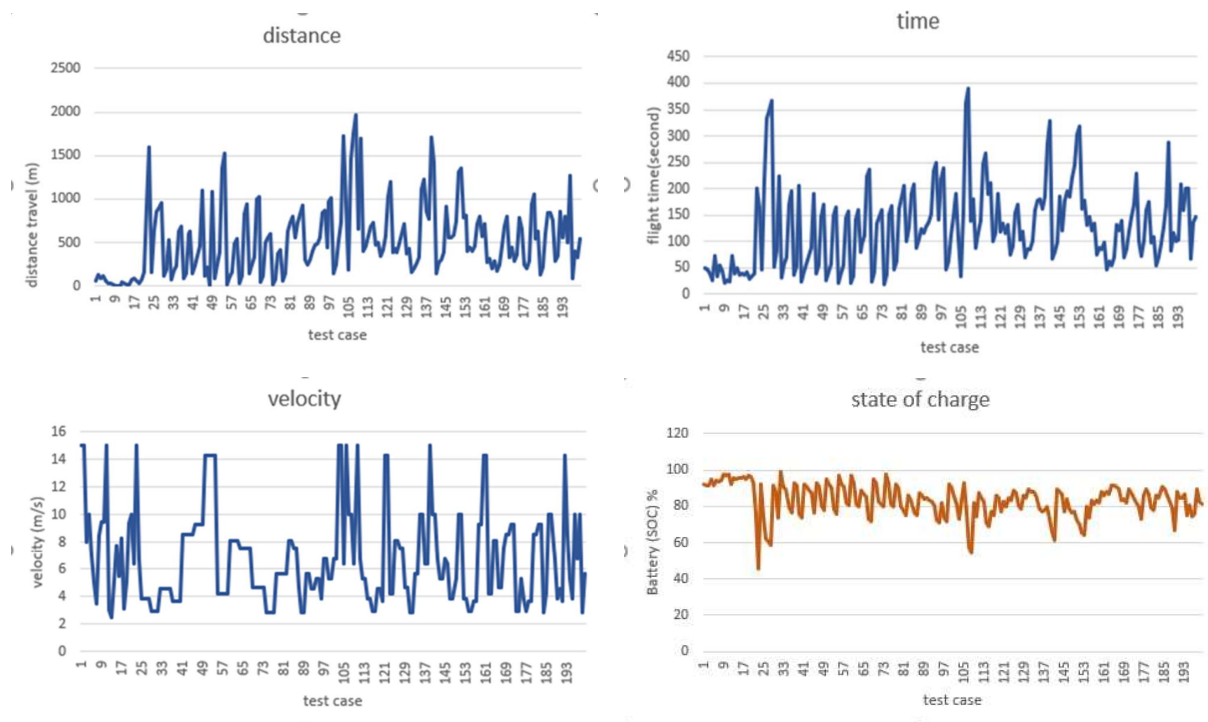


그림 116 System monitoring tested cases and battery(SOC) for success rate

(다) 강건성(1-1-③)

● **측정 및 검증 방법**

- ▶ 총 200번의 테스트를 진행 후 실패율을 계산함.

● **테스트 환경**

- ▶ (가)에서 제시한 AirSim 시뮬레이터 상에서 구현한 300m × 200m 크기의 정적 장애물이 존

재하는 환경에서 테스트를 수행함.

● 성과지표

- ▶ 임무 수행 중, 장애물과 조우 시 임무 재계획에 소요되는 시간이 (가)에서 제시한 실시간 임무 경로 재설계 계산시간의 평균보다 큰 경우를 실패 케이스로 정의함.

$$t_{meet\ obs} > t_{replanning} \quad t_{replanning} = 0.004058sec$$

✓ $t_{meet\ obs}$: 장애물을 마주하였을 때, 임무 재계획에 걸리는 시간(sec)

✓ $t_{replanning}$: (가)에서 제시한 실시간 임무 경로 재설계 평균계산시간(sec)

표 34 Success and fail cases analysis for real-time mission replanning

Test case	real-time mission replanning		
	Success case	Failed case	Failure rate
200	198	2	1%

- ▶ 테스트 결과, 에이전트는 1%의 낮은 실패율을 보이며 임무를 재설계함.

(2) 초기 최적 경로 생성(1-2)

(가) 초기 최적 경로 생성(1-2-①)

● 측정 및 검증 방법

- ▶ 총 200번의 테스트를 진행 후 평균 시간을 측정함.

● 테스트 환경

- ▶ AirSim 시뮬레이터 상에 구현한 300m × 200m의 정적 장애물이 존재하는 환경에서 테스트를 수행함.
- ▶ 본 환경은 건물, 기둥 등의 장애물로 이루어져 있으며, 크기 또는 위치가 테스트 경우에 따라 바뀌지 않음.



그림 121 A Top view of static environment

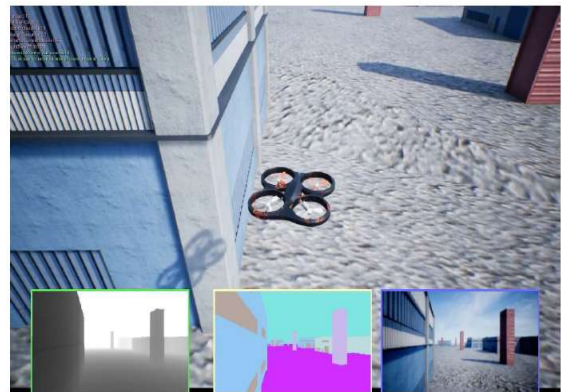


그림 122 Screenshot of static environment

● 성과지표

- ▶ 총 테스트 케이스에 대해 에이전트의 평균 계산시간을 성과지표로 함.
- ▶ 목표치 : 초기 최적 경로의 평균 계산시간 < 120sec
- ▶ 달성치 : 초기 최적 경로의 평균 계산시간 = 0.004058sec

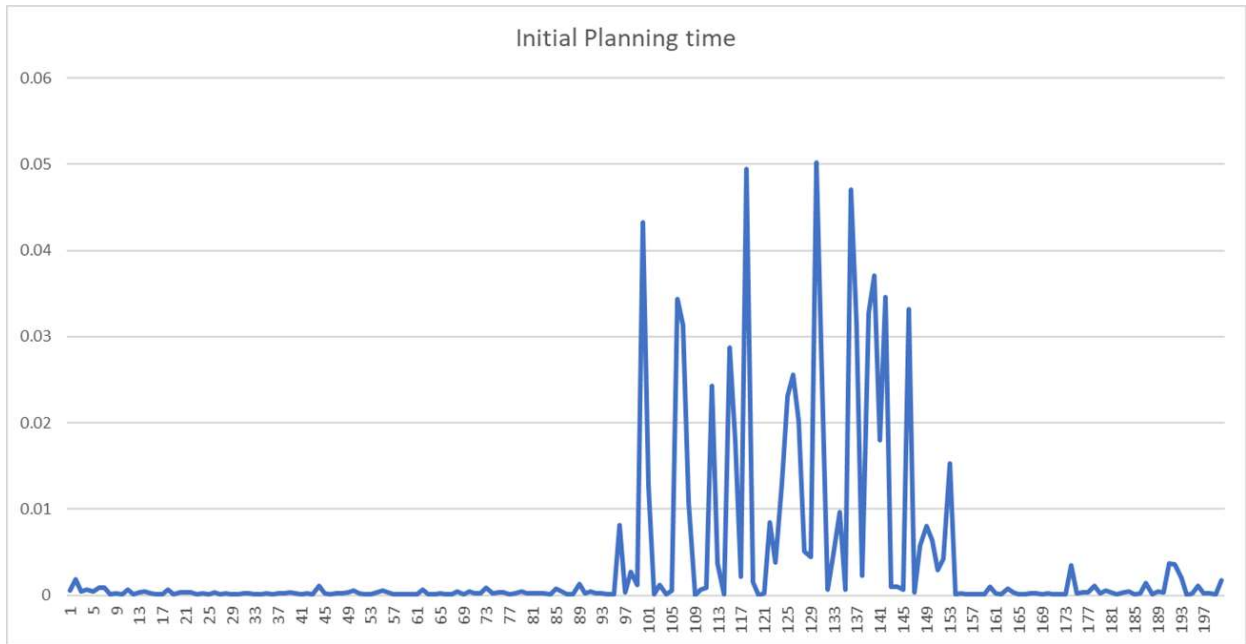


그림 123 Running times for initial path planning in a static environment

- ▶ 초기 경로 계획은 200번의 테스트 케이스 평균 0.004058초가 소요됨.
- ▶ 본 결과는 목표 소요시간인 120초보다 약 30만배 빠른 결과로 iADA* 경로계획기의 우수한 성능을 보여줌.

(나) 강건성(1-2-②)

● 측정 및 검증 방법

- ▶ (가)에서 수행한 테스트 상에서의 실패율을 확인함.

● 테스트 환경

● 성과지표

- ▶ 전역 경로 계획 및 재계획 에이전트가 출발지점으로부터 도착지점까지의 경로를 계획하지 못하는 경우를 실패 경우로 정의함.
- ▶ 따라서 여기서는 전체 테스트 케이스 대비 실패 케이스의 비율을 실패율로 정의함.
- ▶ 목표치 : (경로계획)실패율 < 2%
- ▶ 달성치 : (경로계획)실패율 = 0%

표 35 Success and fail cases analysis for initial optimum path planning

Test case	Static environment		
	Success case	Failed case	Failure rate
200	200	0	0%

- ▶ 표 35는 300m × 200m의 크기의 정적 환경 맵에서 iADA* 전역 경로 계획 에이전트를 활용한 초기 최적 경로 생성 테스트 결과를 보여줌.
- ▶ 본 연구에서 개발한 경로 계획 에이전트를 활용하여 테스트를 수행한 결과 실패 케이스가 존재하지 않는 실패율 0%를 보임.
- ▶ 전역 경로 계획 및 재계획 에이전트가 이미 환경을 파악해 모든 정적 장애물을 고려한 경로를 계획하였으므로 초기 최적 경로 생성은 성공적으로 수행됨.

(3) 시뮬레이션(1-3)

(가) 시뮬레이션 상 무인기 구현(1-3-①)

● 측정 및 검증 방법

- ▶ 총 200번의 테스트 케이스를 진행한 후 실패율을 계산함.

● 테스트 환경

- ▶ AirSim 시뮬레이터 상에 구현한 300m × 200m 크기의 정적 장애물이 존재하는 환경에서 테스트를 수행함.



그림 124 A Top view of static environment



그림 125 Screenshot of static environment

● 성과지표

- ▶ 개발한 다중에이전트 기반 인공지능 알고리즘과 AirSim 시뮬레이터를 연결하는 과정에서 정상적으로 무인기가 구현되지 않는 경우를 실패 케이스로 정의함.
- ▶ 전체 테스트 케이스 대비 실패 케이스의 비율을 실패율로 정의함.
- ▶ 목표치 : 실패율 < 1%
- ▶ 달성치 : 실패율 = 0%

표 36 Success and fail cases analysis for initial optimum path planning

Test case	Static environment		
	Success case	Failed case	Failure rate
200	200	0	0%

- ▶ 초기 경로 계획 성능 테스트 결과, 0%의 실패율로 테스트를 수행함.
- ▶ 결과적으로 시뮬레이션 상 무인기 구현에 오류가 없음을 확인할 수 있음.

나) 장애물 인식 및 회피 인공지능 알고리즘

(1) DQN 기반 장애물 인식 및 회피(2-1)

(가) 실시간 장애물 회피(2-1-①)

● 측정 및 검증 방법

- ▶ 총 200번의 테스트를 진행 후 평균 반응속도를 계산함.

● 테스트 환경

- ▶ AirSim 시뮬레이터 상에 구현한 300m × 300m 크기의 동적 장애물이 존재하는 환경에서 테스트

스트를 수행함.

- ▶ 테스트 환경 1 : 본 환경은 총 11개의 동적 장애물과 11개의 정적 장애물이 존재하는 복잡한 환경으로, 목표 도착지점은 매 테스트 케이스마다 무작위로 할당됨.

