Regular paper
Journal of the KIECS. pp. 95-100, vol. 16, no. 1, Feb. 28. 2021, t. 105, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2021.16.1.95

메타강화학습을 이용한 수중로봇 매니퓰레이터 제어

문지윤* · 문장혁** · 배성훈***

Control for Manipulator of an Underwater Robot Using Meta Reinforcement Learning

Ji-Youn Moon* · Jang-Hyuk Moon** · Sung-Hoon Bae***

요 약

본 논문에서는 수중 건설 로봇을 제어하기 위한 모델 기반 메타 강화 학습 방법을 제안한다. 모델 기반 메타 강화 학습은 실제 응용 프로그램의 최근 경험을 사용하여 모델을 빠르게 업데이트한다. 다음으로, 대상 위치에 도달하기 위해 매니퓰레이터의 제어 입력을 계산하는 모델 예측 제어로 모델을 전송한다. MuJoCo 및 Gazebo를 사용하여 모델 기반 메타 강화 학습을 위한 시뮬레이션 환경을 구축하였으며 수중 건설 로봇의 실제 제어 환경에서의 모델 불확실성을 포함하여 제안한 방법을 검증하였다.

ABSTRACT

This paper introduces model-based meta reinforcement learning as a control for the manipulator of an underwater construction robot. Model-based meta reinforcement learning updates the model fast using recent experience in a real application and transfers the model to model predictive control which computes control inputs of the manipulator to reach the target position. The simulation environment for model-based meta reinforcement learning is established using MuJoCo and Gazebo. The real environment of manipulator control for underwater construction robot is set to deal with model uncertainties.

키워드

Underwater construction robot, Manipulator control, Reinforcement learning, Meta learning, Model based reinforcement learning
수중 건설 로봇, 매니퓰레이터 제어, 강화 학습, 메타 학습, 모델 기반 강화 학습

Ⅰ. 서 론

기술의 발전에 따라 나날이 위험이 높은 지상에서의 작업들은 로봇들로 대체되어 가고 있으나 수중과 같은 특수한 환경에서 요구하는 기술은 수준이 높아 대체율이 현재 낮은 실정이다. 수중건설 작업의 경우 오랜 시간 작업을 진행할 시에 잠수병의 위험이 있으며 수심이 깊거나 강한 해류가 발생할 시에 인명사고가 발생하기도 한다. 특히 해저면 정리, 사석 고르기 등의 잠수사가 깊은 바다에 들어가 손으로 오랜 시간 작업해야 되는 작업은 위험성이 높다. 이에 수중 환경

* 조선대학교 전자공학부 (jymoon@chosun.ac.kr)
** 한양대학교 경영학부 (wisewalkerj@gmail.com)
*** 교신저자 : 레드원테크놀러지 (bsh@urc.kr}
• 접 수 일 : 2021. 01. 22
• 수정완료일 : 2021. 02. 04
• 게재확정일 : 2021. 02. 17
• Received : Jan. 22. 2021, Revised : Feb. 04, 2021, Accepted : Feb. 17, 2021
• Corresponding Author : Sung-Hoon Bae
REDONE technologies
Email : bsh@urc.kr

을 고려한 강인한 매니퓰레이터 제어 기술은 필수적이다.

수중의 환경은 로봇에게 시시각각 변화할 수 있는 압력, 해류 등의 외부 요인 또는 자신의 움직임에 영향을 받아 위치가 변하는 등의 내부 요인 등에 강인한 제어 모델이 필요하다. 하지만, 기존의 강화 학습 모델로는 지속적인 변화에 맞춰 적절한 대응이 어렵다. 모든 경우의 수를 상정한 모델을 구축하는 데에 있어서 한계가 존재하며 또 그에 맞는 방대한 양의 데이터를 제공하기가 제한적이기 때문이다. 즉, 강화학습을 통해 학습을 진행한 로봇은 작은 변화 에서도 작동을 실패할 확률이 높다. 반면에, 모델기반 메타강화학습은 학습의 방법을 학습함으로써 다양한 환경 변화에 유연하게 대처할 수 있다는 강점을 갖는다. 예를 들어, 키보드의 자판의 위치가 제조사마다 차이가 있어도 금방 키보드를 이용하는 법에 대한 이전의 학습의 경험을 통해 새 키보드에 빠르게 적응할 수 있도록 한다. 이와 같은 학습 방법은 사람이 학습하는 과정과 유사하며 최종적으로 사람이 하는 지적인 일마저 대체할 수 있는 AGI(Artificial General Intelligence)를 목표로 하고 있다 [1].

