한국정밀공학회지 제 40 권 제 6 호 pp. 467-472
J. Korean Soc. Precis. Eng., Vol. 40, No. 6, pp. 467-472
June 2023 / 467
http://doi.org/10.7736/JKSPE.022.124
ISSN 1225-9071 (Print) / 2287-8769 (Online)

강화학습을 이용한 초정밀 가공기용 챔버의 온도 제어

Application of Deep Reinforcement Learning to Temperature Control of a Chamber for Ultra-precision Machines

김병섭[1,#], 노승국[1]
Byung-Sub Kim[1,#] and Seung-Kook Ro[1]

1 한국기계연구원 초정밀장비연구실 (Department of Ultra-precision Machines and Systems, Korea Institute of Machinery & Materials)
Corresponding Author / E-mail: bkim@kimm.re.kr, TEL: +82-42-868-7109
ORCID: 0009-0007-2023-8162

KEYWORDS: Temperature control (온도 제어), Reinforcement learning (강화학습), Artificial intelligence (인공 지능), Ultra-precision machines (초정밀 가공기), Chamber (챔버), DQN algorithm (DQN 알고리즘)

Deep reinforcement learning (RL) has attracted research interest in the manufacturing area in recent years, but real implemented applications are rarely found. This is because agents have to explore the given environments many times until they learn how to maximize the rewards for actions, which they provide to the environments. While training, random actions or exploration from agents may be disastrous in many real-world applications, and thus, people usually use computer generated simulation environments to train agents. In this paper, we present a RL experiment applied to temperature control of a chamber for ultra-precision machines. The RL agent was built in Python and PyTorch framework using a Deep Q-Network (DQN) algorithm and its action commands were sent to National Instruments (NI) hardware, which ran C codes with a sampling rate of 1 Hz. For communication between the agent and the NI data acquisition unit, a data pipeline was constructed from the subprocess module and Popen class. The agent was forced to learn temperature control while reducing the energy consumption through a reward function, which considers both temperature bounds and energy savings. Effectiveness of the RL approach to a multi-objective temperature control problem was demonstrated in this research.

Manuscript received: October 18, 2022 / Revised: March 8, 2023 / Accepted: March 9, 2023

NOMENCLATURE

ε	=	Probability of Random Action
ϕ	=	Function Representing State Histories with a Fixed Length
θ	=	Weight Vector in a Network
γ	=	Reward Discount Factor
π	=	Policy
a_t	=	Agent Action at Time Step t
P	=	Probability Distribution
Q	=	Action-value Function
$\hat{Q}$	=	Target Action-value Function
r_t	=	Reward at Time Step t
s_t	=	Environment State at Time Step t
T_{ref}	=	Reference Temperature
T_{room}	=	Ambient Temperature
T_{sp}	=	Set Point Temperature for PID Controller
T_t	=	Chamber Temperature at Time Step t

1. 서론

인공지능 기계학습(Machine Learning)은 크게 지도 학습(Supervised Learning)과 비지도 학습(Unsupervised Learning) 그리고 강화학습(Reinforcement Learning)의 세 개 영역으로 나뉜다.

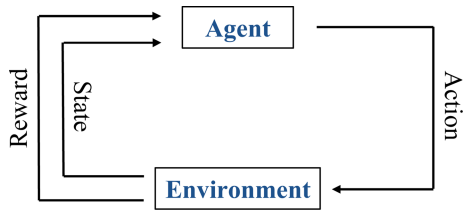

Fig. 1 Reinforcement learning cycle

강화학습 분야는 2016년 알파고의 등장으로 폭발적인 관심과 더불어 다양한 영역에서 활용이 될 수 있을 것으로 기대되었으나 물리적 현실에서의 활용보다는 컴퓨터 게임과 가상 시뮬레이션 환경에서 주로 활용 성과가 나오고 있다.

