ISSN 1975-8359 [Print] / ISSN 2287-4364 [Online]
The Transactions of the Korean Institute of Electrical Engineers, vol. 74, no. 1, pp. 118~126, 2025
https://doi.org/10.5370/KIEE.2025.74.1.118

Development of a Reinforcement Learning-Based Control Education Platform Using Python and LW-RCP

Python과 LW-RCP를 이용한 강화학습 기반 제어 교육 플랫폼 개발

Jongbeom Lee . Taegun Lee . Doyoon Ju . Young Sam Lee
이종범* . 이태건* . 주도윤* . 이영삼†

Abstract

This paper proposes a reinforcement learning-based control education platform utilizing Python and light-weight rapid control prototyping (LW-RCP). The platform employs the Sim-to-Real technique, in which neural networks are trained in a Python-based simulation environment and applied to real systems. The trained networks are converted into a format compatible with Matlab/Simulink. The lab-built LW-RCP is used to implement a real-time controller under the Simulink environment by incorporating the converted networks. The proposed platform allows students to easily apply reinforcement learning theory to real systems, contributing to the integration of reinforcement learning control into control curriculum. The effectiveness of the proposed platform is demonstrated by implementing a reinforcement learning controller for the pendubot system. The implemented controller performs the swing-up and transition control and exhibits strong disturbance rejection and recovery properties.

Key Words

Reinforcement learning, Pendubot, Sim-to-Real Learning, LW-RCP

1. 서 론

2016년 이루어진 알파고와 이세돌 간의 바둑 대국은 4차 산업혁명 시대의 시작을 시사하는 중대한 이정표로 평가된다. 이 사건은 인공지능 기술이 단순한 학문적 연구의 영역을 넘어, 사회 전반에 영향을 미칠 수 있는 현실적 응용 기술로 성장했음을 증명하였다[1]. 이는 인공지능이 인간의 행동과 결정 과정을 모방할 수 있는 능력을 갖추었고, 나아가 인간보다 더 효율적으로 문제를 해결할 수 있다는 가능성에 대한 중요한 시사점을 제공한다. 특히, 알파고의 승리는 강화학습(reinforcement learning)이라는 고도화된 기계학습 방법론의 효용성을 전 세계에 입증한 사례로서 주목할 수 있다. 이때 '강화학습'이란, 인공지능 알고리즘이 동적인 환경과의 시행착오적인 상호작용 과정을 통해 스스로 최적의 행동 패턴을 학습하는 기법을 의미한다[2].

강화학습은 제어공학 분야에서도 그 효용성을 인정받아 자율주행과 로보틱스 등 다양한 분야에서 활발히 연구되고 있다. 이러한 연구를 통해 강화학습을 실제 자율주행 시스템에 성공적으로 적용하였으며[3], 사족 보행 로봇의 강건한 자각 보행 구현에도 성공하는 등 새로운 제어 방법론을 제시하고 있다[4]. 이처럼 강화학습은 복잡한 환경에서의 자율성을 향상시키며, 다양한 제어 시스템에서 그 가능성을 입증하고 있다. 강화학습과 제어공학의 융합적인 연구가 두드러진 발전을 보이는 가운데, 이에 발맞추어 많은 교육기관들이 '강화학습' 혹은 제어공학에 지능형 도구를 활용한다는 의미에서 '지능제어' 등의 이름으로 새로운 교육과정을 개발 및 운영 중에 있다[5-7]. 하지만, 대부분의 교육은 이론과 컴퓨터 시뮬레이션에 중점을 두고 있고, 실물 시스템에 강화학습을 적용하는 것은 기술적 한계로 인한 어려움을 겪고 있다. 실물 시스템을 구동하기 위해서는 마이크로컨트롤러를 통해 데이터를 수집하고 제어 신호를 전달해야 한다. 그리고 제어 신호를 연산하는 강화학습 신경망을 마이크로컨트롤러에 구현해야 하지만, 마이크로컨트롤러는 성능과 자원이 제한적이기 때문에 이를 구현하는 것에는 기술적 한계가 존재한다.

이 문제를 해결하려면 가지치기(Pruning), 양자화(Quantization), 지식 증류(Knowledge Distillation)와 같은 인공신경망 경량화 기술들이 요구된다[8]. 하지만, 이러한 기술들은 강화학습을 실물 시스템에 적용하고자 하는 학생들에게는

† Corresponding Author : Dept. of Electrical and Computer Engineering, Inha University Incheon, Republic of Korea.
E-mail : lys@inha.ac.kr https://orcid.org/0000-0003-0665-1464
* Dept. of Electrical and Computer Engineering, Inha University, Incheon, Republic of Korea.
https://orcid.org/0009-0004-8289-8092 https://orcid.org/0009-0007-3107-2735
https://orcid.org/0000-0001-7011-6779
Received : Sep. 20, 2024 **Revised** : Nov. 28, 2024 **Accepted** : Dec. 04, 2024

너무 높은 진입장벽이 될 수 있으며, 교육의 원래 목표에서 벗어나게 될 가능성이 존재한다. 실제로 교육의 본래 목표는 학생들이 추가적인 전문 지식 없이도 강화학습 알고리즘의 기본 원리를 실물 시스템에 적용할 수 있는 능력을 키우는 것이다.

이러한 문제를 해결하기 위해, 본 논문에서는 Python으로 학습된 강화학습 신경망을 실물 시스템에 쉽게 적용할 수 있도록 돕는 강화학습 기반 제어 교육 플랫폼을 제안한다. 이 플랫폼은 Sim-to-Real 기법을 활용해 시뮬레이션 환경에서 학습된 신경망을 실물 시스템에 적용하는 방식을 사용한다. Sim-to-Real 기법은 강화학습 에이전트가 상호작용하는 환경을 시뮬레이션으로 구축하여 신경망을 학습시키는 방법을 의미한다[9].