2절에서는 모델기반 강화 학습을 소개하고 3절에서는 매니퓰레이터 제어를 위한 시뮬레이션 환경에 대한 기술을 한 이후에 4절에서는 실험 결과에 대해서 소개한다.

II. 모델기반 메타강화학습

2.1 모델기반 강화학습

강화학습 문제는 $(S, A, p, r, \gamma, \rho_0)$로 정의된 마르코프 결정 과정으로 표현된다 [2]. S는 상태변수 집합, A는 행동의 집합, $p(s'|s,a)$는 상태변이분포, $r : S\times A\rightarrow R$은 보상 함수, γ는 감가율, $\rho_0 : S\rightarrow R+$는 초기 상태분포이다. 궤적은$\tau(i,j)$ $(s_i, a_i, \cdots, s_j, a_j, s_{j+1})$ 로 표현되고, 강화학습은 보상의 기댓값을 최대화하는 행동을 결정하는 정책 $\pi : S\rightarrow A$를 찾는 것을 목표로 한다. 비모델(model free) 강화학습은 상태변이분포 없이 로봇의 실행 데이터만을 이용하여 가치함수와 정책을 업데이트하여 강화학습의 목표인 최적 정책을 찾고, 모델기반(model based) 강화학습은 동역학 모델인 상태변이분포 $p(s'|s,a)$를 이용하여 학습 데이터 궤적 (s,a,s')을 생성하고 이 궤적을 이용하여 가치함수와 정책을 업데이트하여 강화학습의 목표인 최적 정책을 찾는다 [3].

2.2 메타학습

메타학습은 학습 알고리즘을 학습하는 기법으로 미리 여러 메타학습작업(meta-training task)에 대한 데이터로 학습한 후 새로운 메타시험작업(meta-test task)에 빨리 적응할 수 있도록 한다 [4]. 이때, 메타학습작업과 메타시험작업은 전혀 상관이 없는 작업이 아니라 같은 작업분포 $\rho(T)$에 속하는 연관성이 있어야 한다. 메타학습의 목표는 손실 함수 L_T를 최소화하는 학습법 $u_\psi(D_T^{tr}, \theta)$를 찾는 것에 있다.

MAML(: Model-agnostic meta-learning) 기법은 최소 단계의 경사하강(gradient descent)으로 새로운 작업에 대하여 빠르게 학습할 수 있는 메타학습이다 [5]. MAML 식은 아래와 같다.

$$\begin{aligned} &\min_{\theta,\psi} \mathrm{E}_{\mathrm{T}\sim\rho(\mathrm{T})}[\mathrm{L}(\mathrm{D}^{\mathrm{test}_{\mathrm{T}}}\theta')] \\ &\text{s.t. } \theta' = u_\psi(D_T^{tr}, \theta) \end{aligned} \tag{1}$$

최적 파라미터 θ^*, ψ^*를 구하면 학습법 $u_\psi(D^{tr}{}_T\theta) = \theta - \alpha \nabla_\theta L(D^{tr}{}_T, \theta)$은 메타시험작업을 학습하는데 사용된다 [6].

2.3 온라인 모델 메타학습

메타학습은 학습된 최적 파라미터 θ^*, ψ^*를 이용하여 새로운 작업을 학습하였다. 메타강화학습은 최대가능도 (maximum likelihood) 메타목적함수 $\hat{p}_{\theta'}(s'|s,a)$에 대하여 θ, ψ를 최적화하는 문제가 되며 식 2와 같다.

$$\begin{aligned} &\min_{\theta,\psi} \mathrm{E}_{\tau_\epsilon(\mathrm{t-M,t+K})\sim \mathrm{D}}[\mathrm{L}(\tau_\epsilon(\mathrm{t,t+K}), \\ &\text{s.t.} \\ &\theta'_\epsilon = u_\psi(\tau_\epsilon(t-M, t-1), \theta) \end{aligned} \tag{2}$$

$\tau_\epsilon(t-M, t+K) \sim D$ 는 이전 데이터 궤적을 나타

내고, 손실 함수는 음의 로그 가능도(log likelihood) 함수로 식3과 같이 나타낼 수 있다.

$$L(\tau_\epsilon(t,t+K),\theta'_\epsilon) = -\frac{1}{K}\sum_{k=t}^{t+K}\log\hat{p}_{\theta'_\epsilon}(s_{k+1}|s_k,a_k) \quad (3)$$

과거 M개 데이터는 θ를 θ'로 학습시키는데 사용되고, 손실 함수는 미래 K개의 데이터로 계산된다.

