Journal of Institute of Control, Robotics and Systems (2023) 29(7):543-549
https://doi.org/10.5302/J.ICROS.2023.23.0045 ISSN:1976-5622 eISSN:2233-4335

LW-RCP를 이용한 실물 시스템에 대한 강화학습 기반의 제어기 개발 환경

Development Environment of Reinforcement Learning-based Controllers for Real-world Physical Systems Using LW-RCP

이 태 건, 주 도 윤, 이 영 삼*
(Taegun Lee[1], Doyoon Ju[1], and Young Sam Lee[1,*])
[1]Department of Electrical and Computer Engineering, Inha University

Abstract: In recent years, reinforcement learning (RL)-based controller design methods have emerged as a powerful alternatives to traditional methods, providing a novel paradigm that overcomes limitations associated with the need for accurate model information. In this paper, we propose a development environment for RL-based controllers in real-world systems by integrating MATLAB/Simulink, Python, and the LW-RCP (Light-weight Rapid Control Prototyping) system developed by the authors' laboratory. The proposed development environment utilizes LW-RCP's library block in a Simulink-based RL controller model, enabling real-time experiments on real-world systems, and stores state information data in MATLAB's workspace. Python obtains this data through the Python API after each episode and uses it to iteratively enhance the RL agent's policy by using RL algorithms. Updated parameter values for the agent's policy neural network are then sent back to MATLAB's workspace, enabling convenient updates to the deep neural network-based policy controller block in Simulink. This development environment greatly reduces the time and trial and error in configuring real-time system controllers by providing LW-RCP with all necessary functions. Moreover, the efficient data acquisition and integration between MATLAB and Python workspaces facilitate the learning process and reflection of results in Simulink-based controllers. We demonstrate the effectiveness and convenience of the proposed environment through its successful application to the swing-up control problem of a single inverted pendulum.

Keywords: LW-RCP, reinforcement learning, development environment, single inverted pendulum

I. 서론

최근 몇 년 동안 인공지능 기술의 급격한 발전에 따라 강화학습을 제어공학 분야에 적용하는 연구가 활발히 진행되고 있다[1-2]. 전통적인 제어기 설계 방식은 정확한 모델 파라미터와 복잡한 수학적 모델링이 요구되는 반면, 강화학습 기반의 제어기는 사전 정보가 없어도 환경과 상호작용을 통해 받는 보상을 최대화하는 방향으로 행동 정책을 지속적으로 개선한다[3]. 특히 대상 시스템에 대한 수학적 모델 정보가 전혀 없이 실물 시스템을 이용하여 취득한 실험 데이터를 기반으로 제어기를 설계할 수 있는 강화학습의 특징은 모델 정보를 필요로 하는 전통적 제어기 설계 방식의 단점을 극복할 수 있게 하는 새로운 패러다임의 제어기 설계법이라 할 수 있다[2]. 또한, 많은 강화학습 알고리즘들이 심층 신경망을 접목하여 연속적이고 복잡한 환경에서도 최적의 행동을 도출할 수 있는 능력을 갖추게 되었다[4]. 이러한 특성에 기반하여 기존의 model-based 접근 방식인 고전제어 알고리즘을 강화학습의 model-free 알고리즘으로 대체할 수 있는 가능성이 제시된다[5].

하지만 강화학습을 실물 시스템에 적용함에 있어 다양한 문제점과 제약 조건들이 발생하기[6] 때문에 최근의 강화학습 연구는 대부분 시뮬레이션 환경에 초점을 맞추고 있다. 이에 따라 많은 연구에서는 시뮬레이션을 통해 실물 시스템과 동일한 동특성을 가진 환경을 구축하고, 이를 바탕으로 강화학습 알고리즘을 적용하여 연구를 진행하고 있다[7-8]. 그러나 시뮬레이션 환경을 구축하기 위해서는 실물 시스템의 파라미터를 정확히 측정하여 동일한 모델을 생성하는 과정이 선행되어야 한다. 즉, 강화학습의 model-free 알고리즘을 사용한 제어기를 구현하기 위해서는 먼저 완벽한 모델이 필요하게 되는 역설적인 상황이 발생한다. 이로 인해 시스템 제어 모델에 대한 명시적인 지식 없이도 복잡한 제어 정책을 학습할 수 있다는 강화학습의 장점이 상대적으로 퇴색된다. 이러한 강화학습의 장점을 최대화하기 위해서는 강화학습 에이전트가 실물 시스템과 직접 상호작용하며 얻은 데이터를 기반으로 학습을 진행하는 방식이 필요하다.