그 이유 중의 하나는 강화학습 훈련 과정에서 에이전트(Agent)가 주어진 환경(Environment)을 임의의 행동(Action)으로 탐험을 하고, 그에 따른 보상(Reward)을 통해서 상태(State)와 행동에 대한 가치(Value)를 배워 나가야 하는데, 가상 환경이 아닌 경우에 에이전트의 임의 행동에 따른 탐험 자체가 환경에 너무 위험할 수 있기 때문이다. 예를 들어 강화학습으로 로켓의 발사 제어를 훈련하고자 한다면 실제 훈련으로는 수없이 추락하는 로켓 비용을 감당할 수 없기 때문에 컴퓨터 시뮬레이션 환경이 필수적일 것이다. 또한 강화학습은 학습에 필요한 데이터 양이 지도 학습과 비지도 학습에 비하여 월등히 많기 때문에 데이터 확보를 위한 시간 제약을 해결하기 위하여 현실 환경보다 컴퓨터 시뮬레이션 환경이 강화학습 연구에 더 적합할 수 있다.

공조 시스템(Heating, Ventilation, and Air Conditioning, HVAC)은 복잡한 다중 정책(Policy)을 학습할 수 있는 강화학습의 장점 때문에 강화학습을 이용하는 연구가 활발히 진행되고 있는 분야 중의 하나이다. 하지만, Wang은 온도 제어, 특히 빌딩을 대상으로 강화학습을 적용하는 연구와 관련하여 조사한 77편의 연구 논문 중 11%만이 실제 빌딩 환경이었고 나머지는 시뮬레이션 환경인 것으로 보고하였다[1]. Natale 등은 방 온도를 제어하는 에이전트를 강화학습으로 1차 시뮬레이션 환경에서 훈련한 후에 실제 빌딩에서 2차 훈련하는 방법을 소개하고 상대적으로 적은 시간 훈련하여 쾌적함과 에너지 절감의 두가지 목적을 동시에 만족하는 제어기를 구성하였다[2]. Zhang 등은 강화학습 Asynchronous Advantage Actor-Critic (A3C) 알고리즘을 이용하여 EnergyPlus라는 시뮬레이션 환경에서 사무용 빌딩의 공조 관리용 신경망을 훈련시킨 후 실제 빌딩에 적용하여 16.7%의 난방 요구를 절감시키는 효과를 거두었다[3]. Vàsquez-Canteli 등은 강화학습 Deep Q-Learning 알고리즘을 CitySim이라는 시뮬레이션 환경에서 히트 펌프(Heat Pump) 제어에 적용하여 기존에 사용하던 법칙 기반의 제어기에 비하여 10%의 에너지 절감을 달성하였다[4]. Brandi 등은 강화학습 Double Deep Q-Learning 구조와 EnergyPlus라는 시뮬레이터를 사용하여 내부 거주자 수의 변화와 내부 온도 설정이 동적으로 변하는 환경에서도 강화학습으로 만든 에이전트가 유연하게 작동하여 시나리오에 따라 5%에서 12%까지 난방 에너지를 절약할 수 있었다고 보고하였다[5].

컴퓨터 시뮬레이션을 통해 훈련된 강화학습 에이전트가 실제 환경에 잘 적용되기 위해서는 현실과 매우 유사한 환경의 시뮬레이션 모델이 필요하며 경우에 따라서는 시뮬레이션 환경을 구축하기 위해서 온도 제어 모델을 설계하는 것보다 더 복잡하고 어려운 시뮬레이션 모델 작업을 해야 할 수도 있다.