메타 목적함수는 학습법 u_ψ와 θ의 경사하강 스텝으로 최적화되고, M개의 데이터를 이용하여 시험단계에서 빠르게 모델을 업데이트할 수 있다. MAML을 이용하여 온라인 모델 메타 학습법을 식4와 같이 표현할 수 있다.

$$\theta' = \theta + \psi\nabla_\theta\frac{1}{M}\sum_{m=t-M}^{t-1}\log\hat{p}_{\theta_\epsilon}(s_{m+1}|s_m,a_m) \quad (4)$$

메타학습된 θ^*,ψ^*와 새로 습득한 데이터를 이용하여 모델 파라미터를 $\theta'_* = u_{\psi_*}(\tau(t-M,t),\theta_*)$ 갱신하면, 현재 동역학 특성을 포함한 모델 $\hat{p}_{\theta'_*}$를 얻게 되고, 이 모델을 보상함수 r과 수평선 H와 함께 매니퓰레이터 MPC 제어(Model predictive control)에 사용한다 [7]. 학습이 시작되면 정책(policy)가 아직 정해진 상태가 아니므로 임의의 정책을 적용하여 궤적 (s,a,s')를 생성한다. 생성된 궤적 (s,a,s') 데이터를 이용하여 모델의 상태변이분포 $\hat{p}_{\theta_\epsilon}(s_{t+1}|s_t,a_t)$를 메타학습하여 θ^*를 구하고, Adaptive learner 블록에서 최근 데이터를 이용하여 $\theta'_* = u_{\psi_*}(\tau(t-M,t),\theta_*)$를 한 번 더 갱신하여 현재 모델과 환경에 맞게 업데이트한다 [8]. 이 모델을 이용하여 모델예측제어 기법으로 매 시간 수평선 H 안에서 최적의 제어를 도출한다 [9]. MPC는 매 시간 측정되는 상태변수를 이용하므로 동역학 모델의 불확실성에 의하여 축적되는 오차를 방지하며 매니퓰레이터의 제어 입력을 업데이트 할 수 있다.

III. 매니퓰레이터 제어

강화학습의 시간적, 비용적 한계로 인해 모델기반 메타강화학습을 매니퓰레이터에 적용하기 위해 에이전트(Agent) 제어 시뮬레이션의 구현이 필요하다. 에이전트 제어 시뮬레이션은 두 가지 방법으로 구현이 가능하며 첫 번째는 Python을 이용하여 MuJoCo에 직접 제어 명령을 내리도록 구현하는 것이고, 두 번째는 Python에서 내리는 명령을 TCP/IP를 이용하여 ROS를 제어하도록 구현하는 것이다. 본 논문에서는 실제 환경의 매니퓰레이터를 모델기반 메타강화학습으로 제어하기 위해 실제 환경의 매니퓰레이터와 연동이 가능한 ROS Gazebo와 Python으로 구현한 모델기반 메타강화학습을 소켓통신을 이용하여 구현한다.

3.1 MuJoCo

MuJoCco(: Multi-Joint dynamics Contact)는 로봇공학, 생체 역학, 그래픽 및 애니메이션 등 빠르고 정확한 시뮬레이션이 필요한 분야의 연구 및 개발을 목표로 하는 물리 엔진이며, MuJoCo를 사용하면 최적제어, 상태 추정, 시스템 식별 및 자동화된 메커니즘 설계 등 계산 집약적인 기술을 확장하고, 동적 시스템에 적용할 수 있다.

강화학습은 환경과의 상호작용을 통해 획득한 데이터로 정책을 유도 유도하기 때문에 수많은 step을 진행해야 한다. 이로 인해 실제 시스템에서는 데이터를 얻기 위해 필요한 시간과 비용이 매우 크고, 실제 시스템을 파손시킬 수 있다. 이러한 이유로 바로 실제 환경에서 학습을 진행하기에는 강화학습의 한계가 있기 때문에 본 논문에서는 실제 매니퓰레이터에 모델기반 메타강화학습을 적용하기 전에 MuJoCo 환경에서 구현한 7-DOF 매니퓰레이터에 모델기반 메타강화학습을 적용하여 성능과 안정성을 검증한다.