강화학습 에이전트가 실물 시스템과 직접 상호작용하는 과정에서 가장 먼저 마주하게 되는 주요 어려움은 실물 시

*Corresponding Author
Manuscript received April 18, 2023; revised May 5, 2023; accepted May 10, 2023
이태건: 인하대학교 전기컴퓨터공학과 대학원생(dlxorjs815@gmail.com, ORCID 0009-0007-3107-2735)
주도윤: 인하대학교 전기컴퓨터공학과 대학원생(seiko.kr@gmail.com, ORCID 0000-0001-7011-6779)
이영삼: 인하대학교 전기컴퓨터공학과 교수(lys@inha.ac.kr, ORCID 0000-0003-0665-1464)

스템과의 상호작용을 위한 인터페이스 구현에 관한 문제로 볼 수 있다. 이러한 인터페이스 구현을 위해 실물 시스템에 부착된 센서의 데이터를 수집하기 위한 DAQ 장치와 관련 소프트웨어가 필요하며, 컴퓨터상에서 처리된 결과 값을 활용해 실물 시스템의 구동기를 제어할 수 있는 제어기를 추가로 개발해야 한다. 또한, 강화학습 알고리즘은 환경의 현재 상태 정보를 바탕으로 최적의 행동을 도출하고, 해당 행동을 환경에 적용한 직후 변화된 환경의 상태 정보를 관찰하는 과정을 반복하므로 데이터 입출력 및 제어기 구동에서의 실시간성이 보장되어야 한다. 이러한 요구사항들로 인해 실물 시스템을 기반으로 한 강화학습 연구는 강화학습 알고리즘 설계 외에도 정보 통신 기술과 제어공학에 대한 지식이 요구되며, 이는 연구자가 본연의 목적인 강화학습 알고리즘 설계에 전념하기 어렵게 만들 수 있다.

본 논문에서는 앞서 언급된 실물 시스템을 활용한 강화학습 연구의 어려움을 해결하기 위해, 저자들이 속한 연구실에서 개발한 LW-RCP (Light-Weight Rapid Control Prototyping) 시스템을 활용하는 강화학습 기반의 제어기 설계 환경을 제안한다. LW-RCP 장치는 실시간성을 보장하는 동시에, 실물 시스템에 부착된 센서 데이터를 수집하는 DAQ 장치 역할과 Matlab/Simulink 상에서 계산된 제어 연산 결과를 실물 시스템의 구동기로 전달하는 출력 장치 역할을 동시에 수행한다[9]. 이를 통해 데이터 입출력 및 제어기 구동의 실시간성을 유지하면서도 추가적으로 요구되는 복잡한 인터페이스 구현 과정을 줄일 수 있다. 또한, LW-RCP를 통해 수집된 데이터는 Matlab을 통해 처리할 수 있으며 Matlab은 Python과의 연동을 위한 Python API를 제공한다. 이를 활용하여 강화학습 연구에 주로 사용되는 Python 환경과 데이터를 원활하게 주고받을 수 있다. 결과적으로 LW-RCP를 활용함으로써 Matlab/Simulink, Python 간의 데이터 전달 시스템을 구축하기 위해 요구되는 추가적인 노력과 시간을 크게 줄일 수 있으며, 실시간으로 모든 데이터를 모니터링하면서 강화학습 기반 제어기 설계에 집중할 수 있을 것으로 기대된다.

본 논문은 다음과 같은 구성을 갖는다. 2장에서는 본 논문에서 제안하는 강화학습기반 제어기 개발 환경에 사용되는 LW-RCP라는 rapid control prototyping 시스템에 대한 소개와 이와 함께 사용되는 Matlab/Simulink 및 Python을 결합한 형태의 개발 환경 설계 방법을 제안한다. 이후 3장에서는 구현된 개발 환경을 사용하여 직선형 1단 도립진자의 강화학습 기반 제어기를 구현하고 성능을 검증한다. 끝으로 4장에서는 논문의 결론을 다루도록 한다.

II. 제안되는 개발 환경의 구조

1. Python을 이용한 강화학습 에이전트의 구현

제안되는 강화학습 기반의 제어기 설계 환경의 구조를 소개하기에 앞서, 본 논문에서 의미하는 강화학습 기반의 제어기에 대한 개념을 먼저 설명하고자 한다.

강화학습 기반의 제어기는 고전적인 제어 방식에서 제어 연산을 수행하는 기존의 제어기 역할을 강화학습 에이전트로 대체한 형태로 정의된다. 강화학습에서 에이전트란 강화학습 알고리즘을 구현한 인공지능 시스템을 의미한다. 이러한 강화학습 에이전트는 주어진 환경에서 자신의 행동 정책에서 도출된 행동을 수행하며, 결과적으로 얻어지는 보상을 기반으로 자신의 행동 정책을 개선하는 과정을 반복한다. 이를 통해 구현되는 강화학습 기반의 제어기는 주어진 시스템의 환경 정보를 입력으로 받아 학습된 행동 정책에 따라 최적의 행동, 즉 제어량을 출력하게 된다.

본 논문에서 제안하는 개발 환경 구축의 핵심 요소 중 하나인 강화학습 에이전트를 구현하는 방법에 있어 Python 언어의 사용이 요구되는데, 이는 Python을 사용함으로써 강화학습 알고리즘을 사용하는데 강력한 편의성을 제공받을 수 있기 때문이다. 강화학습 분야를 포함한 인공지능 연구를 수행함에 있어 Python은 가장 많이 사용되는 프로그래밍 언어이다. TensorFlow[10], PyTorch[11]와 같은 대표적인 딥러닝 및 강화학습 라이브러리와 프레임워크들이 Python을 기반으로 개발되어 있으며, 현재 사용되는 대다수의 강화학습 알고리즘들은 이러한 도구들을 활용하여 구현되어 있다. 이에 근거하여 본 연구는 강화학습 기반 제어기에 사용되는 강화학습 에이전트의 편리하고 신속한 구축을 위해 Python을 사용하였다.