빌딩의 공조 시스템에서 미세 온도 조절 목표 범위는 대략 ±0.5°C 수준인데 반하여 본 연구에서는 그 보다 작은 ±0.1°C 이내로 초정밀 가공기용 소형 챔버의 온도 제어를 하고자 하였으며 또한 컴퓨터 시뮬레이션 환경이 아니라 실제 초정밀 가공기용 온도 제어 챔버에서 강화학습 신경망의 훈련을 시도하였다. C-언어로 구동되는 제어기 하드웨어와 Python으로 구현된 강화학습 알고리즘과의 연동을 위해 Subprocess 모듈과 Popen이라는 클래스를 사용하여 측정 데이터 및 제어 명령의 통로를 구축한 특징이 있다. 많은 제어용 하드웨어들이 C-언어를 기본으로 지원하는데 반하여 인공지능에서는 Python 언어가 주류를 이루고 있기 때문에 동기화를 구현하는 과정에서 어려움이 생기는데 본 연구에서는 두 시스템 사이에 주종(Master-Slave)의 관계로 데이터 파이프라인을 구축하여 1 Hz의 속도에서 충분히 실제 장비에서도 강화학습 알고리즘으로 에이전트를 훈련시키고 다중 목적의 온도 제어기를 구성할 수 있음을 실험적으로 보였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 연구에 사용된 강화학습 Deep Q-Network (DQN) 알고리즘을 소개하고 에너지 절감의 제한 조건 하에서 정밀한 온도 제어를 하기 위하여 보상 함수를 어떻게 구성했는지 설명한다. 3장에서는 강화학습 알고리즘을 실제 장비와 온라인으로 연결하여 훈련시키기 위한 하드웨어 구동부와 강화학습 알고리즘 사이의 데이터 파이프라인의 구축 방안을 제시한다. 4장에 에이전트의 실제 훈련이 이루어진 실험 경과 및 결과를 보이며 5장에 결론을 제시한다.

2. 강화학습 알고리즘

연구에서 사용된 강화학습 알고리즘은 2015년 Nature에 발표된 Google DeepMind 팀의 Deep Q-Network이다[6]. Q함수를 신경망을 통해 근사하고자 할 때 학습의 불안전성과 수렴이 잘 되지 않는 문제를 해결하고자 DQN은 경험 재생(Experience Replay)과 타겟망(Target Network)의 사용을 방안으로 제시하였다.

경험 재생은 연속적으로 수집된 샘플 사이의 종속적인 상관관계(Sample Correlation)를 제거하고 독립적인 관계로 만들기 위해 재생 버퍼(Replay Buffer)를 만들어서 샘플을 많이 축적한 후에 재생 버퍼에서 임의 추출된 샘플들로 훈련을 진행하는 것이며 타겟망 사용은 주신경망(Main Q-Network)의 웨이트(Weight) θ를 업데이트할 때 목표 값이 움직이는 것을 방지하기 위하여 타겟망으로 목표 값을 고정시켜 두고 일정 횟수만큼 주신경망의 θ를 업데이트한 후에 주기적으로 타겟망이 주신경망을

Table 1 DQN algorithm [7] (Adapted from Ref. 7 on the basis of OA)

Algorithm: Deep Q-learning with Experience Replay

Initialize replay memory D to capacity N
Initialize action-value function Q with random weights θ
for episode = 1, M **do**
  Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for t = 1, T **do**
    With probability ε select a random action a_t
    otherwise select $a_t = \text{argmax}_a\, Q(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D
    Set $y_j = r_j$ for terminal ϕ_{j+1}
    otherwise set $y_j = r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta)$ for non-terminal ϕ_{j+1}
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ
    (Refer to [6] for target action-value function $\hat{Q}$ part)
  End for
End for

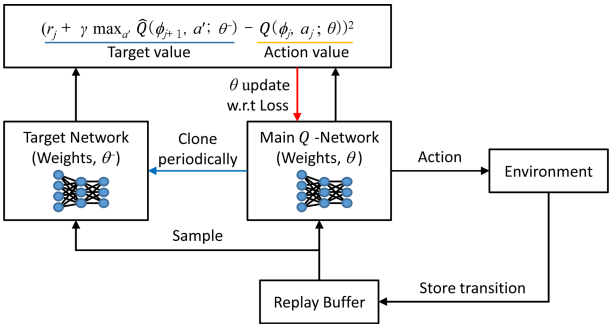

Fig. 2 An illustration of DQN architecture

복제하도록 하면서 훈련을 진행하는 것이다. 개략적인 DQN 알고리즘의 진행 단계와 구조를 Table 1과 Fig. 2에 나타내었다.

