

RDA 기반 보상·하네스 설계 통한 유성기어 역진자의 실기 학습 루프 강화학습 제어

Reinforcement Learning Control of a Planetary-Gear Inverted Pendulum in a Hardware-in-the-Learning-Loop via RDA-based Reward and Harness Design

○진 호 경¹, 박 영 진^{1*}

¹⁾ 경상국립대학교 기계공학부 (TEL: 055-772-1621; E-mail: {wlsghrud001, youngjin.park}@gnu.ac.kr)

Abstract Designing reward functions and training infrastructure remains the main bottleneck when reinforcement learning (RL) is applied directly to physical hardware. This paper presents a hardware-in-the-learning-loop (HILL) framework in which a large language model (LLM) coding agent acts as a reward design agent (RDA) that iteratively revises not only the reward function but also the training harness (safety limits, start-up kick, replay buffer, encoder calibration) from physical diagnoses of real trajectories. The plant is a 3D-printed planetary-gear inverted pendulum in which a BLDC motor drives only the sun gear while the carrier swings freely. A Teensy 4.1 runs a 4-128-128-2 policy at 1 kHz on board, and a PC trains soft actor-critic on transitions streamed over micro-ROS. From a simulation seed that failed to transfer, thirteen reward-harness stages reduced the steady-state angle error to 0.35° with a holding torque RMS of 0.0002 Nm, and the final policy recovered from disturbances that were never part of training.

Keywords Reinforcement learning, Large language model, Reward design, Hardware-in-the-loop, Inverted pendulum

1. 서론

심층 강화학습(reinforcement learning, RL)은 보행, 조작, 비행 등 여러 로봇 과업에서 Real World 성공 사례를 쌓아 왔지만, 그 성공은 대개 대규모 시뮬레이션이나 정교하게 다듬은 학습 절차에 기대고 있다. 최근의 Real World 성공 사례 조사는 실기에서 안정적이고 효율적인 학습과 체계적인 개발·평가 절차가 여전히 부족하다고 지적한다[1]. 실기 학습을 성공시킨 연구들도 알고리즘 자체보다 보상 계산, 환경 리셋, 저수준 제어기 같은 구현 세부가 성과를 좌우한다고 보고하며[2], 사람의 시연과 교정을 학습 루프에 넣어 실기 학습 시간을 수 시간 이내로 줄이기도 한다[3].

실기 학습의 부담을 피하는 일반적인 경로는 시뮬레이터에서 학습한 정책을 실기로 옮기는 sim-to-real 전이다. 그러나 마찰·감쇠 같은 구동 파라미터를 식별하지 않으면 전이가 쉽게 실패한다. 3단 도립진자의 천이 제어에서 파라미터 식별 없이 학습한 정책은 실기 10회 시도를 모두 실패했고, 실측 궤적으로 파라미터를 식별한 뒤에야 모두 성공하였다[4]. 같은 계열 연구는 모델 일치도가 높도록 하드웨어 자체를 설계하여 56가지 평형점 천이를 실기에서 구현하였다[5]. 휴머노이드 보행에서는 구동기 모델이나 구동기 열 모델을 시뮬레이터에 넣어 격차를 줄였고[6, 7], 시뮬레이터만 바꾸어도 보상과 상태 설계를 다시 해야 했다는 보고도 있다[8]. 저가형 회전형 도립진자 플랫폼에서도 구동기 한계와 현실 격차가 강화학습 제어의 주요 제약으로 다루어진다[9]. 즉 정밀한 모델을 세우기 어려운 저가 하드웨어일수록, 보상

과 학습 조건을 실기에 맞게 다시 설계하는 일이 피할 수 없는 과제가 된다.

보상 설계 자체를 대규모 언어모델(large language model, LLM)에 맡기려는 연구도 빠르게 늘고 있다. Eureka는 LLM이 보상 코드를 작성하고 학습 통계를 되먹임 받아 진화적으로 개선하여 다수 과업에서 전문가 보상을 넘어섰고[10], Text2Reward는 환경 설명으로부터 밀집 보상 코드를 만들고 사람의 피드백으로 다듬었다[11]. DrEureka는 보상과 함께 도메인 무작위화 분포까지 LLM이 설계하여 sim-to-real 전이를 자동화하였고[12], CurricuLLM은 하위 과업 커리큘럼과 그 보상 코드를 생성한 뒤 롤아웃으로 정책을 평가하였다[13]. ARCHIE는 LLM이 보상과 성공 판정 기준을 만들고 시뮬레이션에서 학습한 정책을 실제 양팔 로봇에 배치하였다[14]. 국내에서도 LLM이 기존 알고리즘의 전략 파라미터를 생성하거나[15], 로봇 작업 계획과 제어 코드를 생성하는 연구가 확산되고 있다[16]. 한편 보상이 높은 점수를 주더라도 설계자가 원하는 행동과 어긋날 수 있으므로, 보상과 선호의 정합도를 따로 재는 지표도 제안되었다[17].

이 흐름에서 RDA(Reward Design Agent)[18]는 성공률이나 보상 성분 통계 같은 거친 수치만으로는 서로 다른 실패 모드를 구분할 수 없다고 지적하고, 시각-언어 모델이 학습된 궤적을 하위 과업 단위로 직접 보고 실패 모드를 요약한 뒤 보상 코드를 고치는 루프를 제안하였다. 또한 LLM 에이전트의 성능이 모델 자체뿐 아니라 이를 둘러싼 실행 구조, 곧 하네스(harness)에 좌우된다는 관점이 대두되었다. HarnessX는 프롬프트·도구·메모리·제어 흐름으로 이루어진 하네스를 조합·진화 가능한 객체

로 다루고 결정론적 게이트로 회귀를 막았으며[19], HarnessForge는 하네스와 정책을 함께 진화시켜야 둘 사이의 실행 호환성이 유지된다고 보았다[20]. Meta-Harness와 Agentic Harness Engineering은 과거 후보의 코드·점수·실행 기록을 에이전트가 직접 읽고, 각 수정에 검증 가능한 예측을 붙여 하네스를 개선하였다[21, 22].

그러나 이들 연구를 실기 학습에 그대로 적용하기에는 세 가지 공백이 있다. 첫째, LLM 기반 보상 설계는 대부분 대량 병렬 시뮬레이션에서 보상 후보마다 학습을 반복하는 방식으로 평가되며, RDA도 그 비용을 한계로 꼽았다[18]. 실기에서는 에피소드 하나가 실제 시간을 소모하므로 이 방식을 쓸 수 없다. 둘째, 하네스 진화 연구는 텍스트·도구 사용 에이전트에 국한되어 있고, HarnessX는 연속 행동 공간인 로봇 제어로의 확장을 검증하지 않았다고 명시하였다[19]. 그런데 실기 학습에서는 안전 한계, 시동 조건, 리플레이 버퍼, 센서 보정처럼 보상 밖의 학습 조건이 성패를 크게 좌우한다[2]. 셋째, LLM이 결과를 보고 다시 행동하는 되먹임 루프에서는 LLM 자신이 대리 목표를 과최적화하는 보상 해킹이 일어날 수 있고[23], 실기에서는 계측 결함까지 지표를 오염시킨다. 실제 공정 제어에서도 작업자의 관찰 없이 LLM만으로는 결함을 놓치는 경우가 보고되었다[24]. 따라서 스칼라 지표보다 풍부한 물리 진단과 사람의 감독이 함께 필요하다.

본 연구는 이 공백을 유성기어 역진자에서 다룬다. 이 장치는 모터가 태양기어만 구동하고 유성기어를 매단 캐리어는 자유 회전하는 과소구동계이며, 3D 프린팅한 PLA 기어의 정지마찰과 백래시 때문에 정확한 모델링이 어렵다. 본 연구에서는 학습과 추론을 분리하여 실기에서 얻은 데이터로 정책을 계속 학습시키는 구조를 HILL(Hardware-in-the-Learning-Loop)이라 부르고, LLM 코딩 에이전트(Claude Code)가 RDA 역할을 맡아 실기 궤적의 물리 진단을 근거로 보상함수와 학습 하네스를 함께 고치도록 하였다. 여기서 학습 하네스는 신경망가중치를 제외한 학습·실행 조건 전체를 뜻한다.

