Journal of the KIMST, Vol. 24, No. 1, pp. 123-132, 2021
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2021.24.1.123

Research Paper 제어·구동 부문

Advantage Actor-Critic 강화학습 기반 수중운동체의 롤 제어

이 병 준*,1)

1) 국방과학연구소 제4기술연구본부

Roll control of Underwater Vehicle based Reinforcement Learning using Advantage Actor-Critic

Byungjun Lee*,1)

1) *The 4th Research and Development Institute, Agency for Defense Development, Korea*

(Received 18 September 2020 / Revised 14 January 2021 / Accepted 22 January 2021)

Abstract

In order for the underwater vehicle to perform various tasks, it is important to control the depth, course, and roll of the underwater vehicle. To design such a controller, it is necessary to construct a dynamic model of the underwater vehicle and select the appropriate hydrodynamic coefficients. For the controller design, since the dynamic model is linearized assuming a limited operating range, the control performance in the steady state is well satisfied, but the control performance in the transient state may be unstable. In this paper, in order to overcome the problems of the existing controller design, we propose a A2C(Advantage Actor-Critic) based roll controller for underwater vehicle with stable learning performance in a continuous space among reinforcement learning methods that can be learned through rewards for actions. The performance of the proposed A2C based roll controller is verified through simulation and compared with PID and Dueling DDQN based roll controllers.

Key Words : Reinforcement Learning(강화학습), Actor-Critic(행동자-비평자), Underwater Vehicle(수중운동체), Roll Control(롤 제어)

1. 서 론

최근에 수중운동체는 민간 분야에서부터 국방 분야까지 다양한 분야에서 활용되고 있으며 특히 수중운동체가 다양한 임무를 수행하기 위해서는 먼저 수중운동체의 자동조종(autopilot) 제어기를 통한 심도(depth), 경로(course), 롤(roll) 제어가 매우 중요하다. 이러한 제어기를 설계하기 위해서는 수중운동체의 운동 모델을 구성하고 유체 계수를 선정하는 것이 필요하다.

유체력은 속도, 심도, 자세 등의 영향에 따라서 비선형 특성을 가지므로 제한된 작동범위 내에서 비선형 시스템을 선형화하고 이를 바탕으로 제어기를 설

* Corresponding author, E-mail: bj0208@hanmail.net

계하고 적용한다. 제한된 작동범위 내에서 설계가 이루어지기 때문에 정상상태에서의 제어성능은 잘 만족하나 과도상태에서의 제어 성능은 불안정해질 가능성이 존재한다.

이러한 기존 제어기의 문제점 및 설계 방법을 보완하기 위하여 인공지능을 이용한 학습 방법에 대한 연구가 활발히 진행되고 있다[1-4]. 인공지능의 알고리즘은 크게 지도학습(supervised learning)과 비지도학습(unsupervised learning), 강화학습(reinforcement learning)으로 나누어지며 강화학습의 방법 중 가치함수를 추정하는 DQN(Deep Q-Network)[5], DQN을 보다 안정화시킨 방법인 DDQN(Double Deep Q-Network)[6] 그리고 상태와 행동을 나누어 학습하는 Dueling Q-Network[7]이 있다. 또한 직접 정책을 추정하는 강화학습 방법으로는 Critic 신경망에서 추정된 가치함수를 이용하여 Actor 신경망의 정책을 학습하는 Advantage Actor-Critic[8,9]이 연구되었다.

본 논문에서는 기존 제어기 단점을 극복하기 위한 방법으로 제어기 설계가 필요 없는 대신 행동에 대한 보상을 통해 학습할 수 있는 강화학습 방법 중에서 연속 공간(continuous space)에서 안정된 신경망 학습이 가능한 Advantage Actor-Critic 기반의 수중운동체의 롤 제어기를 제안하고 제안된 제어기가 스스로 롤 제어를 학습하도록 시스템을 구성한다. 이렇게 학습된 롤 제어기의 성능을 시뮬레이션을 통하여 검증하고 PID 및 Dueling DDQN 기반 롤 제어기와 비교 분석한다.

2. 수중운동체의 운동방정식

수중운동체의 6자유도 운동에 대한 비선형 미분 방정식[10]으로부터 종방향 및 횡방향 운동방정식으로 선형화할 수 있으며 롤 제어 시스템의 설계를 위한 횡방향 선형 상태공간 모델식은 다음 식 (1)과 같다.