이렇게 학습된 신경망은 Matlab/Simulink 환경에 호환 가능한 데이터로 변환되어 제어기를 구현하는데 사용된다. Matlab/Simulink에서 구현된 강화학습 기반 제어기는 제어량을 계산하게 되며, 이 제어량을 실물 시스템의 구동기로 전달하고 센서 데이터를 수집하는 역할은 본 논문의 저자들이 속한 연구실에서 직접 개발한 LW-RCP(Light-Weight Rapid Control Prototyping)가 담당한다. LW-RCP는 강화학습 알고리즘을 마이크로컨트롤러보다 빠른 연산 속도와 풍부한 자원을 갖춘 PC에서 실행할 수 있게 도와주며, 인공 신경망 경량화와 같은 기술 없이도 기존의 시뮬레이션 기반 교육 방식을 활용할 수 있게 한다. 결과적으로 본 논문에서 제안하는 플랫폼은 강화학습 기반 제어기를 실물 시스템에 적용하는데 발생하는 진입장벽을 효과적으로 낮출 수 있다. 이를 통해 이론 중심 교육과정에서 벗어나, 학생들이 강화학습 알고리즘을 실물 시스템에 적용해보는 실습 중심 학습의 효과를 기대할 수 있다.

본 논문은 제안된 플랫폼의 효용성을 검증하기 위해, pendubot 시스템을 활용한 제어 실험을 진행한다. Pendubot은 두 개의 링크로 구성된 부족 구동 시스템으로, 다양한 제어 이론을 적용하는 테스트베드로 사용되고 있으며, 강화학습 연구에도 활용된 바 있다[10, 11]. 특히, pendubot의 비선형 모델 방정식은 강화학습 기반 제어기의 성능을 평가하기에 적합한 도전과제를 제공한다. 이러한 이유로, 본 논문에서는 제안된 플랫폼의 실효성을 검증하는 데 pendubot을 사용한다.

본 논문의 이후 구성은 다음과 같다. 2장에서는 Python과 Matlab/Simulink, LW-RCP가 결합된 플랫폼의 구조를 서술한다. 이어지는 3장에서는 Sim-to-Real 기법을 적용하기 위한 pendubot의 모델 방정식을 유도하고, pendubot의 균형점에 대해 서술한다. 이후 4장에서는 강화학습 기반 제어기를 설계하고 pendubot 제어 실험을 통해 제안하는 플랫폼의 효용성을 검증하고, 마지막으로 5장에서 결론을 맺는다.

2. 제안하는 플랫폼의 구조

2.1 Sim-to-Real 기법을 활용한 강화학습 기반 제어기

본 논문에서 제안하는 플랫폼에는 강화학습 기반 제어기가 핵심적인 역할을 수행한다. 강화학습 기반 제어기는 고전적 제어 방식에서 제어 연산을 담당하는 제어기의 역할을 강화학습 에이전트로 대체한 형태를 의미한다[12].

강화학습에서 사용되는 개념인 에이전트란 강화학습 알고리즘을 구현한 시스템을 말한다. 에이전트는 특정 목적을 갖고 주어진 환경과 상호작용하며 자신의 행동 정책에서 비롯한 동작을 수행하며, 그 결과로 얻어지는 보상을 통해 행동 정책을 개선하는 과정을 반복한다[13]. 이렇게 완성된 강화학습 기반 제어기는 주어진 환경에서 취득한 데이터를 입력받아 학습된 행동 정책에 따른 최적의 행동인 제어량을 출력하게 된다. 이때, 에이전트와 상호작용이 이루어지는 환경은 주로 실물 시스템을 활용한 물리적 환경, 혹은 시뮬레이션 기반의 가상환경 두 가지로 구성된다.

먼저, 실물 시스템과 직접 상호작용하는 방법은 실물 시스템의 정확한 모델이 필요 없다는 장점이 있다. 하지만, 해당 방식으로 학습을 진행하는 경우, 학습에 요구되는 물리적인 시간이 필요하며, 학습 과정 중 발생하는 시행착오로 인해 안전사고가 발생할 수 있다[14]. 이는 제한된 시간 내에 안전하게 교육을 진행해야 하는 실습 중심의 교육과정 특성상 해당 방식은 도입되기 어렵다.

따라서, 본 논문에서는 에이전트가 시뮬레이션 환경과 상호작용하는 방식인 Sim-to-Real 학습 기법을 활용한 강화학습 기반 제어기를 사용하고자 한다. 해당 기법은 에이전트와 상호작용하는 환경이 시뮬레이션 환경이 되므로 학습에 걸리는 시간을 크게 줄일 수 있다. 이로 인해, 제한된 교육 환경에서 제어기 설계에 드는 시간적 비용을 크게 줄여 더욱 내실 있는 교육 과정을 구성할 수 있게 된다. 또한, 학습 과정 중 발생할 수 있는 모든 시행착오들이 컴퓨터 시뮬레이션에서 발생하기 때문에, 안전하게 에이전트 학습을 진행할 수 있다. 이를 통해 제한된 교육 시간 내에 안전하게 에이전트의 학습을 마칠 수 있도록 도와준다.

본 논문에서 제안하는 플랫폼의 핵심 요소 중 하나인 강화학습 에이전트를 구현하기 위해서는 Python의 사용이 요구된다. 이는 강화학습 분야를 포함한 인공지능 연구를 수행하기 위해 사용되는 대표적 프레임워크인 PyTorch[15], Tensorflow[16]등이 Python으로 만들어졌기 때문이다.

2.2 LW-RCP

Sim-to-Real 기법을 통해 완성된 강화학습 기반 제어기의 동작에 필요한 센서 데이터는 LW-RCP를 통해 획득한다. RCP 시스템은 제어 시스템 엔지니어들이 제어 알고리즘을 빠르고 효율적으로 설계 및 검증하기 위해 사용하는 개발 환경을 의미한다[17, 18].

해당 시스템은 hardware interface를 담당하는 장치와 Simulink에서 사용하는 library block으로 구성되며, 그림 1은 LW-RCP hardware interface 장치의 사진이다. LW-RCP의 hardware interface를 통해 센서에서 관측한 데이터를 PC로 전송할 수 있으며, PC에서 구동되는 Simulink 제어 모델의 연산

결과인 제어량을 실물 시스템의 구동부에 인가할 수 있다. 해당 과정은 high-speed USB통신으로 이루어지며, 최대 2kHz의 샘플링 주파수를 보인다.

교육을 듣는 학생들은 LW-RCP library에서 제공하는 block을 이용하여 하드웨어 접근을 담당하는 알고리즘을 구성하고, Simulink의 'matlab function' block으로 구현된 신경망으로 강화학습 기반 제어기를 구성한다. 센서 데이터 측정과 제어 데이터를 구동부에 적용하는 hardware interface가 필요한 부분은 LW-RCP가 담당하고, Simulink 기반 제어기 모델의 연산은 PC가 담당하는 구조다. LW-RCP와 PC의 동작 원리는 그림 2와 같다.

