사물인터넷융복합논문지 Vol. 7, No. 4, pp. 29-34, 2021 https://doi.org/10.20465/KIOTS.2021.7.4.029

가상 환경에서의 강화학습을 활용한 모바일 로봇의 장애물 회피

이종락*
영남이공대학교 사이버보안계열 교수

Obstacle Avoidance of Mobile Robot Using Reinforcement Learning in Virtual Environment

Jong-lark Lee*
Professor, Division of Cyber Security, YeungNam University College

요 약 실 환경에서 로봇에 강화학습을 적용하기 위해서는 수많은 반복 학습이 필요하므로 가상 환경에서의 시뮬레이션을 사용할 수밖에 없다. 또한 실제 사용하는 로봇이 저사양의 하드웨어를 가지고 있는 경우 계산량이 많은 학습 알고리즘을 적용하는 것은 어려운 일이다. 본 연구에서는 저사양의 하드웨어를 가지고 있는 모바일 로봇의 장애물 충돌 회피 문제에 강화학습을 적용하기 위하여 가상의 시뮬레이션 환경으로서 Unity에서 제공하는 강화학습 프레임인 ML-Agent를 활용하였다. 강화학습 알고리즘으로서 ML-Agent에서 제공하는 DQN을 사용하였으며, 이를 활용하여 학습한 결과를 실제 로봇에 적용해 본 결과 1분간 충돌 횟수가 2회 이하로 발생하는 결과를 얻을 수 있었다.

주제어 : 모바일로봇, 강화학습, ML-agent

Abstract In order to apply reinforcement learning to a robot in a real environment, it is necessary to use simulation in a virtual environment because numerous iterative learning is required. In addition, it is difficult to apply a learning algorithm that requires a lot of computation for a robot with low-spec. hardware. In this study, ML-Agent, a reinforcement learning frame provided by Unity, was used as a virtual simulation environment to apply reinforcement learning to the obstacle collision avoidance problem of mobile robots with low-spec hardware. A DQN supported by ML-Agent is adopted as a reinforcement learning algorithm and the results for a real robot show that the number of collisions occurred less then 2 times per minute.

Key Words : Mobile Robot, Reinforcement Learning, ML-Agent

1. 서론

최근 다양한 산업 분야에서 이동식 로봇에 대한 활용도가 높아지고 있다. 대형 물류창고에서의 물건의 이동, 호텔에서의 룸서비스, 위험한 환경에서의 수색 및 드론의 자율주행[1] 등에서 모바일 로봇이 다양하게 활용되고 있는데 이때 장애물 회피 기능은 필수적이라 할 수 있다.

모바일 로봇에 장애물 회피 문제에 대한 전통적인 접근법은 map-learning, 혹은 path-planning 방법으로써 프로그래머에 의해 잘 설계된 이동 경로를 만들어 로봇이 해당 경로 안에 존재하는 장애물을 만났을 때 어떻게 행동해야 할지를 선택하도록 하는 것이다[2,3]. 하지

본 논문은 2021학년도 영남이공대학교 연구조성비 지원에 의한 것임
*교신저자 : 이종락(jllee@ync.ac.kr)
접수일 2021년 10월 18일 수정일 2021년 11월 30일 심사완료일 2021년 12월 5일

만 이 방법은 사전에 고려하지 않은 상황이 발생하는 경우 대처할 수 없으며, 특정한 환경 내에서만 사용 가능하다는 문제점이 있다. 따라서 근래에는 머신러닝 기술 중 강화학습을 활용하여 모바일 로봇을 학습시켜 충돌을 회피하는 방법이 연구되고 있다[4,5,6].

모바일 로봇 분야에서의 강화학습은 로봇이 학습할 수 있는 다양한 환경들을 에피소드(Episode)로 만들어 시행착오(trial-and error) 방법을 활용해 학습시킴으로써 로봇의 성능을 개선시키는 머신러닝 기법이다. 이를 통해 로봇이 학습되지 않은 새로운 환경에서도 장애물을 회피할 수 있는 능력을 갖추도록하는 것이 목적이다. 이때 로봇을 학습시킬 수 있는 에피소드의 개수가 크면 클수록 로봇의 성능은 좋아지지만, 실제 환경에서 로봇이 훌륭한 성능을 낼 정도로 충분한 개수의 학습 에피소드를 만드는 것은 한계가 있다. 또한 에피소드를 충분히 만들 수 있다 할지라도 로봇이 복잡한 학습 알고리즘을 학습하기 위해서는 복잡한 계산을 수행할 수 있는 하드웨어 성능을 갖추고 있어야 한다. 본 연구에서는 다양한 가상 환경에서 가상의 로봇을 학습시킨 후 이를 하드웨어의 성능이 좋지 않은 실제 모바일 로봇에 적용시켜 장애물 회피 능력을 증명해보고자 한다. 이를 위한 가상 환경은 3D 모델링 표준 플랫폼으로 널리 사용되고 있는 Unity3D에서 제공하는 ML-Agent를 사용하며, 학습 알고리즘은 이 플랫폼 내에서 제공하는 Neural Network를 사용한다.