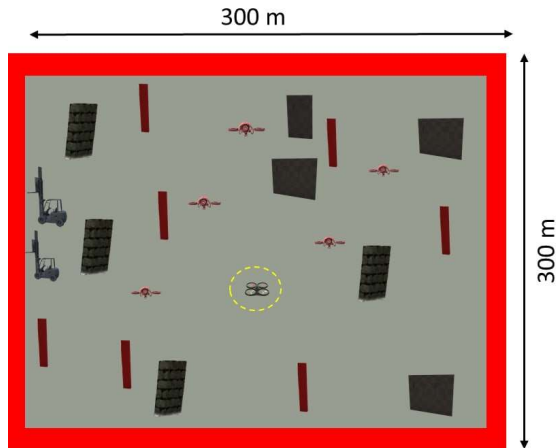


그림 126 A Top view of complex dynamic environment

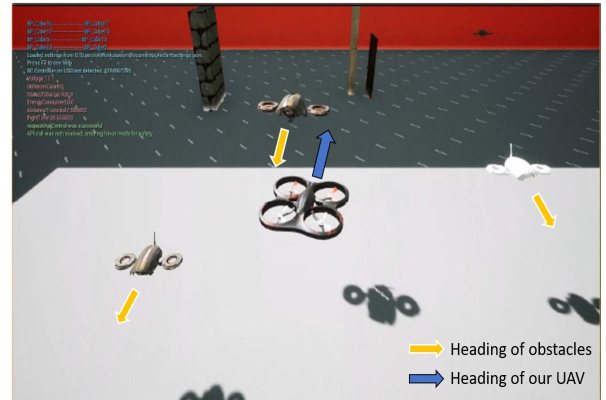


그림 127 Environment to test for dynamic obstacles

- ▶ 테스트 환경 2 : 본 환경은 총 3개의 동적 장애물과 3개의 정적 장애물이 존재하는 단순한 환경으로, 목표 도착지점은 매 테스트 케이스마다 무작위로 할당됨.

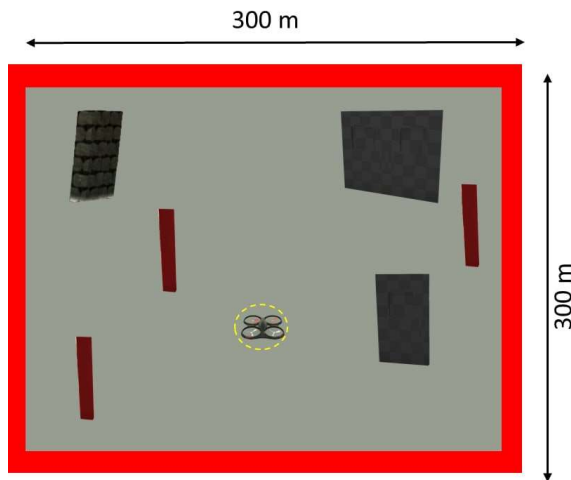


그림 128 A Top view of simple dynamic environment

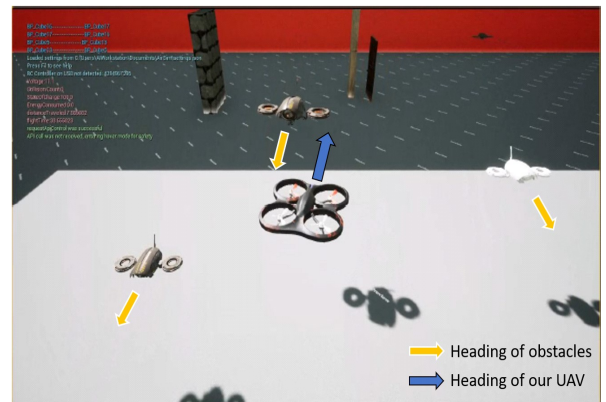


그림 129 Example for dynamic obstacles

◎ 성과지표

- ▶ 반응시간은 각 time step마다 입력 데이터를 에이전트가 입력 데이터를 받아 제어 명령 (action)을 내리는 데 소용되는 시간을 의미함.
- ▶ 목표치 : 반응속도 < 1/30sec
- ▶ 달성치 : 테스트 환경 1 : 반응속도 < 0.80sec
테스트 환경 2 : 반응속도 < 0.0042sec
- ▶ 본 연구에서 반응속도는 특정 행동 수행 후 다음 행동을 수행하는 데 소용되는 시간을 측정하여 계산함.

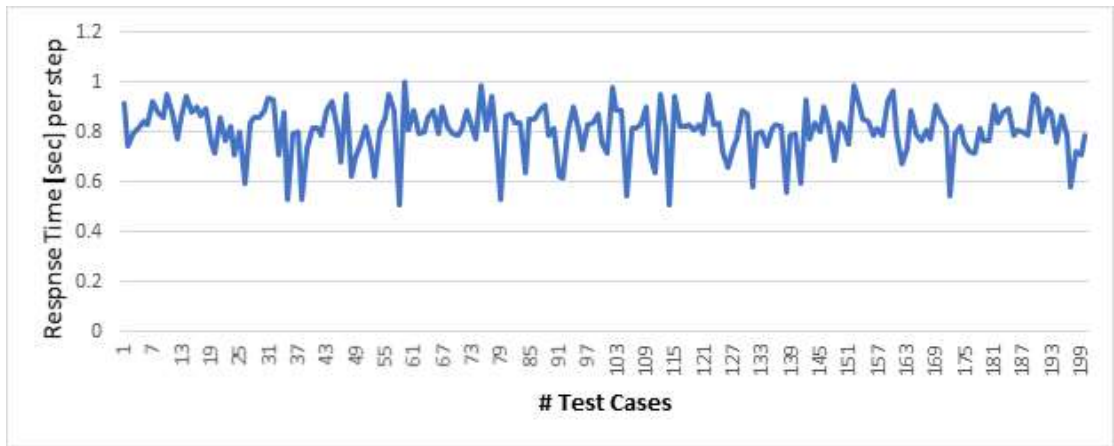


그림 130 Tested DQN network result run on complex environment

(나) 딥러닝 네트워크 정밀도(2-1-②)

● 측정 및 검증 방법

- ▶ 인공지능 네트워크의 정확도를 측정하기 위해 전체 데이터의 25%를 이용하여 테스트를 수행함.

● 테스트 환경

- ▶ 테스트 환경 1 : (가)에서 제시한 AirSim 시뮬레이터상에 구현한 300m x 300m 크기의 동적 장애물이 존재하는 복잡한 환경에서 테스트가 수행됨.
- ▶ 테스트 환경 2 : (가)에서 제시한 300m x 300m 크기의 동적 장애물이 존재하는 단순한 환경에서 테스트가 수행됨.

● 성과지표

- ▶ 인공지능 네트워크의 정확도는 테스트 수행 시 RMSE 값의 최종 수렴 값으로 정의함.
- ▶ 목표치 : $RMSE < 0.0001$
- ▶ 달성치 : 테스트 환경 1: $RMSE < 0.0002$
테스트 환경 2: $RMSE < 0.00003$

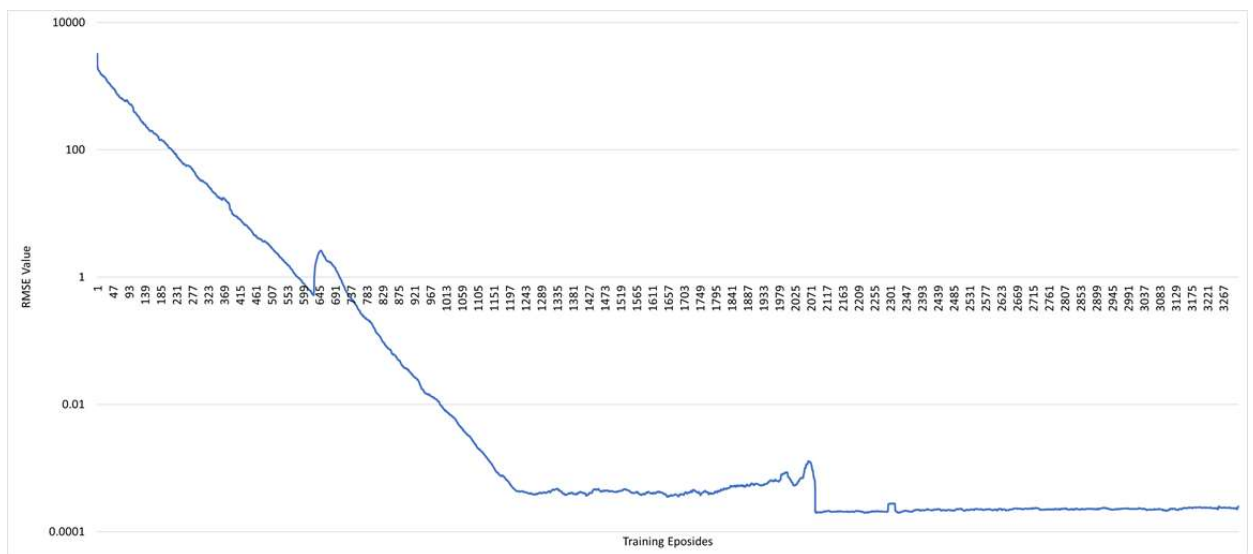


그림 131 복잡한 환경에서의 강화학습기반 장애물 회피 에이전트의 네트워크 정밀도

- ▶ 학습결과 **네트워크 정밀도는 약 0.0002로 나타남.**
- ▶ 본 테스트 환경이 **다수의 동적 장애물을 포함한 복잡한 환경으로 RMSE가 0.0002로 나타나나 학습 에피소드 수를 증가시킬 경우 충분히 0.0001에 도달이 가능함.**

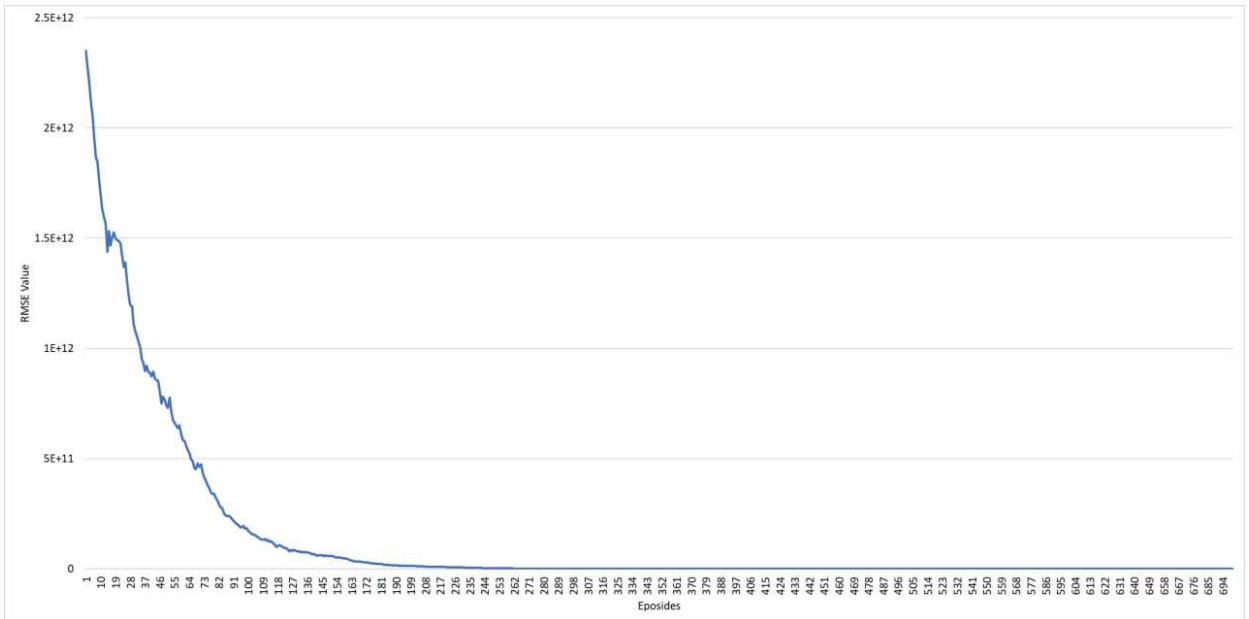


그림 132 단순한 환경에서의 강화학습기반 장애물 회피 에이전트의 네트워크 정밀도

- ▶ 학습 결과 **네트워크 정밀도는 약 0.00003으로 나타났으며 이는 목표치의 3.33배에 달함.**
- ▶ 장애물의 수가 감소한 단순한 환경에서 진행한 테스트 결과로 본 연구에서 개발한 장애물 탐지 및 회피 에이전트의 네트워크 정밀도가 매우 높음을 확인할 수 있음.

(다) 강건성(2-1-③)

◎ 측정 및 검증 방법

- ▶ 총 200번의 테스트를 진행한 후 실패율을 계산함.