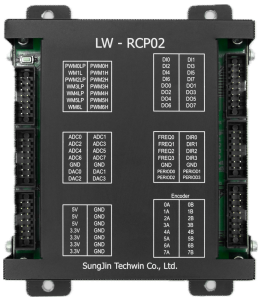


그림 1 LW-RCP02 hardware 장치
Fig 1 LW-RCP02 hardware unit

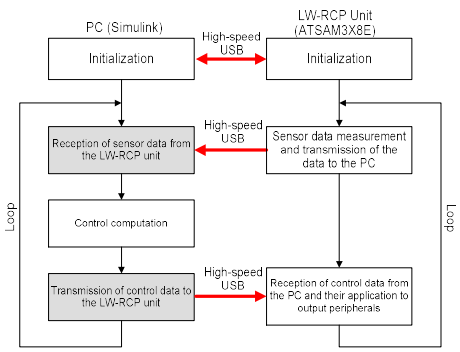


그림 2 LW-RCP의 동작 방식
Fig 2 Flow chart of LW-RCP operation

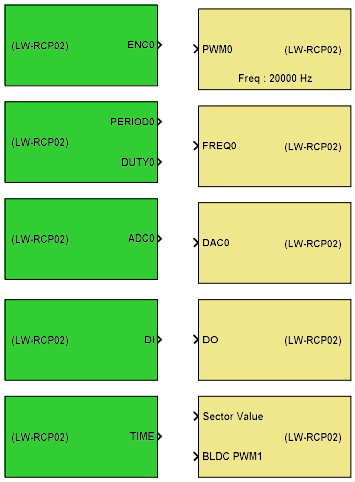


그림 3 LW-RCP02 입출력 library block
Fig 3 Input/Output library block of LW-RCP02

그림 3은 RCP가 제공하는 입출력 library block을 보여준다. 해당 block들은 각각 입출력 기능을 담당하고 있으며, 학생들은 이를 활용하여 하드웨어 접근에 필요한 기능을 손쉽게 만들 수 있다. 좌측 열의 block들은 receive block으로, hardware interface로부터 데이터를 수신하는 기능을 갖고 있다. 우측 열에 위치한 block들은 hardware interface로 데이터를 보내기 위한 Send block들이다. 따라서 LW-RCP를 이용한 제어 알고리즘은 Simulink의 block을 활용하는 방식으로 구현되며, 학생들이 마이크로컨트롤러를 제어하기 위한 code를 만들고 이를 debugging 하는 과정을 없애 제어 알고리즘 구성에만 집중할 수 있는 환경을 제공한다.

2.3 플랫폼 구조

제안하는 교육 플랫폼은 실시간으로 실물 시스템의 데이터 취득 및 전송, 수신 데이터 기반 제어량 연산, 그리고 제어 신호의 입출력을 포함하는 일련의 과정을 구현한다. 이러한 과정은 강화학습 알고리즘을 기반으로 한 에이전트의 생성 및 활용을 목표로 하며, 그림 4의 개념도에 나타난 3가지 시스템의 결합을 통해 이루어진다.

그림 4의 구성도에 따르면, LW-RCP는 센서를 통해 관측한 데이터를 Matlab/Simulink 환경에 전달하고, 동시에 제어 입력을 실물 시스템의 구동기에 인가한다. Matlab/Simulink는 중간에 위치하여 실물 시스템의 실시간 제어 시스템의 역할을 수행한다. Python에서는 강화학습 에이전트의 학습이 이루어지는 시뮬레이션 환경이 구현되어 있으며, 학습 완료된 신경망은 Matlab/Simulink에서 사용 가능한 형식으로 변환되어 강화학습 기반 제어기 구성에 사용된다.

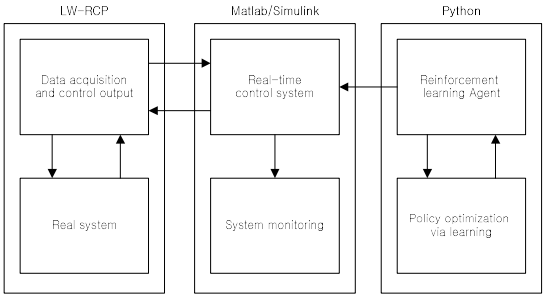


그림 4 제안하는 플랫폼의 구성도
Fig. 4 Concept diagram of the proposed platform.

Python에서 Sim-to-Real 기법을 통해 학습된 신경망과 실물 시스템 제어기가 구현된 Matlab/Simulink가 통합된 플랫폼을 구성하기 위해 MATLAB에서 제공하는 'matlab.engine'과 Python API를 사용한다. 해당 기능은 Matlab의 기능을 Python에서 사용할 수 있도록 Matlab에서 제공하는 기능으로 해당 API를 통해 Python코드 내부에서 Matlab 함수를 호출할 수 있으며, 그 반대의 경우도 가능하다. 또한, Python과 Matlab간의 작업 공간에 서로 접근하여 저장된 변수들을 사용함으로써 더욱 효율적인 개발이 가능해진다. 이를 통해, 본 연구에서는 Python 환경에서 Sim-to-Real 기법으로 학습된 신경망의 파라

미터 값인 가중치 값과 편향 값을 Matlab/Simulink 환경에서 사용가능한 파일로 저장한다. 이후 저장된 파라미터 값들은 Matlab/Simulink 환경에서 강화학습 기반 제어기를 구성하는 요소로 사용된다[12].

상기된 요소들을 결합하여 Python과 Matlab/Simulink가 결합된 형태의 플랫폼의 진행 순서는 다음과 같다. 먼저, Python에서 생성된 강화학습 에이전트는 시뮬레이션 환경에서 학습을 진행하게 된다. 학습 과정은 사용자가 설정한 에피소드의 종단 시간을 넘거나, 미리 설정한 특정 종료 조건에 부합하는 상황이 발생한 경우, 시뮬레이션을 종료한다. 학습이 완료된 에이전트는 자신의 파라미터 값들을 Python API를 통해 Matlab의 작업 공간에 전달한다. 이후 Simulink로 제어 시스템 모델 파일에 전달받은 파라미터들을 매개 변수로 사용한 심층 신경망 블록을 구성한다. 심층신경망으로 구현된 강화학습 기반 제어기는 센서 데이터를 활용하여 제어 연산을 수행하고 그 출력으로 제어량을 출력하게 된다. 출력된 제어량은 LW-RCP를 통해 실물 시스템의 구동기에 전달되어 실물 시스템 제어를 수행하게 된다.