강화학습을 활용한 모바일 로봇의 장애물 회피에 대한 연구는 그 필요성이 증가하는 만큼 활발한 연구가 진행되고 있다. 강화학습에서는 로봇이 주변 환경에 대한 상태(state)를 인식한 후 선택 가능한 액션(action) 내에서의 보상(reward) 정책을 만들어 긍정적인 보상으로 유도하도록 학습시킨다. 이때 실제 환경과 동일한 가상의 환경을 만들어 놓고 학습을 통해 장애물을 회피하여 최적 경로를 찾아가도록 방법이 연구되었다[7,8,9]. 본 연구는 학습된 로봇이 새로운 환경에서 동작하도록 한다는 점에서 이들의 연구와 차별점이 있다. 미지의 환경에서 최적 선택을 유도하기 위한 연구로서 Q-Learning 알고리즘을 사용한 방법이 많이 사용되었다[8,9,10]. 하지만 이 방법은 일반적으로 상태와 액션에 대한 정보가 이산형(discrete)인 경우로 제한되며 된다. Chen Xia는 신경망 모형을 적용한 NNQL(Neural Network Q-Learning (NNQL) 방법을 통해 상태-액션 정보가 연속형인 실제 문제에서 장애물 회피가 성공적으로 가능함을 증명하였다[6]. 하지만 MATLAB이라는 가상의 환경내에서의 증명이며 이를 실제 환경에서 테스트하지는 못하였다.

본 연구에서는 Unity3D라는 가상의 플랫폼에서 학습시킨 학습모델을 실제 모바일 로봇에 적용해봄으로써 실제 환경에서 어느 정도의 효과가 있는지를 확인해 보는 것이 목적이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 강화학습과 적용하고자 하는 Unity 3D, ML agent를 기술하고 3장에서는 우리가 제안하는 시스템의 개념과 구성을 기술한다. 그리고 4장에서는 실험결과를 설명하고 5장에서는 결론과 향후 과제를 기술한다.

2. 관련 연구

2.1 Unity ML-Agents

ML-Agents는 게임 개발자 및 AI 연구자를 위해 Unity에서 개발한 플랫폼으로서 지능형 에이전트(Intelligent agents)를 개발하고 학습시킬 수 있는 환경을 제공하는 시뮬레이션 플러그인이다[12].

ML-Agents 툴킷을 활용하여 지능형 에이전트를 학습시킬 수 있는데, 이때 간단한 Python API를 활용하여 Reinforcement Learning(강화학습), Imitation Learning(모방학습), Neuroevolution(신경진화) 및 기타 머신러닝 알고리즘을 적용시킬 수 있다[13].

본 연구는 ML-Agents 툴킷에서 제공하는 학습 환경 중 강화학습을 활용한다.

2.2 Reinforcement Learning

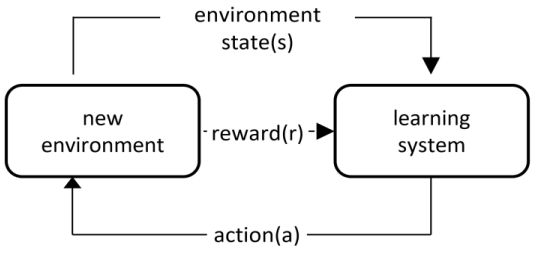
강화학습은 시행착오(trial-and-error) 기법에 근거한 머신러닝 방법으로서 로봇이 주변 환경(environment)과의 상호작용을 통해서 자신의 상태(state)를 인지하고 어떻게 행동(action)할지를 오로지 보상(reward) 시스템에 따라 학습하도록 한다는 점에서 지도학습이나 비지도학습과는 구별된다.