2. Matlab/Simulink와 함께 사용되는 LW-RCP

강화학습 에이전트가 행동 정책을 개선하기 위해 필요한 학습 데이터는 실물 시스템과의 상호작용 과정에서 LW-RCP를 통해 획득할 수 있다.

RCP 시스템은 제어 시스템 개발자들이 단기간에 제어 알고리즘을 효율적으로 설계, 개발 및 검증하기 위해 사용하는 개발 환경을 의미한다. LW-RCP란 저자들이 속한 연구실에서 개발한 오픈소스 하드웨어로 만들어진 경량화 된 RCP 장치로서, 경제성과 편리성 측면에 있어 다른 RCP 시스템에 비해 강점을 가진다[9]. 해당 시스템은 high-speed USB 통신을 사용하여 PC상의 Matlab/Simulink와 데이터를 주고받을 수 있다. LW-RCP의 입력 장치를 통해 수집된 센서 정보는 high-speed USB 통신을 통해 PC로 전송된다. 이후, PC에서 실행 중인 Simulink 기반의 제어기 모델은 전달받은 센서 정보를 기반으로 제어 연산을 수행한다. 연산

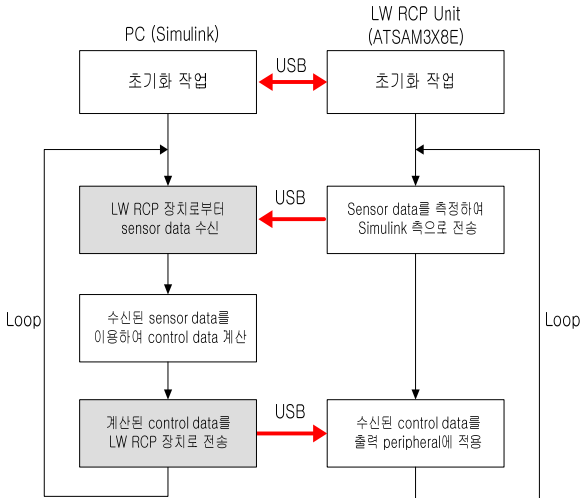

그림 1. LW-RCP의 동작 방식.

Fig. 1. Flowchart of the LW-RCP operation mechanism.

결과로 도출된 제어량은 다시 high-speed USB 통신을 통해 LW-RCP의 hardware unit에 전달되며, 이를 통해 출력 장치와 연결된 실물 시스템 구동기를 물리적으로 제어할 수 있다. 그림 1은 LW-RCP와 PC의 상호작용 과정을 도식화한 흐름도로 나타낸 것이다.

High-speed USB 통신을 기반으로 PC와 LW-RCP가 정보를 교환하기 때문에 두 장치 간의 통신 지연은 high-speed USB 통신의 microframe 발생 주기인 125μs를 초과하지 않고, LW-RCP를 통해서는 최대 2kHz까지의 샘플링 주파수를 갖는 제어 시스템의 실시간 제어 구현이 가능하다.

강화학습을 실물 시스템에 적용하는 연구 결과들을 검토해본 결과[12-13] 로봇 팔 제어의 경우 20Hz의 샘플링 주파수를, 더 복잡한 4족 보행 로봇의 경우 동작에 따라 100-200Hz의 샘플링 주파수를 가지고 제어한 것으로 확인되었다. 강화학습 기반 제어기는 대부분 심층 신경망을 사용한 연산 과정이 필요하여 전통적인 고전제어에 비해 계산 복잡성이 증가하기 때문에 상대적으로 더 긴 샘플링 타임이 요구된다. 그러므로 고전제어를 위해 충분히 빠르게 설계된 2kHz의 샘플링 주파수를 갖는 LW-RCP로도 강화학습기반의 제어기를 사용하여 대부분의 고난도 실물 시스템 제어가 가능하다는 것을 알 수 있다. 이를 통해 LW-RCP를 활용하여 실물 시스템에 강화학습 기반의 제어기를 적용하는 시스템을 구축할 수 있다는 가능성이 입증된다.

그림 2는 LW-RCP에서 지원하는 Simulink 기반의 입/출력 라이브러리 블록의 다양한 종류를 나타낸다. 2열로 구성된 라이브러리 블록들 중 첫 번째 열에 위치한 블록들은 DAQ 장치로부터 데이터를 수신하기 위한 Receive 블록이고, 두 번째 열에 위치한 블록들은 DAQ 장치 쪽으로 데이터를 전송하기 위한 Send 블록들이다. 각 라이브러리 블록이 지원하는 입/출력 장치들의 세부 사항은 표 1에 기술되어 있다. LW-RCP는 다양한 입/출력 기능들을 지원하여 범용성이 높게 설계되었기 때문에, 연구자들이 자신만의 실물 시스템 환경에 대한 상태 정보를 관측하기 위해 필요한 다양한 센서들의 데이터를 취득하는데 용이하다. 이를 바탕으

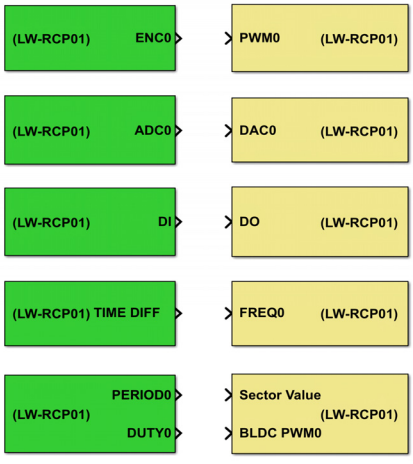

그림 2. LW-RCP에서 지원하는 Simulink 기반의 입/출력 라이브러리 블록.

