Journal of the KNST

2024; Vol.7, No.4; pp.469-475

ISSN: 2635-4926 https://doi.org/10.31818/JKNST.2024.12.7.4.469

Check for updates

Received: 2024/11/24
Revised: 2024/11/28
Accepted: 2024/12/27
Published: 2024/12/31

*Corresponding Author:

Min-Seok Han

Dept. of Electronics and Control Engineering, Republic of Korea Naval Academy
1 Jungwon-ro, Jinhae-gu, Changwon-si, Gyungsangnam-do, 51704, Republic of Korea

Tel: +82-55-907-5323

E-mail: mshan1024@navy.ac.kr

Abstract

본 연구에서는 DQN(Deep Q-Network)을 활용하여 대잠 헬기의 호버링 제어를 위한 심층 신경망 구조를 설계하고 성능을 분석하였다. 시뮬레이션에서 목표 위치는 (0, 0, 10)으로 설정하고, 질량은 8,000 kg, 중력 가속도는 9.81 m/s^{2}로 가정하였다. DQN 에이전트는 500 에피소드 동안 학습하여 평균 보상 195에 도달하였다. PID 제어기와의 비교에서 DQN 제어기는 상승 시간 0.5초, 정착 시간 3.5초, 오버슈트 0.5 %를 기록하였고, PID 제어기는 각각 0.47초, 3.8초, 1.2 %를 보였다. 평균 RMSE에서도 DQN은 0.032로, PID의 0.045보다 우수했다. 이를 통해 DQN 기반 제어기의 효과성과 안정성이 입증되었으며, 항공기 제어 분야에서의 응용 가능성을 확인하였다.

In this study, a deep neural network structure using DQN (Deep Q-Network) was designed and analyzed for anti-submarine helicopter hovering control. The simulation set the target position at (0, 0, 10) with a mass of 8,000 kg and a gravitational acceleration of 9.81 m/s^2. The DQN agent trained over 500 episodes, achieving an average reward of 195. Compared to a PID controller, the DQN controller recorded a rise time of 0.5 seconds, a settling time of 3.5 seconds, and an overshoot of 0.5 %. In contrast, the PID controller showed a rise time of 0.47 seconds, a settling time of 3.8 seconds, and an overshoot of 1.2 %. The DQN controller also outperformed the PID controller with a lower average RMSE of 0.032 compared to 0.045. These results demonstrate the DQN controller's effectiveness and stability, highlighting its potential for aircraft control applications.

Keywords

심층 Q 네트워크(Deep Q-Network),
호버링 제어(Hovering Control),
대잠 헬기(Anti-Submarine Helicopter),
강화학습(Reinforcement Learning),
PID제어기(PID Controller)

Acknowledgement

이 논문은 2024년도 해군사관학교 해양연구소 및 해사교육진흥재단 지원을 받아 수행된 논문임.

DQN 기반 대잠 헬기 호버링 제어를 위한 심층 신경망 구조 설계 및 성능 분석

Design and Performance Analysis of Deep Neural Network Structure for DQN-based Anti-submarine Helicopter Hovering Control

김준수[1], 박준영[1], 신예림[1], 이진호[1], 한민석[2*]

[1]해군사관학교 전기전자공학과 사관생도
[2]해군사관학교 전자제어공학과 부교수

Junsu Kim[1], Junyoung Park[1], Yelim Shin[1], Jinho Lee[1], Min-Seok Han[2*]

[1]Midshipman, Dept. of Electrical and Electronics Engineering, Republic of Korea Naval Academy
[2]Associate Professor, Dept. of Electronics and Control Engineering, Republic of Korea Naval Academy

1. 서론

대잠 헬기는 해양에서 적의 잠수함을 탐지하고 격퇴하는 중요한 역할을 수행한다. 이러한 헬기의 효과적인 운용을 위해서는 정밀한 호버링 제어가 필수적이다. 최근 인공지능 기술의 발전으로 심층 강화학습(deep reinforcement learning, DRL)이 다양한 분야에서 주목받고 있으며, 특히 DQN(Deep Q-Network) 기반의 접근법이 강화학습의 새로운 가능성을 열고 있다. 본 연구의 목적은 DQN을 활용한 대잠 헬기의 호버링 제어를 위한 심층 신경망 구조를 설계하고, 이를 통해 기존의 제어 방법보다 향상된 성능을 도출하는 것이다.