강화학습 알고리즘은 Python 언어로 PyTorch 프레임워크를 이용하여 구현하였으며 주신경망 Q-Network은 완전 연결층(Fully Connected Layer)으로 구성하였다. Q-Network은 입·출력층을 제외하고 2개의 은닉층이 있으며 각각 9→32→64→3개의 노드를 사용하고 활성화 함수(Activation Function)로는 출력층에서만 선형 함수를 쓰고 나머지 층들은 ReLU (Rectified Linear Unit) 함수를 사용하였다.

상태 벡터에 해당하는 입력 요소는 9개(내부 PID 제어기의 온도 설정값 T_{sp}, 현재 챔버 온도 T_i, 온도 차이 T_i - T_{sp}, 내부 PID 제어 출력값, 실내 온도 T_{room}, 과거 챔버 온도들 T_{i-1}, T_{i-2}, T_{i-3}, T_{i-4})이며 에이전트의 환경을 나타낸다. 신경망의 출력은 내부 PID 제어기의 온도 설정값 T_{sp}를 +0.01°C 올리거나, 변동 없이 유지하거나, -0.01°C만큼 떨어뜨리는 3개의 행동을 결정하는 것으로 정의하였다. 내부 PID 제어기는 에너지 소비를 고려하지

Fig. 3 Experimental chamber for ultra-precision machines

않고 빈번하게 히터(Heater)와 쿨러(Cooler)를 번갈아 가며 키고 끄기를 반복하며 두 기기의 작동하는 시간 차를 이용하여 다소 공격적으로 T_{sp}를 추종하여 작동하도록 되어있기 때문에 강화학습 제어기가 외부 환경 및 에너지 소비를 모두 고려하여 T_{sp}를 조절하도록 한 것이다.

Fig. 3에 실험에 사용된 온도 제어 챔버의 모습을 보이고 있다. 챔버 내부에는 초정밀 가공기가 위치해 있으며 챔버 뒤쪽에 설치된 히터와 쿨러가 에어 덕트(Air Duct)를 통하여 챔버 내부의 공기 온도를 조절한다. 챔버 내부의 3개 장소에 저항 온도센서(Resistance Temperature Detector, RTD)를 설치하여 3개 센서의 평균으로 챔버 내부 온도 T_i를 계산한다. 강화학습 제어기와 기존의 내부 PID 제어기 루프는 동기화되어 1 Hz 주기로 작동하며 T_i를 기준온도 T_{ref}에서 ±0.1°C 이내로 유지하면서 에너지를 가능한 적게 사용하도록 보상을 부여하였다. T_{ref}는 초정밀 가공을 위해 필요한 물리적 온도를 말하며 실험에서는 편의상 23.5°C로 설정하였는데 강화학습 제어기는 결국 T_{sp}를 T_{ref} 주변에서 조정하여 보상을 최대로 받도록 훈련된 행동을 한다.

강화학습 DQN 알고리즘은 1 Hz로 온도 제어와 훈련을 반복하여 신경망을 업데이트한다. 훈련은 300초를 한 개의 에피소드(Episode) 단위로 설정하여 매 1초마다 주어지는 보상값을 에피소드 별로 합산하여 훈련 과정을 모니터링하였다. 보상값은 한 사이클마다 계산하는데 T_i가 T_{ref}에서 ±0.1°C 이내일 때는 1점을 주고 선형적으로 감소하여 ±0.5°C에 다다르면 0점이 되도록 하였고, 에너지를 가능한 적게 사용하도록 벌점을 두어 PID의 출력을 최대 출력 절대값이 1이 되도록 정규화(Normalization)한 후에 그 값에 0.2점을 곱하여 보상에서 빼도록 구성하였다. 따라서 한 개 에피소드에서 PID 제어 출력을 100% 사용하고도 온도 범위에서 점수를 전혀 못 받을 경우에는 최저 -60점(300×0.0+300×-0.2 = -60)이며, 한 개 에피소드에서 제어 출력을 100% 사용하여 챔버 온도 T_i를 T_{ref}±0.1°C 이내로 유지하는 경우 +240점(300×1.0+300×-0.2 = +240)을 보상으로 받는다. 비현실적인 상황이지만 한 개 에피소드 동안에 PID 제어 출력을 0으로 유지하여 히터와 쿨러를 전혀 가동시키지 않고도 챔버 온도 T_i가 T_{ref}±0.1°C 사이를 유지한다면 에이전트는 +300점(300×1.0+300×0.0 = +300)의 최대 보상을 받을 수 있다. Fig. 4에 1초마다 계산되는 보상 함수를 그래프로 나타내었다.