Fig. 2. Simulink-based I/O library blocks supported by LW-RCP.

표 1. LW-RCP의 입/출력 기능 세부사항.

Table 1. I/O function details of LW-RCP.

	Block name	Description
Input blocks	Encoder counter	2 channels, 32-bit
	ADC	8 channels, 12-bit
	Digital input	8-bit
	Period and duty	3 channels
	Time/Time difference	Resolution: micro second
Output blocks	PWM	6 channels PWM/Direction interface or Unipolar complementary PWM
	DAC	2 channels
	Digital output	8-bit
	Frequency	2 channels, up to 40,000Hz
	BLDC PWM	2 channels

로 본 논문의 3장에서는 제안되는 개발 환경의 검증을 위해 직선형 1단 도립진자의 실물 시스템과 LW-RCP를 사용하여 실물 시스템으로부터 취득한 데이터를 사용해 강화학습 에이전트의 학습이 가능함을 증명한다. 추가적으로 동일한 계열의 LW-RCP 시스템을 사용하여 고전제어 분야에서 Ball-Plate 시스템[14], 2·3단 직선형 도립진자 시스템[15-16]을 제어한 선례가 존재한다. 해당 시스템들은 직선형 1단 도립진자보다 훨씬 더 많은 센서들을 활용해야 하는 시스템으로, 이를 통해 LW-RCP의 다양한 환경에 대한 활용 가능성을 확인할 수 있다.

LW-RCP와 함께 사용되는 Matlab/Simulink는 실시간으로 데이터를 수집하고 처리하는데 적합한 환경으로, 복잡한 제어 시스템의 설계 및 구현에 널리 사용되고 있다. 본 개발 환경에서 Matlab/Simulink는 그림 1에서 설명하는 바와 같이 LW-RCP를 통해 수신된 실물 시스템의 센서 정보를 처리함과 동시에 실물 시스템에 PC상에서 연산된 제어량을 인가하는 제어 시스템 환경의 역할을 수행한다. 본 논문에서 구현하고자 하는 강화학습 기반의 제어기도 Simulink 상에 블록 형태로 위치하게 된다. 상기된 강화학습 기반의 제어기는 학습된 강화학습 에이전트의 신경망 파라미터들을 이용해 입력으로 들어오는 시스템의 환경 정보 데이터를 기반으로 심층 신경망의 연산 과정을 통해 최적의 행동을 도출하게 된다. 연구자가 선정한 강화학습 알고리즘에서 사용하는 심층 신경망과 동일한 기능을 수행할 수 있는 블록 구현의 편의를 위해 Matlab에서는 Deep Learning Toolbox를 지원하고 있으며, 상대적으로 간단한 구조의 심층 신경망은 Matlab function block을 활용하여 직접 구현할 수 있다. 또한 Matlab/Simulink 환경에서 자체적으로 임베디드 코드 생성을 위한 기능을 제공하고 있으므로, 임베디드 시스템에 적용하는 상황에 있어서도 편의를 제공받을 수 있다.

3. Matlab/Simulink와 Python을 결합한 통합 개발 환경

제안하는 개발 환경은 실시간으로 실물 시스템의 데이터 취득 및 처리, 수신된 데이터 기반 제어량 연산 및 제어 신

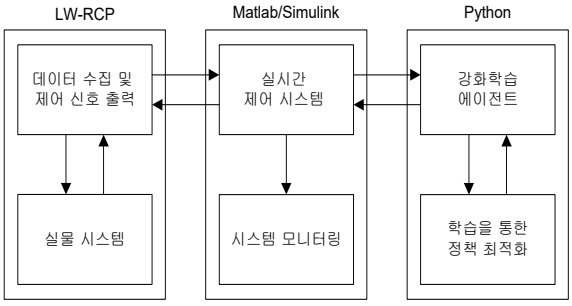

그림 3. 제안된 개발 환경의 구성도.

Fig. 3. Concept diagram of the proposed development environment.

호 입/출력, 이를 바탕으로 하는 강화학습 알고리즘 기반의 강화학습 에이전트 생성 및 사용 등 일련의 과정을 구현하기 위해 그림 3에 보이는 개념도와 같이 세 가지 하부 시스템을 결합하여 사용한다. Matlab/Simulink는 중단에 위치하여 실물 시스템의 실시간 제어 시스템의 역할을 수행하며, LW-RCP에서 관측된 데이터를 Python까지 전달한다.

하부 시스템 간 통합 개발 환경을 구성하기 위해 Matlab이 제공하는 'matlab.engine' Python API를 활용한다. 해당 기능은 Matlab에서 개발한 기능을 Python 환경에서 사용할 수 있도록 지원하기 위해 Matlab 측에서 제공하는 기능으로 이 API를 사용하면 Python 코드 내에서 Matlab 함수를 호출하고 실행할 수 있고, 그 반대의 경우도 가능하다. 더불어, Matlab과 Python간의 작업 공간에 서로 접근하여 저장된 변수들을 자유롭게 공유하고 활용할 수 있어 두 언어의 장점을 모두 활용한 효율적인 개발이 가능해진다.