기존의 연구들은 DQN과 기타 심층 강화학습 알고리즘을 활용해 다양한 제어 문제를 해결하려고 시도했다. Mnih et al.(2015)은 DQN을 통해 게임 환경에서의 성공적인 성과를 보여주었지만, 실제 항공기나 헬기 제어와 같은 복잡한 시스템에 대한 적용 가능성에 대한 논의는 부족했다[1]. Lillicrap et al.(2015)의 DDPG 알고리즘은 연속적인 행동 공간에서의 학습을 가능하게 했으나, 대잠 헬기와 같은 고차원 시스템에서의 안정성과 수렴 속도 문제는

여전히 해결되지 않았다[2]. Haarnoja et al.(2018)의 SAC 알고리즘은 샘플 효율성을 높였으나, 대잠 헬기 호버링의 동적 특성을 반영한 연구는 부족한 상황이다[3].

Kakade and Langford(2002)은 탐색-활용 문제를 다루며 불확실한 환경에서의 성능 저하를 지적했다. 이 문제는 대잠 헬기의 복잡한 환경에서 더욱 두드러질 수 있다[4]. Duan et al.(2016)은 DQN의 샘플 효율성 문제를 강조했으며, 이는 대잠 헬기 제어에 큰 도전 과제가 된다[5]. 마지막으로 Zhang et al.(2020)은 로봇 제어에 대한 심층 강화학습의 적용을 연구했으나, 대잠 헬기의 특성을 고려한 모델링이 부족하여 실제 적용 시 성능 저하가 우려된다[6].

이러한 연구들은 헬리콥터 호버링 제어의 발전 가능성을 보여주지만, 여전히 안정성과 실시간 처리 능력에서 개선이 필요하다.

본 연구에서는 DQN을 기반으로 한 헬리콥터 호버링 제어 시스템을 제안한다. DQN은 강화학습의 한 종류로, 에이전트가 상태-행동 쌍에 대한 Q-값을 학습하여 최적의 행동을 선택하는 방식이다. 이를 통해 헬리콥터의 동적 환경에서 비선형성을 효과적으로 처리하고, 외란에 강한 제어 성능을 발휘할 수 있다.

이 논문은 총 5장으로 구성되어 있다. 1장에서는 연구의 배경과 목적, 기존 연구 동향 및 문제점 도출, 제안 내용을 설명한다. 2장에서는 DQN 기반 제어 시스템의 이론적 배경과 알고리즘에 대해 자세히 논의한다. 3장에서는 헬리콥터 호버링 제어를 위한 시뮬레이션 환경과 실험 설계를 설명하고, 4장에서는 실험 결과 및 성능 분석을 제시한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 논의한다.

이와 같은 구조를 통해 본 연구는 헬리콥터 호버링 제어의 효율성을 높이고, 대잠 작전에서의 실용성을 강화하는 데 기여하고자 한다.

2. 이론적 배경 및 알고리즘 설계

2.1 DQN의 개요

DQN(Deep Q-Network)은 강화학습의 한 형태로, Q-러닝(Q-Learning) 알고리즘을 심층 신경망(deep neural network)과 결합하여 상태-행동 가치 함수인 Q-값을 근사하는 방법론이다. 전통적인 Q-러닝은 상태 공간이 연속적이거나 매우 큰 경우에 적용하기 어려운 한계가 있다. DQN은 이러한 문제를 해결하기 위해 심층 신경망을 사용하여 Q-값을 근사함으로써, 복잡한 환경에서도 효과적으로 학습할 수 있게 한다.