여기서, v, p, r, ϕ, ψ는 각각 sway velocity, rollrate, yawrate, roll, yaw를 가리키며, δ_ϕ, δ_r는 롤 제어 입력, 방향 제어 입력을 나타내며 롤 제어 입력은 우승강타(δ_{er})에 $0.5\delta_\phi$로 좌승강타(δ_{el})에 $-0.5\delta_\phi$로 할당되어 제어된다. 모델링된 수중운동체의 유체계수는 수조 및 해상 시험을 통하여 선정하였으며 Fig. 1은 수중운동체의 몸체고정 및 지구고정 좌표계이다.

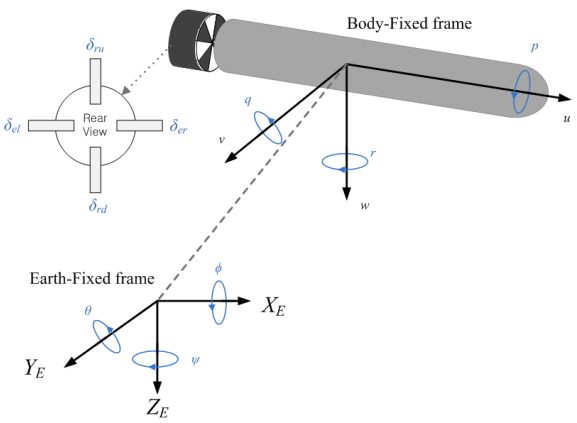


Fig. 1. Body-fixed and Earth-fixed coordinate of Underwater Vehicle

$$\begin{bmatrix} m-\frac{\rho AL Y_{\dot v}}{2} & -\frac{\rho AL^2 Y_{\dot p}}{2} & -\frac{\rho AL^2 Y_{\dot r}}{2} & 0 & 0 \\ -\frac{\rho AL^2 K_{\dot v}}{2} & I_x-\frac{\rho AL^3 K_{\dot p}}{2} & -\frac{\rho AL^3 K_{\dot r}}{2} & 0 & 0 \\ -\frac{\rho AL^2 N_{\dot v}}{2} & -\frac{\rho AL^3 N_{\dot p}}{2} & I_z-\frac{\rho AL^3 N_{\dot r}}{2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot v \\ \dot p \\ \dot r \\ \dot\phi \\ \dot\psi \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\rho AU Y_v}{2} & \frac{\rho ALU Y_p}{2} & -m u_0+\frac{\rho ALU Y_r}{2} & (W-B) & 0 \\ \frac{\rho ALU K_v}{2} & \frac{\rho AL^2 U K_p}{2} & \frac{\rho AL^2 U K_r}{2} & z_B B & 0 \\ \frac{\rho ALU N_v}{2} & \frac{\rho AL^2 U N_p}{2} & \frac{\rho AL^2 U N_r}{2} & -x_B B & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{2}\rho AU^2 Y_{\delta_r} & 0 \\ \frac{1}{2}\rho ALU^2 K_{\delta r} & \frac{1}{2}\rho ALU^2 K_{\delta e} \\ \frac{1}{2}\rho ALU^2 N_{\delta_r} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_r \\ \delta_\phi \end{bmatrix} \quad (1)$$

3. Advantage Actor-Critic 강화학습 기반 롤 제어 시스템 설계

3.1 Deep Q-Network 강화학습

강화학습은 기계학습(machine learning)의 한 영역으로,

Fig. 2에서와 같이 에이전트(agent)와 환경(environment)의 상호작용에 따라 관찰되는 상태(state), 행동(action) 및 보상(reward)을 효과적으로 활용하여 얻는 보상의 합을 최대화하는 정책을 학습하는 것이다. 가치기반 강화학습인 DQN은 큐러닝(Q-Learning)에 심층신경망(Deep Neural Network)을 함께 사용하는 학습방법으로 구조는 Fig. 3과 같다. 큐러닝은 다음의 순서 (a), (b)를 반복하며 Q함수를 업데이트하며 Q함수는 벨만 최적 방정식을 사용하며 업데이트 식은 아래 식 (2)과 같다.