상기된 기법들을 활용하여 Python과 Matlab/Simulink가 결합된 형태의 개발 환경의 진행 순서는 그림 4에 보이는 흐름도를 따른다. 전체적인 과정은 다음과 같이 설명할 수 있다. 먼저, Python에서 생성된 강화학습 에이전트는 자신의 행동 정책에 사용되는 파라미터 값, 즉 심층 신경망의 가중치 값과 편향 값을 변수로 저장하고, Python API를 이용해 Matlab의 작업 공간에 전달한다. 이후, Simulink는 실시간 제어 시스템 모델 파일에 전달받은 파라미터들을 매개변수로 사용하여 강화학습 에이전트와 동일한 행동 정책을 수행하는 심층 신경망 블록을 구성한 뒤 Simulink 모델 파일을 실행하여 실물 시스템과 상호작용한다. 이 과정에서 Simulink는 to Workspace 블록을 사용하여 각 상호작용마다 그 순간의 환경 상태 정보, 취했던 행동, 설계된 보상함수에 의한 보상 값의 데이터를 Matlab의 작업공간에 저장한다. Simulink 모델은 실물 시스템과 상호작용을 반복하다가 연구자가 설정한 에피소드의 종단 시간이 지나거나, 미리 설정해둔 특정 종료 조건에 부합하는 상황이 발생할 경우 제어 모델을 종료한다. Simulink 모델이 종료된 후, Matlab은 한 에피소드동안 저장된 학습용 데이터들을 Python API를 이용하여 Python으로 전송한다. Python은 해당 데이터들을 저장공간에 받아온 다음 지정된 횟수만큼 강화학습 에이전트의 학습, 즉 정책 최적화를 진행한다. 이후 위 과정을 연구자가 원하는 수준의 결과가 나올 때까지 반복하여 강화학습 기반 제어기의 성능 향상을 이룬다.

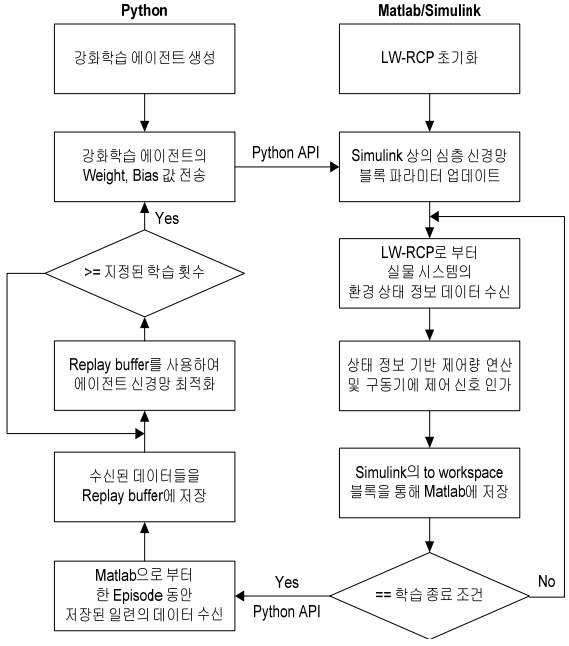

그림 4. Matlab/Simulink와 Python의 흐름도.

Fig. 4. Flowchart of the Matlab/Simulink and Python.

III. 강화학습 기반의 1단 직선형 도립진자의 swing-up 제어기 구현

2장에서 제시한 개발 환경의 유효성을 검증하기 위해 그림 5에 도시된 직선형 1단 도립진자의 실물 시스템을 대상으로 강화학습 알고리즘을 이용한 제어기 구현 실험을 수행하였다. 본 실험에서 강화학습 에이전트는 실물 시스템에 대한 사전 정보를 전혀 알지 못한 상태에서 오로지 실물 시스템과의 상호작용을 통해 얻은 데이터를 기반으로 행동 정책을 개선하게 된다.

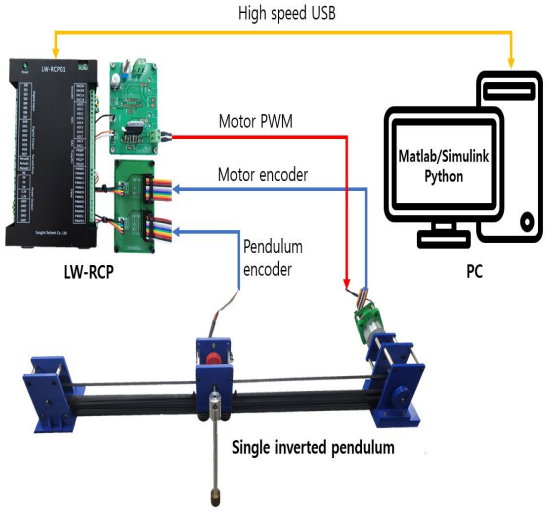

그림 5. 제안된 개발 환경의 하드웨어 배치도.

Fig. 5. Hardware layout of proposed development environment.

