Journal of Korea Robotics Society (2022) 17(1):058-067
https://doi.org/10.7746/jkros.2022.17.1.058 ISSN: 1975-6291 / eISSN: 2287-3961

저가 Redundant Manipulator의 최적 경로 생성을 위한 Deep Deterministic Policy Gradient(DDPG) 학습

Learning Optimal Trajectory Generation for Low-Cost Redundant Manipulator using Deep Deterministic Policy Gradient(DDPG)

이 승 현[1] · 진 성 호[1] · 황 성 현[1] · 이 인 호†
Seunghyeon Lee[1], Seongho Jin[1], Seonghyeon Hwang[1], Inho Lee†

Abstract: In this paper, we propose an approach resolving inaccuracy of the low-cost redundant manipulator workspace with low encoder and low stiffness. When the manipulators are manufactured with low-cost encoders and low-cost links, the robots can run into workspace inaccuracy issues. Furthermore, trajectory generation based on conventional forward/inverse kinematics without taking into account inaccuracy issues will introduce the risk of end-effector fluctuations. Hence, we propose an optimization for the trajectory generation method based on the DDPG (Deep Deterministic Policy Gradient) algorithm for the low-cost redundant manipulators reaching the target position in Euclidean space. We designed the DDPG algorithm minimizing the distance along with the jacobian condition number. The training environment is selected with an error rate of randomly generated joint spaces in a simulator that implemented real-world physics, the test environment is a real robotic experiment and demonstrated our approach.

Keywords: Trajectory Optimization, Low-Cost Manipulator, DDPG

1. 서 론

최근 다양한 구조의 manipulator와 장애물이 있는 환경, 미세 작업 환경, 수중, 모바일 로봇 위와 같은 다양한 환경에서 강인한 manipulator 경로 및 제어에 관한 연구가 진행되고 있다. 이는 전통적인 kinematics/dynamics와 같은 접근 방법을 기초로 연구가 되어 왔으며, Newton-Raphson method를 이용한 역 기구학의 해와 Closed-form의 형태로 수식을 정의하여 역 기구학의 해를 구하여 경로를 생성하는 방법으로도 연구되었다[1-3]. 이를 이용한 연구로는 모바일 manipulator end-effector에 달린 카메라를 이용하여 물건을 잡는 연구[4], 구난 로봇용 양팔 manipulator의 진동이 적게 발생하는 자세를 구하는 연구[5]가 진행되었다.

최근에는 인공 신경망을 사용하여 manipulator 경로, 제어에 관한 연구를 진행하고 있으며, 신경망을 사용한 manipulator의 제어에 대한 가장 기본적인 접근 방식 중 하나는 manipulator의 기저에서 end-effector 변환 행렬을 뉴런 입력으로 사용하고 6개의 joint 각도를 출력으로 하여 inverse kinematics를 해결하는 연구가 있다[6]. 또한, manipulator의 경로에 대한 가장 기본적인 접근으로 다양한 작업 사이에서 최소 주행 시간을 가지는 경로 생성을 신경망으로 해결한 연구[7], multi-arm manipulator의 고차원적인 경로 계획 문제를 SAC (Soft Actor-Critic) 강화학습과 MAMMDP (multi-arm Manipulator Markov decision process)로 해결하는 연구[8], 수중 건설 로봇에 장착된 manipulator의 제어 방법에 대하여 Meta 강화 학습 방법을 사용하는 연구[9], 모바일 manipulator의 pick and place를 문제를 해결

Received : Dec. 16. 2021; Revised : Jan. 21. 2022; Accepted : Feb. 16. 2022
※ This work was supported by BK21FOUR, Creative Human Resource Education and Research Programs for ICT Convergence in the 4th Industrial Revolution
※ This work was supported by Pusan National University Research Grant, 2020
1. MS Student, Pusan National University, Busan, Korea (seunghyeon1696, seongho1696, seonghyeon7379@pusan.ac.kr)
† Assistant Professor, Corresponding author: Dept of Electronics Engineering Pusan National University, Busan, Korea (inholee8@pusan.ac.kr)

하기 위해 DRL (Deep Reinforcement Learning)로 경로를 생성하는 연구[10], 장애물을 회피하는 경로 생성을 위한 신경망 연구 등이 진행되었다[11,12].

하지만, 고전적인 방법을 이용한 경로와 학습을 이용한 경로 바탕으로 모터를 제어하는 방법은 다양한 task에 작업을 정밀하게 해결할 가능성이 있지만, 일반적으로 high resolution, high stiffness, gear ratio를 가지는 manipulator 하드웨어에 의존한다. 이는 manipulator의 동적 움직임을 구현하기 위해 전통적인 kinematics/dynamics 계산 과정에서 작은 오차를 가지지만, 이를 신뢰성 있게 구현하기 위해 manipulator 시스템은 high resolution, high stiffness, gear ratio와 같은 요구사항이 필요하다.

반면에 저가의 manipulator는 위의 요구사항을 만족하지 못한 low resolution, low stiffness를 가지게 된다. 이는 계산 과정에서 작은 오차를 이끌 수 있지만, 실제 환경에서 동적 움직임에서 많은 부정확성을 가져오게 되고 밀접한 사람의 생활 공간에서 개발하는 데 큰 걸림돌이 된다. 예를 들어, manipulator end-effector에 폐쇄회로 텔레비전을 부착하여 광범위한 화각의 공간 관측 시스템을 구성하는 경우, end-effector의 출렁임과 부정확성이 카메라에 영향을 주어 시스템을 비효율적으로 만든다. 이러한 문제점으로 인해, 하드웨어적 오차가 있는 저가형 redundant manipulator의 end-effector가 Euclidean space (x, y, z)에서 높은 정밀도를 이끌 수 있는 최적 경로가 필요하다.

또한, 기구학적인 여자유도를 가지지 못한 manipulator는 end-effector를 원하는 위치와 방향으로 두는데 필요한 자유도보다 적은 자유도를 가지기 때문에 end-effector의 운동에는 영향을 주지 않고, joint 변화만을 이용하여 자세를 바꿀 수 없다. 하지만, 비선형적 오류를 가지는 manipulator는 end-effector의 위치와 방향을 고정하고 많은 자세에 대한 비교가 필요하므로 충분한 여자유도를 가지는 redundant manipulator가 필요하게 된다.

따라서, 본 논문에서는 낮은 비용으로 인한 low resolution, low stiffness와 같은 비선형적인 오류를 가지는 redundant manipulator에 대하여 end-effector가 선형 궤적으로부터 작은 오차를 가지는 최적 경로를 DDPG (Deep Deterministic Policy Gradient) 알고리즘을 사용하여 생성하는 방법을 제안한다[13].