본 논문의 기여는 다음과 같다. 첫째, 실기 HILL 강화 학습에서 LLM 에이전트가 보상과 학습 하네스를 공동 설계하는 프레임워크를 구축하였다. 둘째, MATLAB Simscape Multibody를 이용한 시뮬레이션 강화학습으로 5세대를 거쳐 얻은 정책을 실물 학습의 첫 가중치로 적용하여 HILL 학습을 시작하여 13단계의 보상·하네스 진화로 정착 각오차 0.35° , 유지 토크 RMS 0.0002 Nm 에 도달한 과정을 단계별 실패 모드·처방과 함께 제시한다. 셋째, 학습에 포함되지 않은 외란에 대한 복원과 재스윙업 등 실기에서 관찰된 외란에 대한 거동을 보고한다. 2장에서 시스템을, 3장에서 RDA 기반 설계 방법을, 4장에서 실험 결과를 기술하고, 5장에서 서론의 문제 제기에 비추어 결과와 한계를 논의한 뒤 6장에서 결론을 맺는다.

2. 유성기어 역진자 HILL 시스템

2.1 하드웨어와 제어 목표

그림 1은 SOLIDWORKS로 설계하고 PLA로 3D 프린

팅한 유성기어 역진자다. 링기어를 제외한 태양기어, 유성기어, 두 기어를 잇는 캐리어와 지지대로 구성되며, 베어링의 외륜이 태양기어의 회전을, 내륜이 캐리어의 회전을 담당하여 둘이 같은 회전축을 공유한다. 모터는 태양기어만 돌리고 캐리어는 자유롭게 움직이므로, 모터 토크는 기어 맞물림을 통해 캐리어에 반작용으로만 전달된다. 즉 1자유도 입력으로 캐리어와 모터의 2자유도 운동을 다루는 과소구동 제어 문제가 된다.

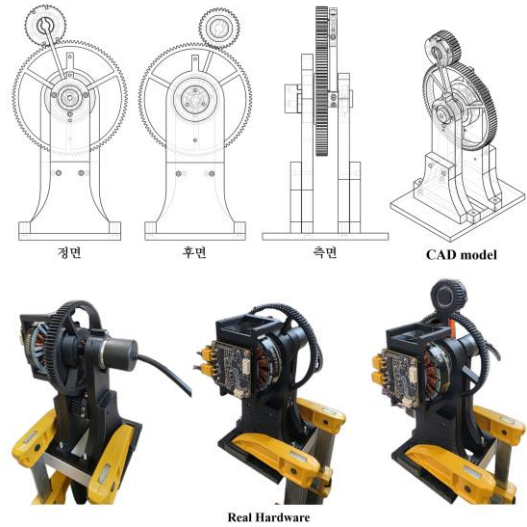


그림 1. 유성기어 역진자의 CAD 모델(위)과 실물(아래)

그림 2는 시스템 구성이다. 주 제어기 Teensy 4.1은 CAN 트랜시버를 거쳐 CAN-FD(데이터 5 Mbps) 통신으로 moteus r4.11 드라이버에 토크를 지령하고, 드라이버는 MJ5208 BLDC 모터를 구동한다. moteus는 위치 이득을 0으로 두어 순수 피드포워드 토크만 내도록 설정하였다. 캐리어 작은 Autonics E30S4-1024 증분형 엔코더를 4채배하여 4096 count/rev로 1 ms마다 읽으며, 엔코더의 Z 인덱스 펄스를 기준으로 12시(태양기어 기준 불안정 평형점)의 절대 영점을 보정한다. 엔코더 신호는 초기에 소프트웨어 인터럽트로 계수하였으나, 고속 회전 시 계수 누락이 확인되어 MCU의 하드웨어 쿼드러처 디코더로 교체하였다(4.4절 참조).

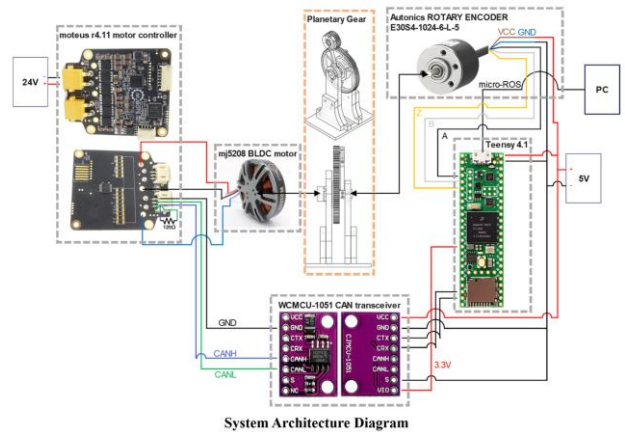


그림 2. 시스템 구성도

캐리어 각 θ 는 6시(안정 평형점) 방향을 $-\pi/2$, 12시(목표) 방향을 $+\pi/2$ 로 정의하며, 각오차는 식 (1)과 같다. 제

어 목표는 6시에 정지한 캐리어를 5 s 에피소드 안에 12 시로 스윙업하여 정지시키는 것이다. 캐리어가 반대 방향으로 돌아 올라가면 $\theta = -3\pi/2 (-270^\circ)$ 에서도 12시가 된다.

$$e = \text{atan2}(\sin(\theta - \pi/2), \cos(\theta - \pi/2)) \quad (1)$$

2.2 학습과 추론을 분리한 HILL 구조

제어 루프에 통신 지연이 끼어들지 않도록 추론과 학습을 물리적으로 분리하였다(그림 3). Teensy 4.1은 1 kHz 타이머 인터럽트에서 엔코더로 $\theta, \dot{\theta}$ 를 갱신하고, moteus 응답에서 모터 속도 ω_m 을 읽어 관측 o 를 구성한 뒤, 정책 신경망을 추론하여 토크를 지령한다.

$$o = [\sin \theta, \cos \theta, \dot{\theta}/10, \omega_m/500] \quad (2)$$

$$a = \tanh(\mu(o) + \sigma(o) \cdot \epsilon) \quad (3)$$

$$\tau = \text{clip}(0.2 \cdot a, -0.2, 0.2) \text{ [Nm]} \quad (4)$$

θ 를 $\sin \cdot \cos$ 쌍으로 넣은 것은 $\pm 180^\circ$ 경계의 불연속을 없애기 위함이고, 속도는 입력 크기가 $O(1)$ 이 되도록 정규화하였다. 정책은 4-128-128-2 완전연결 신경망으로 평균 μ 와 표준편차 σ 를 내며, 가중치 17,410개를 RAM에 상주시켜 온보드에서 추론한다. 추론 181 μs 를 포함한 한 주기 처리시간은 최대 186 μs 이다. 학습 중 탐험에는 식 (3)의 표준정규 잡음 ϵ 을 쓰고, 평가와 배포에는 $\epsilon=0$ 인 결정적(greedy) 행동을 쓴다.

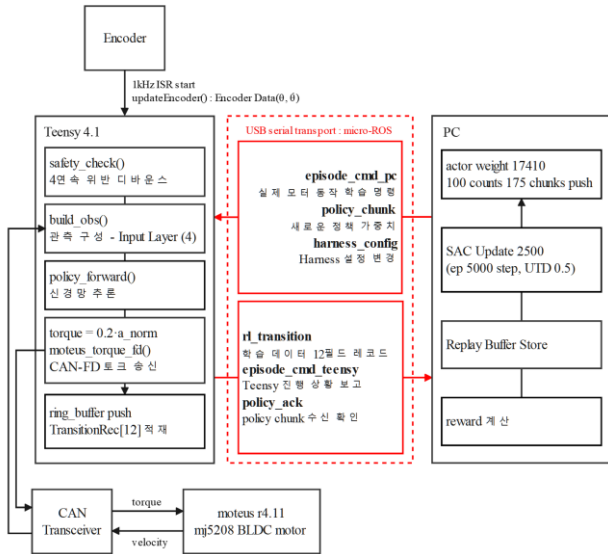


그림 3. Teensy 4.1-PC 통신 아키텍처

Teensy와 PC는 USB 직렬 전송의 micro-ROS로 연결되어 ROS 2 토픽을 주고받는다. micro-ROS는 MCU 수준의 자원에서도 손실 없이 동작함이 보고된 바 있다 [25]. 데이터는 저장 후 전달(store-and-forward) 방식으로 PC에서 Teensy로 보낸다. 에피소드 동안에는 Teensy에서 관측, 행동, 실패토크와 원시 $\theta, \dot{\theta}, \omega_m$ 을 담은 전이 레코드를 링버퍼에만 쌓고, 에피소드가 끝난 뒤 진단 레코드와 함께 PC로 일괄 전송한다. 이때 원시 물리량을 함께 보내는 이유는 보상함수가 바뀌었을 때 리플레이 버퍼의 과거 전이 전체의 보상을 새 함수로 재계산(relabel)하기 위함이다. PC에서 SAC 알고리즘으로 학습이 진행되

어 갱신된 가중치를 Teensy로 100개씩 175개 청크로 내려보낸다.