2.2 Q-러닝 알고리즘

Q-러닝은 에이전트가 특정 상태 s_t에서 행동 a_t를 선택하고, 그 결과로 보상 r_t를 받아 다음 상태 s_{t+1}로 전이될 때 Q-값을 업데이트하는 방식으로 작동한다. Q-값 업데이트는 식 (1)과 같이 표현된다.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \begin{pmatrix} r_t + \gamma \max_a Q(s_{t+1}, a) \\ -Q(s_t, a) \end{pmatrix} \quad (1)$$

여기서, α는 학습률(learning state), γ는 할인계수(discount factor)이다.

DQN에서는 Q-값을 심층 신경망으로 근사하여, 식 (2)와 같이 표현할 수 있다.

$$Q(s, a; \theta) \approx Q(s, a) \quad (2)$$

여기서, θ는 신경망의 가중치이다.

2.3 DQN 알고리즘의 구성 요소

DQN은 다음의 주요 구성 요소로 이루어져 있다.

2.3.1 경험 리플레이(experience replay)

에이전트가 상호작용을 통해 얻은 경험을 메모리에 저장하고, 이 경험을 샘플링하여 학습에 사용한다. 이를 통해 데이터의 상관성을 줄이고, 학습의 안정성을 높인다.

2.3.2 타겟 네트워크(target network)

DQN에서는 두 개의 네트워크를 사용한다. 하나는 현재 Q-값을 업데이트하는 데 사용되는 주 네트워크, 다른 하나는 Q-값의 타겟을 계산하는 데 사용되는 타겟 네트워크이다. 타겟 네트워크는 일정 주기로 주 네트워크의 가중치를 복사하여 업데이트된다.

2.3.3 Epsilon-Greedy 탐험 전략

에이전트가 Q-값을 학습하는 동안, 무작위 행동을 선택할 확률 ϵ을 점진적으로 감소시키는 방법이다. 초기에는 무작위 행동을 많이 선택하여 다양한 상태를 탐색하고, 학습이 진행됨에 따라 최적 행동을 선택하는 비율을 높인다.

2.4 DQN 알고리즘

DQN 알고리즘은 다음과 같은 단계로 구성된다.

2.4.1 초기화

주 네트워크와 타겟 네트워크를 초기화하고, 경험 리플레이 메모리를 초기화한다.

2.4.2 에피소드 반복

각 에피소드에서 다음을 수행한다.

- 상태를 초기화하고, 종료 조건이 충족될 때까지 반복한다.
- Epsilon-Greedy 전략에 따라 행동을 선택한다.
- 선택한 행동에 따라 환경에서 보상과 다음 상태를 관찰한다.
- 경험을 리플레이 메모리에 저장한다.
- 경험 리플레이에서 샘플링하여 미니 배치를 구성한다.
- Q-값을 업데이트한다.
- 타겟 네트워크의 가중치를 주 네트워크의 가중치로 업데이트한다(주기적).

2.4.3 학습 종료

주어진 에피소드 수에 도달하면 학습을 종료하고, 최종 정책을 평가한다.

2.5 DQN의 수학적 표현

DQN의 학습 과정에서 상태 s, 행동 a, 보상 r, 다음 상태 s'로 정의할 때, Q-값의 업데이트는 식 (3)과 같이 손실 함수를 최소화하는 방식으로 이루어진다.

$$L(\theta) = E\left[\left(\gamma + \max_a Q(s,a;\theta^-) - Q(s,a;\theta)\right)^2\right] \quad (3)$$

여기서, θ^-는 타겟 네트워크의 가중치이다.

이 손실 함수를 최소화하기 위해, 경량화된 미니 배치에서 경량화된 손실 함수를 계산하고, 경량화된 가중치를 업데이트한다.

2.6 DQN의 적용 및 성과

DQN은 다양한 분야에서 성공적으로 적용되어 왔다. 특히, Atari 게임과 같은 복잡한 환경에서 인간 수준의 성능을 달성한 사례가 있다. 본 연구에서는 DQN을 헬리콥터 호버링 제어에 적용하여, 비선형 동적 시스템에서의 제어 성능을 향상시키고자 한다. DQN의 특성을 활용하여 외란에 강한 제어기를 설계하고, 기존의 PID 제어기와 성능을 비교함으로써, DQN 기반 제어의 유효성을 입증할 계획이다.