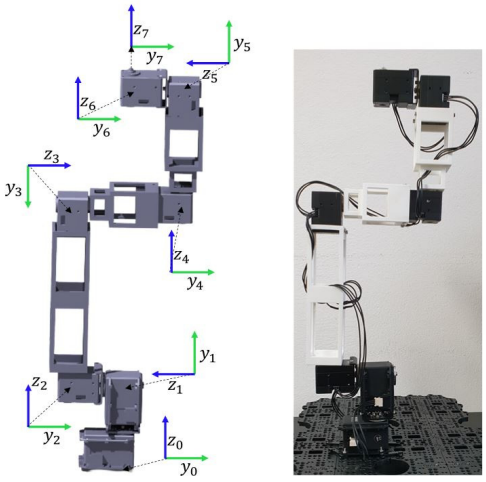
본 연구에서 end-effector의 경로를 생성하고 이를 inverse kinematics 통해 joint 영역으로 변환할 때 발생하는 singularity의 발생 문제를 joint의 변화량을 출력으로 하여 모터의 급가속 및 과부하를 방지하고 추가적인 singularity 회피 알고리즘이 필요하지 않게 하였으며[14], low resolution, low stiffness로 발생하는 비선형적인 오류 문제에 대하여 입력 상대 오차($\Delta\theta$)에 대한 출력 상대 오차(ΔX)의 증폭 인자 Jacobian condition number가 낮은 joint 경로를 생성하여 Euclidean space에서 비선형적인 오류에서 강인한 경로를 생성하여 해결한다[15]. 또한, DDPG 학습 과정에서 manipulator 자유도가 증가함에 따라 목표 성공 빈도가 감소하는 문제점이 있다. 이는 학습 속도를 늦추거나, 신경망이 엉뚱한 방향으로 수렴 및 학습하는 문제점을 일으킨다. 이러한 문제점을 해결하기 위하여, 본 논문에서는 목표 도착 허용 범위를 ε-greedy policy와 유사하게 이용하여 점차 줄이는 방법을 택하여 안정적이고 빠르게 학습하는 결과를 보여준다[16].

2장에서는 system overview를 통하여 redundant manipulator를 사용한 이유를 설명하고 문제 정의 및 시스템 구성에 관해 설명한다. 3장에서는 비선형적인 manipulator의 오차를 줄이려는 방법인 Jacobian condition number의 이론적 배경을 설명한다. 4장에서는 최적 경로를 생성하기 위해 사용한 DDPG 알고리즘에 대하여 설명한다. 5장에서는 실험을 위한 실제 환경과 가상환경의 구성을 설명한다. 6장에서는 실제 환경과 가상환경에서 실험을 통해 본 논문에서 제시한 최적 경로의 타당성을 확인한다. 7장에서는 본 연구의 결론 및 향후 연구계획을 제시한다.

2. System Overview

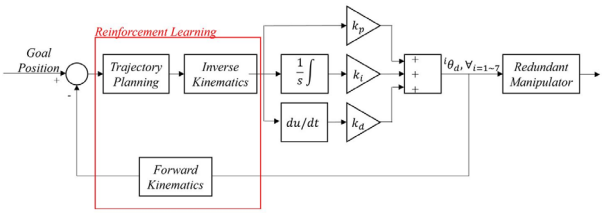
2.1 Redundant Manipulator

[Fig. 1]의 a, b는 실험 테스트를 수행하기 위해 제작된, 저가형 redundant manipulator의 가상 환경과 실제 환경을 보여준다. 1-2 joint는 Dynamixel MX-64 서보 모터, 3-7 joint는 Dynamixel XM430-W350 서보 모터를 포함하며, link는 3D printer를 사용하여 총 220만 원 이하에 설계되었다. [Fig. 1]과 같은 구조로 redundant manipulator를 설계하였을 때, joint의 회전각은 $\pm\pi$, 최대 각속도는 $2^\circ/\mathrm{sec}$가 된다. 이는 강화 학습



(a) (b)

[Fig. 1] (a) Actual Environment, (b) Simulation Environment



[Fig. 2] Manipulator Control Block Diagram

을 할 때, 모터의 한계점을 지정하지 않는다는 장점과 manipulator workspace를 넓게 사용할 수 있다는 장점이 있다. 또한, Redundant manipulator를 사용하여 하나의 end-effector의 위치와 방향을 두고 다양한 자세를 비교할 수 있게 하였다.

2.2 Problem Definition and System Configuration

목표지점까지 도달하기 위한 manipulator의 관절 궤적 생성 제어 기법은 [Fig. 2]와 같다. Goal position이 정해지면 trajectory planning을 통해 다음 경로에 대한 end-effector의 위치와 각도를 결정되고 결정된 위치와 각도는 inverse kinematics를 통해 각 joint의 각도를 결정한다. 결정된 각도는 PID 제어기를 통해 redundant manipulator를 제어에 활용하며, forward kinematics를 통해 현시점의 end-effector 위치와 각도를 feedback 받는다.

여기서, Trajectory planning과 inverse kinematics 계산의 오차가 작더라도 forward kinematics에서 low resolution encoder에 의존하기 때문에 end-effector의 정밀성은 떨어지게 된다. 더불어, low stiffness로 인해 동적인 움직임에서 더 큰 오차가 예상된다. 이러한 오차들은 수학적으로 모델링하기 어려운 비선형성 문제이고 자유도가 높아질수록 효과가 더 증가한다.

이러한 문제점을 해결하기 위하여 본 논문에서 trajectory planning과 inverse kinematics, forward kinematics를 포함한 reinforcement leaning을 사용해 해결하고자 한다.

Manipulator의 궤적 최적화 강화 학습은 agent (manipulator)와 환경 사이에 action (trajectory), reward 주고받는 상호 작용을 통해 최적의 trajectory를 생성한다. 이때, 환경은 manipulator의 low resolution encoder과 low stiffness를 한 축별로 정의해서 만들어진 동적 시뮬레이터이다.

실제 환경 실험에서는 동적 시뮬레이터에서 실험하고 나오는 joint 변화량만을 이용하여 진행하였다.