행동 결정의 주체인 로봇은 자신이 속한 환경을 인식하여 어떻게 행동할지를 선택하게 되는데. 이때 자신의 행동이 어떠한 보상을 받게 될 것인지에 대한 기댓값을 계산하게 된다. 이때 이 보상에 대한 기댓값을 최대화하는 방식으로 로봇이 행동을 선택하도록 학습시키는 것이다.

그 과정과 용어의 정의는 〈Table 1〉 및 [Fig. 1]과 같다.

〈Table 1〉 Key terminology of Reinforcement Learning

Terminology	definition
state	the state that the robot perceives through its surroundings
action	the action how the robot will behave in a particular state
reward	the reward expected when the robot performs a specific action in a specific state



[Fig. 1] Process of Reinforcement Learning

2.3 Q-Learning

Q-Learning은 강화학습을 위한 여러 가지 방법 중 하나로서 특정한 상태에서 특정한 행동을 수행할 결과에 대한 가치(state-action ($Q(s,a)$))를 계산하도록 하여 최적 행동을 선택하도록 하는 방법이다[14].

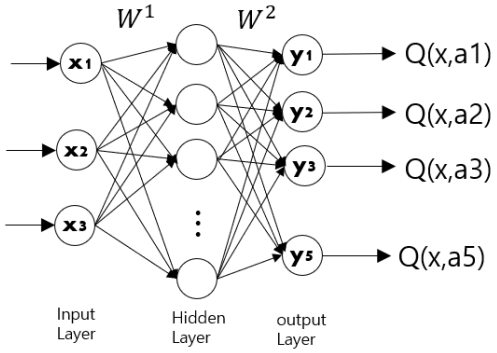
t 시점에서 로봇은 현재의 환경 상태를 s_t 로 관측하고 액션 a_t 를 선택한다. state-action Q-value인 $Q(s_t, a_t)$가 만들어진다. action a_t 가 수행된 후 로봇은 그 다음 상태인 s_{t+1}을 관측하고, 또한 즉각적인 보상 r_t 를 받는다. 마지막으로 현재의 $Q(s_t, a_t)$는 다음의 Q-value 함수에 의해 최적 Q-value인 $Q^*(s_t, a_t)$로 업데이트된다.

$$Q^*(s_t,a_t) = Q(s_t,a_t) + \alpha\left[r_t + \gamma \max_{a_{t+1}\in A} Q(s_{t+1},a_i) - Q(s_t,a_t)\right]$$

여기서 α 는 learning rate로서 0~1사이의 값을 가지며, 되지 않음을 의미하므로 학습되지 않음을 의미한다. 이를 1에 가까운 값으로 설정하는 것은 학습이 빨리 진행됨을 의미한다. γ는 0~1사이의 discount factor로서, 이 값이 0에 가까우면 로봇은 즉각적인 리워드를 고려하려 할 것이며, 반대로 1에 가까우면 더 미래의 리워드를 고려하게 된다.

2.4 Neural Network Q-Learning

Q-learning은 상태와 행동이 이산적(discrete)인 경우를 위해 설계되었으나 우리의 모바일 로봇의 장애물 회피 문제에서는 로봇이 움직임과 함께 센서의 입력으로 인해 환경에 대한 상태의 입력이 연속적이며, 이로 인한 행동 또한 연속일 수밖에 없다. 또한 보상의 최대값을 구하기 위해서는 모든 시점에서의 상태와 행동의 집합을 저장하여야 하는데 이를 위해서는 대용량의 메모리공간과 계산 속도가 빠른 컴퓨터가 필요하다. Chen[15]은 이 문제를 해결하기 위해 Q-learning에 신경망모델을 적용한 NNQL(Neural network Q learning)을 사용하였는데, 3계층의 신경망을 Q-table을 대신하여 사용하였고, output을 Q-value로 매핑하였다. 우리가 실험을 위해 사용하는 ML-agents에서는 DQN(Deep Q network)라는 이름으로 NNQL과 동일한 학습알고리즘을 제공한다.