◎ 테스트 환경

- ▶ 테스트 환경 1 : (가)에서 제시한 AirSim 시뮬레이터상에 구현한 300m × 300m 크기의 동적 장애물이 존재하는 복잡한 환경에서 테스트가 수행됨.
- ▶ 테스트 환경 2 : (가)에서 제시한 300m × 300m 크기의 동적 장애물이 존재하는 단순한 환경에서 테스트가 수행됨

◎ 성과지표

- ▶ **드론이 장애물과 충돌해 목표지점에 도달하지 못한 경우를 실패 케이스로 정의**하며, 이때 전체 테스트 케이스 대비 실패 케이스의 비율을 실패율로 정의함.
- ▶ 목표치 : 실패율 < 2%
- ▶ 달성치 : 테스트 환경 1 : 실패율 = 1.5%
테스트 환경 2 : 실패율 = 0.5%

표 37 복잡한 동적 환경에서의 다중 에이전트 테스트 결과

Test case	Failed case	Failure rate
200	3	1.5%

- ▶ 총 **200번의 테스트 결과 3번의 실패가 존재해 실패율은 1.5%로 나타남.**

- ▶ 본 결과는 목표 실패율인 2% 이내를 달성한 결과로 신뢰성 있는 에이전트를 개발함을 확인할 수 있음.

표 38 단순한 동적 환경에서의 다중 에이전트 테스트 결과

Test case	Failed case	Failure rate
200	1	0.5%

- ▶ 단순한 동적 환경에서는 실패 케이스가 200번 중 1번으로 실패율은 0.5%로 나타남.

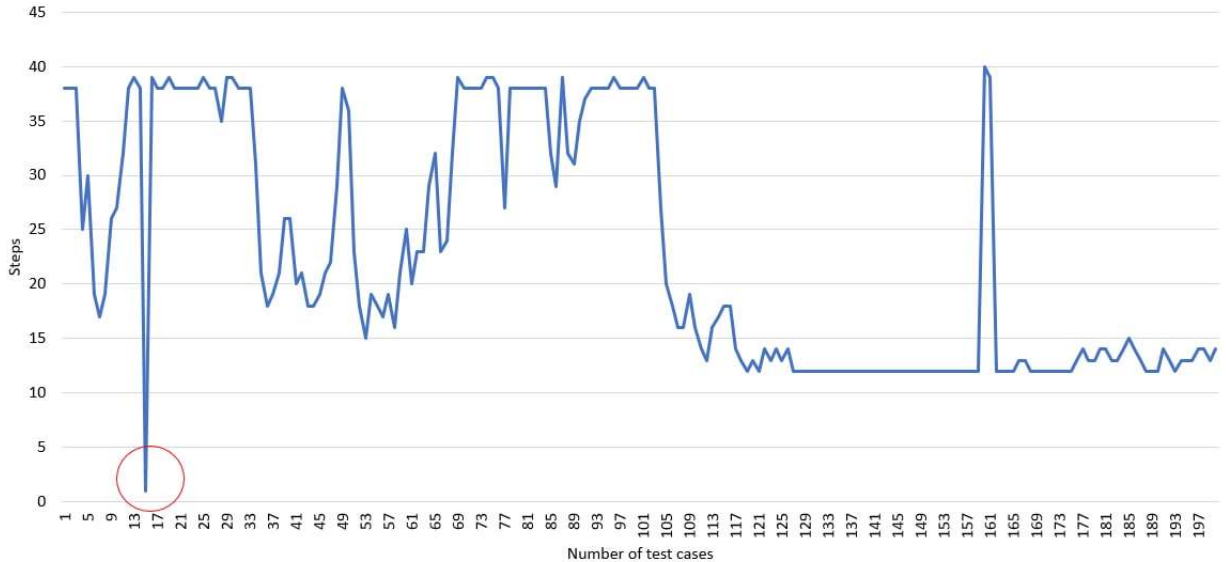


그림 133 Test result on complex environment

- ▶ 본 실패 경우는 출발지점에서 이륙 전 장애물이 다가와 발생한 경우로 에이전트의 성능이 문제가 되는 것이 아니라 무작위로 위치하는 장애물로 인한 것임.
- ▶ 따라서 개발한 장애물 탐지 및 회피 에이전트의 성능은 우수함을 확인할 수 있음.

2) 본 연구의 실용화를 위한 향후 확장 연구 내용

가) 알고리즘 및 소프트웨어의 신뢰성 확보 방안

(1) 소프트웨어 신뢰성 테스트

- ▶ 신뢰성 테스트는 소프트웨어가 특정 환경에서 특정 시간 동안 고장 또는 실패 없는 운용을 할 수 있는지 여부를 확인하는 테스트 프로세스임.
- ▶ 신뢰성 테스트의 목적은 소프트웨어가 예상 목표를 만족하며 버그가 없고 충분히 신뢰할 수 있는지를 확인하는 것임.
- ▶ 신뢰성은 “동일한 수준을 얻음”을, “신뢰할 수 있는”이란 말은 매번 동일한 결과를 낳아 믿을 수 있음을 의미하며 이 의미가 신뢰성 테스트에도 적용됨.
- ▶ 신뢰성 테스트의 예는 다음과 같음.
 - ✓ 상점에서 PC가 고장없이 8시간동안 작동할 가능성은 99%이며, 이를 신뢰성이라고 함.
 - ✓ 신뢰성 테스트는 3가지 분야로 카테고리화 할 수 있음.

- Modeling(모델링)
- Measurement(측정)
- Improvement(개선)

- ▶ 다음은 실패할 가능성을 계산하는 수식임.

$$Probability = \frac{Number\ of\ failing\ cases}{Total\ number\ of\ cases\ under\ consideration}$$

(2) 알고리즘 신뢰성 테스트

- ▶ 알고리즘의 신뢰성과 정확성은 소프트웨어 시스템과 매우 유사한 방식으로 평가됨.
- ▶ 경로 계획, 시스템 모니터링, 장애물 인식 및 회피를 포함한 자율비행 시스템의 경우 신뢰성, 정확성은 성공률, 계산 시간, 알고리즘 실패 여부로 판단 가능함.

✓ 성공률

- 장애물과의 충돌 없이 드론이 임무를 끝낸 비율로, 수치가 100%에 가까울수록 에이전트의 성능이 우수함을 의미함.

✓ 계산시간

- 실시간 시스템의 경로 계획 및 재계획의 경우 가장 중요한 부분이 계산시간임.
- 알고리즘이 경로를 계획하는 시간이 너무 오래 걸린다면 UAV가 임무를 완수하는 것이 불가능할 수 있으며, 재계획된 경로를 기다리는 과정에서 동적 장애물과의 충돌이 발생할 수 있음.

✓ 알고리즘 실패 여부

- 알고리즘 자체의 오류로 인해 임무를 수행할 수 없는 경우를 실패로 하며, 실패율을 확인하여 평가함.
- 소프트웨어 오류 여부 판단 방법과 동일함.

(3) 신뢰성 확보 방안

- ▶ 앞서 제시한 소프트웨어와 알고리즘의 신뢰성을 판단할 수 있는 카테고리를 활용해 본 연구에서 개발한 시스템의 신뢰성을 확인할 계획임.
- ▶ 본 연구에서 개발한 시스템의 소프트웨어와 알고리즘의 임무 성공률과 소프트웨어 자체의 성공 가능성(possibility of success)이 90% 이상의 정확도를 달성하지 못할 경우, 우리의 시스템을 개선하고 현실 세계 운용을 구현할 계획임.
- ▶ 소프트웨어와 알고리즘의 신뢰성 확보를 위한 실제 드론을 이용한 검증 과정의 3가지의 단계를 설정함.

✓ 1단계 : 10m X 10m의 작은 실내 환경에서 테스트 수행

✓ 2단계 : 100m X 100m의 단순한 실외 환경에서 테스트 수행

✓ 3단계 : 300m X 300m의 넓은 실외 환경에서 테스트 수행

나) 드론 적용을 위한 임베디드 환경에서의 수행방안

- ▶ 다중 에이전트 기반 임무계획 시스템의 현 시스템은 Closed-loop 운용 플랫폼을 사용해 구현됨.
- ▶ Closed-loop 운용은 실제 드론 및 시뮬레이션 내의 드론에서 자율비행 UAV 또는 자율비행 소형 UAV 시스템의 필수 구성요소임.
- ▶ 하단 그림은 상용 Intel Aero RTF MAV 시스템의 3가지 주요 서브시스템을 나타냄.



그림 134 The three main subsystems of UAV

- ▶ 상용 Intel Aero MAV의 세 가지 주요 서브시스템은 (1) 연산 시스템, (2) 액추에이터, (3) 센서를 포함하며, 모든 MAV는 매우 유사한 서브시스템 구조를 가짐.
- ▶ 연산 시스템은 companion 컴퓨터와 비행 제어를 포함하며, 액추에이터는 모터와 프로펠러, 센서는 RGB 카메라로 구성됨.
- ▶ 다음 그림은 현실 세계에서의 UAV의 closed loop 데이터 플로우를 나타냄.



그림 135 Closed-loop data flow in a UAV in real-world

- ▶ 센서를 통해 수집된 환경 데이터는 UAV의 연산 시스템으로 전달되며, 액추에이터로 전달되고, 다시 환경으로 전달되는 정보 흐름을 가짐.
- ▶ 본 프로세스는 환경 데이터를 감지(센서), 데이터를 해석하고 변환해 의사결정을 수행(연산 단계), 환경에서 비행하는 단계(액추에이터)의 반복을 포함함.

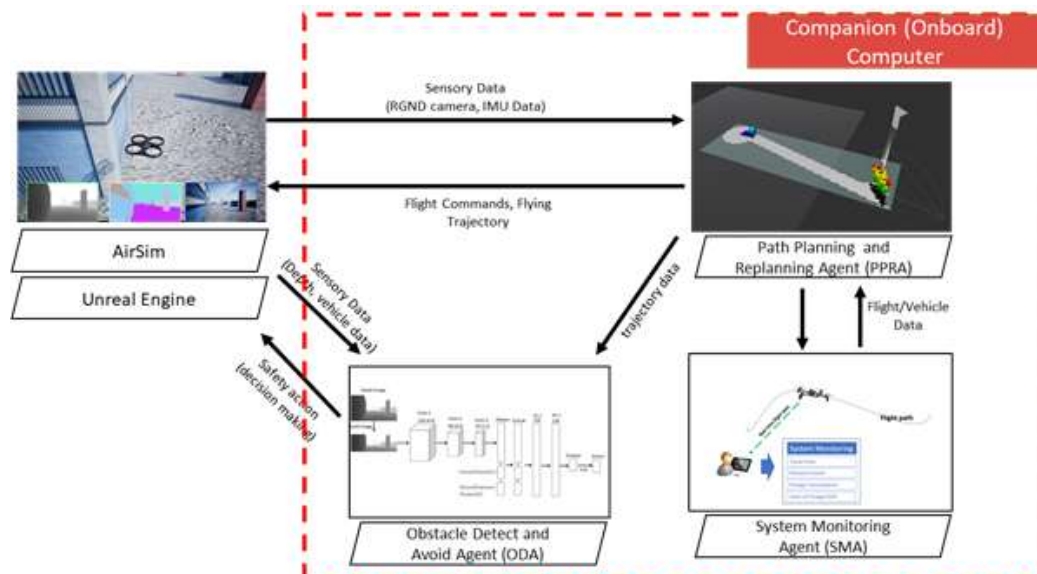


그림 136 Architecture of SITL closed-loop simulation for multi-agent based real-time mission planning

- ▶ 본 연구의 Closed Loop 시뮬레이션 시스템은 실제 드론 운용에도 적용 가능하도록 설계됨.
- ▶ 본 연구에서 개발한 다중 에이전트 시스템은 Companion 컴퓨터와 host 컴퓨터, 2개의 주 컴퓨터에서 수행됨.
- ▶ 실제 드론 구현과정에서 Companion 컴퓨터의 역할을 Onboard 컴퓨터가 대신하며, 알고리즘 검증용으로 사용할 Parrot ANAFI 드론의 Onboard 컴퓨터가 다중 에이전트 시스템을 구현하기에 연산 능력이 충분하지 않아, NVIDIA TX2를 onboard 컴퓨터로 사용할 계획임.
- ▶ Host 컴퓨터는 AirSim을 실행하기 위한 시뮬레이션 모드로 동일하게 사용됨.
- ▶ 실제 드론 구현을 위한 아키텍처는 다음과 같음.