안전 계층은 펌웨어에 있다. 토크는 $\pm 0.2 \text{ Nm}$ 로 상시 제한되며, 캐리어 각속도 30 rad/s, 모터 속도 250 rad/s, 공전 3π , 드라이버 20 ms 무응답 중 하나가 4주기 연속 위반되면 에피소드를 중단하고 감쇠 토크로 안전 정지한다. 한 에피소드는 토크 0으로 6시에 정지할 때까지 기다리는 정착 단계(15~25 s)와 5,000스텝(5 s)의 구동 단계로 이루어진다.

2.3 SAC 학습기

PC의 학습기는 연속 제어에서 여전히 강력한 기반으로 쓰이는 off-policy 최대 엔트로피 알고리즘 SAC(soft actor-critic)를 사용한다[26]. Actor 신경망 1개와 Twin-Q Critic 신경망을 쓰며, 감가율 $\gamma=0.99$, 학습률 1×10^{-4} (Actor)/ 5×10^{-4} (Critic), 엔트로피 계수 $\alpha=0.1$ 고정, 배치 1,024, 리플레이 버퍼 100k로 설정하였다. UTD(Update to Data)=0.5로 설정하여 한 에피소드(5,000 전이)가 도착할 때마다 5000개의 데이터에서 무작위로 1024개를 뽑아서 약 2,500회 갱신한 뒤 새 가중치를 Teensy로 보낸다.

3. RDA 기반 보상-하네스 공동 설계

3.1 학습 구조 정의

RDA[18]는 보상 설계 문제를 지시 I , 환경 E , 보상함수 공간 R , 학습 알고리즘 L , 정책이 지시에 부합하는 정도를 재는 점수 S 로 정의하고, L 과 S 를 고정한 채 $S(L(E, R), I)$ 를 최대화하는 보상을 찾는다. 본 연구는 여기에 학습 하네스 H 를 탐색 변수로 더한다.

$$(R, H) \in \arg\max_{R, H} S(L(E_H, R, H), I) \quad (5)$$

E_H 는 하네스가 적용된 실제 플랜트이고, S 는 3.4절의 물리 지표와 과형 관측으로 평가한다. 실기에서는 L 을 한 번 실행하는 데 실제 시간이 들기 때문에 보상 후보를 병렬로 학습시키는 대신 한 번에 하나의 처방을 적용하고 결과를 관측하는 순차 루프를 택하였다.

3.2 학습 하네스

HarnessX와 HarnessForge의 하네스가 LLM 에이전트의 실행 구조를 가리키는 것과 달리[19, 20], 본 연구는 그 관점을 강화학습 루프로 옮겨 신경망 가중치를 제외한 학습·실행 조건 전부를 하나의 하네스 객체로 정의하였다(표 1). 이 가운데 시동 킥(kick)은 에피소드 시작 시 정지마찰을 넘기도록 가하는 개루프 토크 펄스이고, Z 보정(zcal)은 2.1절의 12시 절대 영점 오프셋이며, 넉다운(knock)은 에피소드가 12시에서 끝났을 때 다음 에피소드를 6시에서 시작하도록 진자를 넘여쓰리는 리셋 동작이다. Teensy 측 값은 ROS 토픽으로 전달되어 펌웨어 재빌드 없이 바뀐다.

표 1. 학습 하네스의 10개 차원(기본값)

차원	위치	주요 파라미터
obs	Teensy	θ 정규화 10, ω_m 정규화 500
action	Teensy	토크 스케일·클립 0.20 Nm
reward	PC	보상 항 계수·게이트 폭 (표 2)
explore	PC	엔트로피 관련 설정
safety	Teensy	$\dot{\theta}$ 30 rad/s, ω_m 250 rad/s, 공전 3π
episode	Teensy	정착 15~25 s, 5,000 스텝
train	PC	γ , 학습률, 배치, 버퍼
kick	Teensy	0.18 Nm $\times$ 150 ms
zcal	Teensy	Z 인덱스-12시 오프셋
knock	Teensy	에피소드 후 6시 복귀 펄스

3.3 RDA 루프와 역할 분담

그림 4는 RDA 환경 구조다. RDA는 Claude Code(Anthropic, Opus 계열 모델)로 구성하였고, 학습 결과를 읽고 판독 후 분석 및 처방을 진행하고 코드를 수정하는 구간을 맡는다. 수정 대상은 보상, 하네스, 학습 설정의 세 영역이며 이를 위해 파일 읽기·편집·셀 실행과 MATLAB 실행(matlab-mcp) 도구를 쓴다. 사람은 목표 설정, 모터 구동 허가, 최종 채택을 맡는다. 실제 모터를 돌리는 모든 명령에 사용자의 명시적 허락을 요구하여, 되먹임 루프 속 LLM의 과최적화 위험[23]을 사람의 판단으로 차단하였다.

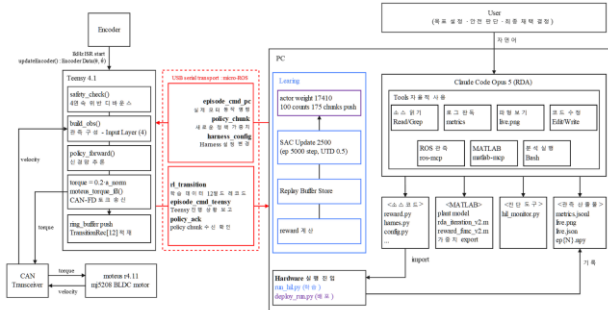


그림 4. RDA 환경 구조

실기 원시 데이터는 에피소드당 약 6만 개 수치여서 LLM의 문맥에 그대로 넣을 수 없다. 이에 관측 도구에 에피소드마다 압축 지표 한 줄(metrics.jsonl)과 과형 대시보드(live.png)를 만들고 RDA는 둘을 함께 읽는다. 이는 제적을 직접 보고 실패 모드를 요약하는 RDA의 방식[18], 실행 기록을 근거로 하네스를 고치는 하네스 진화 방식[21, 22]을 실기 과형 판독으로 옮긴 것이다

3.4 평가 지표

단계별 정책은 10 에피소드마다 수행하는 greedy 평가 제적으로 비교한다. 정착 각오차는 에피소드 마지막 1 s 동안의 평균 $|e|$, 유지 토크 RMS는 같은 구간 토크의 표준편차로 정의한다. 정착 각오차가 작은 정책을 정밀하다고, 유지 토크 RMS가 작은 정책을 정속하다고 부른

다. 유지 토크가 진동하면 기어 마모가 커지므로 정속성도 정밀도와 함께 목표로 삼았다. 이 밖에 $|e| < 1^\circ$ 체류율, 스윙업 시간, 중단 사유를 함께 기록한다.

3.5 보상함수 구조

최종 보상은 12개 항으로 이루어진다(표 2). 이 보상함수의 구조와 계수는 모두 LLM의 자율 판단으로 설계 및 수정을 진행했다. 항들은 광역 유도, 상단 포획, 정밀, 정속, 에너지 펌핑, 과에너지 억제역역의 역할로 묶이며, 상단 근처나 저속 구간에서만 커지도록 다음 게이트를 쓴다.

$$p = \exp(-e^2/(2\sigma_p^2)), s = \exp(-(\dot{\theta}/0.7)^2) \quad (6)$$

$$q = \exp(-(\tau/0.04)^2) \quad (7)$$

$$\tilde{E} = (0.5\dot{\theta}^2 + \omega_0^2(\sin \theta - 1)) / \omega_0^2 \quad (8)$$

p 는 상단 근접, s 는 저속, q 는 정속 게이트이며 $\sigma_p=0.15$ rad이다. $\tilde{E}$ 는 상단 도달에 필요한 에너지를 0으로 하는 정규화 에너지로, $\omega_0^2=128$ 은 실측한 캐리어 진자 고유진동수의 제곱이다. 펌핑 항 (6)과 (8)은 진자 에너지를 상단 에너지로 끌어올리는 행동에 보상을 준다. 표 2의 항들은 처음부터 있던 것이 아니라 4.3절의 진화 과정에서 하나씩 추가된 것이다.