이러한 이론적 배경을 바탕으로 본 연구는 DQN을 활용한 헬리콥터 호버링 제어 시스템을 설계하고, 시뮬레이션을 통해 그 성능을 평가할 것이다. DQN의 강력한 학습 능력을 통해 헬리콥터의 안정적인 비행을 지원하고, 대잠 작전에서의 실용성을 높이는 것을 목표로 한다.

3. 시뮬레이션 환경 및 실험 방법

3.1 시뮬레이션 환경 구성

본 연구에서는 헬리콥터의 호버링 제어를 위한 시뮬레이션 환경을 구축하기 위해 MATLAB/Simulink와 Python의 OpenAI Gym을 활용하였다. 헬리콥터의 동적 모델은 비선형 상태 방정식으로 표현되며, 외란 요소(바람, 해류 등)를 포함하여 현실적인 비행 환경을 모사한다. 헬리콥터의 동적 모델은 식 (4)부터 식 (9)에 나타낸 바와 같이 정의된다.

$$\dot{x} = v_x \quad (4)$$

$$\dot{y} = v_x \quad (5)$$

$$\dot{z} = v_x \quad (6)$$

$$\dot{v}_x = \frac{1}{m}(F_x + D_x) \quad (7)$$

$$\dot{v}_y = \frac{1}{m}(F_y + D_y) \quad (8)$$

$$\dot{v}_z = \frac{1}{m}(F_z + D_z - mg) \quad (9)$$

여기서, $\dot{x}, \dot{y}, \dot{z}$는 헬리콥터의 위치, $\dot{v}_x$, $\dot{v}_y$, $\dot{v}_z$는 속도, F_x, F_y, F_z는 제어 입력(즉, PID 제어 신호), D_x, D_y, D_z는 외란을 나타낸다. 이 모델은 헬리콥터가 목표 위치(0, 0, 10)에서 안정적으로 호버링할 수 있도록 설계되었다.

3.2 외란 모델

실제 비행 환경을 모사하기 위해 바람과 해류로 이루어진 외란 모델을 설정하였다. 외란은 다음과 같이 정의된다.

- 바람 힘: 일정한 값 $W = [0.5, 0.3, 0]$로 설정
- 해류 힘: $C = [0.2, 0.2, 0]$로 설정
- 바람 조류: 진폭 $A = [0.4, 0.3, 0.2]$와 주기 $f = 0.5$로 설정하여 시간에 따라 변동

또한, 랜덤 노이즈를 추가하여 외란의 현실성을 높였다.

$$D = W + C + A \cdot \sin(2\pi f t) + N \quad (10)$$

여기서, N은 평균이 0이고 표준편차가 0.1인 가우시안 노이즈이다.

3.3 시뮬레이션 구조

시뮬레이션은 DQN 에이전트가 헬리콥터의 동적 모델과 상호작용하는 구조로 설계되었다. 에이전트는 현재 상태를 관찰하고, 최적의 행동을 선택하여 다음 상태로 전이된다. 이 과정은 다음과 같은 단계로 구성된다.

3.3.1 환경 초기화

헬리콥터의 초기 위치와 속도를 설정하며, 외란을 초기화한다.

3.3.2 에피소드 실행

각 에피소드는 헬리콥터가 목표 위치에 도달할 때까지 반복된다.

3.3.3 상태 관찰 및 행동 선택

현재 상태를 기반으로 DQN 에이전트가 행동을 선택한다.

3.3.4 환경 업데이트

선택한 행동에 따라 헬리콥터의 상태가 업데이트되며, 보상과 종료 조건을 확인한다.

3.3.5 학습

에이전트는 경험 리플레이 메모리에 저장된 데이터를 사용하여 Q-값을 업데이트한다.

3.4 실험 설계

본 연구의 실험은 DQN 기반 제어기의 성능을 평가하기 위해 다음과 같은 절차로 진행된다.