3. Pendubot의 모델 방정식과 균형점

3.1 Pendubot의 모델 방정식 유도

2장에서 언급된 Sim-to-Real 학습 기법을 사용하기 위해서는 시뮬레이션 환경에서 사용될 pendubot의 모델 방정식이 요구된다. 따라서, 본 절에서는 pendubot의 수학적 모델 방정식을 유도한다. 그림 5는 pendubot의 기구적 개념도를 나타내며, 그림에서 사용되는 변수들은 SI 단위계를 따르고 있으며, 그림 5에 기술된 변수들의 세부적 의미는 다음과 같다.

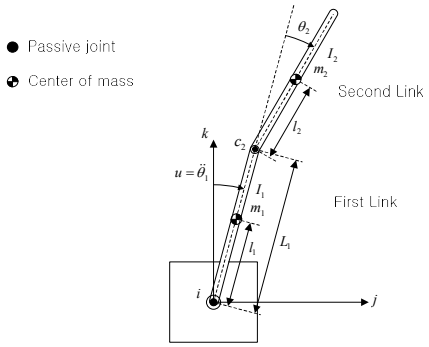


그림 5 Pendubot의 기구적 개념도
Fig 5 Mechanical conceptual diagram of pendubot

T는 1단 링크에 가해지는 토크, m_1, m_2는 각각 1단 링크와 2단 링크의 질량, l_1, l_2는 각각 1단 링크와 2단 링크의 회전축으로부터 무게 중심까지의 길이를 나타낸다. θ_1는 지면의 법선 방향으로부터 1단 링크의 회전 변위를 의미하며, θ_2는 2단 링크가 1단 링크와 이루는 상대적인 회전 변위를 의미한다. L_1은 1단 링크의 회전축부터 2단 링크의 회전축까지의 길이를 의미한다. c는 2단 링크의 회전축에 존재하는 마찰계수를 의미하며, I_1, I_2는 1단 링크 2단 링크의 관성 모멘트다. u는 1단 링크의 각가속도를 의미하며, i, j, k는 1단 링크의 회전축을 중심점으로 한 직각 좌표계의 좌표축들을 의미한다.

Pendubot의 모델 방정식은 Euler-Lagrange Equation을 이용하여 유도하면 식 (1)로 나타낼 수 있다.

$$\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} T \\ 0 \end{bmatrix} \tag{1}$$

여기서 식 (1)을 구성하는 요소는 식 (2)와 같다.

$$\begin{aligned} m_{11} &= h_1 + h_2 + h_3 + 2h_4\cos(\theta_1) \\ m_{12} &= h_3 + h_4\cos(\theta_2) \\ m_{21} &= h_3 + h_4\cos(\theta_2) \\ m_{22} &= h_3 \\ r_1 &= -h_4\sin(\theta_1)(\dot{\theta}_2^2 + 2\dot{\theta}_1\dot{\theta}_2) - h_5\sin(\dot{\theta}_1) \\ &\quad -h_6\sin(\theta_1 + \theta_2) \\ r_2 &= h_4\sin(\theta_2)\dot{\theta}_1^2 - h_6\sin(\theta_1 + \theta_2) + c\dot{\theta}_2 \end{aligned} \tag{2}$$

h_1~h_6는 다음과 같이 정의되며, g는 중력가속도 $9.81[m/s^2]$를 나타낸다.

$$\begin{aligned} h_1 &= m_2 L_1^2 \\ h_2 &= m_1 l_1^2 + I_1 \\ h_3 &= m_2 l_2^2 + I_2 \\ h_4 &= m_2 L_1 l_2 \\ h_5 &= m_2 L_1 + m_1 l_1 \\ h_6 &= m_2 l_2 \end{aligned} \tag{3}$$

식 (1)을 재배열하면, 식 (4)의 형태로 정리가 가능하다.

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = -\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \begin{bmatrix} T \\ 0 \end{bmatrix} \right\} \tag{4}$$

이 때, 1단 링크의 각가속도 $\ddot{\theta}_1$을 제어입력 u로 하는 각가속도 모델을 유도하면 pendubot의 운동방정식을 식 (5)의 형태로 나타낼 수 있다.

$$\begin{aligned} \ddot{\theta}_1 &= u \\ \ddot{\theta}_2 &= \frac{r_2 - m_{21}u}{m_{22}} \end{aligned} \tag{5}$$

이 때, 식 (5)를 바탕으로 구한 운동방정식을 통해 상태변수를 각각 $x_1 = \theta_1, x_2 = \theta_2, x_3 = \dot{\theta}_1, x_4 = \dot{\theta}_2$로 정의하면 최종적으로 pendubot의 모델방정식은 식 (6)과 같은 비선형 상태공간 방정식으로 나타낼 수 있다.

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} x_3 \\ x_4 \\ u \\ \dfrac{r_2 - m_{21}u}{m_{22}} \end{bmatrix}}_{f(x,u)} \tag{6}$$

Pendubot의 모델식은 각 링크의 회전축을 중심으로 한 회전 운동만이 발생할 수 있고 그 외의 직선 및 회전운동은 발생하지 않는 것을 가정한다. 모델식에서 고려한 마찰력은 회전속도에 선형적인 관계를 갖는 회전 마찰력만을 가정하며, coulomb 마찰은 고려하지 않는다.

3.2 Pendubot의 균형점

Pendubot은 2개의 링크를 갖는 형태로, 1단 링크와 2단 링크의 회전 변위에 따라 다양한 균형점을 갖는다. 균형점은 2단 링크가 바닥에 늘어진 안정적인 상태와 도립 상태가 있으며, 본 논문에서는 도립 상태의 균형점만을 다룬다. 그 중 1단 링크의 각변위에 따라 $\theta_1 = 0, \theta_1 = -\frac{\pi}{6}, \theta_1 = \frac{\pi}{6}$인 도립 상태의 균형점만을 실험에서 구현한다. 그림 6은 해당 균형점을 시각화한 그림이다.

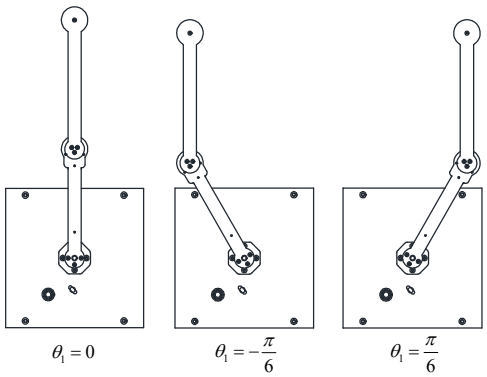


그림 6 Pendubot 시스템의 균형점
Fig 6 Equilibrium points of pendubot system.