표 2. 최종 보상함수(⑬)의 12개 항

항	식	역할
(1)	$+2.0 \cdot \cos e$	광역 유도
(2)	$+9.0 \cdot p \cdot \exp(-(\dot{\theta}/4)^2)$	상단 포획
(2b)	$+6.0 \cdot (1 - \tanh(e /0.03)) \cdot \exp(-(\dot{\theta}/1.5)^2) \cdot q$	정밀
(3)	$-2.5 \cdot p \cdot \tanh((\dot{\theta}/1.0)^2)$	상단 속도 억제
(4)	$-0.6 \cdot \tanh((\dot{\theta}/8)^2)$	스핀 억제
(5)	$-0.5 \cdot \tau^2$	제어 비용
(5b)	$-20 \cdot p \cdot s \cdot \tau^2$	토크 채터링 억제
(5c)	$-2.0 \cdot p \cdot s \cdot \tanh((\omega_m/20)^2)$	모터 공전 억제
(5d)	$-0.02 \cdot p \cdot s \cdot \min(\omega_m , 100)$	공전 포화 보완
(6)	$+2.0 \cdot \exp(-\tilde{E}^2/(2 \cdot 0.7^2)), e > 0.35$	에너지 펌핑
(7)	$-2.0 \cdot \tanh(\max(0, \tilde{E}))$	과에너지 억제
(8)	$+3.0 \cdot \tanh(-\tilde{E} \cdot P/3), P = 1050 \cdot \tau \cdot \dot{\theta}/128$	펌핑 파워

4. 실험 결과

4.1 시뮬레이션 사전 강화학습

무작위 정책으로 실기를 탐험하면 기구 파손 위험이 크므로, 먼저 CAD 모델을 MATLAB Simscape Multibody로 옮겨 사전 강화학습을 진행하였다. Simscape Multibody 모델(그림 5)은 실측 PLA 밀도로 질량을 보정하고 토크 한계를 실기와 같은 0.2 Nm로 설정하고 SAC 알고리즘으로 학습하였다. 이때 유성기어 모델링에 대해서 시뮬레이션을 시물과 최대한 유사하게 구성하여

sim2real gap을 최대한 줄이기 위해 기어 마찰·백래시 모델을 추가했고 실기와 같은 관측(식 (2))과 신경망을 사용하여 학습을 진행하였다. 여기서 실물 학습과의 차이는 제어 주기를 50 Hz(에피소드당 250 스텝)로 진행한 것이다. 정책은 상태를 토크로 바로 사상하는 기억 없는 함수여서 학습 주기와 무관하게 가중치를 1 kHz 실기로 옮길 수 있는 반면, 1 kHz로 시뮬레이션하면 에피소드당 스텝과 갱신 횟수가 20배로 늘어 세대당 약 20분이던 학습 시간이 수 시간으로 늘어나기 때문이다. 또한 할인을 0.99에서 1 kHz의 실효 계획 지평은 약 0.1 s에 불과해 약 1 s가 걸리는 스윙업의 보상 전파가 어려워진다. 실기와 주기 차이는 HILL 학습에서 1 kHz로 재학습하여 보완하였다. 250 에피소드를 한 세대로 하여 세대마다 Claude Code(model Opus 5)가 궤적 그래프와 시뮬레이션 화면을 판독해 보상을 재설정하였다.

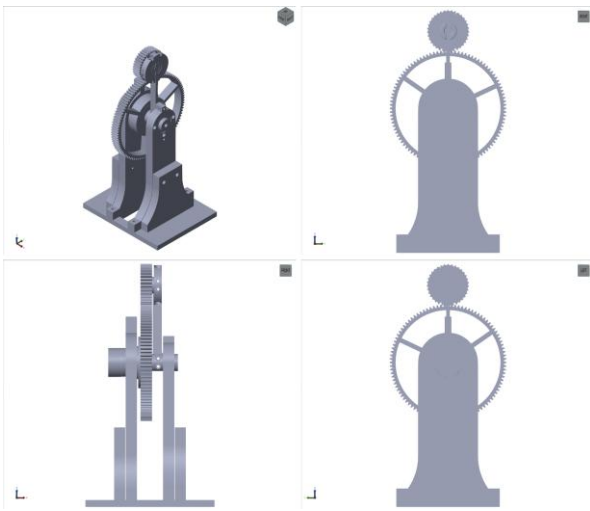


그림 5. MATLAB Simscape Multibody 유성기어 모델링

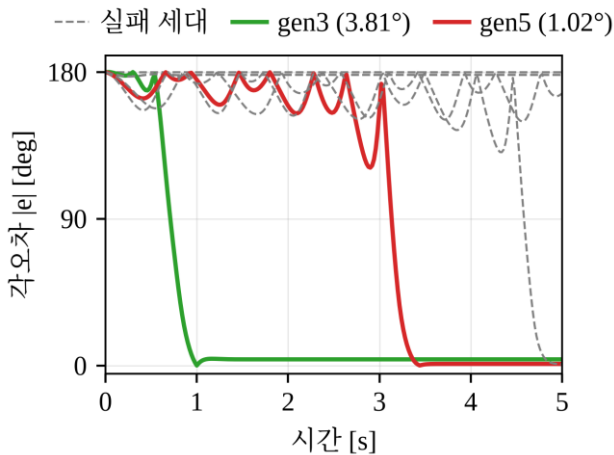


그림 6. Simscape 사전학습 7개 세대의 greedy 궤적

그림 6과 같이 7개 세대 가운데 gen3(최종 각오차 3.81°)과 gen5(1.02°)만 상단에 정착하였다. 나머지 세대는 바닥에 머물거나(gen1·2·4·7) 상단을 잡지 못하고 회전을 반복하였다(gen6). 정착 오차가 가장 작은 gen5의 가중치 17,410개를 펌웨어와 PC 학습기에 함께 넣어 실기 HILL 학습의 초기 가중치로 사용하였다.

4.2 가중치 실기 이식과 HILL 학습 진행 과정

실물 유성기어 하드웨어에서 MATLAB Simscape Multibody 강화학습의 gen5 정책을 초기 가중치로 이식한 뒤 HILL 학습을 시작하자 처음에 6시에서 전혀 움직이지 않았다(그림 7(a)). 정책 토크는 0.009~0.010 Nm로 시뮬레이션의 시작 토크와 같았지만, 시뮬레이션의 매끄러운 마찰 모델에는 없던 실기 기어열의 정지마찰을 넘지 못하였다. 이에 Claude Code는 사용자의 요청 없이 스스로 판단하여 펌웨어에 3.2절의 시동 킥(0.18 Nm×150 ms)을 추가하였다. 그러자 진자는 상단을 지나쳐 약 400°까지 과회전하였다(그림 7(b)). 같은 토크에서 실기가 더 큰 운동에너지를 얻었고 제어 주기도 50 Hz에서 1 kHz로 바뀌었기 때문이다. 이는 마찰을 식별하지 않은 sim-to-real 전이가 실패한다는 선행 보고[4, 5]와 일치한다. 즉 시뮬레이션 정책은 실기에서 그대로는 동작하지 않았으므로, 이를 출발점으로 삼아 실기 데이터로 정책을 계속 학습시키면서 보상과 하네스를 함께 진화시켰다.