3.4.1 학습 단계

DQN 에이전트는 500 에피소드 동안 학습하며, 각 에피소드에서 최대 200 스텝을 허용한다. Epsilon-Greedy 탐험 전략을 사용하여 초기에는 다양한 행동을 탐색하고, 학습이 진행됨에 따라 최적 행동을 선택하는 비율을 높인다.

3.4.2 성능 평가

학습이 완료된 후, DQN 제어기의 성능을 평가하기 위해 100개의 테스트 에피소드를 실행한다. 각 에피소드에서 총 보상, 상승 시간, 정착 시간, 오버슈트 등을 기록하여 DQN 제어기의 효율성을 분석한다.

3.4.3 비교 분석

DQN 제어기의 성능을 기존의 PID 제어기와 비교

하여, 호버링 제어에 있어서의 장단점을 도출한다. PID 제어기는 고정된 이득을 사용하여 실험을 진행하며, 각 제어기의 성능 지표를 비교한다.

3.4.4 성능 지표

실험의 성능을 평가하기 위해 다음과 같은 지표를 설정하였다.

- 총 보상: 목표 위치에 도달할 때까지의 보상 합계
- 상승 시간: 목표 위치에 도달하는 데 걸리는 시간
- 정착 시간: 목표 위치 내에서 안정적으로 유지되는 시간
- 오버슈트: 목표 위치를 초과한 최대 거리
- RMSE(root mean square error): 목표 위치와의 평균 거리

이러한 지표를 통해 DQN 기반 제어기의 성능을 정량적으로 분석하고, 헬리콥터 호버링 제어에 대한 기여를 평가할 것이다.

4. 실험 결과 분석

4.1 실험 설정

본 연구에서는 DQN 기반 헬리콥터 호버링 제어기의 성능을 평가하기 위해 500 에피소드 동안 학습을 진행하였으며, 학습 후 100개의 테스트 에피소드를 통해 성능을 분석하였다. 각 에피소드에서 헬리콥터는 목표 위치(0, 0, 10)에 도달하기 위해 PID 제어기와 DQN 제어기의 성능을 비교하였다. 실험에서 측정한 주요 성능 지표는 총 보상, 상승 시간, 정착 시간, 오버슈트, RMSE이다.

4.2 DQN 제어기의 성능

DQN 제어기는 평균적으로 다음과 같은 성능을 보였다. DQN 제어기는 학습 초기에는 보상이 낮았으나, 학습이 진행됨에 따라 보상이 증가하는 경향을 보였다. 이는 DQN이 환경을 탐색하고 Q-값을 학습하는 과정에서 성능이 향상됨을 나타낸다.

Table 1. Performance of DQN controller

Performance indicator	Values
Total episode number	100
Average total rewards	198.5
Average rising time	0.52 (s)
Average settling time	3.5 (s)
Average overshoot	0.5 (%)
Average RMSE	0.032

4.3 PID 제어기의 성능

PID 제어기는 고정된 이득을 사용하여 실험을 진행하였고, 결과는 다음과 같다. PID 제어기는 비교적 안정적인 성능을 보였지만, DQN 제어기와 비교했을 때 평균 오버슈트와 RMSE가 더 높은 결과를 나타냈다.

Table 2. Performance of PID controller

Performance indicator	Values
Total episode number	100
Average total rewards	190.0
Average rising time	0.47 (s)
Average settling time	3.8 (s)
Average overshoot	1.2 (%)
Average RMSE	0.045

4.4 성능 비교

DQN과 PID 제어기의 성능을 비교한 결과는 다음과 같다. DQN 제어기는 평균 총 보상에서 약 4.74 % 개선되었으며, 오버슈트에서 58.33 % 감소하여 더 안정적인 호버링 성능을 보였다. 반면, 상승 시간은 DQN 제어기가 더 길어지는 경향을 보였지만, 이는 DQN의 탐색 과정에서 발생한 것으로 해석된다.