4. 실험 및 결과

본 절에서는 앞서 서술한 모델 방정식과 pendubot 시스템을 이용하여, Sim-to-Real 기법을 활용한 강화학습 기반 제어기를 설계하고 이를 실제 시스템에 적용하여 제안하는 플랫폼의 효용성을 검증한다. 본 절의 구성은 다음과 같다. 먼저 시뮬레이션 환경을 만들기 위한 설정과, 강화학습 기반 제어기를 구현하는 과정에 대해 서술한다. 이후, 실험 환경 및 실험 결과에 대해 서술한다.

4.1 시뮬레이션 환경 및 설정

강화학습 에이전트가 학습을 위해 직접 상호작용하는 환경은 3장에 서술한 모델 방정식을 바탕으로 Python상에 시뮬레이션으로 구현하였다. 해당 환경을 구축하는데 사용된 pendubot의 물리적 파라미터는 표 1에 나열되어 있으며, 비선형 상미분 방정식의 해를 구하기 위한 solver는 ode4 Runge-kutta 방법을 선택하였다. 본 시뮬레이션 환경에서 학습을 진행한 강화학습 에이전트는 학습을 완료한 후, 2장에서 언급된 'matlab.engine'과 Python API를 활용하여 Matlab/Simulink에서 사용 가능한 형태로 변환되어 강화학습 기반 제어기에 사용된다.

시뮬레이션으로 구동되는 학습 환경에서 한 에피소드의 길이는 10초로 설정되었고, 시뮬레이션은 1ms 주기로 업데이트가 이루어지며, 에이전트는 10ms 마다 상태정보를 관측한다. 따라서 에이전트는 한 에피소드당 환경과 최대 1000번 상호작용을 하게 되고, 매 상호작용 시점의 보상을 바탕으로 자신의 행동 정책을 개선한다. 추가적으로 1000번의 상호작용이 이루어졌거나 1단 링크의 각속도의 절댓값인 $|\dot{\theta}_1|$이 $25 rad/s$을 초과하는 경우 학습이 종료된다. pendubot의 시뮬레이션 환경에서 관측 가능한 상태 정보는 3장에 기술된 비선형 상태방정식에 따라 $<\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2>$로 구성된다. 이때 θ_1, θ_2는 보상함수의 원활한 설계를 위해 나머지 연산을 적용하여 $-\pi < \theta < \pi$의 범위로 제한한다. 추가적으로, θ_1, θ_2는 학습 과정에서 정규화와 연속성의 이점을 얻기 위해 $\sin(\theta_i), \cos(\theta_i)$의 형태로 재구성한다.

그리고 3장에서 언급한 3가지 균형점을 $\tau \in \left\{-\frac{\pi}{6}, 0, \frac{\pi}{6}\right\}$로 정의한다. τ의 각 요소는 3개의 균형점과 일대일 대응되며, 해당 값에 따라 에이전트는 자신이 어느 목표 균형점으로 이동해야 하는지 알 수 있게 된다. 결과적으로 강화학습 에이전트가 매 timestep마다 관측하는 재구성된 상태 정보는 $<\sin(\theta_1), \cos(\theta_1), \sin(\theta_2), \cos(\theta_2), \dot{\theta}_1, \dot{\theta}_2, \tau>$로 구성된 7개의 데이터가 된다. 강화학습 에이전트는 해당 상태 정보를 입력으로 받아 행동 정책에 따른 행동인 제어량을 출력하게 된다. 이때 출력되는 제어량은 모터의 각가속도 값 u이며, 실제 시스템 구동기의 한계를 고려하여 $-100 < u < 100$으로 제한한다.

표 1 실험에 사용된 pendubot의 파라미터
Table 1 Parameters of the pendubot used in the experiment

Parameter	Value
L_1	0.1645 [m]
I_2	7.9454e-04 [kgm^2]
m_2	0.1592 [kg]
l_2	0.1512 [m]
c	1.6626e-05

4.2 강화학습 기반 제어기 구현

4.2.1 강화학습 알고리즘

본 연구에서 사용되는 강화학습 에이전트는 Truncated Quantile Critics(TQC) 알고리즘으로 구현되었다[19]. TQC 알고리즘은 연속적 행동 공간을 대상으로 한 정책 최적화 문제에 효과적인 알고리즘으로, 기존 Quantile Regression DeepQ-Network(QR-DQN)[20], Soft-Actor-Critic[21] 알고리즘의 장점들을 결합하여 과대평가 문제를 완화하고, 더 정밀한 보상 분포 추정을 가능하게 한다. 이를 통해 알고리즘이 극단적으로 높은 보상을 기대하는 행동을 선택하는 것을 방지하여 더욱 현실적인 기대를 기반으로 정책을 결정하도록 유도한다.

해당 알고리즘을 구현하기 위해 사용된 하이퍼 파라미터 값들은 표 2에서 확인할 수 있다.

TQC 알고리즘을 통해 구현된 강화학습 에이전트는 Python으로 구현된 pendubot 시뮬레이션 환경과 지속적인 상호작용을 하며 swing-up 제어를 위한 기법을 학습한다. 해당 알고리즘을 구현하는데 사용된 하이퍼 파라미터 값들은 표 2에서 확인할 수 있다. 참고문헌 [19]에 따르면, Number of ciritics인 N이 3일 때의 학습 결과가 $N=5, N=10$일 때의 학습 결과와 유사하므로 $N=3$으로 선정하여 학습을 진행했다. 또한, 3가지 균형점에 대한 제어를 모두 학습시키기 위해 제어기에 해당하는 policy network의 size는 키웠다. 이 외의 값들은 모두 참고문헌 [19]의 저자들과 동일한 하이퍼 파라미터 값들을 사용했다.