(a) ϵ-탐욕 정책을 통해 샘플 $[S_t, A_t, R_t, S_{t+1}, A_{t+1}]$을 획득

(b) 획득한 샘플로 아래 식 (2)을 통해 Q함수를 업데이트

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1},a') - Q(S_t,A_t)) \quad (2)$$

Fig. 2. The agent-environment interaction process

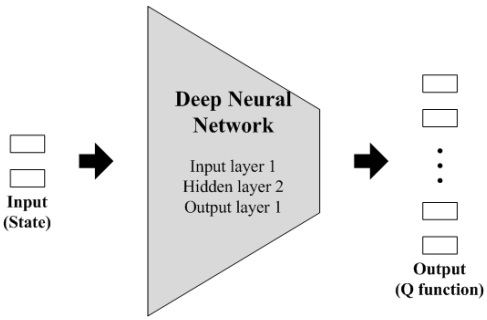


Fig. 3. Structure of Deep Q-Network

DQN은 Fig. 4와 같이 에이전트가 환경에서 탐험하며 얻은 샘플 <s, a, r, s'>을 경험 리플레이 메모리(experience replay memory)에 저장한다. 저장된 데이터는 에이전트의 학습에 쓰이며, 무작위로 여러 샘플을 뽑아 심층신경망을 업데이트한다. 에이전트는 매 타임스텝(time step)마다 메모리에서 샘플을 추출하여 미니배치(mini batch)로 학습에 사용하며 Q함수는 식 (2)를 이용하여 업데이트한다. 심층신경망을 업데이트할 때는 경사하강법을 사용하며 DQN 에이전트가 학습에 사용하는 오차함수(loss function)는 아래 식 (3)과 같다.

$$MSE = (\text{정답} - \text{예측})^2 = (R_{t+1} + \gamma \max_{a'} Q(s',a',\theta) - Q(s,a,\theta))^2 \quad (3)$$

식 (3)의 문제점은 업데이트 목표가 되는 정답과 학습되는 신경망이 동일하여 심층신경망의 학습이 업데이트 될 때마다 계속 변하는 것으로 이를 방지하기 위해서 타깃신경망을 따로 만들어서 타깃신경망에서 정답에 해당하는 값을 구하고 심층신경망을 계속 학습시키며 타깃신경망은 일정한 타임 스텝마다 학습된 심층신경망으로 업데이트한다. 타깃신경망을 이용한 DQN 오차함수는 식 (4)와 같으며 θ^-는 타깃신경망의 매개변수, 현재 상태 S_t, 탐욕 정책을 통해 선택된 A_t, θ는 심층신경망의 매개변수이다[4,9].

$$MSE = (\text{정답} - \text{예측})^2 = (R_{t+1} + \gamma \max_{a'} Q(S_{t+1},a',\theta^-) - Q(S_t,A_t,\theta))^2 \quad (4)$$

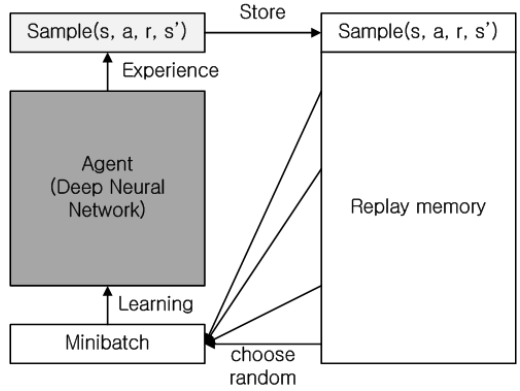


Fig. 4. Learning of Deep Q-Network using experience replay memory

DQN 알고리즘은 동작 정책을 정의하는 심층신경망($Q(s,a,\theta)$), DQN 오차함수에 대한 타깃 Q 값을 생성하는 데 사용되는 타깃 신경망($Q(s,a,\theta^-)$), 에이전트가 심층신경망 학습을 위해 무작위로 샘플링하는 데 사용하는 리플레이 메모리의 세 가지 주요 구성 요소로 Fig. 5와 같이 구성된다[11].

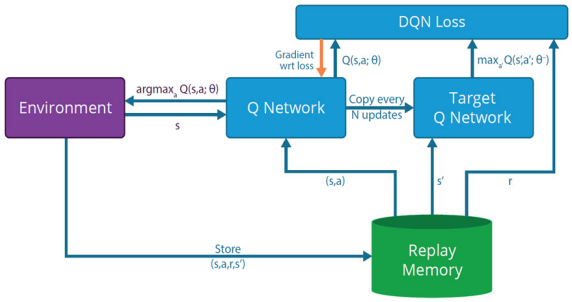


Fig. 5. Configuration of Deep Q-Network algorithm

식 (4)의 DQN 학습방법을 더욱 안정화시킨 DDQN의 수정식은 아래 식 (5)와 같다. 즉, 다음의 상태 S_{t+1}에서 Q값이 최대가 되는 행동 a'은 학습되는 심층신경망에서 구하고, 그때의 Q값은 타깃신경망에서 구하는 것이다.