3. Jacobian Condition Number

Jacobian condition number는 입력 상대 오차($\Delta\theta$)에 대해 출력 상대 오차(ΔX)의 증폭 인자이다[15]. 따라서, 증폭 인자를 정량화하기 위하여, Jacobian을 이용한 식 (1)을 사용할 수 있다.

$$J^{\dagger}\Delta X = \Delta\theta \tag{1}$$

여기서 Jacobian(J)은 ΔX(x, y, z, roll, pitch, yaw)와 각 joint의 각도($\Delta\theta$) 사이의 선형 관계를 나타내는 행렬 식으로 본 논문의 Jacobian은 6×7 행렬이 된다. 직사각행렬 모양을 가진 Jacobian의 역행렬을 구하기 위하여, $J^{\dagger}$은 7×6 inverse kinematic Jacobian matrix로 Moore-Penrose Pseudo inverse를 사용하였다. 또한, norm 중 가장 많이 사용되는 2-norm을 통해 식 (2)를 유도할 수 있으며, 식 (1), (2)를 통해 식 (3)을 유도한다[15].

$$\|J^{\dagger}\Delta X\| \le \|J^{\dagger}\|\|\Delta X\| \tag{2}$$

$$\frac{\|\Delta X\|}{\|X\|} \le \|J^{\dagger}\|\,\|J\|\frac{\|\Delta\theta\|}{\|\theta\|} \tag{3}$$

따라서, 증폭 인자 Jacobian condition number(κ)는 식 (4)와 같이 정의된다[15].

$$\kappa(J) = \kappa(J^{\dagger}) = \|J^{\dagger}\|\,\|J\| \tag{4}$$

증폭 인자 $\kappa(J)$가 적은 방향일수록 joint encoder(θ)의 부정확성과 link의 뒤틀림에서 end-effector의 부정확성은 줄어든다. 따라서, Jacobian condition number를 고려한 joint의 경로를 생성하였을 때, 우리가 줄이고자 하는 저가형 redundant manipulator의 end-effector 부정확성은 낮아진다.

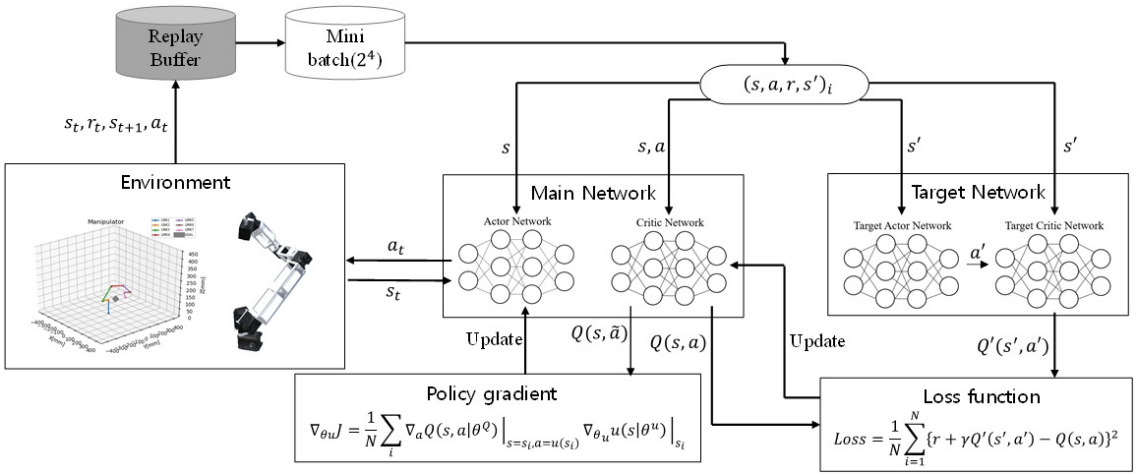
4. Optimal Trajectory Generation Method based on DDPG

4.1 DDPG

DDPG는 정책 신경망(actor)에 대한 평가와 정책 개선을 다른 신경망(critic)을 통해 반복하는 강화 학습의 한 알고리즘으로 최적의 policy parameter(main actor network)를 찾는 목적을 가진다[13]. 이는 actor 신경망과 critic 신경망으로 이루어져 있으며, actor 신경망은 state를 확인하고 이에 대한 policy parameter를 통해 action을 출력하는 신경망이다. 또한, critic 신경망은 action과 state를 통하여 행동 가치를 평가함으로써 actor 신경망을 조율하는 역할을 한다.

[Fig. 3]에서 특정 상태 s_t에서 Q-learning의 ε-greedy policy에 의해 action(a_t)이 생성되고 environment로 들어가 manipulator를 움직인다. ε-greedy policy는 식 (5)와 같다[16].

$$\pi(s,a) = \begin{cases} 1-e+\dfrac{e}{|A(s)|} & (a = A^{*}) \\ \dfrac{e}{|A(s)|} & (a \neq A^{*}) \end{cases} \tag{5}$$



[Fig. 3] Deep Deterministic Policy Gradient schematic

여기서, $|A(s)|$는 state(s)에서 가능한 행동의 개수, A^*은 Q 값을 최대로 하는 action(a)이며, e는 $0 \le e \le 1$로 수렴된다. 즉, $e=1$이면 행동 100%를 랜덤하게 선택하게 되고 $e=0$이면 식 (6)을 이용하여 greedy policy로 행동을 결정한다.

$$u(s) = \arg max_a Q(s,a) \tag{6}$$

이는 state에서 Q 값이 가장 크게 나오는 action을 선택하는 의미가 있다. Q 값은 행동 가치함수로써 policy π를 따랐을 때 action과 state에 대한 기댓값으로 다음과 같이 계산한다.

$$Q^{\pi}(s_t,a_t) = E_{r_{i\ge t}, s_{i>t}\sim E, a_{i>t}\sim\pi}[R_t|s_t,a_t] \tag{7}$$

이를 통해 행동(a_t)을 결정하게 되고 environment와 main actor network는 상호 작용을 통해 현재 상태(s_t), 다음 상태(s_{t+1}), 보상(r_t)을 생성한다. 데이터[s_t, a_t, r_t, s_{t+1}]는 replay buffer에 저장하고 일정 수준 이상 데이터가 쌓이면 mini batch 형식으로 추출하게 된다. 이는 신경망을 update 하기 위하여 데이터들은 서로 독립적이라고 가정해야 하지만, replay buffer의 데이터는 독립적이지 않기 때문에 독립성을 부여하기 위한 작업이다.

다음 순서로 독립 표본 데이터 [s_t, a_t, r_t, s_{t+1}] 기반으로 target network를 통해 행동 가치함수 Q 값을 만드는 main critic network를 update 하는 과정이 필요하다. 이 과정에서 Time Difference target ($r_t + \gamma Q(s_{t+1}, \pi_\theta(s_{t+1}))$)이 영향을 받기 때문에 신경망을 복사한 target critic network, target actor network를 별도로 두어 손실함수를 다음과 같이 계산한다.

$$Loss = \frac{1}{N}\sum_{i=1}^{N}(r_i + \gamma Q'(s_{i+1}, u'(s_{i+1})|\theta^{u'})|\theta^{Q'}) - Q(s_i,a_i))^2 \tag{8}$$

여기서, Q'은 target critic network이며, u'은 target actor network, γ는 감쇠비로 표기하며, $u'(s_{i+1}) \mid \theta^{u'}$는 s_{i+1}의 상태를 보고 target actor network (u')를 통해 나온 a_{i+1}가 되어 [Fig. 3]의 loss function이 된다. 이를 정리하면 target actor network를 통해 다음 상태(s_{t+1})에 대한 다음 행동(a_{t+1})을 추출하고, 이를 이용하여 target critic network를 통한 평갓값 $Q'(s',a')$과 main critic network를 통한 평갓값 $Q(s,a)$을 차이를 통해 back propagation을 이용한 main critic network를 update 하게 된다.