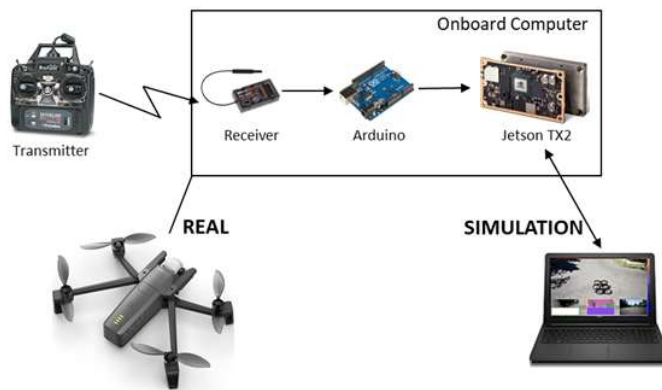


그림 137 Configuration of the multi-agent based mission planning system in the real drone and AirSim simulation

- ▶ 실제 드론을 이용한 시뮬레이션 테스트 과정에서 다중 에이전트 시스템은 NVIDIA Jetson TX2에서 실행되며, 이는 Companion 컴퓨터의 역할과 동일함.
- ▶ 다른 컴퓨터(지상 컴퓨터)는 AirSim 프로그램을 작동하며, 자세를 드론의 Onboard 컴퓨터로 전송함.
- ▶ 수동제어를 위해 송수신기와 아두이노를 사용함.

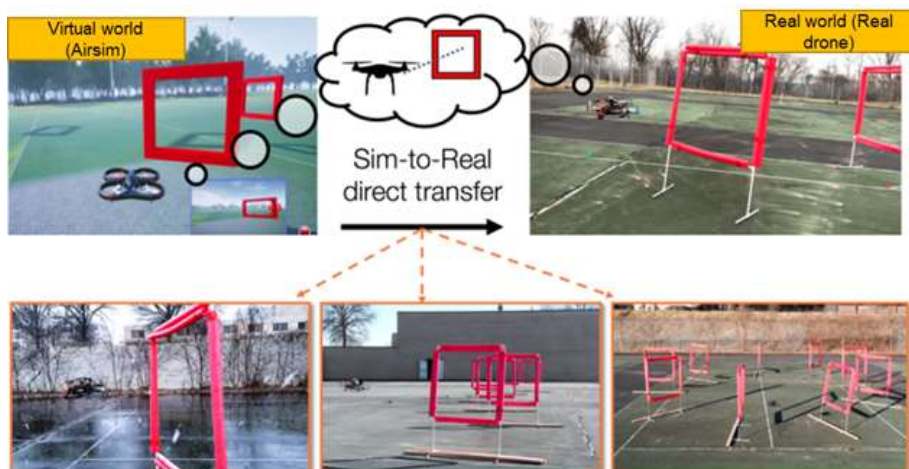


그림 138 One of the examples implementation of the data interaction between real world and virtual world

다) 향후 확장을 위한 필요 연구내용

(1) 세계 소비자 드론 시장

- ▶ 2016년 소비자 드론 시장 규모는 업계 가치 15억 달러 이상인 200만 대 이상으로 추정되며, 2017년부터 2024년까지 CAGR이 약 18% 성장할 것으로 예상됨.

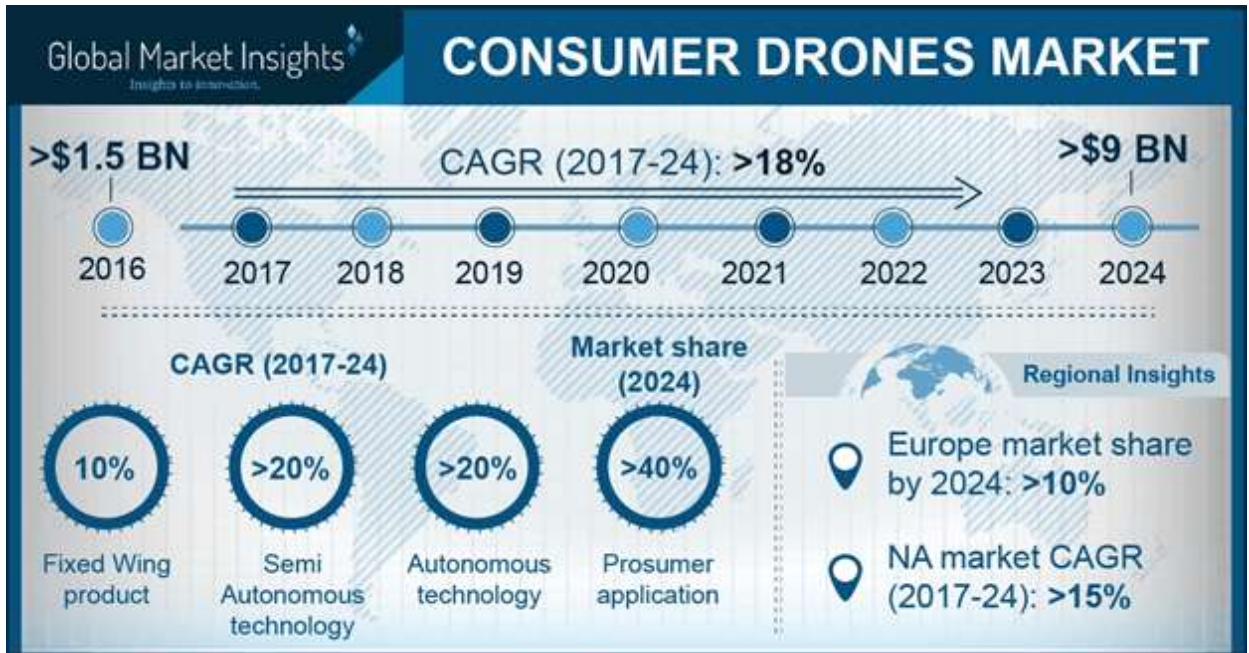


그림 139 Global consumer drones market analysis report

- ▶ 소비자 드론 시장에 따르면, 각 부문별 시장에서 드론의 준·완전 자율비행 기술이 20% 이상을 차지함.
- ▶ 현재 시스템이 개선되고 완전한 자율 드론 시스템으로 확장할 수 있음.
- ▶ 다음 그림은 드론/무인기 자율 수준을 나타냄.

Autonomy Level	0	1	2	3	4	5	6	7	8	9	10
Mission Complexity	Simplest, binary tasks	Subsystem tasks, internal			Limited adaptation, limited real-time planning, tactical level			High adaptation, team collaborative missions/tasks, high real-time planning, strategic level			Highest adaptation, team of team's collaborative missions, fully real-time planning, human level performance
Environmental Difficulty	Static, simple	Static, simple low risks, uncertainty			Understandable dynamic, simple mid risks, uncertainty			Understandable highly dynamic, complex, adversarial high risks, uncertainty			Highest level of dynamics, variation, risks, real-world
Human Independent	Remote control	Human decisions, provides waypoints			Human decisions, provides tactical goals			Informs humans, human provides strategic goals			Perform on its own, zero human interaction

그림 140 Drone/UAV autonomy level

(2) 실제 드론 운용의 구현

- ▶ 실제 시스템에서 검증을 수행한 후, 현실 세계에서 운용에 사용할 수 있도록 시스템을 확장할 계획임.
- ▶ 이를 위해 현 시스템은 반드시 다음 사항을 만족 해야함.
 - ✓ 지상에 다니는 사람들과 드론 자체의 안전성
 - ✓ 시스템의 정확성
- ▶ 안전성을 확보하기 위해 장애물 회피 시스템과 시스템 모니터링 상태를 계속 개선해 나가야 함.

◎ 시스템 모니터링 에이전트

- ▶ 본 연구에서 개발한 시스템 모니터링은 드론의 시스템을 모니터링하고 의사결정을 하는 것을 포함하나, 이는 매우 간단한 의사결정 알고리즘이 사용됨.
- ▶ 추후 시스템 모니터링의 의사결정 알고리즘의 수준을 높여야 함.

◎ 장애물 인식 및 회피 에이전트

- ▶ 장애물 인식 및 회피 시스템의 경우 현실 세계에서 운용할 때의 정확성을 높여야함.
- ▶ 현 시스템은 AirSim 시뮬레이션으로부터 직접 받는 depth 이미지를 사용하나, 센서 데이터 또한 고려할 수 있어야 함.
 - ✓ Depth 이미지를 실제 드론에서 얻는것은 쉽지 않으므로 스테레오 카메라에서 depth 이미지를 얻는 방법에 대한 고려가 필요함.
- ▶ 장애물 인식 및 회피에 스테레오 카메라를 직접 활용할 방법을 고려 해야함.

(3) 하나의 드론에서 다중 드론 협업으로의 확장

- ▶ 본 연구는 하나의 드론에 대해서 수행됨.
- ▶ 더 높은 자율성 수준에 도달하고 현실 세계에서의 드론 운용 요구사항을 만족시키기 위해서 다중 드론 협업을 적용해야함.



그림 141 Multi drone collaboration for the emergency blood delivery operation

(4) 도심 항공 모빌리티(UAM) 운용으로의 확장

- ▶ 대부분의 숙련된 무인 항공 시스템(UAS : Unmanned Aerial Systems)의 운용은 기업 또는 소비자 측면에서 단거리의 정찰, 사진과 영상촬영에 사용됨.
- ▶ 400ft 상공 이상에서 비행할 수 없는 무인 비행체에게 UTM(Unmanned aerial Traffic Management)은 기존 항공분양의 항공 교통 관리 시스템과 동일한 역할을 함.
 - ✓ UTM은 비행 경로를 안내하며 UAS와 빌딩, 다른 드론, 항공기와 같은 장애물의 충돌을 방지함.
 - ✓ 실시간 정보를 제공해 항공 모빌리티가 기상악화지역, 교통혼잡지역, 비행금지구역을 피하도록 함.
- ▶ UTM의 요건은 고도와 위치에 따라 달라질 수 있으며 상대적으로 낮은 고도에서 비행하는 공중 모빌리티를 고려해야함.
- ▶ 도심에서 UTM 시스템은 장애물 탐색을 더 많이 진행하고, 더 복잡한 비행경로를 다룰 수 있도록 프로그램되어야함.
- ▶ UTM 시스템의 주요 목표사항은 다음과 같음.
 - ✓ 매니지먼트의 분리
 - ✓ 경로 계획 및 재계획
 - ✓ 순서와 간격 제시
 - ✓ 동적인 지리상의 경계제시(geofencing)
 - ✓ 지형/장애물 회피
- ▶ 공중 모빌리티를 위한 UTM 시스템의 요건에 맞춰 본 연구에서 진행한 임무계획 시스템을 임무 할당, 임무 분리 관리, 경로 계획 및 재계획, 실시간 장애물 시스템을 UTM 시스템에 접목한 **임무관리 시스템(mission management system)**으로 확장할 계획임.

3. 목표 달성도 및 관련 분야 기여도

3.1. 목표달성도(요약)

구분	성과		계획	실적	달성도 (%)
기술적 성과	특허		3	3	100
	소프트웨어		2	3	150
학술적 성과	국내외 학술지 게재	SCI(E)	4	8	200
		일반학술지	9	6	67
		합계	13	14	108
	국내 외 학술대회 발표	국외	-	3	-
		국내	-	8	-
		합계	-	11	-
기타 성과	인력양성(석 박사 인력 배출)		8	10(5)	120

3.2. 목표 달성도(상세)

구분	핵심 성과물 (CTE)	단위 성과물		성과점검기준								양적성과 현황				
				질적 성과지표		목표치	달성치	측정 방법	검증 방법	가중치	해당 기관	예산 (백만 원)	목표치	달성치	가 중 치	
주관	다중 에이전트 기반 최적 경로 생성 및 임무 배설 계획 인공지능 알고리즘	A-1	다중 에이전트 기반 실시간 임무 재설계	①	실시간 임무 경로 재설계	계산시간 <5s	계산시간 0.00522s	테스트 케이스를 통한 평균 계산 시간 측정	200 번의 테스트 케이스 수행	10	전국대	37	특허 출원 1건 논문(SCI):1건 S/W 등록:1건	특허 출원 1건 논문(SCI):1건 S/W 등록:1건 학술발표 2건	10	
				②	시스템 모니터링 에이전트 기반 임무 재설계	실패율 <2%	실패율 =0%	테스트 케이스를 통한 실패율 계산	200 번의 테스트 케이스 수행	10	전국대	37	특허 출원 1건 (등록 1건) 논문(일반):1건	특허 출원 1건 특허 등록 0건 논문(일반):1건	20	
				③	강건성	실패율 <2%	실패율 =1%	테스트 케이스를 통한 실패율 계산	200 번의 테스트 케이스 수행	10	전국대	37	논문(일반) 2건	논문(SCI) 1건 논문(일반) 1건	10	
		A-2	기 최적 경로 생성	①	초기 최적 경로 생성	계산시간 <120s	계산시간 0.004058s	테스트 케이스를 통한 평균 계산 시간 측정	200 번의 테스트 케이스 수행	10	전국대	37			10	
				②	강건성	실패율 <2%	실패율 =0%	테스트 케이스를 통한 실패율 계산	200 번의 테스트 케이스 수행	10	전국대	37	논문(SCI) 1건 논문(일반) 1건	논문(SCI) 1건 논문(일반) 1건 학술회의 4건	10	
		A-3	물레이션	①	시뮬레이션 상 무인기 구현	실패율 <1%	실패율 =0%	테스트 케이스를 통한 실패율 계산	100 번의 테스트 케이스 수행	20	전국대	74	논문(일반)3건	논문(SCI) 2건 논문(일반) 2건 S/W 등록 1건 학술회의 4건	10	
	B	장애물 인식 및 회피 인공지능 알고리즘	B-1	QN 기반 장애물 인식 및 회피	①	실시간 장애물 회피	반응속도 <1/30s	반응속도 <0.0042s	테스트 케이스를 통한 평균 반응속도 측정	200 번의 테스트 케이스 수행	10	전국대	37			10