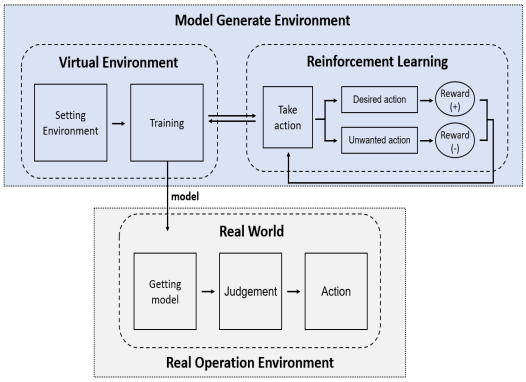


[Fig. 2] 3-layer NNQL model

3. NNQL을 활용한 모바일 로봇 장애물 회피

가상환경의 강화학습을 활용하여 실제 모바일 로봇이 장애물을 회피하도록 하는 우리의 문제에서는 Unity에서 제공하는 ML-Agent를 활용하기로 한다. 또한 모바일 로봇의 학습을 위해서는 ML-Agent에서 제공하는 DQN(Deep Q network)을 사용하되 입력센서를 3개 부착한 모바일 로봇을 사용하기로 한다. 이를 위한 시스템의 구성은 [Fig. 3]과 같다. 실제 환경을 시뮬레이션 할 수 있는 가상 환경(Virtual Environment) 모듈과, 가상환경과 상호작용을 통해 모델을 생성하는 강화 학습(Reinforce ment Learning) 모듈로 구성된다. 또한 이러한 2개의 환경을 통해 실제 환경(Real World)에서 사용할 모델을 생성한 후 이를 모바일 로봇에 이식하여 실

제 환경에서 사용하도록 한다.



[Fig. 3] Overall System Structure

〈Table 2〉 Overall Process of Experiment

1. Configure real environment and check the measured values 2. Configure virtual environment reflected the real environment 3. Training in the virtual environment to make model 4. Convert the model to using in the real environment 5. Apply the model to the real environment

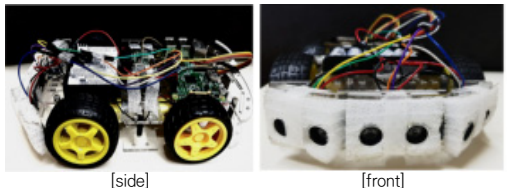
4. 실험 및 결과

우리의 실험 구성은 다음과 같다. 먼저 실제 장애물 회피를 위해 사용할 스마트 로봇을 만든다. 이 로봇은 거리를 측정할 수 있는 3개의 입력 센서를 가지고 있는데, 해당 센서로부터 환경을 인지하여 장애물을 회피 및 무작위 탐색(원더링)을 수행한다.

가상의 학습 환경으로는 Unity에서 제공하는 ML-Agents 모듈을 사용한다. 가상 환경에서 로봇이 장애물을 회피하며 원활하게 탐색을 진행하면 학습을 종료시킨다. 이후 실제 환경에 이 모델을 적용하여, 로봇이 장애물을 잘 회피하며 환경을 탐색하도록 하는 것이 목적이다.

4.1 실제 환경

실제 환경에서 구동되는 로봇은 전면에 3개의 거리 측정 센서(HC-SR04)를 달고, 움직이는 자동차 로봇이다. 제어 컴퓨터는 라즈베리파이(4B)를 사용하였다. 또한 강화학습을 원만하게 사용하기 위해 edge TPU(Coral TPU)를 사용했다.



[Fig. 4] Appearance of Mobile Robot

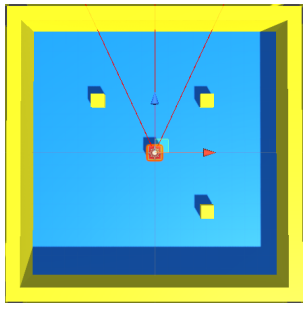
로봇의 거리 센서는 최대 약 450cm까지 측정하며, 1초에 약 48cm를 이동한다. 구현한 로봇의 기본적인 움직임은 아래 5개의 명령으로 정의하고 구현하였다.

〈Table 3〉 5 Actions of Mobile Robot

action	definition
go()	move forward
back()	move backward
turnLeft()	turn left
turnRight()	turn right
stop()	stop

4.2 가상 환경

가상 환경에서의 로봇은 사각 박스로 대응시켰는데, 중요한 것은 외형이 아니고 실제 환경과 유사한 운동 및 센싱이기 때문이다. 장애물 역시 4각 박스로 표현했으며 거리 측정을 위한 가상의 센서는 물리적 센서의 한계와 유사하게 시뮬레이션하였다. 초음파 센서 3개가 위치한 각도는 Quaternion 모듈의 Euler() 함수를 통해 가상 환경과 유사하게 적용하였다. 로봇의 속도도 실제 환경과 유사하게 반영하였다. 가상의 로봇도 실제 로봇과 동일하게 5개의 행동을 가지고 있다.