실험에서 강화학습 훈련은 72시간, 총 864개의 에피소드를 통하여 진행되었다. 사용된 재생 버퍼의 크기는 10,000으로 2시간

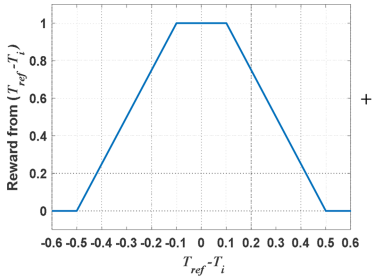

Fig. 4 Computation of reward r_i at each time step

47분간 수행하면 버퍼 전체가 새로운 데이터로 갱신되며 재생 버퍼가 완전히 채워진 후에 주신경망의 θ를 업데이트하기 위한 과정을 시작하였다. 에이전트가 임의의 행동을 취할 확률 ε는 시작할 때 100%로 설정하였고 150,000 스텝(41시간 40분에 해당)이 지나면 최종 1%로 선형적으로 감소되도록 하였으며 점차 훈련된 신경망의 정책(Policy) π에 따라 행동을 결정하도록 하였다. 훈련 과정에서 챔버 온도가 지나치게 오르거나 내려가지 않도록 정책에 상관없이 T_{sp}는 T_{ref}±0.5°C를 상한과 하한으로 설정하여 범위 밖으로 나가지 못하도록 하였다. 훈련 시 보상 할인 인자(Reward Discount Factor) γ는 0.99, 샘플의 배치 크기(Batch Size)는 32, 학습률(Learning Rate)은 1e-4, 타겟망이 주신경망을 복제하는 주기는 1,000 스텝(16분 40초 마다 수행)으로 설정하였다.

3. Python 기반 강화학습 알고리즘과 C-언어 구동 제어 하드웨어의 연동

Python이 인공지능 연구에 주류 언어로 사용되고 있지만 제어용 보드들은 처리 속도때문에 Python보다 C-언어를 지원하는 경우가 대부분이다. Fig. 5에 실험에 사용된 NI 사의 Real-time 제어기 PXIe-8135를 나타내었는데 이 장비 또한 Python 언어가 아니라 C-언어로 프로그래밍하도록 되어 있다.

전체 프로그램의 구성은 호스트 PC에서 구동하는 Python 언어로 된 강화학습 DQN 코드와 같은 호스트 PC에서의 NI 호스트용 C-프로그램, 그리고 NI PXIe-8135 하드웨어에서 구동하는 Real-time용 C-프로그램까지 총 3개의 프로그램 모듈로 되어있다. 내부 PID 제어기가 NI Real-time 하드웨어에서 1초 주기로 사이클이 반복되고 있는 상황에서 호스트 PC에서 돌아가는 Python DQN 알고리즘의 명령이 같은 PC에서 돌아가고 있는 NI 호스트 프로그램으로 명령을 넘기거나 측정된 온도값 등을 전달받기 위해서 Python 언어에서 제공하는 Subprocess.Popen() 함수를 사용하였다. 또한 호스트 PC와 NI Real-time 하드웨어 사이는 네트워크 변수를 사용하여 값들을 넘겨주도록 구성하였는데 3개 프로그램 모듈 간의 데이터 통신 관계를 Fig. 6에 개략적으로 나타내었다.