다음으로 policy gradient를 통해 최적의 policy (main actor network)의 parameter를 찾는 단계가 된다. 이는 objective function을 이용하여 gradient decent를 최대화하고 최대화된 gradient decent로 update 방향 및 크기를 결정한다. 여기서 objective function은 trajectory 동안 받을 것이라고 기대하는 보상의 합이며, trajectory는 독립 표본 데이터[s_t, a_t, r_t, s_{t+1}]이다. 따라서 식 (9) 과 같은 objective function을 기초로 식은 시작된다.

$$J(\theta) = E[Q(s,a)|_{s=s_t, a_t=u(s_t)}] \tag{9}$$

여기서, policy loss를 계산하기 위해 policy parameter와 chain rule을 적용하여 objective function을 미분하며 이는 식 (10)과 같다.

$$\nabla_{\theta u} J(\theta) = \nabla_a Q(s,a) \nabla_{\theta u} u(s|\theta^u) \tag{10}$$

식 (11)을 사용하여 독립 표본 데이터들의 기울기의 합계 평균으로 main actor network를 update 한다.

$$\nabla_{\theta u} J = \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=u(s_i)} \nabla_{\theta u} u(s|\theta^u)|_{s_i} \tag{11}$$

따라서 main network의 actor, critic network는 update가 되었으며, 식 (12-13)과 같이 target network는 main network를 기반으로 느린 속도로 update 된다.

$$\theta^{Q'} = \tau_c \theta^{Q} + (1-\tau)\theta^{Q'} \tag{12}$$

$$\theta^{u'} = \tau_a \theta^{u} + (1-\tau)\theta^{u'} \tag{13}$$

낮은 update factor (τ_c, τ_a)로 느리게 target network를 update 하여 안정성을 확보하였다.

즉, DDPG 알고리즘은 앞서 설명한 과정을 거쳐 actor, critic 신경망의 가중치를 update 함으로써 학습하며, 최적의 Policy parameter (Main Actor Network)를 찾게 된다. 학습이 종료된 후 상태(s)를 입력받았을 때 Main Actor Network의 결과인 행동(a)를 도출하는 알고리즘이다.

DDPG 알고리즘은 DQN (Deep Q Learning), DPG (Deterministic Policy Gradient)와 비교할 수 있다[17,18]. 두 강화 학습과 비교하여 DDPG의 장점은 continuous action space를 가진 문제에서 높은 학습률을 보이며, policy gradient를 통해 policy를 바로 학습 가능하다는 것과 policy gradient를 수행할 때, action에 대한 integral을 수행하지 않음으로 계산 과정이 단순화된다. 이러한 장점으로 인하여, 본 논문에서 DDPG를 구현하였고, low resolution, low stiffness에서 발생한 비선형적 오류를 가지는 저가형 redundant manipulator에 대한 end-effector가 선형 궤적으로부터 작은 오차를 가지는 최적 경로 생성 문제를 Markov decision process로 변환하여 해결하였다[19].

4.2 The State Space

State space는 DDPG의 agent가 인지하는 환경 정보를 나타내며 해당 agent의 action을 결정하는 요소이다. 하지만, 현실 세계에서 state 집합은 방대하거나 무한할 수 있으므로 모든 상태를 이용하는 것은 비현실적이다. 따라서, 최적 경로 생성의 문제를 해결하기 위한 특정 state를 결정해야 한다. 본 논문에는 DDPG의 agent가 관찰 가능한 state(S)는 식 (14)와 같은 25개로 결정하였다.

$$s_t = \{G, x_i, y_i, z_i, X, Y, Z\} \in S \tag{14}$$

여기서, G는 goal의 약자로 end-effector가 목표지점 허용 범위에 들어와 있음을 0과 1로 나타내며, x_i, y_i, z_i는 각 link 끝단의 위치 21개, X, Y, Z는 목표지점과 manipulator의 end-effector와의 각 좌표축의 Euclidean distance 3개이다.

4.3 The Action Space

Action space는 DDPG의 state의 기반으로 환경에 직관적으로 행하는 부분이다. 따라서, 본 논문에서 DDPG의 agent가 취할 수 있는 action (A)은 각 joint의 변화량으로 식 (15)와 같은 7개가 된다. 따라서, Environment의 manipulator는 각도의 변화량(ω_i)과 current angle (θ_i)의 합으로 이동하며 최적의 경로를 생성한다. 이는 joint space 상의 경로에서 제어하기 때문에 singularity의 생성 조건을 사전에 차단할 수 있는 장점으로 다음 경로에 대한 급가속과 모터의 과부하를 방지할 수 있으며, 추가적인 singularity 회피 경로를 생성이 필요하지 않다.

$$a_t = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7\} \in A \tag{15}$$

4.4 Reward Function

Agent의 reward function은 action의 각도의 변화량에 대한 요구사항을 충족하는지를 평가한다. 따라서, 본 논문에서 제시하는 저가형 redundant manipulator에 대한 공간상 선형 궤적으로부터 end-effector가 작은 오차를 가지는 경로를 이끌 수 있는 reward이며, reward의 인자와 계수를 변형해가며 실험한 결과 제일 좋은 결과를 보인 식 (16-20)과 같이 정하였다.

$$R_1 = -\frac{d}{R_{workspace}/2} \tag{16}$$

$$R_2 = \begin{cases} 0, \text{if}(d > L) \text{ or } (step \le 3) \\ 20, \text{if}(d < L) \text{ and } (step > 3) \end{cases} \tag{17}$$

$$L = MAX(L^{*}0.999, L_{target}) \tag{18}$$

$$R_3 = -\frac{\kappa(J)}{Mean(\kappa(J))} \tag{19}$$

$$R_{total} = R_1 + R_2 + R_3 \tag{20}$$

여기서, 식 (16)의 R_1은 end-effector가 목표지점을 찾아가게 하는 reward이며, d는 목표지점과 end-effector의 Euclidean distance이다. $R_{workspace}/2$을 통해 reward를 scaling 하여, R_1의 보상을 $-1 \le R_1 \le 0$로 수렴하게 하였다. 이는 거리(d)와 동일하게 reward로 주어졌을 때, 식 (19)의 R_3보다 높은 reward를 받게 된다. 이때, 학습은 Jacobian condition number를 고려하지 않게 되며, 본 논문에서 제시하는 공간상 선형 궤적으로부터 end-effector가 작은 오차를 가지는 경로를 생성할 수 없게 한다.