표 2 본 실험에 사용된 하이퍼 파라미터

Table 2 Hyperparameter that used in this experiment

HyperParameter	Value
Optimizer	Adam
Learning rate	0.0003
Discount factor (γ)	0.99
Replay buffer size	1e-6
Number of critics (N)	3
Number of hidden layers in critic networks	3
Size of hidden layers in policy networks	512
Size of hidden layers in 1st policy networks	2
Size of hidden layers in 2nd policy networks	400
Minibatch size	300
Nonlinearity	ReLU
Target smoothing coefficient (β)	0.005
Number of atoms (M)	25

4.2.2 보상함수 설계 및 학습 결과

강화학습 에이전트는 환경과 상호작용이 일어나는 순간마다 그 시점의 보상에 기반하여 자신의 행동 정책을 개선한다. 이때 보상 값을 산출하기 위해 사용되는 보상함수는 pendubot에 존재하는 이번 실험에 사용될 3개의 균형점(equilibrium point) 중 어떤 균형점에 도달할지에 따라 달라진다. 여기서 균형점은 2단 링크가 바닥으로 늘어진 안정한 균형점과 도립 상태의 균형점이 있으며, 본 실험에서 구현하고자 하는 균형점은 도립 상태의 균형점이다.

표 3은 목표 균형점을 나타내는 값인 τ에 따라 달라지는 각 링크의 각도 θ_i^*를 나타내고, 보상함수는 식 (7)의 형태로 나타낸다.

표 3 목표 균형점에 따른 각 링크의 각도

Table 3 Each link's angle according to target equilibrium point

τ	target angle	
	θ_1^*	θ_2^*
$-\frac{\pi}{6}$	$-\frac{\pi}{6}$	$\frac{\pi}{6}$
0	0	0
$\frac{\pi}{6}$	$\frac{\pi}{6}$	$-\frac{\pi}{6}$

$$
\begin{aligned}
R_{\theta_1} &= 0.5+0.5*\cos(\theta_1-\theta_1^*)\\
R_{\theta_2} &= 0.5+0.5*\cos(\theta_2+\theta_2^*)\\
R_{\dot{\theta}_1} &= \exp(-0.1*|\dot{\theta}_1|)\\
R_{\dot{\theta}_2} &= \exp(-0.1*|\dot{\theta}_2|)\\
R_u &= \exp(-0.003|u|)
\end{aligned}
\tag{7}
$$

최종적인 보상 함수는 각 요소를 모두 곱한 형태로 표현된다.

$$
Reward = R_{\theta_1}R_{\theta_2}R_{\dot{\theta}_1}R_{\dot{\theta}_2}R_\tau \tag{8}
$$

보상함수를 이루는 각각의 요소는 그림 7에서 확인할 수 있으며, 모든 요소들은 [0, 1]의 값으로 정규화 되어있다. 따라서 해당 요소들의 곱들로 이루어진 $Reward$ 또한 [0, 1]의 값을 가지게 된다. 이때 한 에피소드는 최대 1000개의 timestep으로 이루어져 있으므로, 한 에피소드에서 얻을 수 있는 최대 보상은 1000이 된다. 이때, 도달하고자 하는 균형점의 각도에 따라 변하는 값인 τ에 따라 변하는 값인 θ_i^*를 포함하지 않는

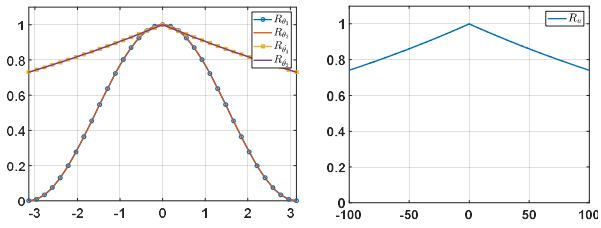


그림 7 보상함수 그래프

Fig 7 Reward function graph

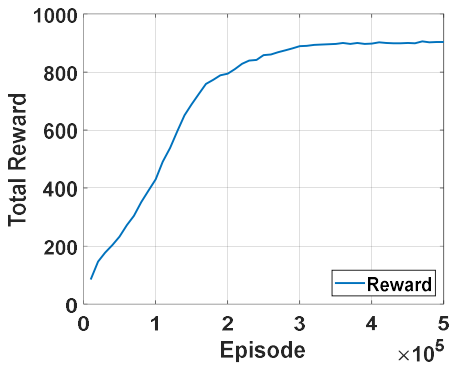


그림 8 학습 결과 그래프

Fig 8 Learning results graph

$R_{\theta_1}, R_{\theta_2}$는 0에 수렴할수록 보상이 증가하는 형태를 갖도록 한다. 이를 통해 pendubot은 균형점에 최소한의 움직임을 갖도록 하여 최소의 제어량을 갖도록 한다. θ_i^*에 따라 보상이 달라지는 $R_{\theta_1}, R_{\theta_2}, R_t$는 목표한 균형점에 가까울수록 높은 보상을 받는다. 이를 통해 에이전트는 상태 변수 τ로 결정되는 목표 균형점에 도달하기 위해 자신의 행동 정책을 개선한다. 그림 8은 앞서 설명된 조건에서 매 에피소드마다 τ값에 변화를 주며 반복적인 학습을 진행한 결과다.

4.3 실험 환경

제안하는 플랫폼의 효용성을 검증하기 위한 실험은 다음과 같다. 먼저, pendubot이 $\theta_1 = 0$인 균형점으로 이동하는 swing-up 제어를 수행한 뒤, 차례대로 $\theta_1 = -\frac{\pi}{6}$, $\theta_1 = \frac{\pi}{6}$인 균형점으로 이동하는 천이제어를 수행한다. 마지막으로 $\theta_1 = 0$인 균형점에서 외란을 인가해 2단 링크가 균형점을 이탈했을 때, 원래의 균형점으로 돌아오는 Recovery 특성[13]을 확인한다.

그림 9는 플랫폼의 효용성 검증 실험인 pendubot 제어 실험을 위한 Simulink 모델이며, 그림 10은 hardware interface 기능을 LW-RCP의 library block으로 구현한 모습이다. 강화학습 기반 제어기는 matlab function block으로 구현되었으며, 이는 학생들이 사용할 수 있도록 library block의 형태로 제공된다. 해당 제어기는 pendubot의 4가지 상태변수, 목표 균형점, 제어량을 입력으로 받는다. 모든 상태변수 값들은 그림 10의 receive block을 통해 수집되며, 목표 균형점은 사용자의 직관적인 이해를 위해 60분법으로 표현된 각도를 호도법으로 변환하여 입력된다. 제어기는 입력된 정보들을 바탕으로 1단부 링크의 회전 각가속도인 $\ddot{\theta}_1$을 출력하며, 이는 적분되어 구동부 모터의 속도 지령치로 사용된다. 구동부 모터는 PI 속도 제어기를 통해 제어되며, 모터 구동을 위한 PWM 신호는 그림 10의 Send block을 통해 출력된다.