Python DQN 코드와 NI 호스트용 C-프로그램 사이의 데이터

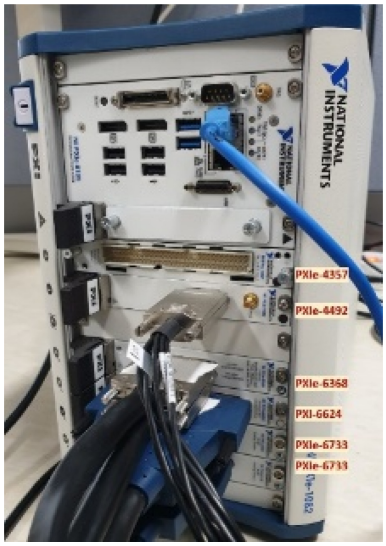

Fig. 5 NI real-time hardware (PXIe-8135)

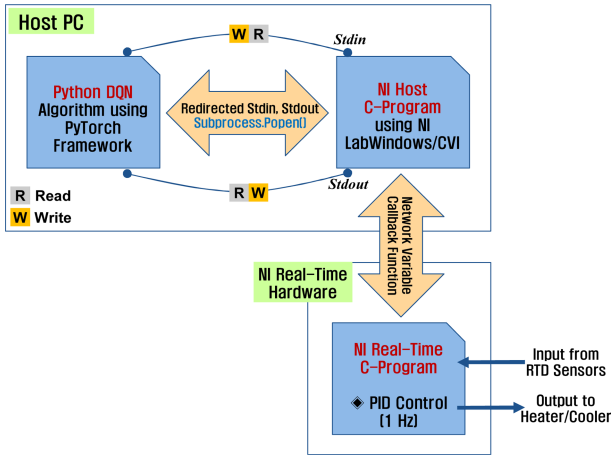

Fig. 6 Data pipeline structure between the Python-code and C-programs

파이프라인 구축 방법은 Ref. 8을 참조하여 콘솔(Console)형 입출력 함수로 Python 코드에서 C-프로그램으로 명령을 내려서 NI 호스트용 C-프로그램이 명령을 직접 수행하게 하거나 NI Real-time C-프로그램 쪽으로 명령을 전달하는 구조로 만들었으며, NI 호스트용 C-프로그램과 NI PXIe-8135 Real-time용 C-프로그램 사이의 통신은 Ref. 9의 방법으로 연결하여 호스트 C-프로그램의 요구를 Real-time C-프로그램이 1 Hz로 돌면서 수행하도록 하였다.

계산 시의 부하 측면에서 NI Real-time 하드웨어는 온도 센서 신호 수집 및 PID 제어기 구동에 여유가 많은 상태였고, PC 쪽은 Python DQN 코드와 NI 호스트용 C-프로그램의 동시 수행을 인텔 Xeon E3-1535M 2.9 GHz가 탑재된 노트북을 이용하였는데 Real-time 하드웨어 쪽에서 수집된 데이터를 다음 사이클에 넘겨받을 때까지 한 사이클 안에서 강화학습 훈련과 데이터 기록을 무난하게 처리하는 것을 확인할 수 있었다.