MuJoCo 환경에서 강화학습을 적용하기 위하여 OpenAI에서 제공하는 Python API인 mujoco-py를 사용한다. mujoco-py의 함수는 XML으로 저장된 매니퓰레이터 3D Model을 불러오기와 position과 velocity로 행동(Action)을 제어, 에이전트(Agent)의 상태(State)를 수집, 시뮬레이션 reset, 시뮬레이션을 진행하는 step 등이 구현되어 있다. 이러한 함수들을 이용하여 MuJoCo에 구현된 7-DOF 매니퓰레이터 환

경에서 Target과 End Effect가 근접할 수 있도록 7개의 Joint(shoulder_pan, shoulder_lift, upper_arm_roll, elbow_flex, forearm_roll, wrist_roll, wrist_roll)를 관절값으로 제어하는 모델기반 메타강화학습을 적용한다.

3.2 ROS 시뮬레이터

ROS(: Robot Operating System)는 로봇용 오픈소스 메타 운영체제로, 일반 운영체제에서 제공하는 하드웨어 추상화, 저수준 기기 제어, 빈번히 사용되는 기능들이 구현되어 있으며, 프로세스간 메시지 전달, 패키지 관리 기능을 제공한다. 본 논문에서는 ROS의 응용 프로그램 중에서 Gazebo를 이용하여 3차원 시뮬레이션을 구현하였다. Gazebo는 ROS의 3차원 시뮬레이션을 위한 로봇, 센서, 환경 모델 등을 지원하고 물리 엔진을 탑재하여 실제와 근사한 결과를 얻을 수 있는 3차원 시뮬레이터이다.

ROS Gazebo 시뮬레이션 또한 3.1절에서 소개한 강화학습의 한계인 수 많은 step을 진행하는데 필요한 시간과 비용을 최소화 하기 위해 사용되며 모델기반 메타강화학습의 성능과 안정성을 확인 후 실제 환경에 적용하도록 한다.

ROS에서 매니퓰레이터 제어는 관절(Joint)값으로 제어를 하고 있지만 실제 매니퓰레이터의 제어는 내부의 절대(absolute) 엔코더(Encoder)값으로 제어하고 있다. 따라서 실제 환경과 ROS를 연동하여 매니퓰레이터를 제어하기 위해서 실제 엔코더값과 ROS의 관절값을 변환하도록 변환 함수를 구현했다. 이를 이용하여 OpenAI에서 제공하는 Python API인 mujoco-py와 MuJoCo로 구현한 환경에 적용한 모델기반 메타강화학습을 소켓통신을 이용하여 ROS와 행동과 상태를 송수신하여 에이전트인 실제 환경의 로봇에 적용할 수 있도록 한다.

3.2. 실제 환경

실제 환경 ROV의 매니퓰레이터는 절대(absolute) 엔코더(Encoder)값으로 제어한다. 따라서 MuJoCo, ROS 등 시뮬레이션과 연동하기 위해서는 엔코더(Encoder)값과 관절(Joint)값을 변환하는 과정이 필요하기 때문에 관절(Joint)값에 따른 엔코더(Encoder)값을 구하여 변환함수를 적용한다.

실제 환경의 매니퓰레이터는 변환함수를 통해 ROS Gazebo 시뮬레이션과 연동되어 있으며 ROS Gazebo에서 제어가 가능하다. 따라서 실제 환경의 매니퓰레이터에서 바로 적용하지 않고 ROS 시뮬레이션에서 먼저 실제와 동일하게 엔코더 범위를 설정 후 학습을 진행하여 학습과정 중에 시스템이 파손되는 경우를 방지한다.

ROS에서 학습이 완료 된 후 실제 환경에서 매니퓰레이터를 사용하기 위해 학습이 완료된 모델을 불러와서 실제 환경의 매니퓰레이터에 모델기반 메타강화학습을 적용하여 학습하도록 한다 [10].