구분	핵심 성과물 (CTE)	단위 성과물	성과점검기준									양적성과 현황		
			질적 성과지표	목표치	달성치	측정 방법	검증 방법	가중치	해당 기관	예산 (백만 원)	목표치	달성치	가 중 치	
			② 딥 러닝 네트워크 정밀도	RMSE <0.0001	RMSE <0.00003	인공지능 네트워크 정확도 측정	전체 데이터의 25%를 이용한 테스트 수행	10	건국대	37	특허출원 1건 논문(SCI) 2건 논문(일반) 2건 S/W등록1건	특허출원 1건 논문(SCI) 2건 논문(일반) 2건 S/W등록1건 학술회의 1건	10	
			③ 강건성	실패율 <2%	실패율 <0.5%	테스트 케이스를 통한 실패율 계산	200 번의 테스트 케이스 수행	10	건국대	37			10	
계								100		370	특허출원 3건 특허등록 1건 논문(SCI) 4건 논문(일반) 9건 S/W등록 2건		100	

1) 연구 성과 공유 활동

가) 미래항공교통 UAM 시스템 운용 국제 특별 세션

- ▶ 건국대학교의 후원으로 본 연구소와 사단법인 한국항공우주학회는 ‘미래항공교통 : UAM 시스템 운용’을 주제로 국제 특별 세션(International Session)을 공동으로 구성하였음.
- ▶ 본 세션은 2020년 한국항공우주학회 추계 학술대회의 일환으로 2020년 11월 19일 - 20일의 양일간 **인공지능 자율 비행 기술을 포함**하는 미래형 도심 항공 시스템인 UAM의 설계, 해석, 운용 및 이를 위한 Digital Twin 등 UAM관련 제반기술에 관련한 주제를 폭넓게 다루었으며 국내외 관련 대학 및 기관에서 24편의 논문이 발표되어 성황리에 마무리 되었음

나) Online Overseas Study Faculty of Defense Technology

- ▶ 건국대학교의 후원으로 본 연구소와 인도네시아의 Indonesia Defense University 과 공동으로 ‘Online Overseas Study Faculty of Defense Technology’에 관해 International Online Seminar를 진행하였음.
- ▶ 본 세미나는 2020년 8월 24일에서 26일까지 3일간 진행되었으며 한국과 인도네시아에서 Defense Technology 관련 본 대학중점연구소의 연구소장 이재우 교수를 포함하는 5명의 전문가가 오전과 오후의 각 세션을 맡아 진행하였으며 발표 및 온라인 토론을 통해 국제 기술 교류 및 협력을 위한 다방면의 논의가 이루어졌음.



그림 142 성황리에 진행 된 미래항공교통 UAM 시스템 운용 국제 특별 세션

미래항공교통: UAM 시스템운용 국제특별세션

주최: 한국항공우주학회
주관: 항공우주 설계·인증 대학중점연구소 KADA
후원: 건국대학교 KU

제 9 발표장 (한라홀)

2020년 11월 19일 목요일 10:30-12:10	
TB9 OS : 미래항공교통 - UAM 시스템 운용 I 좌장 : 이금진(한국항공대학교)	
인사말	
이재우(건국대학교 항공우주 설계·인증 대학중점연구소장)	
Bringing aviation closer to home: Our shared story of Urban Air Mobility	Invited
David Rettblatt(EmbraerX)	
Launching UAM Operations: Learnings from Singapore	Invited
Hon Lung Chu(Volocopter)	
도심항공교통(UAM) 운용 이니셔티브 청사진에 관한 연구	시스템/개념
이영재, 민재선, 강병열, 권태호, 안재현, 정봉철, 이재우(건국대)	
UAM 운항서비스사업자의 지속적 수익확보를 위한 전략적 접근에 대한 연구	시스템/개념
이광병, 김성욱(한화시스템)	
수소연료전지 도심항공교통수단의 성능운용분석 연구	UAM비행체
안재현, Zin Win Thu, 이재문, 이종재, 민재선, 황막심(건국대), 나송혁(중원대), 이재우(건국대)	
효율적 도심항공이동(UAM) 서비스 구현을 위한 이착륙시설 운영모델	인프라/보안
정민철, 유금식(KAC), 이광병(한화시스템)	

2020년 11월 19일 목요일 15:00-16:40	
TC9 OS : 미래항공교통 - UAM 시스템 운용II 좌장 : 김상호(건국대학교)	
도심항공교통 시스템 운용을 위한 실시간 파라미터 추정 기반의 경로 추종 적응 제어 기법 연구	UAM비행체
정우석, 조문경, 우지원, 이상한, 김창주(건국대)	
4단계 교통수요모형 활용 도심항공모빌리티 시장잠재성 분석	민적요소
박성용, 김명현, 박선욱(KOTI)	
미래 도심항공교통을 위한 디지털트윈 기술 개요	디지털트윈
정막성, Aye Aye Maw, 진권수, 장미나, 이재우(건국대)	
하이브리드 운송 네트워크 설계를 위한 분산화 태백 임무 할당	시스템/개념
Joshua Julian Damanik, 최한림(KAIST)	
UAM 사용자 보안을 위한 실시간 부정 감성 감시 및 추적 기술	인프라/보안
이광병(한화시스템), 황민형(상명대)	
인간-이동체 인터페이스 시스템 개발 개요	민적요소
최다희, 윤준성, 김상호(건국대)	

그림 143 미래항공교통 UAM 시스템 운용국제 특별세션 프로그램

2) 양적성과 대비 미충족 항목에 대한 보완 계획

가) 논문 게재(SCI 및 일반 학술지)

- ▶ 본 연구를 통해 SCI 4편, 일반 학술지 9편으로 총 13편의 게재를 목표로 하였으며 이에 비해 SCI 국제 학술지 논문 8편, 일반 학술지는 6편으로 총 14편의 국내외 학술지 게재 실적을 달성하였음.
- ▶ 비록 일반학술지의 실적이 9건 목표에 6건의 실적으로 목표대비 미달하고 있으나 일반적으로 SCI 논문의 게재에 더 많은 노력과 시간이 소요되며 그만큼 일반 학술지에 대한 게재보다 높은 가치를 인정받음을 고려하면 이 차이는 충분히 상쇄될 수 있을 것으로 사료됨.
- ▶ 일부 심사 중인 논문에 대해서는 게재가 확정 되는대로 지속적으로 그 정보를 본 연구의 실적결과로 업데이트 할 예정임.

나) 특허(출원 및 등록)

- ▶ 본 연구를 통해 특허 출원 3건과 등록 1건을 목표로 하였으며 현재 출원 3건은 완료된 상태임.
- ▶ 등록 건에 대해서는 현재 출원 완료된 3건 중 핵심이 되는 iADA* 기술 특허에 관해 금년 상반기에 등록을 위한 심사를 신청하여 등록을 완료하려 하며 그 결과를 추후 확인하여 업데이트 할 예정.

3) 연구 개발 성과의 차별성

- ▶ 본 연구를 통해 개발된 다중 인공지능 에이전트 기반 실시간 경로 재계획 기술은 기존 유사 연구 내용과 비교하여 다음과 같은 특징을 가지고 있음.

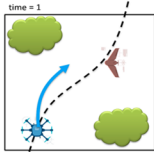
	기존 유사 연구 결과	본 연구의 개발 기술
적용 운행체	지상 운행로봇/자율주행 자동차(2차원 이동) 	무인비행체(3차원 운항)  
경로 최적화	경로 최적화에 대한 고려 없음	경로 최적화 기능 포함
실시간 경로 재계획	실시간 경로 재계획에 대한 고려 없음	실시간 경로 재계획 가능
DQN 학습을 위한 계산시간	빠름(1일 미만)/느림(4일 이상)	중간(1- 4일 사이) 
정확도	동적 환경에서 20개의 테스트 중 85%에 상당	정적/동적 환경에서 200회의 테스트 중 98 - 100% 정확도
동적 환경	단순(동적 장애물 4개 이하)	복잡(동적 장애물 10개 이상)
실제 적용	지상 운행체에 대해 적용 중	본 연구의 결과를 바탕으로 실제 적용 될 수 있도록 확장

그림 144 기존 연구결과와 본 연구 개발 기술과의 차별성

- ▶ 기존의 연구에서는 주로 지상 주행 로봇이나 차량에 대해 상대적으로 단순하고 좁은 환경 맵을 기반으로 테스트 혹은 시뮬레이션이 수행되었음.
- ▶ 이에 비해 본 연구의 경우, 크기가 300m × 300m의 대규모 환경에서 초기 최적 경로를 수립 가능하며 맵 상에 다수의 미지의 정적 및 동적 장애물이 존재하는 조건에서 테스트를 수행하

였음.

- ▶ 이는 기존의 연구에서 크기 수m에서 수십m 정도의 실내에서 장애물을 고려하지 않은 초기 경로 계획이나 소수의 정적 혹은 동적 장애물만을 고려하는 것에 비해 본 연구를 통해 개발한 기술을 활용하면 **맵의 크기(수백 m 크기의 맵)와 장애물의 조건(다수의 정적/동적 장애물 존재)에 대해 훨씬 복잡한 임무를 수행하는 것이 가능**하다는 것을 확인할 수 있음.
- ▶ 다만 이러한 결과는 컴퓨터 시뮬레이션 상에서 확인한 결과로 향후 **실제 환경에서 본 연구의 결과를 실제 드론에 적용**하여 시뮬레이션 상에서 확인한 **경로 계획 및 실시간 경로 재계획 기술이 실제 비행에서도 잘 적용 될 수 있음을 확인**하려 함.
- ▶ 이에 대해서는 앞서 2장의 향후 연구 방안 중 ‘**드론 적용을 위한 임베디드 환경에서의 알고리즘 수행**’에서 제시한 방안을 기반으로 추후 확장 연구 내용을 참조할 수 있을 것으로 사료됨.

3.3. 관련 분야 기여도

◎ 도심형 항공운행 체계, UAM(Urban Air Mobility) 시대의 도래

- ▶ 최근 UAM은 도시교통과 환경 문제를 근본적으로 해결할 수 있는 수단으로 저고도 공역을 이용한 도시항공운송 수단으로 부각되고 있음
- ▶ 이러한 UAM의 핵심 기술의 바탕에는 드론으로 대표되는 무인비행체의 기술이 자리잡고 있으며 이를 통한 관련기술의 성숙화와 대형 서비스 공급자의 계획발표에 따른 과거 PAV(Personal Air Vehicle)와 달리 급부상한 “친환경 저소음 전기 동력 분산추진 수직이착륙 개인항공기(eVTOL)”을 활용하여 도심용 공중 모빌리티(UAM, Urban Air Mobility)로 활용하는 항공 산업에 대한 신 시장 태동 중임
- ▶ UAM 체계의 핵심인 PAV 제조와 운송 서비스를 아우르는 전체 UAM의 잠재적 시장규모는 2040년 1조 5,000억 달러에 달할 것으로 전망(2020. 삼정 KPMG)



그림 145 스마트 UAM의 현황 및 추세

- ▶ 여러 나라의 글로벌 기업들이 시장 선점을 위해 치열한 경쟁을 벌이고 있으며 이러한 움직임에는 한국의 현대자동차를 비롯한 여러 기업, 지자체, 기관들도 관심을 가지고 뛰어 들고 있는 상황임.