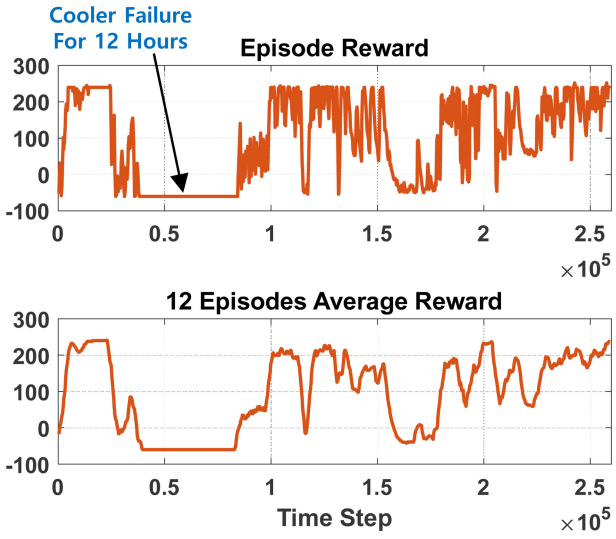

Fig. 7 Episode reward during the training

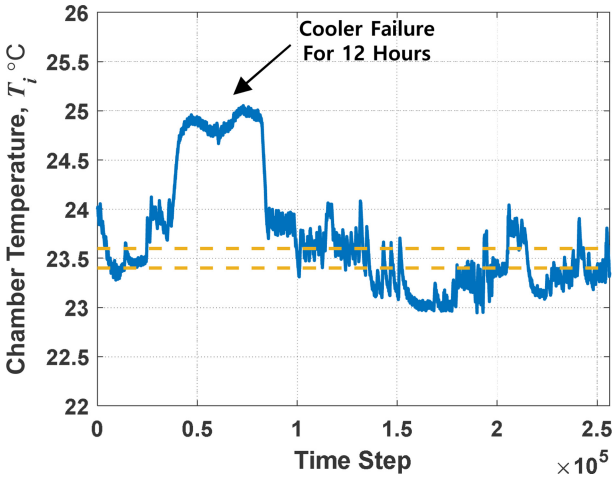

Fig. 8 Chamber temperature T_i during the training

4. 강화학습 에이전트의 훈련 결과

강화학습 에이전트가 72시간, 864개의 에피소드를 거쳐 훈련하면서 얻은 각 에피소드 보상과 12개의 이동 평균으로 본 에피소드의 변화 추이를 Fig. 7에 나타내었다. 훈련 시작 11시간 후에 약 12시간 동안 쿨러 고장으로 온도 제어 및 학습이 제대로 이루어 지지 않았기 때문에 40,000-80,000시간 스텝 동안에는 에피소드 보상이 벌점만 받아서 -60점을 유지하였다. 임의 행동을 취하는 확률 ε가 1%로 떨어진 150,000시간 스텝 이후에 학습된 정책에 의해 행동할 때의 에피소드의 점수는 출렁이기는 하지만 우상향 240점 이상으로 점차 증가하는 추이가 뚜렷함을 확인할 수 있다. 최종회에서 받은 에피소드 보상 점수는 240.8을 기록하였다. Fig. 8에 챔버 온도 T_i의 변화를 나타내었는데 느린 속도지만 가능한 높은 보상을 받을 수 있는 T_{ref}±0.1°C 영역으로

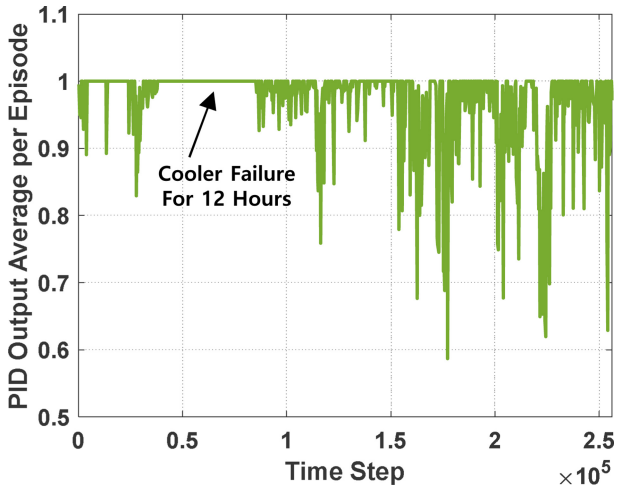

Fig. 9 PID output average per episode during the training

T_i가 움직여가는 모습을 볼 수 있다. Fig. 9에는 PID 제어 출력의 변화를 나타내었다. 값의 진폭이 커서 변화 추이를 알아보기 힘들지만 임의 행동을 취하는 확률 ε가 1%로 떨어진 150,000시간 스텝 이전과 이후를 비교하면 강화학습 에이전트가 PID의 출력을 떨어뜨리는 방향으로 행동하고 있음을 알 수 있다.