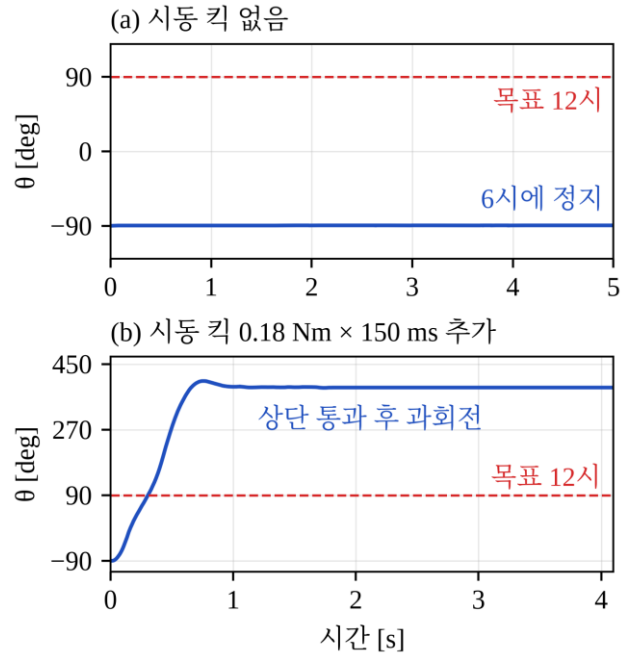


그림 7. HILL 학습 초기 가중치 적용의 거동
(a) 정지마찰에 막힘, (b) 킥 추가 후 과회전

4.3 실기 보상·하네스 진화 13 단계 과정

실기 진화는 2026년 7월 16일부터 8월 18일까지 13단계(①~⑬)로 진행되었다(표 3). 단계는 RDA의 처방이 한 번 적용된 학습 구간을 뜻하며, 처방 영역을 보상(R), 하네스(H), 학습 설정(P)으로 표시하였다. ①은 gen5 가중치로 실물 하드웨어에서 학습을 진행한 후 학습과 관계없는 모터 구동 테스트 중 태양기어 파손 문제가 발생하여 ②부터는 새로운 태양기어로 교체하여 학습을 진행하였으며 각 단계의 Actor는 직전 단계까지의 최선 정책에서 이어 학습(웜스타트)하였다. 정착 각오차는 진단 레코드까지 수신이 확인된 greedy 평가 궤적에서 계산하

였다. ②와 ④는 궤적이 보존되지 않아 기록값만 표시하였고, 특히 ④는 에피소드 끝부분 데이터가 수신되지 않은 상태(꼬리 유실, 4.4절)에서 계산된 값이어서 정확 오차로 인정하지 않는다.

표 3. RDA 보상·하네스 진화 이력

단계	영역	주요 조치	정착 [°]
①	-	gen5 로 구 기어 HILL 학습	1.04
②	-	새 기어 교체 ① 정책을 초기 가중치로 학습	2.9 [†]
③	R	보상 계수 자동 탐색	2.87
④	R	탐색 계수 고정	1.33 [‡]
⑤	R	(5b)(5c) 신설, σ_p 0.25→0.15	0.71
⑥a/b	-	같은 보상에서 정밀/정속 두 극단	0.31/2.52
⑦	R	(2b) 정밀 항, 정속 게이트 q	1.50
⑧	R-H	킵 부호 ± 교대, (5d)(8) 신설	0.35
⑨	P	리플레이 버퍼·비평자 초기화	0.62
⑩	H	시동 킵 제거	0.86
⑪	H	킵 ± 교대 복원, Z 보정 재측정	0.18
⑫	H-P	킵 제거, 버퍼 100k→300k	0.70
⑬	R	(2b) 폭 0.05→0.03	0.35

† 궤적 미보존(기록값). ‡ 꼬리 유실로 무효.

아래 내용은 구 기어 학습과 기어 교체(①~④)에 대한 내용이다. gen5 가중치는 구 기어에서의 HILL 학습으로 정착 1.04°에 도달하였으나(①) 유지 토크 RMS가 0.0516 Nm로 토크가 끊임없이 진동하였다(그림 8). 새로운 태양기어로 교체 후에 구 기어에서 학습한 ① 정책을 초기 가중치로 삼아 학습을 이어 갔다. 그러나 마찰·백래시가 달라진 새 기어에서 구 기어용 보상을 그대로 쓰자 2.9° 부근에서 정체되었다(②). ③에서는 HarnessX의 진화 엔진(AEGIS)[19]을 본떠 보상 계수를 자동 탐색하고 개선된 값만 채택하는 방식으로 근접 항 계수를 6.0→9.0, 근접 게이트 폭을 0.40→0.25 rad로 바꾸었으며, ④에서 이를 고정하였다.

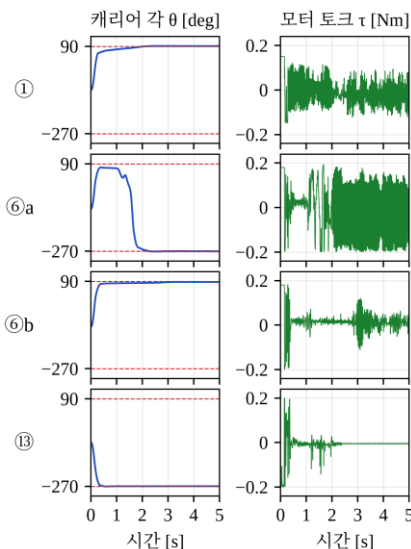


그림 8. 대표 단계의 greedy 평가 궤적 (+90°와 -270°는 모두 12시)

다음은 정속 항과 보상 해킹(⑤)에 대한 내용이다. 토크 진동을 줄이려고 상단·저속 구간의 토크에만 별점을 주자 유지 토크 지표는 목표를 달성한 것처럼 보였다. 그러나 RDA가 과형을 판독한 결과, 정책은 각오차 15.3°를 유지한 채 모터를 55.6 rad/s로 계속 돌려 반동 질량의 운동량으로 토크를 대체하고 있었다. 기존 보상이 모터 속도를 쓰지 않아 모터 공전에 대한 선택압이 없었던 것이 원인으로, 대리 지표만 좋아지는 보상 해킹이다. 보상 점수와 의도한 행동이 어긋날 수 있다는 지적[17]이 실기에서 나타난 사례다. RDA는 모터 공전 항 (5c)를 신설하였고, ⑤는 정착 0.71°에 도달하였다.

다음은 정밀-정속 트레이드오프와 기울기 소멸(⑥, ⑦)에 대한 내용이다. ⑥에서는 보상을 바꾸지 않았는데도 정착 0.31°이지만 유지 토크 RMS 0.1135 Nm로 시끄러운 정책(⑥a)과, 0.0195 Nm로 조용하지만 2.52°에 머무는 정책(⑥b)이 함께 나왔다(그림 8, 그림 9). RDA는 유지 구간의 토크를 회귀하여 정책이 약 -0.047 Nm의 일정 토크와 비례 이득 K_p 의 합으로 근사되고, 평형 편차가 $K_p=1.33$ 에서 2.04°, $K_p=3.94$ 에서 0.68°가 됨을 보였다. 즉 정밀하려면 이득을 높여야 하고, 이득이 높으면 마찰·백래시 구간에서 토크가 진동한다. 또한 가우시안 근접 항 (2)는 상단 근처에서 거의 평탄하여, 오차를 0.5°에서 0으로 줄여도 보상 이득이 +0.015에 불과하였다. ⑦에서 RDA는 $e=0$ 에서도 기울기 $-1/\sigma$ 를 유지하는 $1-\tanh$ 형태의 정밀 항 (2b)를 도입하고(그림 10), 정속 게이트 q를 곱해 시끄러운 해에는 정밀 보상을 주지 않았다. 기존 궤적으로 보상 순위를 미리 계산하여, 게이트가 없으면 시끄러운 정책이 이기지만(14.23 대 13.18) 게이트가 있으면 의도한 순위가 보존됨을 확인한 뒤 적용하였다. 그 결과 ⑦은 정착 1.50°, 유지 토크 RMS 0.0008 Nm로 ⑥b보다 정밀·정속이 함께 개선되었다.