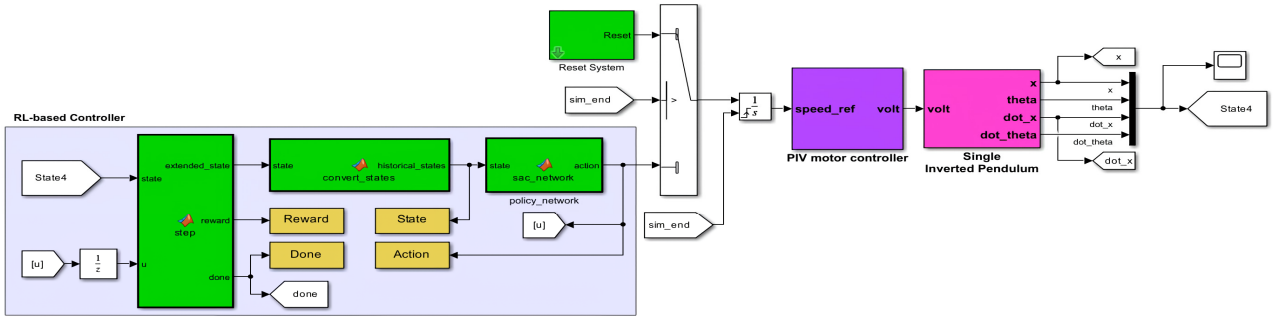

그림 6. Simulink로 구현된 강화학습 기반 제어기를 사용하는 실물 시스템 제어 환경.

Fig. 6. Real-world system control environment utilizing the reinforcement learning-based controller implemented in Simulink.

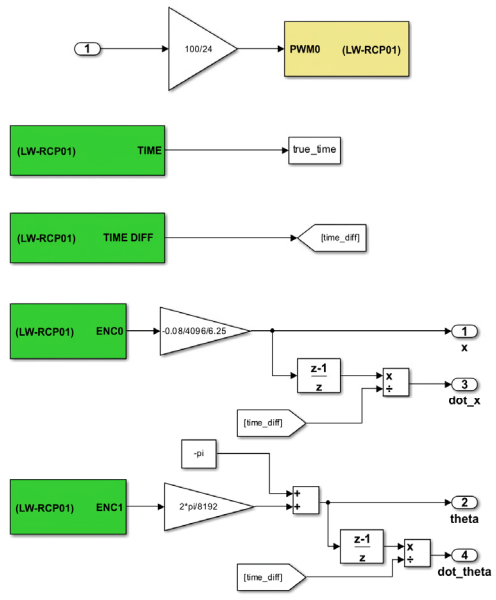

그림 7. LW-RCP의 I/O 블록들을 이용해 구성된 single inverted pendulum 서브 시스템.

Fig. 7. Single inverted pendulum subsystem constructed using LW-RCP's I/O Block.

본 실험에서는 model-free 강화학습 알고리즘 중 많이 사용되는 SAC (Soft Actor Critic)을 채택하였다. 연속적인 행동 공간을 지닌 복잡한 환경에서 높은 효율성과 안정성을 입증한 SAC 알고리즘은 최대 엔트로피 항을 학습 과정에 추가함으로써 탐험을 통한 행동 정책의 다양성과 안정성을 향상시킬 수 있는 특징을 갖고 있다[17]. 본 논문은 강화학습 기반의 제어기 개발 환경 구조 제안에 중점을 두고 있으므로, 강화학습 알고리즘의 세부적인 수식과 증명은 별도로 다루지 않기로 한다. SAC 알고리즘에 대한 상세한 내용은 참고문헌[17]에서 확인할 수 있다.

그림 6은 Simulink를 통해 구현된 강화학습 기반의 제어기를 사용하는 실물 시스템 제어 환경을 나타낸다. 본 실험에서 LW-RCP를 활용하여 관측 가능한 실제 시스템의 환경 정보는 카트의 초기 위치로부터의 변위 x와 진자의 회전 변위인 지면에 대한 법선과 이루는 각 θ이다. 이때, θ값은 추후 원활한 보상함수의 설계를 위해 나머지 연산을 적용하여 $-\pi<\theta<\pi$의 범위로 제한한다. 이 값들을 활용하여 Simulink 상에서 연산을 통해 추가적으로 카트의 속도 $\dot{x}$와 진자의 각속도 $\dot{\theta}$를 도출할 수 있으며, 이는 그림 7을 통해 확인할 수 있다. Simulink는 해당 정보를 재구성하여 x, $\sin(\theta)$, $\cos(\theta)$, $\dot{x}$, $\dot{\theta}$로 이루어진 5개의 데이터로 해당 시점의 환경 상태 정보를 구성한다. 그렇게 구성된 상태 정보는 참고문헌[18]의 저자들이 사용했던 방식에서 착안하여, 과거의 상태 정보들과 결합한 형태로 변환되어 심층 신경망으로 이루어진 강화학습 기반 제어기에 입력으로 전달된다. 입력을 기반으로 제어기는 행동 정책, 즉 심층 신경망의 연산을 통해 제어량을 출력한다. 출력되는 제어량은 모터의 가속도 값 u에 해당하며, 구동기의 작동 능력을 고려하여 -15에서 15 사이의 범위로 제한한다.

본 실험에서는 한 에피소드의 길이를 15초로 설정한다. 강화학습 기반의 제어기는 10ms 주기로 1단 도립진자의 상태 정보를 입력받아 행동 정책에 따른 모터 가속도 값을 계산한다. 이때 산출된 가속도 값은 적분기를 통해 모터의 속도 추종치로 전환되며, 이는 1ms 주기로 PIV 속도 제어기를 통과한 뒤 LW-RCP의 Send 블록을 사용하여 모터에 PWM 신호로 인가된다.