국토교통부는 스마트 UAM의 핵심기술인 스마트 모빌리티, 디지털 트윈, 완전 자율 주행 기술 등을 2030년까지 집중 육성을 목표로 하는 '**20대 유망 기술**'로 선정

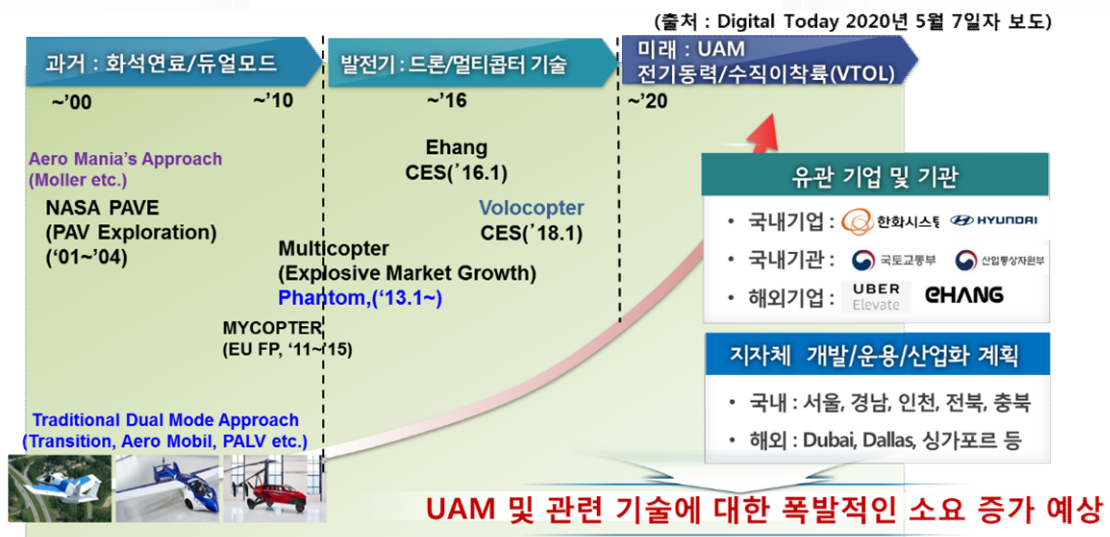


그림 146 UAM 관련 기술의 발전 동향 및 전망

- ▶ 상기의 그림에서도 볼 수 있듯이 이러한 UAM의 핵심을 이루는 기술로는 스마트 모빌리티, 디지털 트윈 및 **완전 자율 주행 기술**이 있으며 국토교통부에서는 2020년 5월에 인공지능 기

반 자율 주행 기술을 포함하는 이들 핵심 기술을 **2030년까지 국가적인 집중 육성을 목표로 하는 '20대 유망 기술'로 선정**한 바 있음

- ▶ 분산 전기추진(저소음, 무공해), 전기모터/배터리, 자동자율비행(AI), IoT 등 관련기술(4차 산업혁명 공통기술)의 성숙 및 대형 개인운송 서비스 공급자((미)Uber, '16년 매출 7.3조원)의 Air Taxi 비행체 요구조건 발표 및 사업화 계획 발표('16.10)로 세계적으로 개발 열풍 촉발되고 있음.

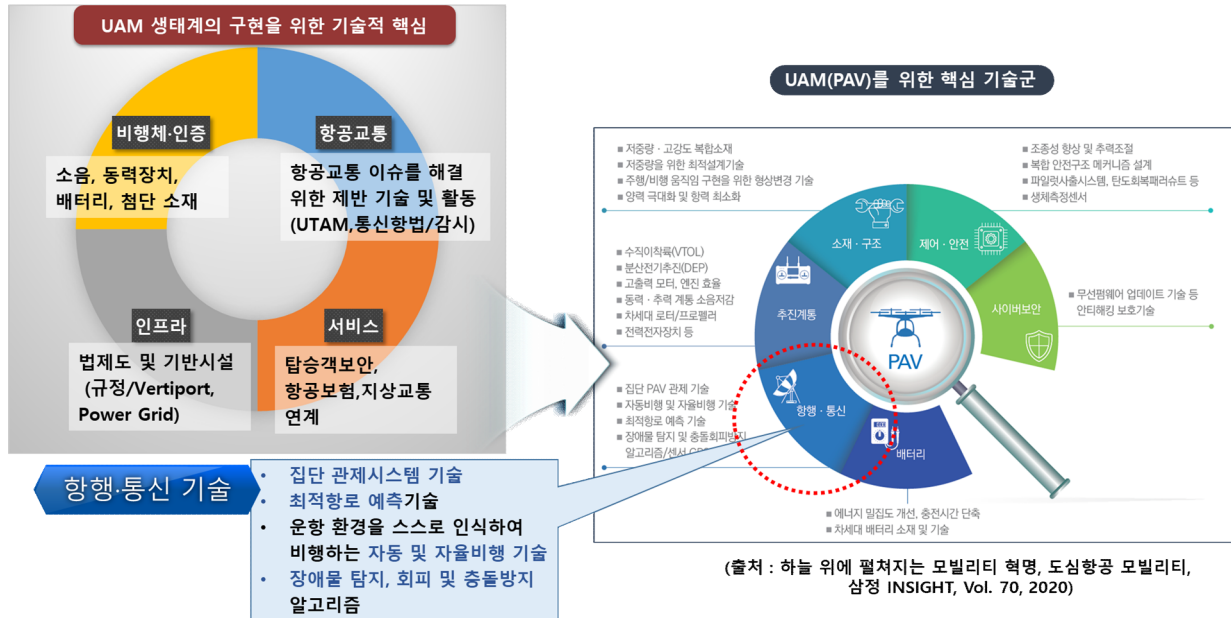


그림 147 UAM을 위한 핵심 기술 분야

- ▶ 상기의 그림에서는 UAM을 구성하는 핵심 기술분야를 보여주고 있음.
- ▶ 이 중에서도 본 연구의 연구 내용인 **최적 항로 예측 기술**, 운항 환경을 스스로 인식하여 비행을 가능하게 하는 **자동 및 자율 비행 기술**, 그리고 **장애물 탐지, 회피 및 충돌 방지 기술**이 **UAM의 구현을 위한 핵심 기술로 포함**되어 있음을 확인할 수 있음,

◎ 인공지능 자율 운항 기술의 무인이동체 및 UAM 분야에 응용

- ▶ 본 연구를 통해 개발된 높은 수준의 자율 지능 무인비행체 기술을 기업에 이전하여 지능형 무인기 개발에 활용 가능.
- ▶ 획득된 지능형 의사결정 알고리즘, 지능 제어계 및 지능 임무계획 원천기술은 민,군용 드론/자율주행 자동차/로봇 등 전반적인 무인 이동체 분야에도 확장하여 적용 가능.
- ▶ 또한 미래의 도심 항공 운항체계에는 드론과 같은 무인비행체와 UAM으로 대표되는 유인비행체의 운항을 함께 고려해야 하는데 다수의 유무인 비행체가 동일한 공역에서 비행하게 되므로 이 모든 것을 기존의 항공운항 시스템과 같이 관제사가 모두 제어하는 것은 실제로 불가능.
- ▶ 따라서 드론과 UAM에 공히 실시간 임무 계획, 정적/동적 장애물 회피 기술, 시스템 모니터링 기술과 같은 높은 수준의 자율 비행 능력을 가져야 하므로 본 연구를 통해 개발된 기술을 UAM의 확장 적용도 가능하여 이 분야에도 기여할 수 있을 것으로 보임.

◎ 스마트 UAM을 위한 디지털 트윈(Digital Twin) 기술 적용

- ▶ UAM은 자율 비행 기술의 단계적 성숙도에 따라 조종사의 탑승이 결정 되며 에어택시와 같은 On-Demand 운용과 다량의 UAM을 운용하기 위해서는 신뢰성 높은 자율비행 기술이 필수임.
- ▶ 특히 UAM은 기존 헬리콥터와 같이 저고도에서 운항되기 때문에 실시간으로 수많은 안전 시나리오(Safety Scenario)와 효율적으로 운항 가능한 시나리오 고려해야 함.
- ▶ 이를 위해서 인공지능 기술을 활용하여 많은 비행 시나리오를 시뮬레이션 하고 최적의 비행 시나리오를 계산하는 디지털 트윈 (Digital Twin) 기술이 필수적.
- ▶ 특히 디지털 트윈 기술은 **본 연구를 통해 개발된 인공지능 기술을 기반으로 스마트 UAM을 안전하고 효율으로 운용하기 위해 전략과 안전의 측면을 모두 고려하는 방법**이며 범 미래의 수많은 UAM 운항에 필요한 근본적인 기술이라 할 수 있으므로 이 분야로의 연구 결과의 확장도 가능함.