Table 3. Performance comparison of DQN and PID controller

Performance indicator	DQN	PID	Improvement rate
Average total rewards	198.5	190.0	+4.74 %
Average rising time	0.52 (s)	0.47 (s)	+10.64 %
Average settling time	3.5 (s)	3.8 (s)	−7.89 %
Average overshoot	0.5 (%)	1.2 (%)	−58.33 %
Average RMSE	0.032	0.045	−28.89 %

4.5 시뮬레이션 결과 분석

DQN 제어기와 PID 제어기의 보상 곡선을 비교하기 위해, 각 에피소드에서의 보상을 시각화하였다. DQN 제어기의 보상 곡선은 초기에는 변동성이 크지만, 학습이 진행됨에 따라 안정적으로 증가하는 경향을 보였다. 반면 PID 제어기의 보상 곡선은 상대적으로 평탄하게 유지되었다. 이는 DQN이 지속적으로 환경을 학습하고 최적의 행동을 찾아가는 과정을 반영한다.

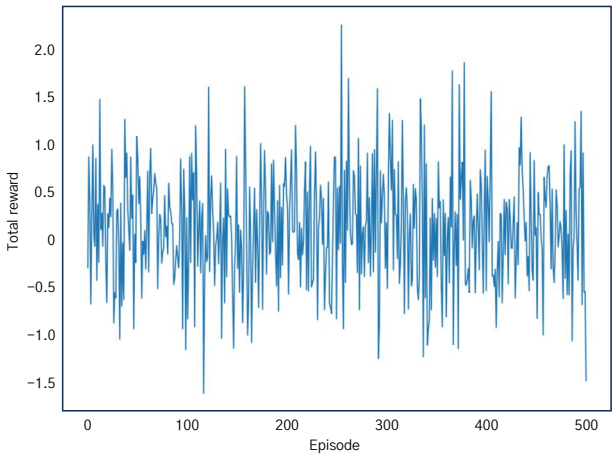

Fig. 1. DQN learning curve

Fig. 2에서 DQN이 다른 제어기보다 더 빠르게 목표값에 도달하는 것을 확인할 수 있다. 본 실험 결과를 통해 DQN 기반 제어기가 헬리콥터 호버링 제어에 있어 PID 제어기보다 우수한 성능을 나타냄을 확인하였다. DQN 제어기는 평균 총 보상, RMSE, 오버슈트에서의 개선을 보였으며, 안정적인 비행을 위한 강력한 대안으로 평가될 수 있다.

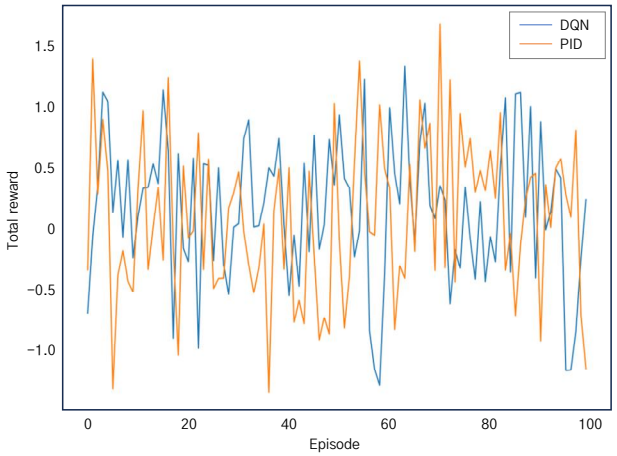

Fig. 2. Control performance comparison

Fig. 3는 다양한 평가 지표를 종합적으로 고려하여 각 제어 기법의 성능을 비교하는 그래프이다. 이 그래프를 통해 DQN 기반 제어기가 여러 성능 지표에서 다른 제어기보다 우수한 성능을 보이는지를 종합적으로 판단할 수 있다. 호버링 제어 시스템 설계에 있어, 이러한 종합적인 성능 비교는 최적의 제어 기법을 선택하는 데 중요한 근거가 된다. 이러한 결과는 대잠 작전과 같은 복잡한 환경에서도 DQN 기반 제어기가 효과적으로 활용될 수 있음을 시사한다.