식 (17)의 R_2는 목표지점 허용 범위에 들어왔음에 따른

reward로 목표 거리(L)이내에서 4 step을 유지했을 경우 보상을 받게 된다. 4 step을 유지하는 의미는 s_{t-3}에서 s_t까지 고려하는 의미가 있다. 1 step(s_t)만을 고려하였을 때, 로봇의 quasi-static한 행동에서만 높은 학습률을 보였으나, 동적인 행동에서는 높은 오차가 여전히 발견되었다. 이는 강화 학습이 속도나 가속도로 인한 end-effector의 오차를 고려하지 않는 의미가 있다. 3 step (s_t, s_{t-1}, s_{t-2}) 이상을 고려하였을 때는 강화 학습은 위치와 속도, 가속도를 고려하게 된다. 이는 low resolution encoder과 low stiffness를 통해 생성되는 추가적인 동적 행동에 대해 현시점의 위치와 속도를 알 수 있게 되며, 추가로 속도가 점점 느려지는가에 대해 강화 학습은 고려하여 학습을 할 수 있게 된다.

식 (18)의 L는 식 (5)의 ε -greedy policy와 유사하게 이끌어 목표 허용 범위를 점차 줄이는 방법을 선택하였다[16]. ε -greedy policy는 무작위로 움직이는 탐험, 경험의 양을 점차 줄이고 학습된 데이터의 기반으로 움직이는 양을 늘리는 방식이다. 강화 학습은 manipulator의 자유도가 증가할수록 높은 차원의 space 안에서 성공에 대한 빈도가 낮게 된다. 이는 학습의 속도는 늦어지게 만들고 강화 학습은 계속 탐험과 경험만을 택할 수 있다. 또한, 목표에 대한 탐험과 경험의 부재로 인해 엉뚱한 방향으로 수렴할 수 있다. 이를 방지하기 위하여 ε -greedy policy와 유사한 목표 허용 거리를 설계하여 학습 초반에는 목표 허용 범위를 크게 주고 점차 작게 줌으로써 성공 경험의 빈도를 높여 빠르게 학습할 수 있게 하였으며, 약 2,000 episode 이후로는 목표로 하는 허용 범위(L_{target})를 유지할 수 있게 하여, 최종적으로 목표로 하는 목표 허용 범위를 도착 할 수 있게 하였다. 실험 과정에서는 목표 허용 범위를 L_{target}로 사용하여 진행하였다.

식 (19)의 R_3은 Jacobian condition number에 관한 reward이며, Jacobian condition number ($\kappa(J)$)는 입력 상대 오차($\Delta\theta$)에 대한 출력 상대 오차(ΔX)의 증폭 인자로, 높을수록 low resolution encoder과 low stiffness에서 발생하는 비선형적인 오차를 증폭시킨다. 따라서, R_3의 reward 통해 비선형적인 오차에도 Jacobian condition number가 작은 방향으로 경로 계획을 하여 manipulator의 end-effector에서의 비선형적 오차를 낮게 한다. 또한, 식 (16)의 R_1과 비슷한 값을 보상받게 하도록 Jacobian condition number($\kappa(J)$)의 평균으로 나누어 낮은 값으로 수렴하게 하였다. 이로 인해 목표지점을 찾아가는 경로와 Jacobian condition number가 낮은 경로를 동시에 얻을 수 있다.

4.5 Hyperparameters

Hyperparameters는 DDPG 알고리즘 framework의 구성을 조절하는 역할을 한다. 이는 learning rate η, discount factor γ, update factor τ 등을 포함하여 [Table 1]과 같다.

[Table 1] Hyperparameters

Parameters	Value
Learning rate η_{actor}	1×10^{-4}
Learning rate η_{critic}	1×10^{-4}
Noise type D	Ornstein Uhlenbeck
Batch size m	2^4
Discount factor γ	0.9999
Update factor τ_{actor}	9×10^{-4}
Update factor τ_{critic}	1×10^{-4}

5. Experiment & Simulation Environment

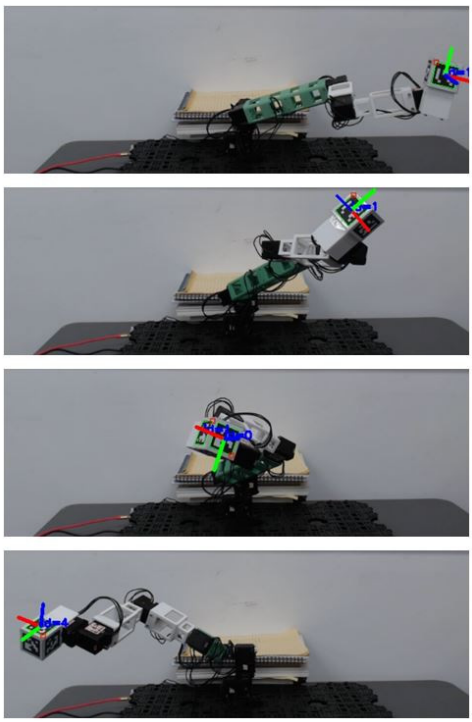
본 논문에서 제안하는 low resolution, low stiffness에서 발생한 비선형적인 오류를 가지는 저가형 redundant manipulator에 대해 end-effector가 선형 궤적으로부터 작은 오차를 가지는 경로를 실험하기 위하여 가상 환경과 실제 환경을 구성하였다.

5.1 Actual Environment

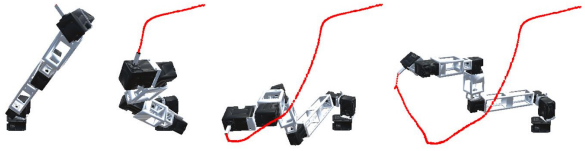
실제 환경에서 저가형 redundant manipulator end-effector의 위치를 추종하는 SLAM (Simultaneous Localization and Mapping) 기술이 필요하다. 이는 cameras, laser radar, optical sensor, IMU (Inertial Measurement Unit) 등을 이용한 다양한 방법이 있다[20,21]. 하지만, manipulator의 workspace 안에서 manipulator end-effector의 위치를 측정하기 힘들다. 따라서 본 논문에서는 [Fig. 4]와 같이 Aruco Markers를 기반으로 manipulator end-effector의 위치를 추종과 하드웨어를 구축하였으며[22], 가상환경에서 실험하고 나오는 joint 변화량만을 이용하여 진행하였다. Aruco Markers는 카메라 기준 x, y, z 순으로 0.25 mm, 0.31 mm, 0.51 mm의 오차가 보였으나, 단안 카메라의 한계로 인하여 먼 거리에서 z 축은 불안정한 결과를 보였다. 또한, 하드웨어는 link의 비선형적인 뒤틀림이 포함되어 있으며, $0.879[\mathrm{deg}/pulse]$의 오차를 허용하여 저가형 redundant manipulator를 구현하였다.