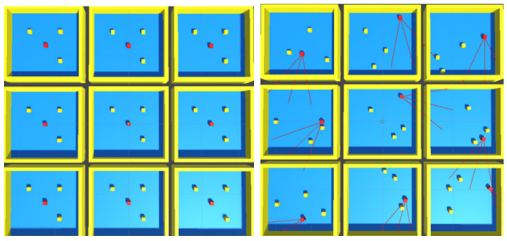


[Fig. 5] Virtual Environment of Simulation

4.3 강화학습을 통한 모델 생성

학습 모듈은 ML-Agent 0.15.1 버전을 사용하였다. 장애물은 실세계가 실시간으로 환경이 바뀐다는 가정하에 한 에피소드가 끝날 때마다 랜덤한 위치에서 새로 시작하도록 함으로써, 로봇이 고정된 환경에서 과적합 되는 것을 방지하고 장애물에 즉각적인 반응을 보이도록 설정하였다. 강화학습을 진행하기 위해서는 로봇의 행동과 보상이 필요한데, 우리는 다음과 같이 구현하였다.

우선, 로봇이 장애물을 만났을 경우의 행동은 학습된 모델이 상태를 판단하여 선택한다. 로봇이 장애물에 부딪히거나 back action을 취할 경우, reward에 (-)값을 주었으며 로봇이 go action을 취할 경우, (+)값을 부여하였다. (-)값은 장애물에 부딪혔을 경우, back action을 취했을 경우의 순으로 큰 값에서 작은 값(값의 크기는 절댓값을 기준으로 비교)을 부여하였다.

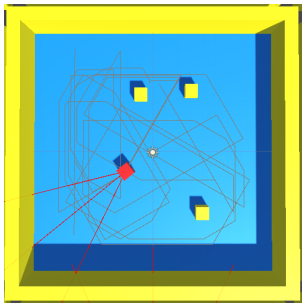


[Fig.6-(a)] virtual environment in unity

[Fig.6-(b)] learning in virtual environment

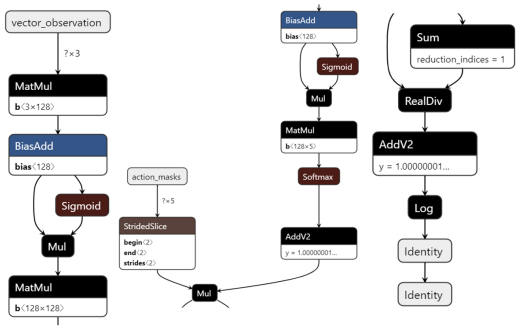
[Fig. 6] Simulation in virtual environment

가상 환경에서 로봇이 원만한 원더링을 보일 때 학습을 종료하였으며, 아래 그림은 가상 환경에서 로봇의 원더링 궤적의 예시를 보여준다.



[Fig. 7] Trajectory of Virtual mobile robot

실험에서 사용한 DQN 모델의 신경망 구조는 아래 그림과 같다.



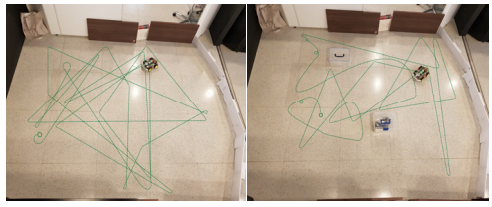
[Fig. 8] Structure of DQN model

4.4 실세계에서 적용

실제 환경에서 로봇은 tensorflow light 버전을 사용하므로 해당 모델을 .tflite 파일로 변환하는 과정이 필요하다. Edge TPU는 이렇게 변환된 모델에서 작동하게 되어 있다.

실험에서는 Unity3D의 agent와 로봇의 시간당 충돌 횟수를 비교해보아 학습이 원활히 진행되었는가와 원더링이 무사히 이뤄지고 있는가를 확인해 볼 것이다. 시간은 1분을 기준으로 한다.

해당 로봇은 가상 환경의 모델에 비하여 1분간 충돌 횟수가 2~3회 많은 것을 확인할 수 있었다(Table 4). 따라서 실세계에서의 적용은 가상 환경보다 정확도는 떨어지지만 학습된 모델대로 동작한다는 것을 알 수 있다. [Fig. 9]는 실세계에서의 로봇의 궤적을 나타낸 것이다.