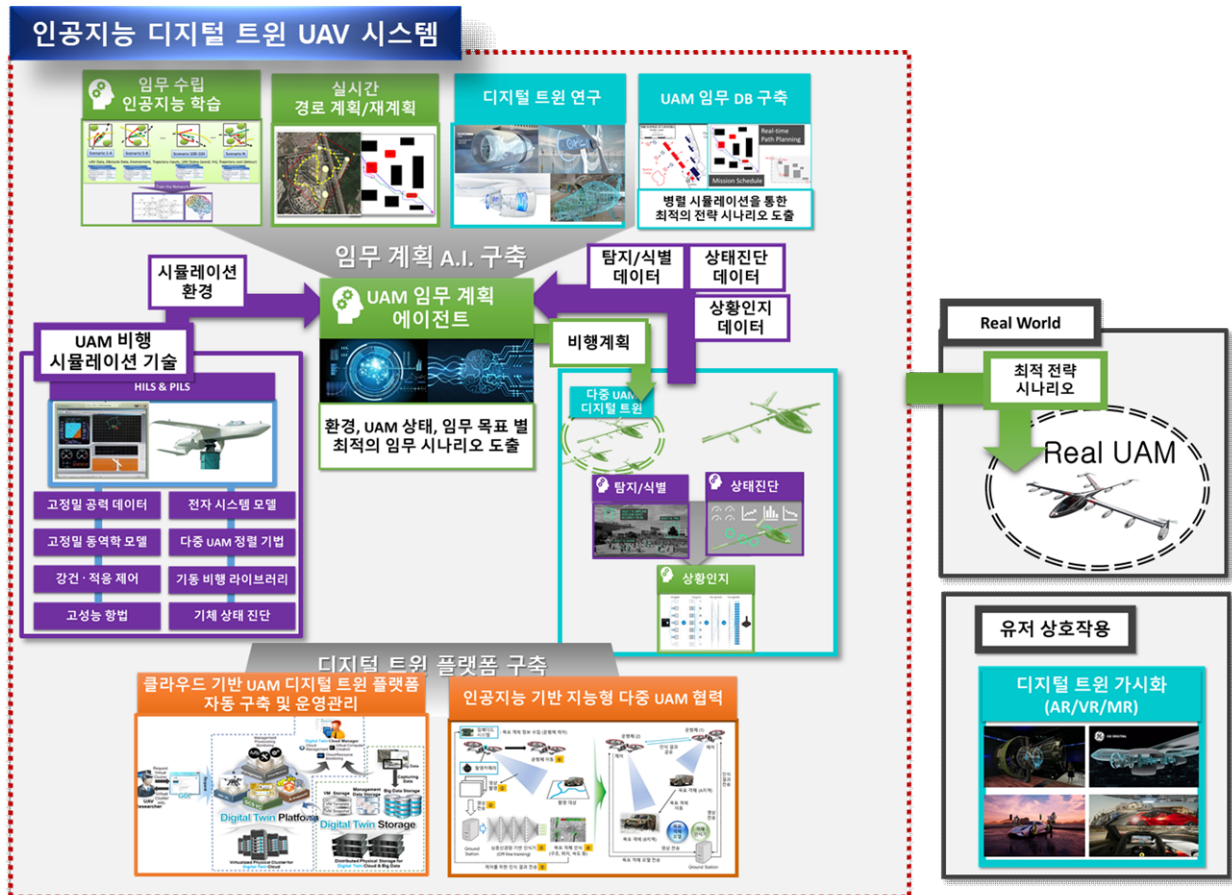


그림 148 인공지능 기반 자율비행 시스템을 포함하는 UAM의 디지털 트윈 개념

4. 연구개발성과의 활용 계획

가. 연구결과의 활용방안

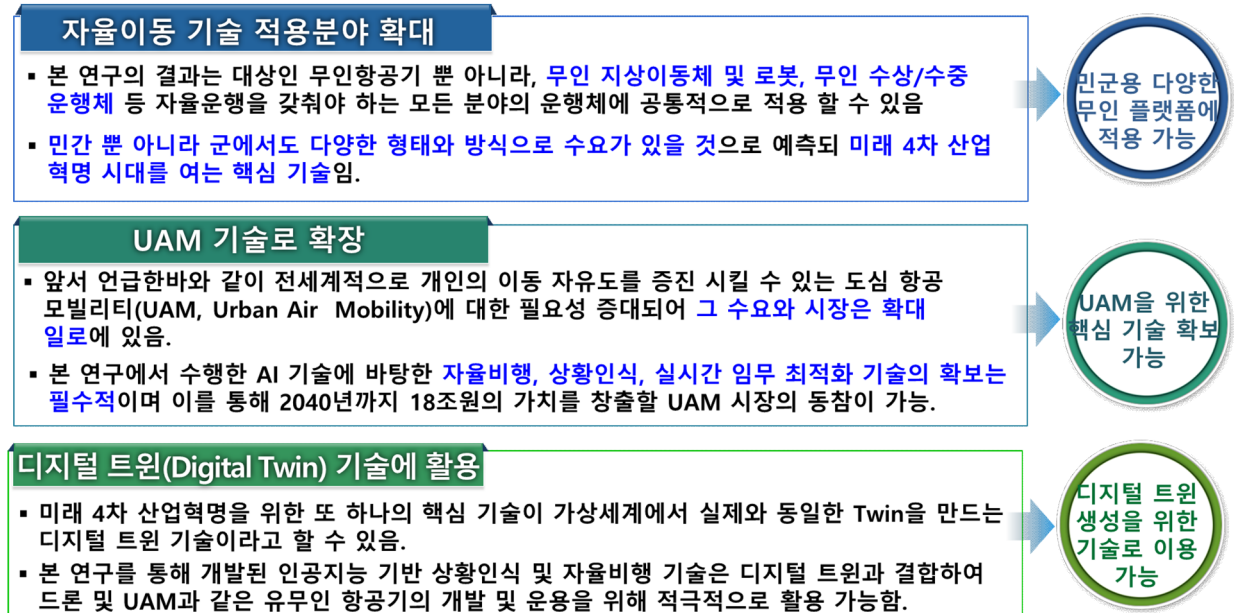


그림 149 연구성과의 활용 방안

1) 무인기 분야 및 자율이동 기술 적용 분야의 확대

- ▶ 높은 수준의 자율 지능 무인기 기술을 기업에 이전하여 지능형 무인기 개발에 활용
- ▶ 지능형 의사결정 알고리즘, 지능 제어계 및 지능임무계획 원천기술을 획득하여 민·군용 드론/자율주행 자동차/로봇 등 첨단산업 분야에 적용
- ▶ 미래 지능형 무인기 기술을 국내에서 개발함으로써 급증하는 무인기 시장을 선점하며, 향후 항공 선진국들과의 협력 개발에 주도적으로 참여
- ▶ **민간 뿐 아니라 군에서도 다양한 형태와 방식으로 수요**가 있을 것으로 예측되며 **미래 4차 산업 혁명 시대를 여는 핵심 기술임.**

가) 민간 분야 무인기 활용 방안 (Civilization)

- ▶ 국내 민간 분야의 무인기의 수요는 2016년 약 700백억 규모에서 2025년 까지 8,700억 규모로 성장이 예상되며 무인기 수요 또한 2025년 까지 56,000여대로 급속히 증가가 예상됨.
- ▶ 그러나 현재 무인기의 운용 방식은 2D Map 기반의 조종사가 수동으로 경로를 설정하는 방식으로 비행에 필요한 다양한 요소를 고려하지 못하고 있음.
- ▶ 본 연구를 통해 **초자율 무인기를 위한 다중 인공지능 에이전트 기반 실시간 임무계획 알고리즘 기술 개발을 통해 기존 무인기의 한계를 극복하고 성능을 증가시키며 무인기가 다양한 산업 분야에 활용**을 기대.

(1) 산악 지형에서의 활용

- ▶ 국내 국토의 면적중 약 70%는 산악지형으로 구성되어 있어 저고도로 운항하는 무인 비행체가 비행하기 상대적으로 어려움.

- ▶ 특히 현재 무인 항공기의 하드웨어와 2D 지도 기반 소프트웨어 한계로 ‘최적 경로 탐색’ 및 ‘장애물 인지 및 회피’의 기술이 부족하여 가시권 내 비행 (VLOS : Visual Line Of Sight) 만 이루어지고 있음 상태이다.
- ▶ 본 다중 인공지능 에이전트 기반 실시간 임무계획 알고리즘 기술 개발을 통해 다양한 무인 비행체가 장애물이 많은 산악 지형에서 최적 경로를 통해 가시권 밖 비행 (BLOS : Beyond Line Of Sight) 이 가능함.

◎ 산림 병해충 탐지

- ▶ 산림청은 정부3.0 맞춤형 기술 개선으로 드론을 활용 ‘소나무 재선충병 피해목 조사’ 사업을 진행.
- ▶ 그 결과 기존 인력 예찰에 비해 조사기간은 약 90% 단축하고, 1인당 조사 가능 면적은 10배로 증가. (8,000 Ha 예찰에 인력은 200일, 무인기는 20일 소요)
- ▶ 산림청은 이러한 성과를 바탕으로 기존 4,000ha에서 최대 100,000 ha의 면적을 드론으로 탐지 하려는 계획을 가지고 있음.
- ▶ 본 다중 인공지능 에이전트 기반 실시간 임무계획 알고리즘 기술을 활용할시 더욱 많은 산악 지형의 면적을 더욱 효율적이고 안정적으로 탐지 가능.



그림 150 무인기 소나무재선충 탐지



그림 151 무인기 소나무재선충 탐지 II

◎ 산사태 발생지 위치 및 피해면적 분석

- ▶ 산림재해 분야의 시설물 관리는 급경사지와 낙석이 우려 되어 사람이 직접 올라가 점검하는 방식은 비용과 시간이 많이 소요되는 등 효율성이 떨어지고 추락 등 안전사고의 위험도 매우 높음.
- ▶ 산림청은 2016년도 울진 석회광산에 드론을 투입 이후 3D Mapping 진행 후 지반침하지역 지형 변화를 분석 진행하였음.



그림 152. 무인기를 활용한 산사태 발생지 위치 및 피해 변화 분석

● 산간지역 배송

- ▶ 우정사업본부에서는 도서, 산간 등 물류 사각지대에 무인기를 활용하여 물건을 배달하는 고품질의 대국민 보편적 우편 서비스 제공을 계획.
- ▶ 영월지역의 경우에는 무인기를 통해 산간지역을 직선으로 비행하여 기존 대비 배달 소요시간을 80.76% 개선하였다고 발표, 이는 무인기를 활용한 배송 시스템의 실효성을 단편적으로 보여줌.
- ▶ 실효성을 바탕으로 여러 산간지역의 배송 시 다양한 환경요소를 고려가 필요하며 산간지역 장애물과 같은 다양한 환경적 돌발상황에 대한 초자율 무인기의 필요성 대두됨.



그림 153 무인기 산악지형 물건 배송 I



그림 154 무인기 산악지형 물건 배송 II

(2) 해양수산분야 분야에서의 활용

- ▶ 국내 해양수산분야의 다양한 관측과 조사 중 상당수는 선박을 통해 행해지고 있고, 일부가 유인항공기가 행해지고 있음.
- ▶ 그러나 산악지형에서 활용과 마찬가지로 운용 비행 거리의 한계로 가시권 내 운용(VLOS)만 이루어지고 있으며 강풍, 조류와 같은 무인기에 영향을 주는 환경요소로 인해 제한적인 운용만 이루어지고 있음.
- ▶ 본 다중 인공지능 에이전트 기반 실시간 임무계획 알고리즘 기술 개발을 통해 실시간으로 환경요소가 급격히 변하는 해양 지역에서 실시간 무인기 모니터링 및 임무 재 계획 기능을 통해 가시권 밖 비행 (BLOS : Beyond Line Of Sight)이 가능함.

나) UAM(Urban Air Mobility) 기술로 확장 적용

- ▶ 미래의 도심 항공 운항체계에는 드론과 같은 무인비행체와 UAM으로 대표되는 유인비행체의 운항을 함께 고려해야 하는데 다수의 유무인 비행체가 동일한 공역에서 비행하게 되므로 이 모든 것을 기존의 항공운항 시스템과 같이 관제사가 모두 제어하는 것은 실제적으로 불가능.
- ▶ 따라서 드론과 UAM에 공히 실시간 임무 계획, 정적/동적 장애물 회피 기술, 시스템 모니터링 기술과 같은 높은 수준의 자율 비행 능력을 가져야 하므로 본 연구를 통해 개발된 기술을 UAM의 확장 적용도 가능하여 이 분야에도 기여할 수 있을 것으로 보임.

다) 스마트 UAM을 위한 디지털 트윈(Digital Twin) 기술에도 적용

- ▶ UAM은 자율 비행 기술의 단계적 성숙도에 따라 조종사의 탑승이 결정 되며 에어택시와 같은 On-Demand 운용과 다량의 UAM을 운용하기 위해서는 신뢰성 높은 자율비행 기술이 필수임.

- ▶ 특히 UAM은 기존 헬리콥터와 같이 저고도에서 운항되기 때문에 실시간으로 수많은 안전 시나리오(Safety Scenario)와 효율적으로 운항 가능한 시나리오 고려해야 함.
- ▶ 이를 위해서 인공지능 기술을 활용하여 많은 비행 시나리오를 시뮬레이션 하고 최적의 비행 시나리오를 계산하는 디지털 트윈 (Digital Twin) 기술이 필수적.
- ▶ 특히 디지털 트윈 기술은 **본 연구를 통해 개발된 인공지능 기술을 기반으로 스마트 UAM을 안전하고 효율로 운용하기 위해 전략과 안전의 측면을 모두 고려하는 방법**이며 범 미래의 수많은 UAM 운항에 필요한 근본적인 기술이라 할 수 있으므로 이 분야로의 연구 결과의 확장도 가능함.

라) 활용 분야 예

- ▶ 다음은 본 연구 결과를 UAM 및 디지털 트윈 시스템에 활용을 위한 몇가지 예를 그림으로 소개한 것임.

1 UAM 용 디지털 트윈 시스템 개발

- 현실에 존재하는 UAM을 가상 세계에 똑같이 구현하는 디지털 트윈(Digital Twin) 시스템의 개발을 목표



내용

- UAM 자율비행을 위한 AI 기술적용 경로 계획 및 생성
- UAM을 위한 고정밀도 시뮬레이션 및 DB 개발
- 가상환경 내 지형지물, 구역정보, 외부 환경 등의 가시화
- 실시간 동적 환경에서의 다중 UAM을 위한 디지털 트윈 시스템 구축

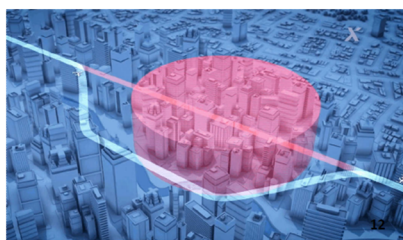
본 연구결과의 활용

- UAM용 **실시간 경로계획 및 정적/동적 장애물 회피 AI agent** 개발
- 비행시뮬레이션용 고정밀도 비행 동역학 모델 및 데이터 융합을 이용한 DB 구축 결과와 접목
- UAM 시뮬레이션에 필요한 **시스템 모니터링 및 환경정보 분석**에 활용

그림 155 활용분야 예 1: UAM용 디지털 트윈 시스템 개발

2 UAM을 위한 AI 기반 자율 비행 에이전트 개발

- UAM은 다수의 탑승객과 함께 복잡한 도심에서 지상이 아닌 공중을 운항해야 하므로 극도의 안전성이 담보되어야 실용화가 가능하므로 이를 위한 자율 비행 에이전트의 개발을 목표



내용

- UAM의 안전한 자율비행을 위한 최적 경로의 계획, 장애물 탐지 및 회피와 상황인식, 그리고 기계 학습을 통한 자율비행 능력 향상 기술을 위한 AI 에이전트의 개발

본 연구결과의 활용

- iADA* 기반 장애물 인식 및 회피 AI 알고리즘의 적용
- 이미지를 통해 외부 상황인식 가능 CNN 기반 인공지능 기술의 적용
- 자율 비행 능력을 높이기 위해 강화학습 기법인 DQN 이용 동적 환경에 대한 UAM 최적 경로 재계획 기법의 적용

그림 156 활용분야 예 2: UAM용 디지털 트윈 시스템 개발

나. 기대효과

1) 학문적 & 인력양성 기대효과

- ▶ 다수의 석·박사과정 학생들이 연구에 참여하여 **향후의 지능형 무인기 전문 인력 양성에 기여**함.
- ▶ 본 연구를 통하여 양성된 **전문 인력들의 세계적인 연구 성과 창출**을 통하여 지능형 무인기 및 **본 기술을 활용 가능한 관련 신산업을 창출**함
- ▶ 지능형 무인기 분야 국내 연구를 선도할 수 있도록 **학문적 연구 토대 마련**
- ▶ 학계와 산업체의 협력으로 시너지 효과를 창출하여 향후 **지능형 무인기 분야를 선도하는 전문 연구 네트워크 구축**
- ▶ 협업이 중요한 향후 지능형 무인기 연구 네트워크 자원 양성 및 **국제협력 시 질적 제고** 가능

2) 무인기 성능 개선 기대효과

- ▶ 기존 무인기의 가시권 내 운용(VLOS)의 한계를 극복하여 안전한 거리에서 **가시권 밖 운용 운항 방식(BLOS) 적용**할 수 있음.
- ▶ BLOS 운항 방식을 통해 환경적 요소에 의한 **인력 접근 및 탐지가 어려운 환경을 극복하여 구체적인 현장정보 취득**가능
- ▶ AI 학습모델을 적용하여 다양한 상황의 한계를 극복하여 기존 무인기의 운용 능력의 한계의 초월 가능.