강화학습 기반의 제어기가 swing-up 동작을 수행한 뒤, 카트의 초기 위치에서 진자의 도립 상태를 성공적으로 유지하도록 학습시키기 위해 참고문헌[19]을 참조하여 다음과 같이 네 가지 요소를 결합한 형태의 보상함수를 설계하였다.

$$\begin{aligned} R_x &= 0.4 + 0.6\,e^{-5x^2} \\ R_\theta &= 0.5 + 0.5\cos\theta \\ R_{\dot{\theta}} &= 0.4 + 0.6\,e^{-0.01\dot{\theta}^2} \\ R_u &= 0.7 + 0.3(1-(u/15)^2) \end{aligned} \tag{1}$$

$$Reward = R_x \times R_\theta \times R_{\dot{\theta}} \times R_u \tag{2}$$

상기된 보상함수를 이루는 각각의 요소는 그림 8에서 확인할 수 있으며, 모든 항은 0에 수렴할수록 값이 증가하는 특성을 나타낸다. 이를 통해 카트는 원점에 최대한 근접하게 위치하도록, 진자는 도립된 상태에서 최소한의 움직임을 유지하도록 하며, 제어 입력량을 최소화하는 방향으로 행동 정책을 학습하게 된다.

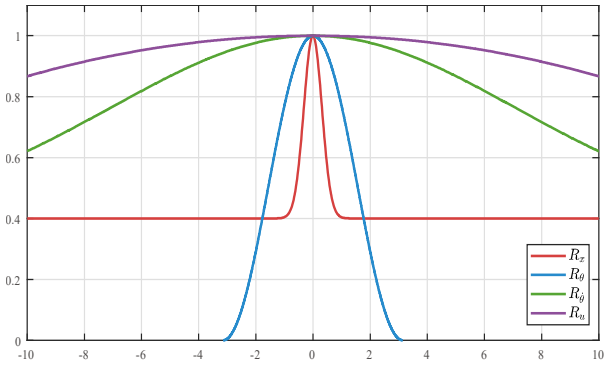

그림 8. 보상함수 그래프.

Fig. 8. Reward function graph.

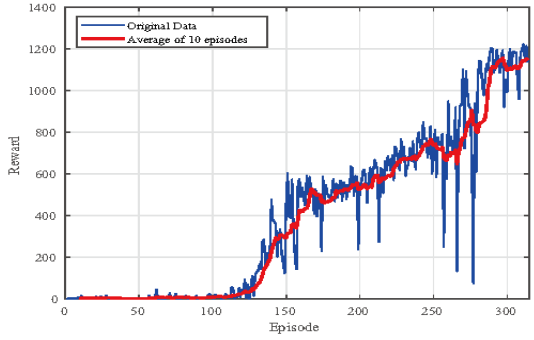

그림 9. 학습 결과 그래프.

Fig. 9. Learning results graph.

추가적으로 x의 값이 0.2[m]를 초과하거나, $\dot{\theta}$의 값이 25[rad/sec]를 초과하는 경우 해당 에피소드는 학습에 도움이 되지 않다고 판단하여 해당 시점에서 조기 종료하고, 이때까지 얻어진 데이터만을 가지고 바로 Python의 학습 과정으로 넘어가게 된다.

상기된 조건의 실험 환경에서 에피소드를 반복하며 실험을 진행하였을 때, 그림 9에서 보이는 바와 같이 약 280회의 에피소드 반복을 거쳐 성공적으로 도립 상태를 유지하는 행동 정책을 학습함을 보였다.

학습이 종료된 해당 시점에서의 환경 상태 정보를 Simulink의 시각화 블록을 사용해 확인한 결과는 그림 10과 같이 나타난다. 이는 강화학습 기반의 제어기를 통해 1단 직선형 도립진자 제어기의 설계 목표였던 swing-up 제어 이후 원점에 근접한 상태에서 도립 상태를 완벽하게 유지할 수 있음을 나타낸다.

IV. 결론

본 논문에서는 LW-RCP라는 rapid control prototyping 시스템과 Matlab/Simulink 및 강화학습 연구에서 가장 활발하게 사용되고 있는 Python을 결합하여 실물 시스템에 대한 강화학습 기반의 제어기 개발 환경을 제안하였다.

제안된 개발 환경을 통해 강화학습 연구자들은 강화학습 에이전트와 실물 시스템 간 상호작용에 필요한 인터페이스 구현에 소모되는 추가적인 노력과 시간을 절약할 수 있다.