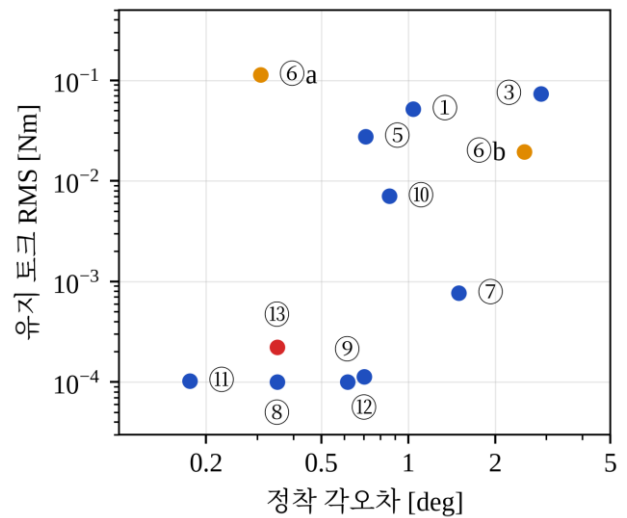


그림 9. 단계별 정착 각오차-유지 토크 RMS (로그 축)

그림 10은 각속도·토크·모터 속도가 0인 12시 근처에서 단계별 전체 보상을 비교한 것이다. 이 조건에서는 ⑧에서 추가된 항 (5d), (8)이 0이므로, ⑥의 보상은 최종 보상에서 (2b)를 뺀 것과 같다. 이렇게 재현한 ⑥의 이득 (0.5°→0: 0.015, 2.48°→0: 0.369)은 학습 당시 기록한 값 (0.015, 0.365)과 일치한다. (2b)를 더한 ⑦은 정점이 11.0

에서 17.0으로 높아질 뿐 아니라, 같은 $0.5^\circ \rightarrow 0$ 개선에 대한 이득이 1.05로 약 70배 커졌고 $e=0$ 부근의 기울기도 오차 1° 당 약 2.1을 유지한다. 정점의 높이 자체보다 이 기울기가 정책이 오차를 줄이도록 미는 힘이다.

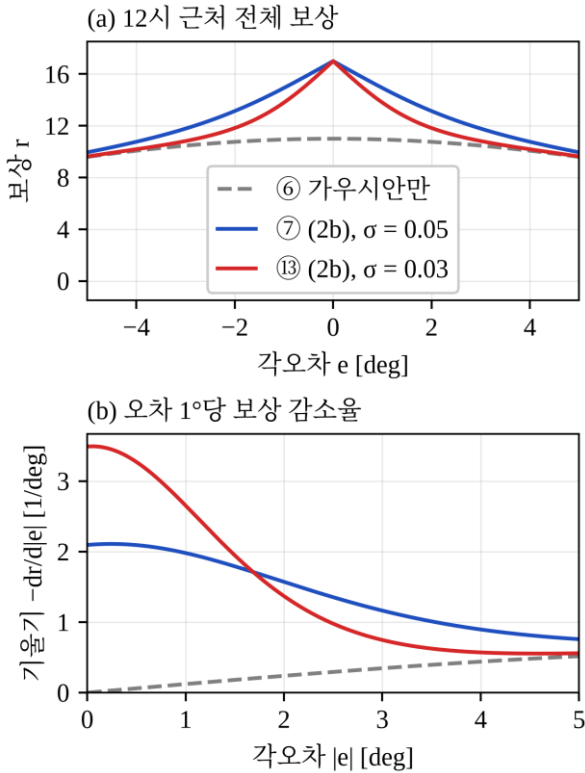


그림 10. 12시 근처 전체 보상의 단계별 비교
(각속도·토크·모터 속도 0)

(a) 보상 단면, (b) 오차 1° 당 보상 감소율

하네스 진화(⑧~⑫). ⑧에서는 시동 킥의 부호를 에피소드마다 $\pm$ 교대시키고, $\tanh$ 포화로 생긴 모터 공전벌점의 사각을 선형 항 (5d)로 메우며, 에너지를 상단으로 보내는 순간 파워에 보상을 주는 항 (8)을 추가하였다. 그 결과 정착 0.35° , 유지 토크 RMS 0.0001 Nm, 스윙업 0.41 s로 처음으로 모든 목표를 만족하였다. ⑨~⑫는 보상을 바꾸지 않고 하네스와 학습 설정만 조정하였다. 리플레이 버퍼를 분석한 결과 상단 근처 전이가 한쪽 방향으로 편중되어($-2 \sim 0^\circ$: 16,323개, $0 \sim +2^\circ$: 939개) 한 방향으로만 포획을 잘하였고 ⑨에서 버퍼와 비평자를 초기화하고 재학습하여 처음으로 양방향 포획을 얻었다. ⑩에서 킥을 없애자 정책은 스윙업을 스스로 배웠으나 역방향 복귀에 실패하였고 ⑪에서 킥을 복원하자 양방향 복귀가 회복되었다. ⑫에서는 버퍼를 300k로 늘린 뒤 다시 킥을 제거하여 킥 없이도 양방향으로 스윙업·포획하는 정책을 얻었다.

다음 내용은 샤프닝(⑬) 내용이다. 마지막으로 정밀 항의 폭을 0.05에서 0.03으로 줄여, 0.75° 에서 0° 로 갈 때의 전체 보상 이득을 +1.57에서 +2.50으로 키웠다. 그림 10(b)와 같이 ⑬은 약 1.7° 안쪽에서 ⑦보다 기울기가 커서($e=0$ 부근 오차 1° 당 약 3.5), 보상의 기울기를 마지막 $1 \sim 2^\circ$ 에 집중시킨다. 최종 정책은 킥 없이 0.34 s에 스윙업하고 정착 0.35° , $|e| < 1^\circ$ 체류율 100%, 유지 토크 RMS 0.0002 Nm, 모터 속도 0.04 rad/s로 거의 정지한 상태에서

진자를 유지하였다. 같은 단계의 평가 3회(ep410/420/430)의 중앙값도 0.35° 로, 직전 단계 기준(0.71°)의 절반이었다.

최종 보상은 $\theta, \dot{\theta}, \omega_m, \tau$ 네 변수의 함수이다. 그림 11은 12시 정지 상태($\theta=90^\circ, \dot{\theta}=0, \omega_m=0, \tau=0$)를 기준으로 한 변수씩만 바꾼 단면이다. 기준점에서 보상은 최대값 17.0을 갖는다. (a) 캐리어 각 단면에서 보상은 6시의 -1.97 에서 12시로 갈수록 완만하게 오르다가 $\pm 20^\circ$ 안에서 급·정밀 항이 더해져 뾰족한 정점을 이룬다. 오차가 1° 만 생겨도 13.8로 떨어지므로, 그림 10의 ⑥ 보상과 달리 상단 바로 근처에서도 정밀도 방향의 기울기가 유지된다. 정점 양옆의 작은 골은 펌핑 항 (6)이 $|e| > 0.35$ rad에서만 켜지는 게이트 경계에서 생긴다. (b) 각속도 단면에서는 $\dot{\theta}$ 가 1, 2, 4 rad/s로 커지면 보상이 12.4, 7.5, 2.5로 빠르게 줄어 8 rad/s 이상에서 음수가 되므로, 정책은 상단을 빠르게 지나치지 않고 멈추도록 유도된다. (c) 모터 속도 단면은 ⑤의 보상 해킹을 막은 항 (5c), (5d)의 효과로, 12시에 정지해 있어도 모터가 20 rad/s로 돌면 15.1, 100 rad/s 이상이면 13.0으로 낮아진다. (d) 토크 단면에서는 유지 토크가 0.02, 0.05, 0.1 Nm로 커지면 보상이 15.7, 12.2, 10.8로 낮아지며, 0.05 Nm 부근에서 정숙 게이트 q 가 닫혀 정밀 항이 거의 사라진다. 즉 최종 보상은 진자가 12시에 정확히 정지하고, 모터도 멈추고, 토크도 거의 쓰지 않는 상태에만 최대 보상을 준다. 상단에서 먼 영역에서는 남은 에너지 없이 12시에 도달하는 에너지 궤적($\dot{E}=0$) 위의 보상이 주변보다 높아($e=-120^\circ$ 에서 $\dot{E}=0$ 일 때 0.40, 속도가 그 절반이면 $-0.99, 1.4$ 배면 -3.15) 스윙업 경로를 유도한다. 토크가 0이면 보상은 회전 방향($\dot{\theta}$ 의 부호)에 대칭이고, 토크가 있으면 펌핑 파워 항 (8)에 의해 토크가 운동 방향과 같아 에너지를 넣는 쪽만 보상을 받는다(6시 부근 $\tau=+0.1$ Nm에서 $\dot{\theta}=+5$ rad/s이면 0.41, -5 rad/s이면 -4.76).