5.2 Simulation Environment

본 논문에서 가상환경을 [Fig. 5]와 같이 Unity에서 구현하였다. 이는 중력, 가속도, 외력, 강도 등을 실제 환경과 유사한 환경을 이끌 수 있다는 장점이 있다. 저가형 redundant manipulator의 낮은 정밀도를 가지는 encoder을 구현하기 위하여



[Fig. 4] Actual environment simulation with Aruco Markers

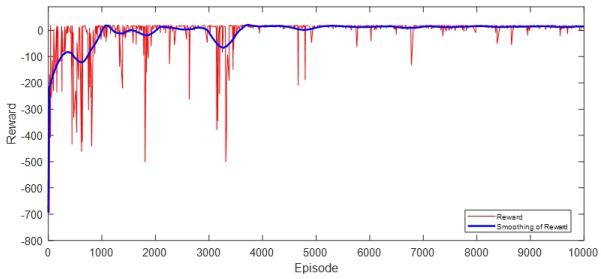


[Fig. 5] Simulation environment

joint마다 $\pm 0.5[\mathrm{deg}/pulse]$의 오차를 허용하였으며, 제어 주기마다 end-effector의 위치를 추종하게 하였다.

6. Experimental Results

본 논문에서 제안하는 low resolution, low stiffness에서 발생한 비선형적인 오류를 가지는 저가형 redundant manipulator에 대해 end-effector가 선형 궤적으로부터 작은 오차를 가지는 경로를 생성하는 DDPG 알고리즘을 학습하기 위하여 10,000 episode 동안 학습을 진행하였으며, 한 episode 당 1,000 step 동안 움직이도록 설정하였다. 목표지점은 한 episode마다 manipulator workspace 안에서 무작위로 위치를 변경하였다. [Fig. 6]는 각 episode에서 얻은 reward의 합을 보여주며 약 2,000 episode 이전에도 높은 reward를 받는 것을 확인 할 수 있다. 이는 식 (18)을 이용하여 ε -greedy policy 방식과 유사한 목표 허용 거리를 줄이는 방법을 사용하였기 때문에 빠른 학습률을



[Fig. 6] 10,000 episodes learning result

보인다. 학습은 10,000번의 episode를 학습하는 데 약 9시간이 소요되었으며, 수렴 양상을 보이는 episode 2,000번대에 진입하는 시간은 약 2시간 반이 소요되었다.

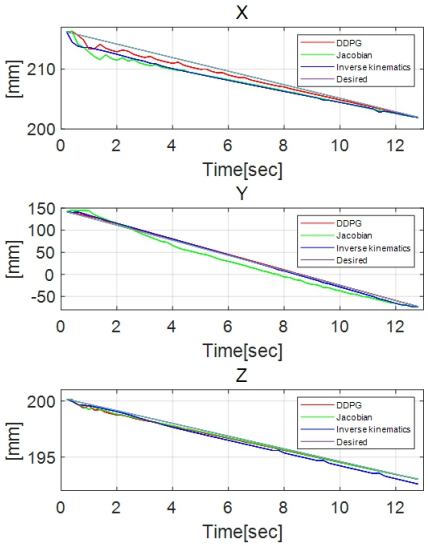
DDPG 학습을 통해 생성된 경로를 검증하기 위하여 가상 환경과 실제 환경에서 Jacobian을 이용한 경로, Newton-Raphson method inverse kinematic을 이용한 경로를 비교 실험을 진행하였다.

6.1 Comparison of the three results in Simulation environment

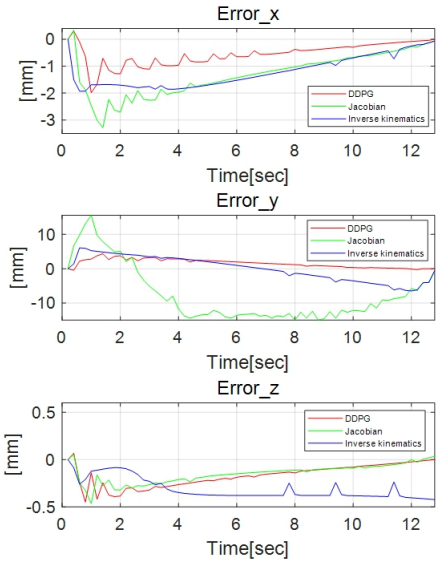
3가지의 경로에 대한 모의실험은 가상환경을 통해 다양한 시작점과 목표지점을 변경해가며 50번의 실험을 진행하였다. 또한, 제어 주기마다 end-effector의 위치를 추종하여 결과를 확인하였다. 가상환경에는 $\pm 0.5[\mathrm{deg}/pulse]$의 오차가 각 joint에 있으며, manipulator의 link 부분의 비선형적인 뒤틀림을 포함하지 않는다. [Fig. 7]과 [Fig. 8]은 50번의 임의의 시작점과 임의의 목표 위치를 바꾸어가며 실험한 결과 중 오차가 가장 작은 end-effector가 $[216.23, 142.02, 200.16]$에서 $[200.96, -72.68, 193]$로 이동한 결과를 보여주며, [Fig. 7]은 Euclidean space에서 end-effector 경로 좌표, [Fig. 8]은 desired 값과 오차를 보여준다. [Table 2]은 50번의 가상환경 실험에서 desired 값과의 분산 평균을 나타낸다.

50번의 실험을 통한 각 축에 대하여 상대 최대 오차의 평균은 Jacobian 경로에서 $[3.4652, 15.8956, 0.5662]$, inverse kinematics 경로에서 $[2.0189, 6.3216, 0.6213]$가 나왔으며, DDPG 경로는 $[2.0023, 4.5632, 0.6512]$나왔다. 또한, 경로 중 가장 큰 Euclidean distance는 Jacobian 경로에서 $16.2340mm$, inverse kinematics 경로에서 $6.6823mm$가 나왔으며, DDPG 경로는 $4.8321mm$가 나왔다.

위의 실험 결과를 통해 DDPG는 Jacobian과 inverse kinematics를 이용한 경로보다 $\pm 0.5[\mathrm{deg}/pulse]$의 joint 오차 안에서 선형 궤적으로부터 작은 오차를 보였다. 이를 통해 가상환경에서 DDPG를 이용하여 생성된 최적 경로에 대한 적합성을 증명하였다.