Ⅳ. 시뮬레이션 및 검토

본 논문에서는 시간과 비용, 그리고 시스템의 파손을 우려하여 실제 매니퓰레이터와 연동되는 ROS Gazebo 7-DOF 매니퓰레이터 시뮬레이션에 모델기반 메타강화학습을 적용하였으며, 부족한 컴퓨터 파워로 인한 실시간성 문제를 해결하기 위해 시뮬레이션 컴퓨터, 모델기반 강화학습 컴퓨터 총 2대의 컴퓨터를 사용하여 분산 시스템을 구성하였다.

시뮬레이션과 강화학습에 필요한 연산량이 매우 많기 때문에 컴퓨터 파워가 부족하여 실시간성을 잃을 우려가 있다. 본 논문에서는 매니퓰레이터와 소캣통신을 통해 학습을 하기 때문에 실시간성 유지를 위해 분산 시스템을 적용한다. 분산 시스템에 사용된 컴퓨터는 표1과 같다. 시뮬레이션은 ROS Gazebo 시뮬레이션을 사용하였으며 실제 환경의 매니퓰레이터와 연동이 가능하다.

시뮬레이션의 환경은 수중에서 ROV에 장착되어 있는 7-DOF 매니퓰레이터의 End Effector가 Target인 Ball에 근접하도록 학습하는 환경이며, ROV와 Target은 수중에 고정되어 있다. 그리고 원활한 학습을 위해 Target의 Collider를 사용하지 않아서 End Effector는 Target을 통과 할 수 있다. 본 논문에서 사용되는 파라미터는 step, epoch, iteration, candidate, meta_batch_size, adapt_batch_size가 있으며, 학습에 사용되는 reward는 $\sqrt{x^2+y^2+z^2}$, $(x, y, z = End\,Effector - Target)$ 로 정의하였으며 End Effector와 Target의 거리, $(Joint0)^2 + ... +$

$(Joint6)^2$ $(Joint = Action)$ 매니퓰레이터를 제어하는 제어 신호의 크기로 이루어져 있다. 시뮬레이션의 실험을 위해 Agent(ROS Gazebo)와 Env(Step environment)에서 소캣통신을 이용하여 Action과 Observation을 송수신한다.

실험 초기에 ROV와 타겟의 위치를 고정하지 않고 해저면에서 학습을 진행한 경우 매니퓰레이터가 움직이면서 바닥과 타겟에 충돌함에 따라 ROV와 타겟이 밀려나는 현상이 발생하여 그림 1과 같은 리워드 분포를 보였다. 따라서, ROV와 타겟의 위치를 해저면에 올려둔 상태에서 수중에 고정하도록 변경하였다.

ROV와 타겟을 수중에 고정하여 학습을 12시간 진행한 결과 여전히 타겟에 도달하지 못하고 Loss가 수렴하지 않는 현상이 발생하였다. Action Space가 너무 큰 값이여서 수렴하지 않는다고 판단하여 Action Space를 (-1 ~ 1)의 값에서 (-0.02 ~ 0.02)로 변경하여 실험을 진행하였고, Reset 함수를 추가하여 매 에피소드마다 환경이 reset되도록 구현하였다.

Action Space를 (-0.02 ~ 0.02)로 적용 후 학습을 12시간 진행한 결과 Target에 End Effector가 도달하는 모습을 관찰하고 리워드 분포를 보였지만 Loss가 수렴하지 않아서 학습이 제대로 진행되지 않았다고 판단했다. Action Space를 (-0.01 ~ 0.01)로 변경하고, 장시간 학습을 위해 최대 Iteration을 1000으로 설정하여 학습을 다시 진행하였다.

60시간 학습 결과 Reward는 증가했지만 End Effector가 Target에 도달하지 못하고 Loss도 분산하는 결과를 확인했다. 원인 분석 결과 Action이 Discrete하여 Policy gradient인 해당 알고리즘에서 적합하지 않다고 판단하여 Continuous한 Action으로 변환하여 실험을 진행하였다. 12시간 학습 결과 End Effector가 Target에 도달하는 모습과 그림 2과 같은 리워드 분포를 관찰 할 수 있었으며, 그림 3와 같이 Loss가 수렴함을 관찰 할 수 있었다.