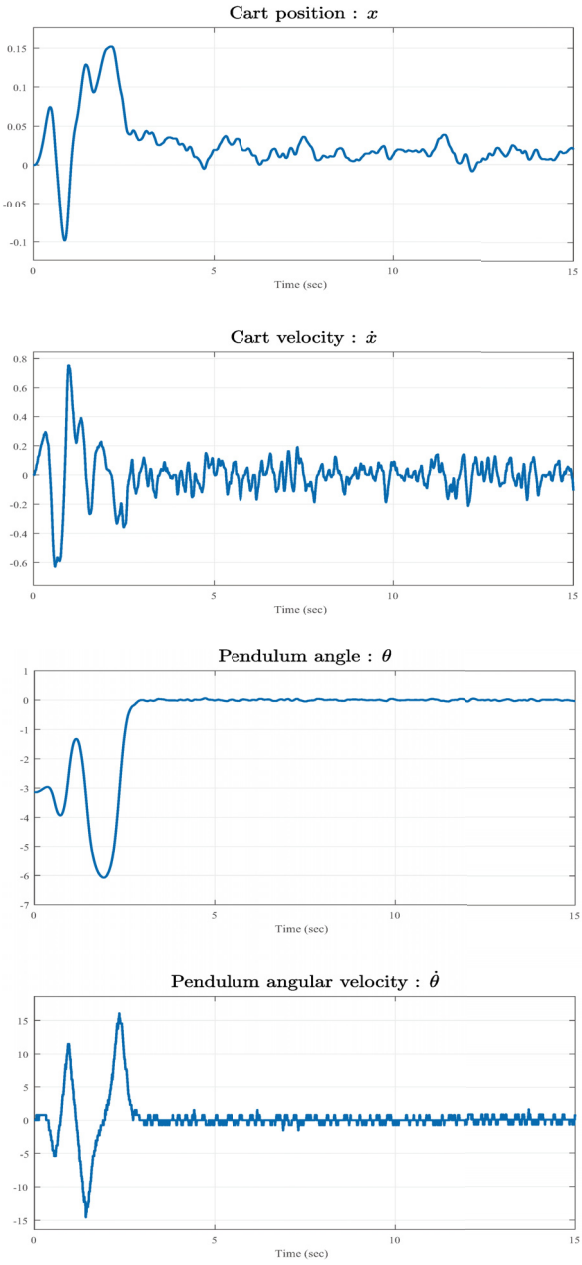

그림 10. 학습 완료 시점의 환경 상태 정보 그래프.

Fig. 10. Environment state information graph at the point of learning completion.

결과적으로 연구자들은 강화학습 알고리즘 설계라는 핵심 연구 목적에 충실할 수 있게 되며, model-free 강화학습 알고리즘의 본질적인 장점을 활용한 연구 수행에 접근하기 쉬워진다. 더불어, 개발 환경을 통해 강화학습 에이전트와 실물 시스템이 실시간으로 어떻게 상호작용하고 있는지 직관적으로 관찰할 수 있다. 해당 시점에서 측정된 실물 시스템의 환경 상태 정보에 관한 데이터와 본인이 설계한 보상함수의 값 등 연구에 사용되는 다양한 데이터를 Simulink상의 시각화 블록들을 활용하여 즉각적으로 확인할 수 있다. 이 같은 기능은 알고리즘 및 보상함수의 설계 성능과 실험의 진행 향방을 실시간으로 파악할 수 있게 함으로써, 연구자의 통찰력을 향상시킬 것으로 기대된다.

REFERENCES

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238 - 1274, 2013.

[2] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 29, no. 6, pp. 2042 - 2062, 2017.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction,* 2nd ed., The MIT Press, 2018.

[4] 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 preprint arXiv:1509.02971*, 2015.

[5] V. Franois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Foundations and Trends in Machine Learning*, vol. 11, no. 3-4, pp. 219 - 354, 2018.

[6] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *arXiv preprint arXiv:1904.12901*, 2019.

[7] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026 - 5033, 2012.

[8] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," *arXiv preprint arXiv:1606.01540*, 2016.

[9] Y. S. Lee, B. Jo, and S. Han, "A light-weight rapid control prototyping system based on open source hardware," *IEEE Access*, vol. 5, no. 1, pp. 11118 - 11130, 2017.

[10] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," *Osdi*. vol. 16. no. 2016, 2016.

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems.* vol. 32, 2019.

[12] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," *2017 IEEE international conference on robotics and automation (ICRA)*. pp. 3389-3396, 2017.

[13] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[14] J. Park, H. Bang, and Y. S. Lee, "A study on the implementation of a ball and plate system using LW-RCP and machine vision based on odroid," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 26, no. 4, pp. 213 - 221, 2020.

[15] D. Ju, C. Choi, J. Jeong, and Y. S. Lee, "Design and parameter estimation of a double inverted pendulum for model-based swing-up control," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 28, no. 9, pp. 793 - 803, 2022.

[16] C. Choi, D. Ju, J. Jeong, and Y. S. Lee, "Structural proposition for a triple inverted pendulum and implementation of swing-up control using an LW-RCP02," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 28, no. 10, pp. 916 - 925, 2022.

[17] 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.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, pp. 529 - 533, 2015.

[19] J. Baek, H. Jeon, J. Park, and S. Han, "Control of inverted pendulum on a cart via reinforcement learning," *Proc. of 2019 34th ICROS Annual Conference (in Korean)*, pp. 313 - 314, 2019.

이 태 건

2023년 인하대학교 전기공학과 졸업.
2023년~현재 인하대학교 대학원 전기컴퓨터공학과 석사과정 재학 중. 관심분야는 강화학습, 임베디드 시스템, 최적제어.

주 도 윤

2023년 인하대학교 대학원 전기컴퓨터공학 석사 졸업.
2023년~현재 동 대학원 박사과정 재학 중. 관심분야는 최적제어, 임베디드 시스템, 강화학습.

이 영 삼

제어 · 로봇 · 시스템학회 논문지, 제 15권 제 4호 참조.