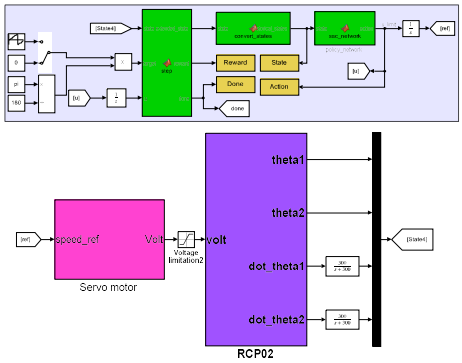


그림 9 Pendubot 시스템 제어 실험 Simulink 모델
Fig 9 Pendubot system control experiment Simulink model

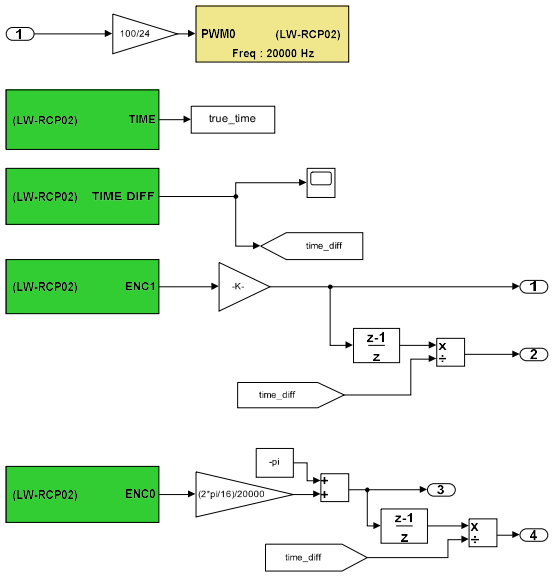


그림 10 Hardware interface Simulink 모델
Fig 10 Hardware interface Simulink model

4.4 실험 결과

그림 11은 pendubot 제어 실험의 결과를 보여주는 연구실 Youtube 영상의 일부다. 실제 Youtube 영상의 주소는 https://youtu.be/G9ik_Ha4E70 이다. (영상 제목 : Sim-to-real reinforcement learning control of a pendubot, Channel 이름: Embedded Control Lab.) 그림 12는 4.3절에서 언급된 swing-up 및 천이 제어, Recovery 특성에 관한 실험 결과를 그래프로 나타낸 것이다. 그래프의 x축은 시간을, y축은 각각 θ_1, θ_2를 나타낸다. swing-up 및 천이제어 실험에서는 강화학습 기반 제어기가 pendubot이 목표 균형점으로 이동하도록 swing-up과 천이제어를 수행하며, 평균적으로 0.02[rad]의 오차를 가지는 것을 확인할 수 있다. 그림 12에서 점선으로 표기된 시점은

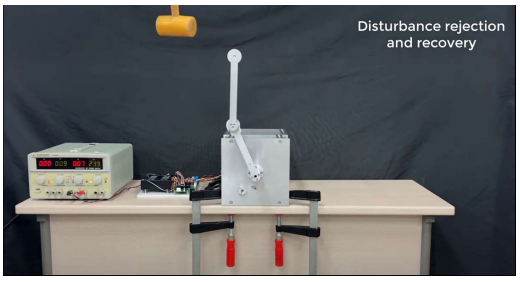


그림 11 실험 과정을 보여주는 Youtube 영상
Fig 11 Youtube video for the experiment

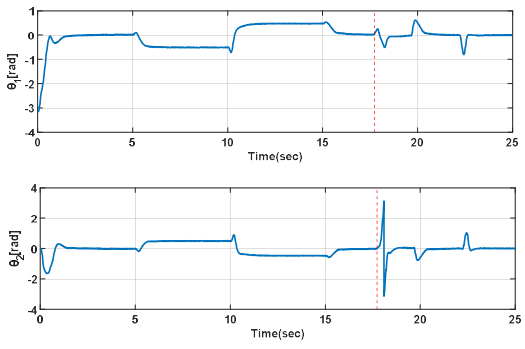


그림 12 Pendubot 제어 실험 결과
Fig 12 Result of pendubot control experiment

외란이 가해진 순간을 나타내며, 실험 시작 후 약 17초경에 외란을 인가한다. 외란이 가해진 이후, 강화학습 기반 제어기가 $\theta_1 = 0$인 균형점으로의 swing-up 제어를 시도하여 목표 균형점으로 회복하는 Recovery 특성을 확인할 수 있다. 영상과 그래프를 통해 확인할 수 있듯 다양한 실험을 손쉽게 수행할 수 있었으며, 강화학습 기반 제어기의 Recovery 특성을 직접 확인할 수 있었다. 이를 통해, 제안한 플랫폼의 효용성을 검증할 수 있었다.

5. 결 론

본 논문에서는 Python과 LW-RCP를 이용한 강화학습 기반 제어 교육 플랫폼을 제안하고 그 효용성을 검증하고자 했다. 강화학습 기반 제어기를 구성하는 신경망은 Python에서 Sim-to-Real 기법을 통해 학습되었으며, 학습이 끝난 신경망은 Matlab/Simulink 환경에 호환되는 형태로 변환된다. 그리고 실물 시스템 제어를 위한 Simulink 모델은 크게 구동부 제어를 담당하는 부분과 강화학습 기반 제어기로 나뉘는데, 구동부 제어는 LW-RCP에서 제공하는 library block으로 구성되었으며, 강화학습 기반 제어기는 matlab function block으로 구현된 신경망으로 제작되었다.

이후, 해당 플랫폼의 효용성 검증을 위한 pendubot 제어 실험을 수행한 결과, swing-up 및 천이 제어, Recovery 특성 실험 모두 성공적인 결과가 나왔음을 확인할 수 있었다. 이를 통해 본 논문에서 제안하는 플랫폼이 강화학습 기반 제어 교육에 사용될 수 있음을 확인할 수 있었다. 따라서, 본 논문에서 제안하는 플랫폼을 통해 강화학습을 실물 시스템에 적용하는 교육과정 운영에 큰 도움을 줄 수 있을 것으로 기대한다.

Acknowledgements

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2024-00347193).