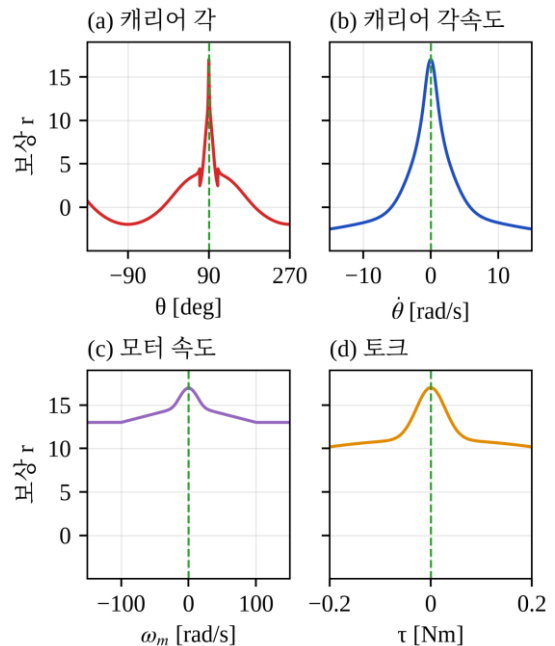


그림 11. 최종 보상함수(⑬)의 변수별 단면
(나머지 변수는 12시 정지값)

4.4 계측 결함의 발견

실기 루프에서는 지표 자체가 틀릴 수 있다. 진행 중 RDA와 사용자는 다음 결함을 발견하여 고쳤다. 에피소드 종료 신호가 전이 스트림의 꼬리를 앞질러 마지막 배치가 버려지는 꼬리 유실(7월 20일 평가 15회 중 6회), 소프트웨어 엔코더 계수의 고속 누락, 험거운 캐리어 축 구멍의 미끄럼(접근 방향에 따라 최대 8.61°), Z 영점 오차(지표 0.69°가 실제 약 6.5°), 보상 재계산 오류(버퍼의 43.7% 과대 보상)가 그것이다. 또한 지표 기반 진단이 반대 방향의 처방을 가리킨 경우가 세 번 있었다. "포획 후 낙하"로 판정된 경우는 실제로 상단을 지나친 과회전이었고, 토크 지표 달성은 앞선 보상 해킹이었으며, 또 다른 토크 지표 개선은 상단 미도달과 미분 이득 붕괴였다. 세 경우 모두 과형 판독이 지표를 교정하였다.

4.5 최종 정책의 외란 응답과 창발적 거동

HILL 학습의 모든 에피소드는 6시에 정지한 상태에서 시작하였고 외란을 주입하거나 외란 복원을 보상하는 항은 없었다. 그럼에도 최종 정책 ⑬은 배포 시험에서 수동으로 가한 외란에 강인하게 반응하였다(그림 12(a)). +7.2°와 -10.3°의 외란은 각각 0.080 s, 0.219 s 만에 복원되었고, 진자를 6시 부근까지 넘어뜨린 세 번의 큰 외란에서도 0.315~0.551 s 만에 킥 없이 재스윙업하여 12시에서 포획하였다. 외란 없이 9.1 s 동안 유지한 시험에서는 평균 0.35°(최대 0.62°), $|e| < 1^\circ$ 체류율 100%였다.

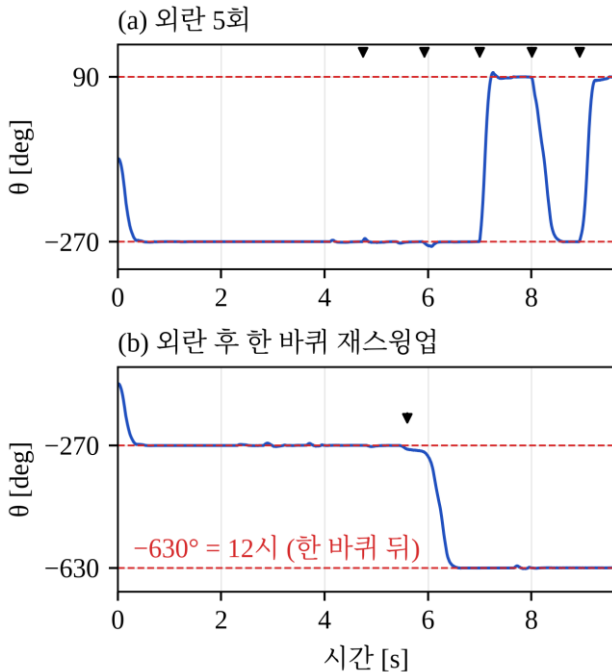


그림 12. 최종 정책 ⑬의 실기 외란 시험(▼: 외란)

흥미로운 점은 큰 외란에 대한 대응 방식이다. 정책은 외란을 거슬러 진자를 되돌리지 않고, 진자를 6시 쪽으로 떨어뜨린 뒤 그 운동 방향 그대로 반대편으로 스윙업하여 한 바퀴 돈 위치의 12시에서 다시 포획하였다. 그림 12(b)에서도 약 5.6 s의 외란 뒤 진자는 -270°에서 -630°

까지 한 바퀴를 돌아 다시 정착하였다. 토크가 ± 0.2 Nm로 제한된 상황에서는 외란으로 얻은 운동에너지를 제동으로 없애고 되돌아오기보다 그 에너지를 보존한 채 넘겨 재스윙업하는 편이 적은 토크로 상단 에너지에 도달할 수 있고 이를 신경망이 스스로 학습을 통해 알아낸 것으로 보인다. 펌핑 항 (6), (8)이 회전 방향을 구분하지 않고 보상 지형이 회전 방향에 대칭이므로(그림 12(b)), 정책은 이 전략을 명시적 지시 없이 찾아낸 것으로 해석된다.

또 다른 관찰은 엔코더 계수 누락이 있던 초기(7월 17일)의 거동이다. 강한 외란 뒤 측정 각이 실제 각과 어긋나자, 데이터상으로는 진자가 12시에 있었지만 실제 진자는 10시 부근(연구 기록상 12시에서 약 20° 어긋난 위치)에서 넘어지지 않고 유지되었다. 정책은 측정된 상태를 기준으로 상단을 정의하므로, 어긋난 기준점을 새 평형점처럼 지켜낸 셈이다. 중력 평형점이 아닌 자세를 유지하는 데에는 기어열의 정지마찰과 반동 질량의 운동이 기여한 것으로 추정되나 정확한 기전은 추가 분석이 필요하다. 이 현상은 하드웨어 디코더로 교체한 뒤 사라졌다.

5. 고찰

서론의 세 공백에 대해 본 연구의 결과는 다음과 같다. 첫째, 병렬 탐색이 불가능한 실기에서도 LLM 에이전트가 단계마다 하나의 처방을 적용하고 판독하는 순차 루프로 보상을 설계하여, 그대로는 동작하지 않던 시물레이션 정책 초기 가중치로부터 기어 교체를 거쳐 정착 0.35°까지 도달하였다. 둘째, 양방향 포획과 무킥 스윙업 같은 질적 향상은 보상이 아닌 하네스 조치(⑨~⑫)에서 나왔다. 이는 하네스와 정책을 함께 진화시켜야 한다는 HarnessForge의 주장[20]과 구현 세부가 실기 학습을 좌우한다는 보고[2]가 연속 제어 실기에서도 성립함을 보여 준다. 셋째, 보상 해킹과 계측 결함은 스칼라 지표로는 드러나지 않았고 물리 진단과 과형 판독, 사람의 감독으로 발견되었다. 이는 궤적을 직접 보는 진단이 거친 수치보다 낫다는 RDA의 결과[18] 및 작업자 관찰이 LLM 제어의 결함 검출을 돕는다는 보고[24]와 같은 방향이다.

그러나 한계도 존재한다. 첫째, 단계별 정착 오차는 대부분 greedy 평가 궤적 1개에서 계산되었고 초기 가중치 이식 반복이나 신뢰구간이 없다. 둘째, 고전 제어 기준선, 사람이 설계한 보상, 보상을 고정한 SAC와의 비교가 없으며 LLM의 기여를 사용자 판단·하드웨어 수정과 분리하는 절차 실험도 없다. 셋째, 기어 교체, 엔코더 디코더 교체, 캐리어 축 재출력, Z 영점 보정으로 학습 중 플랜트와 계측 기준이 바뀌어 단계 간 비교가 엄밀히 통제되지 않았다. 넷째, 상용 LLM은 비결정적이므로 같은 절차의 재현을 보장하는 프로토콜이 필요하다. 비용 면에서 실기 학습은 약 150 에피소드에 1시간이 걸렸고 최종 학습 정책은 약 6시간이 소요되었으나, LLM 토큰 사용량은 기록하지 않았다.