[Fig. 9-(a)] Without obstacles

[Fig 9-(b)] With obstacles

[Fig. 9] Robot trajectory

〈Table 4〉 number of collision after Experiments

environment	number of collision	
	without obstacles	with obstacles
virtual	0	0
real	1	2

5. 결론

실 환경 로봇에 강화학습을 적용하기 위해서는 수많은 학습이 필요하므로 가상 환경 시뮬레이션을 사용할 수밖에 없다. 우리는 Unity에서 제공하는 ML-agents라는 강화학습 프레임을 이용하여 강화학습을 수행한 후 이를 실제 환경에서 로봇의 장애물을 회피 문제에 적용하였다.

로봇의 장애물 충돌 회피 문제를 위해 DQN 알고리즘을 사용하였으며, 가상 환경에서의 학습 모델이 실제 환경에서도 원만하게 작동함을 실험으로 확인하였다. 이러한 시스템의 성능은 실 환경을 얼마나 유사하게 가상 환경에 모델링 하는가가 중요하며, 특히 입력되는 센서의 값이 실제 환경과 비례한다. 그리고 가상 환경에서는 데이터의 노이즈가 없지만, 실제 환경에서는 일정 수준의 노이즈가 발생한다는 점을 고려할 필요가 있다.

또한 좀 더 복잡한 로봇 강화학습의 문제를 이러한 플랫폼에서 쉽고 효과적으로 적용하여 활용할 수 있는지 확인하는 추가적인 연구가 필요하다.

REFERENCES

[1] D.W.Lee, K.M.cho and S.H.Lee, "Comparison & Analysis of Drones in Major Countries based on Self-Driving in IoT Environment," Journal of The Korea Internet of Things Society, Vol.6, No.2, pp.31-36, 2020.

[2] D. Filliat and J.A.Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," Cognitive Systems Research, Vol.4, No.4, pp.243-282, 2003.

[3] J.A. Meyer and D. Filliat, "Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies," Cognitive Systems Research, Vol.4, No. 4, pp. 283-317, 2003.

[4] R.S.Sutton and A.G.Barto, "Reinforcement Learning: An Introduction," A Bradford Book, MIT Press, 2th ed., 2017.

[5] A.E.Sallab, M.Abdou, E.Perot and S.Yogamani, "Deep reinforcement learning framework for autonomous driving," Journal of imaging Science and Technology, Vol.1, No.7, pp.70-76, 2017.

[6] X.B.Peng, G.Berseth, K.Yin and M.V.Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," ACM Transactions on Graphics, Vol.36, No.41 pp.1-13, 2017.

[7] J.H.Woo and N.K.Kim, "Collision Avoidance for an Unmanned Surface Vehicle Using Deep Reinforcement Learning," Graduate School of Seoul National University, Doctoral Dissertation, 2018.

[8] A.Coates, P.Abbeel and A.Y.Ng, "Apprenticeship learning for helicopter control," Communications of the ACM, Vol.52, No.7, pp.97-105, 2009.

[9] S.Y.Park, "Object-spatial layout-route-based hybrid nap and its application to mobile robot navigation," Graduate School of Yonsei University, Doctoral Dissertation, 2010.

[10] N.J.Cho, "Learning, improving, and generalizing motor skills for autonomous robot manipulation : an integration of imitation learning, reinforcement learning, and deep learning," Graduate School of Hanyang University, Doctoral Dissertation, 2020.

[11] B.G.Ahn, "An Adaptive Motion Learning Architecture for Mobile Robots," Graduate school of SungKyunKwan University, Master's Thesis, 2006.

[12] https://github.com/Unity-Technologies/ml-agents

[13] A.B.Juliani, E.Teng, A.Cohen, J.Harper, C.Elion, C.Goy, Y.Gao, H.Henry, M.Mattar and D.Lange, "Unity: A General Platform for Intelligent Agents," arXiv:1809.02627, 2020.

[14] J.C.H.Watkins, D.Peter, "Q-learning," Machine Learning, Vol.8, No.1, pp.272-292, 1992.

[15] X.Chen, "A Reinforcement Learning Method of Obstacle Avoidance for Industrial Mobile Vehicles in Unknown Environments Using Neural Network," Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management, Vol.1, No.1, pp.671-67, 2014.

이 종 락(Jong-Lark Lee) [정회원]

- 1998년 2월 : 성균관대학교 대학원 통계학과(통계학 석사)
- 2012년 2월 : 성균관대학교 대학원 통계학과(통계학 박사)
- 2001년 3월 ~ 2014년 2월 : 서울호서전문학교 사이버보안과 교수
- 2014년 3월 ~ 현재 : 영남이공대학교 사이버보안계열 교수

〈관심분야〉
모바일로봇, 인공지능, 사이버보안