References

[1] D. Silver, et al., “Mastering the game of Go with deep neural networks and tree search,” Nature, vol. 529, no. 7587, pp. 484-489, 2016.
DOI:10.1038/nature16961

[2] L. P. Kaelbling, M. L. Littman, A. W. Moore, “Reinforcement learning: A survey,” Journal of artificial intelligence research, vol. 4, pp. 237-285, 1996.
DOI:10.1613/jair.301

[3] B. Osiński et al., “Simulation-Based Reinforcement Learning for Real-World Autonomous Driving,” 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, pp. 6411-6418, 2020.
DOI:10.1109/ICRA40945.2020.9196730

[4] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” Science robotics, vol. 7, no. 62, eabk2822, 2022.
DOI:10.1126/scirobotics.abk2822

[5] S. A, A. N. Rafferty, et al., “Reinforcement learning for education: Opportunities and challenges,” arXiv preprint arXiv:2017.08828, 2021.
DOI:10.48550/arXiv.2107.08828

[6] B. Fahad Mon, A. Wasfi, et al., “Reinforcement Learning in Education: A Literature Review,” Informatics, vol. 10, no. 3, pp. 74-95, 2023.
DOI:10.3390/informatics10030074

[7] H. Gharbi, L. Elaachak, A. Fennan, “Reinforcement Learning Algorithms and Their Applications in Education Field: A Systematic Review,” The Proceedings of the International Conference on Smart City Applications, pp. 410-418, 2023.
DOI:10.1007/978-3-031-54376-0_37

[8] H. Shin, and H. Oh, “Neural Network Model Compression Algorithms for Image Classification in Embedded Systems,” The Journal of Korea Robotics Society, vol. 17, no. 2, pp. 133-141, 2022.
DOI:10.7746/jkros.2022.17.2.133

[9] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey,” 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 737-744, 2020.
DOI:10.1109/SSCI47803.2020.9308468

[10] M. A. Perez-Cisneros, R. Leal-Ascencio, and P. A. Cook, “Reinforcement learning neurocontroller applied to a 2-DOF manipulator,” Proceeding of the 2001 IEEE International Symposium on Intelligent Control (ISIC’01), pp. 56-61, 2001.
DOI:10.1109/ISIC.2001.971484

[11] Y. Cheng, P. Zhao, F. Wang, D. J. Block, and Hovakimyan, “Improving the Robustness of Reinforcement Learning Policies With L1 Adaptive Control,” IEEE Robotics and Automation Letters, vol. 7, no. 3, pp. 6574-6581, 2022.
DOI:10.1109/LRA.2022.3169309

[12] T. Lee, D. Ju, and Y. S. Lee, “Development Environment of Reinforcement Learning-based Controllers for Real-world Physical Systems Using LW-RCP,” Journal of Institute of Control, Robotics and Systems (in Korean), vol. 29, no. 7, pp. 543-549, 2023.
DOI:10.5302/J.ICROS.2023.23.0045

[13] T. Lee, D. Ju, and Y. S. Lee, “Sim-to-Real Reinforcement Learning Techniques for Double Inverted Pendulum Control with Recovery Property,” The transactions of The Korean Institute of Electrical Engineers (in Korean), vol. 72, no. 12, pp. 1705-1713, 2023.
DOI:10.5370/KIEE.2023.72.12.1705

[14] G. Dulac-Arnold, N. Levine, D. J. Mankowits, J. Li, C. Paduraru, S. Gowal, and T. Hester, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," Machine Learning, vol. 110, no. 9, pp. 2419-2648, 2021.
DOI:10.1007/s10994-021-05961-4

[15] A. Paszke, S. Gross, F. Massa, and et al., "Pytorch: An impressive style, high-performance deep learning library," Advances in Neural Information Processing Systems, vol. 32, 2019.

[16] M. Abadi, P. Barham, J. Chen, and et al., "Tensorflow: A system for large-scale machine learning," Osdi, vol. 16, no. 2016, pp 265-283, 2016.

[17] Y. S. Lee, D. Ju, C. Choi, "Development of Educational Environment to Improve Efficiency of Online Education on Control Systems," Journal of Institute of Control, Robotics and Systems (in Korean), vol. 27, no. 12, pp. 1056-1063, 2021.
DOI:10.5302/J.ICROS.2021.21.0199

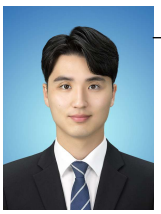
[18] Y. Fujiyama, D. Ju and Y. S. Lee, "The Implementation of a Ball and Plate System using a 3-DOF Stewart Platform and LW-RCP," The transactions of The Korean Institute of Electrical Engineers (in Korean), vol. 72, no. 8, pp. 943-951, 2023.
DOI:10.5370/KIEE.2023.72.8.943

[19] A. Kuznetsov, P. Shvechikov, A. Grishin and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in International Conference on Machine Learning, PMLR, pp. 5556-5566, 2020.

[20] W. Dabney, M. Rowland, M. Bellemare, and Munos R, "Distributional reinforcement learning with quantile regression," in Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1, pp. 2892-2901, 2018.
DOI:10.1609/aaai.v32i1.11791

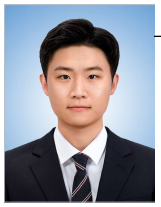
[21] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," International conference on machine learning. PMLR, pp. 1861-1870, 2018.

저자소개



이종범(Jongbeom Lee)

He received B.S. degree in electrical engineering from Inha university in 2024. He is now a M.S. candidate in electrical and computer engineering at Inha university. His research interests include optimal control, reinforcement learning and embedded systems.



이태건 (Taegun Lee)

He received B.S. degree in electrical engineering from Inha university in 2023. He is now a M.S. candidate in electrical and computer engineering at Inha university. His research interests include reinforcement learning, embedded systems and optimal control.

주도윤 (Doyoon Ju)

He received M.S. degree in electrical and computer engineering from Inha university in 2023. He is now a Ph.D. candidate in electrical and computer engineering at Inha university. His research interests include optimal control, embedded systems and reinforcement learning.

이영삼 (Young Sam Lee)

He received B.S. and M.S. degrees in electrical engineering from Inha University, Incheon, South Korea, in 1999, and the Ph.D. degree in electrical engineering from Seoul National University, South Korea, in 2003. From 2003 to 2004, he was a Senior Researcher with Samsung Electronics Co. Since 2004, he has been with the Department of Electrical and Computer Engineering, Inha University. He is the author of four books and more than 60 articles. His research interests include computer– aided control system designs, rapid control prototyping, control and instrumentation, robot engineering, and embedded systems.