향후 최종 하드웨어를 고정한 조건에서 주요 정책을 각 20회 이상 평가하여 평균과 표준편차를 보고하여 앞서 얘기한 한계를 극복할 예정이며 이렇게 만들어진 실

물 하드웨어에서의 정책을 사람이 이해할 수 있고 무거운 신경망 추론을 사용할 필요없는 제어기 수식을 추출하여 유성기어 하드웨어에서 신경망 추론을 통한 정책으로 역진자 제어를 진행하는 것과 추출한 제어기로 구동하여 역진자 제어를 진행하는 것을 비교하여 추출한 제어기의 사용 적합성 확인을 수행할 계획이다.

6. 결론

본 논문은 LLM 코딩 에이전트가 RDA 역할을 맡아 실기 제작의 물리 진단에 근거해 보상함수와 학습 하네스를 함께 진화시키는 HILL 강화학습 프레임워크를 제안하고, 이를 3D 프린팅 유성기어 역진자에 적용하였다. Teensy 4.1이 1 kHz로 정책을 온보드 추론하고 PC가 micro-ROS로 받은 실기 전이로 SAC를 학습하는 구조에서 시뮬레이션 사전학습 정책을 초기 가중치로 이식하여 구 기어에서 시작하여 기어 교체 뒤 구 기어 정책을 다시 초기 가중치로 이어 간 13단계 진화는 정각 각오차 0.35° , $|e| < 1^\circ$ 체류율 100%, 유지 토크 RMS 0.0002 Nm, 0.34 s 스윙업에 도달하였다. 이 과정에서 RDA는 보상해킹, 정밀-정속 트레이드오프, 가우시안 보상의 기울기 소멸, 리플레이 버퍼의 방향 편중, 계측 결함을 찾아 처방하였다. 최종 정책은 학습하지 않은 외란에서도 복원하였고, 큰 외란에는 진자를 떨어뜨려 반대편으로 제스윙업하는 전략을 스스로 보였다. 이 결과는 LLM 기반 보상·하네스 진화가 시뮬레이션을 넘어 복잡한 실제 하드웨어의 강화학습 제어에 적용될 수 있음을 보여 준다. 다만 일반화 가능한 결론을 위해서는 5장에서 제시한 반복평가, 기준선 비교, 절제 실험이 뒤따라야 한다.

※ 본 연구는 (과제명 및 지원기관)의 지원에 의하여 연구되었음.

참고문헌

- [1] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, no. 1, pp. 153-188, May, 2025.
- [2] J. Luo, Z. Hu, C. Xu, Y. L. Tan, J. Berg, A. Sharma, et al., "SERL: A software suite for sample-efficient robotic reinforcement learning," *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 16961-16969, May, 2024.
- [3] J. Luo, C. Xu, J. Wu, and S. Levine, "Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning," *Science Robotics*, vol. 10, no. 105, eads5033, Aug., 2025.
- [4] J. Koo, S. Lee, D. Ju, and Y. S. Lee, "MuJoCo-based sim-to-real reinforcement learning for transition control of a triple inverted pendulum," *Electronics*, vol. 15, no. 17, 4048, Sep., 2026.
- [5] 임창석, 주도운, 이영삼, "Sim-to-Real 강화학습 기법을 활용한 직선형 3단 도립진자의 56가지 천이 제어 구현," *전기학회논문지*, 제74권, 제8호, pp. 1363-1372, 2025.
- [6] 조용현, 이관우, 박경재, 임서현, 신재용, 최현준, 박상혁, 박재홍, 허성문, "접촉력 추정 기반 강화학습을 활용한 휴머노이드 이족 보행의 순응적 착지 프레임워크," *로봇학회 논문지*, 제21권, 제3호, pp. 269-282, 2026.
- [7] 신성준, 김종원, 윤지민, 공경철, "고충실도 Sim-to-real 전환을 위한 물리 시뮬레이터 내 구동기 열 모델 통합 연구," *제어·로봇·시스템학회 논문지*, 제32권, 제1호, pp. 52-58, 2026.
- [8] 강대열, 박종윤, 김필은, "강화학습을 이용한 Sim-to-real을 위한 격차에 대한 연구: Gazebo와 Isaac Sim을 이용한 TD3 최적화," *제어·로봇·시스템학회 논문지*, 제31권, 제11호, pp. 1290-1298, 2025.
- [9] 이재욱, 주도운, 이영삼, "LW-RCP 기반 회전형 도립진자의 강화학습 제어 실험 및 교육 플랫폼 개발," *제어·로봇·시스템학회 논문지*, 제32권, 제9호, pp. 1196-1203, 2026.
- [10] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2024.
- [11] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "Text2Reward: Reward shaping with language models for reinforcement learning," *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2024.
- [12] Y. J. Ma, W. Liang, H.-J. Wang, S. Wang, Y. Zhu, L. Fan, O. Bastani, and D. Jayaraman, "DrEureka: Language model guided sim-to-real transfer," *Proc. of Robotics: Science and Systems (RSS)*, July, 2024.
- [13] K. Ryu, Q. Liao, Z. Li, P. Delgosha, K. Sreenath, and N. Mehr, "CurricuLLM: Automatic task curricula design for learning complex robot skills using large language models," *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4470-4477, May, 2025.
- [14] N. Turcato, M. Iovino, A. Synodinos, A. Dalla Libera, R. Carli, and P. Falco, "Towards autonomous reinforcement learning for real-world robotic manipulation with large language models," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 8850-8857, Sep., 2025.
- [15] 장재준, "대규모 언어 모델 기반의 전략적 파라미터 생성을 통한 상황 적응형 군집 무인기 임무 할당," *로봇학회 논문지*, 제21권, 제3호, pp. 351-362, 2026.
- [16] 이준기, 박상준, 김낙우, 김에텐, 고석갑, "거대언어 모델 기반 로봇 인공지능 기술 동향," *전자통신동향분석*, 제39권, 제1호, pp. 95-105, 2024.
- [17] C. Muslimani, K. Johnstonbaugh, S. Chandramouli, S. Booth, W. B. Knox, and M. E. Taylor, "Towards improving reward design in RL: A reward alignment metric for RL practitioners," *Reinforcement Learning Journal*, vol. 6, pp. 2342-2367, 2025.

- [18] H. Lee, A. Subramanian, B. Abbatematteo, V. Veerabadrar, P. Matias, K. Ridgeway, and N. Kamra, “RDA: Reward design agent for reinforcement learning,” arXiv preprint arXiv:2606.01672, June, 2026.
- [19] T. Chen, S. Lu, K. Zhao, W. Meng, K. Shao, J. Luan, et al. (Darwin Agent Team), “HarnessX: A composable, adaptive, and evolvable agent harness foundry,” arXiv preprint arXiv:2606.14249, June, 2026.
- [20] M. Chen, C. Lv, G. Zhang, H. Chang, and S. Zhou, “HarnessForge: Joint harness and policy evolution for adaptive agent systems,” arXiv preprint arXiv:2606.01779, June, 2026.
- [21] Y. Lee, R. Nair, Q. Zhang, K. Lee, O. Khattab, and C. Finn, “Meta-Harness: End-to-end optimization of model harnesses,” arXiv preprint arXiv:2603.28052, Mar., 2026.
- [22] J. Lin, S. Liu, C. Pan, L. Lin, S. Dou, Z. Xi, et al., “Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses,” arXiv preprint arXiv:2604.25850, Apr., 2026.
- [23] A. Pan, E. Jones, M. Jagadeesan, and J. Steinhardt, “Feedback loops with language models drive in-context reward hacking,” Proc. of the 41st Int. Conf. on Machine Learning (ICML), 2024.
- [24] 전성윤, 김태환, 김남훈, “적층제조에서 작업자 개입이 대형 언어 모델 기반 공정 제어에 미치는 영향 분석,” 한국정밀공학회지, 제43권, 제9호, pp. 907-923, 2026.
- [25] 유승원, 고영훈, 성민영, “큐브위성 비행소프트웨어를 위한 micro-ROS 응용 및 성능 평가,” 정보과학회 컴퓨팅의 실제 논문지, 제32권, 제8호, pp. 373-378, 2026.
- [26] H. Lee, D. Hwang, D. Kim, H. Kim, J. J. Tai, K. Subramanian, et al., “SimBa: Simplicity bias for scaling up parameters in deep reinforcement learning,” Proc. of the Int. Conf. on Learning Representations (ICLR), 2025.