5. 결론

본 연구에서는 강화학습 알고리즘을 컴퓨터 시뮬레이션 환경이 아니라 직접 장비에 연결하여 초정밀 가공기용 챔버의 온도 제어에 적용한 결과를 제시하였으며, 이를 통하여 강화학습을 위한 시뮬레이션 모델링 자체가 어려운 기계 장비에서도 다중 목적의 제어기 설계 등에 강화학습이 유용한 도구가 될 수 있음을 보였다. 강화학습 알고리즘을 사용하여 1 Hz의 속도로 챔버의 정밀한 온도 제어와 에너지 절감이라는 목적을 가진 보상 함수를 통해 에이전트를 훈련하였으며, 쿨러가 완벽히 제 기능을 발휘하지 못하는 상황에서도 의도한 방향으로 에이전트가 훈련되고 작동함을 확인하였다. 또한 Python 언어로 된 강화학습 코드와 실제 장비에서 사용되는 C-언어로 구동되는 제어용 하드웨어 사이에 데이터 파이프라인을 구축하는 방안을 소개하였다.

강화학습 훈련 과정에서 최상의 성능을 나타내는 에이전트의 훈련 종료 시점을 판단하기 어려운 문제와 훈련된 에이전트의 강인성(Robustness)을 보장하기 위한 방안에 관하여 추가적인 연구를 진행할 계획이다.

ACKNOWLEDGEMENT

이 논문은 2022년도 산업통상자원부 소재부품 기술개발사업(소재부품패키지형)에서 지원을 받아 수행된 연구임(No. 20021890).

REFERENCES

1. Wang, Z., Hong, T., (2020), Reinforcement learning for building controls: The opportunities and challenges, Applied Energy, 269, 115036.
2. Di Natale, L., Svetozarevic, B., Heer, P., Jones, C., (2021), Deep reinforcement learning for room temperature control: A black-box pipeline from data to policies, Journal of Physics: Conference Series, 2042, 012004.
3. Zhang, Z., Chong, A., Pan, Y., Zhang, C., Lam, K. P., (2019), Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning, Energy and Buildings, 199, 472-490.
4. Vázquez-Canteli, J. R., Ulyanin, S., Kämpf, J., Nagy, Z., (2019), Fusing TensorFlow with building energy simulation for intelligent energy management in smart cities, Sustainable Cities and Society, 45, 243-257.
5. Brandi, S., Piscitelli, M. S., Martellacci, M., Capozzoli, A., (2020), Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings, Energy and Buildings, 224, 110225.
6. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., (2015), Human-level control through deep reinforcement learning, Nature, 518(7540), 529-533.
7. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., (2013), Playing Atari with deep reinforcement learning, arXiv preprint arXiv:1312.5602.
8. Stack Overflow, Subprocess readline hangs waiting for EOF. https://stackoverflow.com/questions/7897202/subprocess-readline-hangs-waiting-for-eof
9. National Instruments, Connections to network variables. https://www.ni.com/docs/ko-KR/bundle/labwindows-cvi/page/cvi/libref/cvinetvarconnections.htm

Byung-Sub Kim
Principal researcher in the Department of Ultra-precision Machines and Systems, Korea Institute of Machinery & Materials. His research interests include control and dynamic systems for ultra-precision machine tools, integrated dynamic system analysis, and artificial intelligence.
E-mail: bkim@kimm.re.kr

Seung-Kook Ro
Principal researcher in the Department of Ultra-precision Machines and Systems, Korea Institute of Machinery & Materials. His research interests are control system and actuator design for precision machines.
E-mail: cniz@kimm.re.kr