[Fig. 7] Comparison of the three results in simulation environment



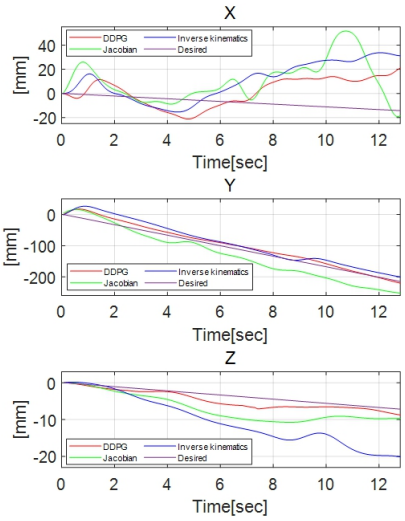
[Fig. 8] Comparison of the three results error in simulation environment

[Table 2] Dispersion in simulation environment

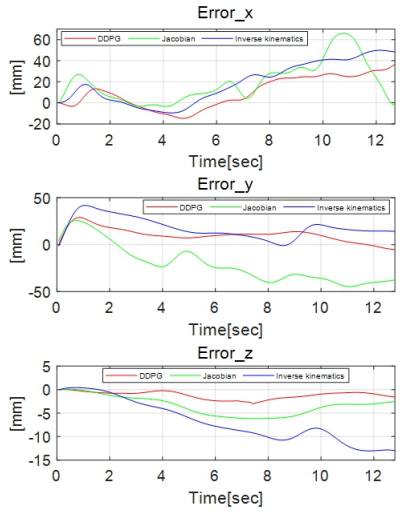
	σ_X	σ_Y	σ_Z	σ_{Total}
DDPG	0.5862	4.0234	0.2364	4.846
Jacobian	2.4656	112.6543	0.9322	116.0521
Inverse kinematics	2.1658	13.0341	0.5465	15.7463

6.2 Comparison of the three results in actual environment

3가지의 경로에 대한 모의실험은 실제 환경을 통해 가상환경에서 실험 후 가장 작은 오차를 가지는 joint 변화량만을 이용하여 진행했으며, end-effector를 추종하기 위하여 단안 카메라를 사용하였다. Calibration과 Aruco Markers를 통해 manipulation 기준 Y, Z의 값은 안정성 있게 받아올 수 있었지만, 단안 카메라의 한계로 인하여 X는 불안정하므로 low pass filter를 적용하여 고주파 영역을 제거하여 [Fig. 9]와 [Fig. 10]에서만 값을 보여준다. 실제 환경에는 joint마다 $0.879[\mathrm{deg}/pulse]$의 오차가 허용되며, 각 link는 비선형적인 뒤틀림이 포함되어 있다.



[Fig. 9] Comparison of the three results in actual environment



[Fig. 10] Comparison of the three results error in actual environment

[Fig. 9]은 가상 환경과 동일하게 $[216.23, 142.02, 200.16]$ 시작하여 $[200.96, -72.68, 193]$의 목표로 이동 때 Euclidean space에서 end-effector 변화량을 보여주며, [Fig. 10]는 실제 환경에서 desired 값과의 오차를 보여준다. 또한, [Table 3]는 실제 환경에서 desired 값과의 분산을 보여준다.

[Table 3] Dispersion in actual environment

	σ_Y	σ_Z	σ_{Total}
DDPG	161.9971	2.0287	164.0258
Jacobian	836.8058	15.6875	852.4933
Inverse kinematics	440.5068	65.7323	506.2391

각 축에 대하여 상대 최대 오차는 Jacobian 경로에서 $[44.6687, 6.1556]$, inverse kinematics 경로에서 $[41.7764, 13.0375]$가 나왔으며, DDPG 경로는 $[28.8696, 2.9945]$나왔다. 또한, 경로 중 가장 큰 Euclidean distance는 Jacobian, inverse kinematics, DDPG 순으로 $41.7764mm$, $40.4587mm$, $28.8737mm$가 나왔다.

위의 실험 결과를 통해 DDPG 경로는 Jacobian과 inverse kinematics를 이용한 경로보다 $0.879[\mathrm{deg}/pulse]$의 허용 오차, link의 비선형적인 뒤틀림을 가지는 redundant manipulator에서 선형 궤적으로부터 작은 오차를 보였다. 이를 통해 실제 환경에서 DDPG를 이용하여 생성된 최적 경로에 대한 적합성을 증명하였다.

7. Conclusion

본 논문에서는 낮은 비용으로 인한 low resolution, low stiffness에서 발생하는 비선형적인 오류를 포함한 redundant manipulator에 대하여 end-effector가 선형 궤적으로부터 작은 오차를 가지는 최적 경로를 DDPG (Deep Deterministic Policy Gradient) 알고리즘을 사용하여 생성하는 방법을 제안하였다.

Redundant manipulator는 하드웨어의 한계점이 발생하지 않는 구조로 설계하여 강화 학습 시 모터의 한계점을 지정하지 않고 manipulator workspace를 넓게 사용할 수 있게 하였다. DDPG는 manipulator의 최적 경로를 생성하기 위하여, Markov decision process를 기반하여 문제를 변환하였으며, Reward를 Jacobian condition number와 목표지점과의 Euclidean distance를 이용하여 설계하였다. 이는 목표지점을 찾아가는 경로와 Jacobian condition number가 낮은 경로를 동시에 학습할 수 있게 한다.

강화 학습은 manipulator의 자유도가 증가할수록 높은 차원의 space 안에서 성공에 대한 빈도가 낮게 되는 문제점을 ε-greedy policy와 유사하게 목표 허용 거리를 생성하여 해결하였으며, 이에 따라 빠른 학습률을 보인 것을 확인하였다. 또한, DDPG 기반 최적 경로를 증명하기 위하여 Jacobian과 Newton-Raphson method inverse kinematic을 이용한 경로를 가상 환경과 실제 환경에서 비교하였다. 여기서 trajectory planning은 Jacobian과 inverse kinematics가 동일하게 진행되지만, Jacobian의 경우 joint 영역의 이동 후 forward kinematics를 통해 오차를 확인하는 과정이 없고 manipulator의 low resolution, low stiffness와 같은 비선형적인 오류를 고려하지 않아 큰 오차를 보이는 것이며, inverse kinematics의 경우 forward kinematics를 통해 오차를 줄이는 과정을 반복하여 계산 오차를 가질 수 있으나 manipulator의 low resolution, low stiffness와 같은 비선형적인 오류를 고려하지 못함으로 오차를 보인다. 하지만 DDPG를 이용한 joint 영역의 경로는 위의 상황을 고려하였기 때문에 Jacobian과 inverse kinematics보다 좋은 결과를 보였다.

본 연구는 직선운동에 대하여 우선으로 강화 학습을 활용한 오차 해소에 집중하였으며, 후속 연구로서 다양한 경로에 대한 일반화를 계획 중이다. 따라서, 현재로서는 여러개의 way point를 활용하여 다양한 경로에 대한 움직임을 구현하고 있다.