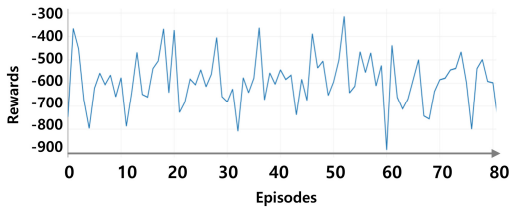

그림 1. ROV를 고정하지 않은 경우 Reward
Fig. 1 ROV Reward (not fixed)

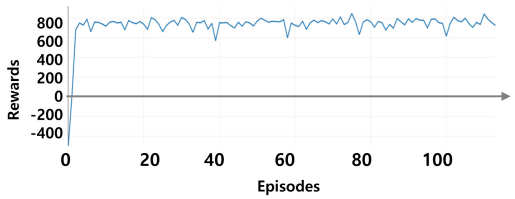

그림 2. Continuos action으로 변경한 경우 reward
Fig. 2 Reward (Continuos action)

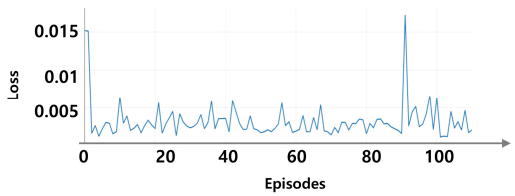

그림 3. Continuos action으로 변경한 Loss
Fig. 3 Loss (Continuos action)

V. 결 론

수중건설로봇의 매니퓰레이터를 제어하기 위하여 모델기반 메타강화학습 기법을 선정하고 이를 적용하기 위한 ROS Gazebo 시뮬레이션 환경과 실제 ROV 매니퓰레이터 환경을 구축하였다. 메타강화학습을 이용하여 모델을 도출함으로써 실제 환경에 적용 시 여러 가지 상황에 빠르게 모델을 갱신할 수 있도록 하였으며, 모델예측제어 기법을 적용하여 모델링의 불확실성을 극복할 수 있도록 하였다. 이를 MuJoCo와 ROS의 응용 프로그램 중에서 Gazebo를 이용하여 3차원 시뮬레이션을 구현하고, 실제 환경에서 수중건설

로봇 매니퓰레이터의 동역학 모션을 Gazebo에서 유사하게 모사하는지 확인하였으며 모델기반 메타강화학습 기법을 ROS Gazebo와 소캣통신을 통해 적용하여 적은 샘플 데이터로 학습이 가능한지, 새로운 환경에 빠르게 적응하는지 확인하였다.

References

[1] A. Nagabandi, I. Clavera, S. Liu, R. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to Adapt in Dynamic, Real-World Environments Through Meta-Reinforcement Learning," *arXiv preprint arXiv:1803.11347*, 2018.

[2] M. Hausknecht and P. Stone, "Deep Recurrent Q-Learning for Partially Observable MDPs," *arXiv preprint arXiv:1507.06527*, 2017.

[3] C. Finn and S. Levine, "Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm," *arXiv preprint arXiv:1710.11622*, 2017.

[4] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," *Int. Conf. on Learning Representations*, 2018.

[5] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *Int. Conf. on Machine Learning*, 2017.

[6] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-Learning in Neural Networks: A Survey," *arXiv preprint arXiv:2004.05439*, 2020.

[7] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. Rehg, B. Boots, and E. Theodorou, "Information theoretic mpc for model-based reinforcement learning," *IEEE Int. Conf. on Robotics and Automation*, 2017.

[8] S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. on Automatic Control*, 1989.

[9] G. Williams, A. Aldrich, and E. Theodorou, "Model Predictive Path Integral Control using Covariance Variable Importance Sampling," *arXiv preprint arXiv:1509.01149*, 2015.

[10] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," *arXiv preprint arXiv:1710.03641*, 2017.

저자 소개

문지윤 (Ji-Youn Moon)

2014년 광운대학교 로봇학부 졸업(공학사)
2020년 서울대학교 대학원 전기정보공학부 졸업(공학박사)
2020년 조선대학교 전자공학부 조교수
※ 관심분야 : 일반인공지능, 인지 로보틱스, 뉴로-심볼릭, 강화 학습, 임베디드 AI 컴퓨팅

문장혁 (Jang-Hyuk Moon)

2021년 한양대학교 경영학부·로봇공학과 졸업(공학사)
※ 관심분야 : 강화학습, 인공지능, 멀티-에이전트 시스템, 다개체 인공지능 의사결정 시스템

배성훈 (Sung-Hoon Bae)

2020년 순천대학교 전자공학과 졸업(공학사)
2021년 레드원테크놀러지
※ 관심분야 : 멀티-에이전트 강화학습, 로보틱스, 자율 공장 자동화 시스템