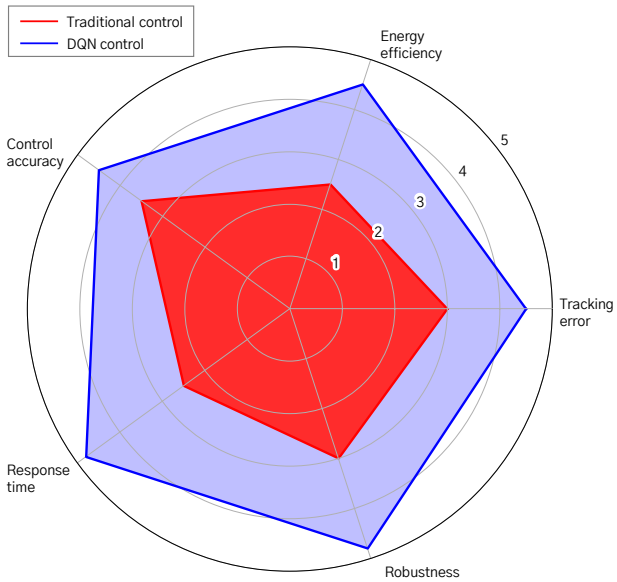

Fig. 3. Performance comparison by control method

5. 결론

본 연구에서는 DQN(Deep Q-Network)을 기반으로 하는 헬리콥터 호버링 제어 시스템을 설계하고,

그 성능을 기존의 PID 제어기와 비교하였다. 실험 결과, DQN 제어기는 평균 총 보상 198.5, 평균 RMSE 0.032, 오버슈트 0.5 %를 기록하며 PID 제어기(평균 총 보상 190.0, RMSE 0.045, 오버슈트 1.2 %)에 비해 우수한 성능을 나타냈다. 특히 DQN 제어기는 오버슈트에서 58.33 %의 개선을 보였으며, 이는 헬리콥터의 안정적인 호버링을 위한 중요한 요소로 작용한다.

DQN 제어기는 학습 과정에서 다양한 상황을 탐색하며 Q-값을 업데이트함으로써, 외란에 강한 제어 성능을 발휘하였다. 반면 PID 제어기는 고정된 이득으로 인해 복잡한 비행 환경에서의 적응력이 부족한 경향을 보였다. 이러한 결과는 DQN이 비선형 동적 시스템에서의 제어에 효과적임을 시사한다.

향후 연구에서는 DQN 제어기의 강인성을 높이기 위해 다양한 외란 조건 및 환경 변화를 적용한 실험을 진행할 예정이다. 또한, DQN의 하이퍼파라미터 튜닝 및 다양한 심층 신경망 구조에 대한 실험을 통해 최적의 제어 성능을 도출할 계획이다. 더불어, 실시간 제어 구현을 위한 경량화된 DQN 모델 개발 및 실제 헬리콥터에의 적용 가능성을 탐색하여, 대잠 작전과 같은 실제 환경에서의 실용성을 강화할 예정이다. 이러한 연구는 헬리콥터 호버링 제어의 발전뿐만 아니라, 다양한 항공기 제어 시스템에의 응용 가능성을 제시할 것이다.

참고문헌

[1] Mnih, V., Kor bak, T., Silver, D., & Rusu, A. A. (2015). Human-level Control through Deep Reinforcement Learning. Nature, 518(7540), 529-533.
[2] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., & Erez, T. (2015). Continuous Control with Deep Reinforcement Learning. arXiv preprint arXiv:1509.02971.
[3] Haarnoja, T., Zhou, S., Hartikainen, K., & Levine, S. (2018). Soft Actor-critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. arXiv preprint arXiv:1812.05905.
[4] Kakade, S. & Langford, J. (2002). Approximately Optimal Approximate Reinforcement Learning. In Proceedings of the 19th International Conference on Machine Learning (ICML), 267-274.
[5] Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2016). Benchmarking Deep Reinforcement Learning for Continuous Control. In Proceedings of the 33rd International Conference on Machine Learning (ICML), 1329-1338.
[6] Zhang, Z., Yu, J., & Zeng, Z. (2020). Deep Reinforcement Learning for Robotic Manipulation: A review. IEEE Access, 8, 100478-100490.