References

[1] M. M. Fateh and H. Farhangfard, “On the Transforming of Control Space by Manipulator Jacobian,” *Institute of Control, Robotics and Systems*, vol. 6, no.1, pp. 101-108, 2008, [Online], https://www.koreascience.or.kr/article/JAKO200809906440883.pdf.

[2] D. L. Pieper, “The Kinematics of Manipulators Under Computer Control,” *Stanford University*, 1968, [Online], https://www.proquest.com.

[3] M. A. Ali, H. Andy Park and C. S. G. Lee, “Closed-form inverse kinematic joint solution for humanoid robots,” *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, DOI:10.1109/IROS.2010.5649842.

[4] D. Jang and S. Yoo, “Integrated System of Mobile Manipulator with Speech Recognition and Deep Learning-based Object Detection,” *The Journal of Korea Robotics Society*, vol. 16, no. 3, pp. 270-275, Sept., 2021, DOI:10.7746/jkros.2021.16.3.270.

[5] D. I. Park, C. H. Park, D. H. Kim, and J. H. Kyung, “Analysis and Design of the Dual Arm Manipulator for Rescue Robot,” *The Journal of Korea Robotics Society*, vol. 11, no. 4, pp. 235-241, Dec., 2016, DOI:10.7746/jkros.2016.11.4.235.

[6] Z. Bingul, H. M. Ertunc, and C. Oysu, “Applying Neural Network to Inverse Kinematic Problem for 6R Robot Manipulator with Offset Wrist,” *Adaptive and Natural Computing Algorithms*, pp. 112-115, 2005, DOI: 10.1007/3-211-27389-1_27.

[7] Y.-H. Kim, H. Kang, and H.-T. Jeon, “Planning a minimum time path for robot manipulator using Genetic Algorithm,” *The Institute of Electronics and Information Engineers*, pp. 698-702, 1992, [Online], https://www.koreascience.or.kr/article/CFKO199211919700824.pdf.

[8] E. Prianto, M. S. Kim, J.-H. Park, J.-H. Bae, and J.-S. Kim, “Path Planning for Multi-Arm Manipulators Using Deep Reinforcement Learning: Soft Actor–Critic with Hindsight Experience Replay,” *Sensors 2020*, vol. 20, no. 20, 2020, DOI: 10.3390/s20205911.

[9] J.-Y. Moon, J.-H. Moon, and S.-H. Bae, "Control for Manipulator of an Underwater Robot Using Meta Reinforcement Learning," *The Journal of the Korea institute of electronic communication sciences*, vol. 16, no. 1, pp. 95-100, 2021, DOI: 10.13067/JKIECS.2021.16.1.95.

[10] A. Iriondo, E. Lazkano, L. Susperregi, J. Urain, A. Fernandez, and J. Molina, "Pick and Place Operations in Logistics Using a Mobile Manipulator Controlled with Deep Reinforcement Learning," *Applied Sciences*, vol. 9, no. 2, 2019, DOI:10.3390/app9020348.

[11] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan, "Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 132-146, Apr., 2012, DOI:10.1016/j.rcim.2011.07.004.

[12] J. D. Norwood, "A neural network approach to the redundant robot inverse kinematic problem in the presence of obstacles," PhD thesis. Rice University, 1991, [Online], https://www.proquest.com.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971 [cs.LG]*, 2015, [Online], https://arxiv.org/abs/1509.02971.

[14] J. Lee, K. Kim, Y. Kim, and J. Lee, "Singularity Avoidance Path Planning on Cooperative Task of Dual Manipulator Using DDPG Algorithm," *The Journal of Korea Robotics Society*, vol. 16, no. 2, pp. 137-146, Jun., 2021, DOI: 10.7746/jkros.2021.16.2.137.

[15] J. P. Merlet, "Jacobian, Manipulability, Condition Number, and Accuracy of Parallel Robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199-206, 2005, DOI: 10.1115/1.2121740.

[16] M. Tokic, "Adaptive ε-Greedy Exploration in Reinforcement Learning Based on Value Differences," *Annual Conference on Artificial Intelligence*, pp. 203-210, 2010, DOI: 10.1007/978-3-642-16111-7_23.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv:1312.5602 [cs.LG]*, 2013, [Online], https://arxiv.org/abs/1312.5602.

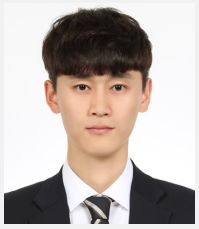
[18] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *31st International Conference on Machine Learning*, vol. 32, no.1 pp. 387-395, 2014, [Online], http://proceedings.mlr.press/v32/silver14.html.

[19] C. C. White III and D. J. White, "Markov decision processes," *European Journal of Operational Research*, vol. 39, no. 1, pp. 1-16, 1989, DOI: 10.1016/0377-2217(89)90348-2.

[20] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," *Scandinavian conference on Image analysis*, pp. 363-370. 2003, DOI: 10.1007/3-540-45103-X_50.

[21] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," *1999 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1322-1328, 1999, DOI: 10.1109/ROBOT.1999.772544.

[22] A. Poroykov, P. Kalugin, S. Shitov, and I. Lapitskaya, "Modeling ArUco Markers Images for Accuracy Analysis of Their 3D Pose Estimation," *30th International Conference on Computer Graphics and Machine Vision*, vol. 2, 2020, DOI: 10.51130/graphicon-2020-2-4-14.

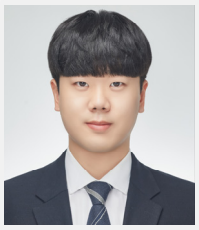


이 승 현

2019 동아대학교 전자공학과 학사

2020~현재 부산대학교 전기전자공학과 석사과정

관심분야: 자동 로봇 제어, 강화 학습, 시스템 설계



진 성 호

2019 경남과학기술대학교 메카트로닉스공학과 학사

2020~현재 부산대학교 전기전자공학과 석사과정

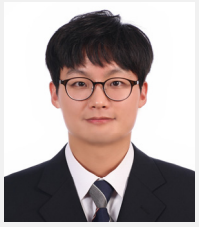
관심분야: 매니퓰레이터 제어, 영상 인식

황 성 현

2017 경남과학기술대학교 메카트로닉스공학과 학사

2020~현재 부산대학교 전기전자공학과 석사과정

관심분야: 선형시스템, 로봇공학, 마이크로프로세서 응용 및 제어



이 인 호

2009 한국과학기술원 기계공학과 학사

2011 한국과학기술원 기계공학과 석사

2016 한국과학기술원 기계공학과 박사

2020~현재 부산대학교 전자공학과 조교수

관심분야: Robotics, 기계 시스템 자동화, 시스템 모델링