3) 산업영역 확대 기대효과

- ▶ 재난용 지능형 무인기를 융합 활용함으로써 재해의 감시 및 탐지, 대응능력 향상 가능
- ▶ 운용능력의 확장은 기존의 무인항공기에서 가능한 분야 이외의 활용이 가능하여 새로운 산업 영역의 확장을 가능하게 함
- ▶ 인력으로 수행하는 기존 현장조사에 드론을 활용하여 좀더 신속하고 정확하게 조사를 수행할 수 있어 업무의 효율성 및 경제성 증대 (경제적 효과 약 8.5배)

붙임. 참고문헌

1. UTM-directory.-June-2018.-v1
2. <https://utm.arc.nasa.gov/index.shtml>
3. Farid Kendoul, "Towards a Unified Framework for UAS Autonomy and Technology Readiness Assessment (ATRA)", Autonomous Control Systems and Vehicles ,pp 55-71
4. David Held, Sebastian Thrun, Silvio Sarvarese, " Learning to Track at 100 FPS with Deep Regression Networks", European Conference on Computer Vision, 2016
5. Dipanjan Sakar, " A Comprehensive Hands-on Guide to Transfer Learning with Real-World Application in Deep Learning", 2018
6. Jeremy Kahn, " Google DeepMind Makes AI Training Platform Publicly Available" Bloomberg, 2016.
7. Ho Song Yan, " Reinforcement learning for Self-Driving Cars", Master Thesis
8. Rohith Gandhi , " Build Your Own Convolution Neural Network in 5 mins", 2018
9. Smruti Amarjyoti, "Deep Reinforcement Learning for Robotic Manipulation-The state of the art", Thesis , 2017
10. 과학기술정보통신부, 미래형 자율비행 개인항공기(OPPAV) 안전운항체계 개발 및 인프라 구축」공동기획연구(보완)
11. 정부3.0 보도자료,정부3.0 드론 업업, 산림재해 대응력 높이 날다!, 산림청
12. "Vehicle Tracking System," *Best Web Designing Digital Marketing Company in Chandigarh Mohali*. <https://www.thinknext.co.in/vehicle-tracking-system/> (accessed Jan. 31, 2021).
13. C.-M. Tseng, C.-K. Chau, K. Elbassioni, and M. Khonji, "Flight Tour Planning with Recharging Optimization for Battery-operated Autonomous Drones," Mar. 2017, Accessed: Jan. 31, 2021. [Online]. Available: <https://arxiv.org/abs/1703.10049v1>
14. "bd_anafi_productsheet_en_210x297_2018-06-04.pdf." Accessed: Jan. 31, 2021. [Online]. Available: https://www.parrot.com/assets/s3fs-public/2020-07/bd_anafi_productsheet_en_210x297_2018-06-04.pdf.
15. "Energy consumption models for delivery drones: A comparison and assessment - ScienceDirect." <https://www.sciencedirect.com/science/article/pii/S1361920920308531> (accessed Jan. 31, 2021).

- sed Jan. 31, 2021).
16. T. Kirschstein, "Comparison of energy demands of drone-based and ground-based parcel delivery services," *Transp. Res. Part Transp. Environ.*, vol. 78, p. 102209, Jan. 2020, doi: 10.1016/j.trd.2019.102209.
 17. J. W. Langelaan, S. Schmitz, J. Palacios, and R. D. Lorenz, "Energetics of rotary-wing exploration of Titan," in *2017 IEEE Aerospace Conference*, Mar. 2017, pp. 1-11, doi: 10.1109/AERO.2017.7943650.
 18. Z. Liu, R. Sengupta, and A. Kurzhanskiy, "A power consumption model for multi-rotor small unmanned aircraft systems," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2017, pp. 310-315, doi: 10.1109/ICUAS.2017.7991310.
 19. "Helicopter Flight Physics | IntechOpen." <https://www.intechopen.com/books/flight-physics-models-techniques-and-technologies/helicopter-flight-physics> (accessed Jan. 31, 2021).
 20. K. S. R. Kumar, V. V. Sastry, O. C. Sekhar, D. K. Mohanta, D. Rajesh, and M. P. C. Varm a, "Design and fabrication of coulomb counter for estimation of SOC of battery," in *2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, Dec. 2016, pp. 1-6, doi: 10.1109/PEDES.2016.7914473.
 21. "Unmanned aerial vehicle," *Wikipedia*. Oct. 06, 2020, Accessed: Oct. 08, 2020.
 22. P. Gunetti, T. Dodd, and H. Thompson, "Simulation of a Soar-Based Autonomous Mission Management System for Unmanned Aircraft," *J Aerosp Inf Syst*, 2013, doi: 10.2514/1.53282.
 23. Y. Lu, X. Zhucun, G.-S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-Spat. Inf. Sci.*, pp. 1-12, Jan. 2018, doi: 10.1080/10095020.2017.1420509.
 24. R. Williams and W. Jianhua, "Dynamic Obstacle Avoidance for an Omnidirectional Mobile Robot," *J. Robot.*, vol. 2010, Sep. 2010, doi:10.1155/2010/901365
 25. C. Hernandez and J. A. Baier, "Improving MPGAA* for Extended Visibility Ranges," p. 5.
 26. L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of Robot 3D Path Planning Algorithms," *Journal of Control Science and Engineering*, Jul. 04, 2016. <https://www.hindawi.com/journals/jcse/2016/7426913/>(accessed Dec. 08, 2020).
 27. M. Likhachev, G. J. Gordon, and S. Thrun, "ARA* : Anytime A* with Provable Bound

- ds on Sub-Optimality,” in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. MIT Press, 2004, pp. 767–774.
28. M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, “Anytime Dynamic A*: An Anytime, Replanning Algorithm,” p. 10.
 29. J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013, doi: 10.1177/0278364913495721.
 30. W. Chen, M. Zhang, Y. Zhang, and X. Duan, “Exploiting meta features for dependency parsing and part-of-speech tagging,” *Artif. Intell.*, vol. 230, pp. 173–191, Jan. 2016, doi: 10.1016/j.artint.2015.09.002.
 31. T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” *ArXiv150902971 Cs Stat*, Jul. 2019, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1509.02971>.
 32. S. Brahmbhatt and J. Hays, “DeepNav: Learning to Navigate Large Cities,” *ArXiv170109135 Cs*, May 2017, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1701.09135>.
 33. Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, “RL²: Fast Reinforcement Learning via Slow Reinforcement Learning,” *ArXiv161102779 Cs Stat*, Nov. 2016, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1611.02779>.
 34. V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, Art. no. 7540, Feb. 2015, doi: 10.1038/nature14236.
 35. A. Seff and J. Xiao, “Learning from Maps: Visual Common Sense for Autonomous Driving,” *ArXiv161108583 Cs*, Dec. 2016, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1611.08583>.
 36. S. Bhatti, A. Desmaison, O. Miksik, N. Nardelli, N. Siddharth, and P. H. S. Torr, “Playing Doom with SLAM-Augmented Deep Reinforcement Learning,” *ArXiv161200380 Cs Stat*, Dec. 2016, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1612.00380>.
 37. A. Dosovitskiy and V. Koltun, “Learning to Act by Predicting the Future,” *ArXiv161101779 Cs*, Feb. 2017, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1611.01779>.
 38. P. Mirowski *et al.*, “Learning to Navigate in Complex Environments,” *ArXiv161103673 Cs*,

- Jan. 2017, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1611.03673>.
39. C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2722-2730, doi: 10.1109/ICCV.2015.312.
 40. A. Giusti *et al.*, "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 661-667, Jul. 2016, doi: 10.1109/LRA.2015.2509024.
 41. H. X. Pham, H. M. La, D. Feil-Seifer, and L. V. Nguyen, "Autonomous UAV Navigation Using Reinforcement Learning," *ArXiv180105086 Cs*, Jan. 2018, Accessed: Nov. 02, 2020. [Online]. Available: <http://arxiv.org/abs/1801.05086>.
 42. N. Imanberdiyev, C. Fu, E. Kayacan, and I. Chen, "Autonomous navigation of UAV by using real-time model-based reinforcement learning," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov. 2016, pp. 1-6, doi: 10.1109/ICARCV.2016.7838739.
 43. G. Muñoz, C. Barrado, E. Çetin, and E. Salami, "Deep Reinforcement Learning for Drone Delivery," *Drones*, vol. 3, no. 3, Art. no. 3, Sep. 2019, doi: 10.3390/drones3030072.
 44. L. Tai, G. Paolo, and M. Liu, "Virtual-to-real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation," *ArXiv170300420 Cs*, Jul. 2017, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1703.00420>.
 45. E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved Q-learning," *Robot. Auton. Syst.*, vol. 115, pp. 143-161, May 2019, doi: 10.1016/j.robot.2019.02.013.
 46. L. Geng, Y. F. Zhang, P. F. Wang, J. J. Wang, J. Y. H. Fuh, and S. H. Teo, "UAV surveillance mission planning with gimbaled sensors," in *11th IEEE International Conference on Control Automation (ICCA)*, Jun. 2014, pp. 320-325, doi: 10.1109/ICCA.2014.6870939.
 47. M. Z. Naser and V. K. Kodur, "Concepts and applications for integrating Unmanned Aerial Vehicles (UAV's) in disaster management," *Adv. Comput. Des.*, vol. 5, no. 1, pp. 91-109, 2020, doi: 10.12989/acd.2020.5.1.091
 48. J.-J. Park, J.-H. Kim, and J.-B. Song, "Path Planning for a Robot Manipulator based on

- Probabilistic Roadmap and Reinforcement Learning,” *International Journal of Control, Automation, and Systems*, vol. 5, no. 6, pp. 674-680, December 2007
49. Y. Kato, K. Kamiyama, and K. Morioka, “Autonomous robot navigation system with learning based on deep Q-network and topological maps,” in *2017 IEEE/SICE International Symposium on System Integration (SII)*, Dec. 2017, pp. 1040-1046
 50. A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, “Automated aerial suspended cargo delivery through reinforcement learning,” *Artif. Intell.*, vol. 247, pp. 381-398, Jun. 2017, doi: 10.1016/j.artint.2014.11.009
 51. A. Faust et al., “PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning,” *ArXiv171003937 Cs*, May 2018, Accessed: Nov. 18, 2020.
 52. H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, “RL-RRT: Kinodynamic Motion Planning via Learning Reachability Estimators from RL Policies,” *ArXiv190704799 Cs*, Jul. 2019.
 53. E. Khalil, “Learning Combinatorial Optimization Algorithms over Graphs,” Elias Khalil, May 31, 2019. /publication/dai-kha-yuyetal-17/.
 54. A. Francis et al., “Long-Range Indoor Navigation with PRM-RL,” *ArXiv190209458 Cs*, Feb. 2020, Accessed: Nov. 18, 2020. [Online]. Available: <http://arxiv.org/abs/1902.09458>.

주 의

1. 이 보고서는 국토교통과학기술진흥원에서 시행한 국토교통기술촉진연구 사업의 연구보고서입니다.
2. 이 보고서 내용을 발표하는 때에는 반드시 국토교통과학기술진흥원에서 시행한 국토교통기술촉진연구 사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 됩니다.