$$\begin{aligned} a' &= \arg\max_a Q(S_{t+1},a,\theta) \\ Q(S_t,A_t) &\leftarrow Q(S_t,A_t)+\alpha(R_{t+1} \\ &\quad +\gamma Q(S_{t+1},a',\theta^-)-Q(S_t,A_t,\theta)) \end{aligned} \tag{5}$$

3.2 Dueling Double Deep Q-Network 강화학습

기존의 DQN 학습방법은 수중운동체가 어떤 행동을 취하던 받게 되는 할인총보상이 상태 s에 의해서만 결정되는 면이 있다. Dueling Q-Network[7]은 Q함수를 상태 s만으로 결정되는 부분 V(s)와 행동에 따라 결정되는 Advantage인 A(s,a)로 나눠서 학습한 다음 마지막 출력층에서 V(s)와 A(s,a)를 더해 Q(s,a)를 아래 식 (6)와 같이 계산하며 Dueling Q-Network의 구조는 Fig. 6과 같다.

$$Q(s,a)=A(s,a)+V(s) \tag{6}$$

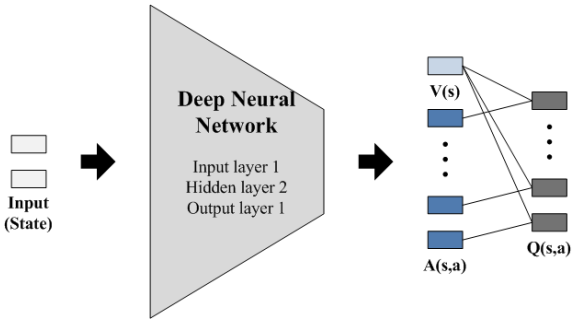


Fig. 6. Structure of Dueling Deep Q-Network

기존의 DQN과 비교했을 때 V(s)로 이어지는 결합가중치를 행동 a와 상관없이 매 단계마다 학습할 수 있어서 DQN보다 적은 수의 에피소드만으로 학습을 마칠 수 있다. 그러나 DDQN과 Dueling Q-Network을 결합한 Dueling DDQN을 적용한 가치함수 추정 방법은 이산 공간(discrete space)에 적합한 방식으로 연속공간에 적용하기에는 어려움이 있다.

3.3 Advantage Actor-Critic 강화학습

정책 기반 강화학습의 목표는 누적 보상을 최대화하는 것이며 정책을 근사하는 방법 중 하나로 정책신경망을 이용할 수 있다. 누적 보상은 정책신경망의 가중치에 따라 달라질 것이며 이는 최적화하고자 하는 목표함수 $J(\theta)$가 되며 식 (7)과 같이 나타낼 수 있다.

$$\theta_{t+1}=\theta_t+\alpha\nabla_\theta J(\theta) \tag{7}$$

Policy Gradient는 목표함수의 경사상승법을 따라서 근사된 정책을 업데이트하는 방식으로 Policy Gradient의 업데이트 식은 다음과 같이 나타낸다.

$$\theta_{t+1}\approx\theta_t+\alpha[\nabla_\theta \log\pi_\theta(a|s)\,Q_\pi(s,a)] \tag{8}$$

식 (8)에서 보듯이 Q함수 $Q_\pi(s,a)$의 근사가 필요하며 이를 가치신경망으로 추정하여 정책의 성과를 평가하는 방법이 Actor-Critic 강화학습이다. Actor-Critic에서 정책의 발전은 Actor 신경망의 업데이트로 정책의 평가는 Critic 신경망을 사용한다.

특히 $Q_\pi(s,a)$의 값에 따라 정책신경망의 학습에 영향을 많이 받으므로 목표함수 Gradient의 분산을 줄이기 위하여 Dueling Q-Network에서 설명한 상태가치함수 V(s)를 활용하여 베이스라인으로 사용한다. 식 (6)으로부터 Q함수를 가치함수를 사용하여 근사해서 정의한 advantage 함수는 식 (9)와 같다.

$$A(s_t,a_t)=R_{t+1}+\gamma V(s_{t+1})-V(s_t) \tag{9}$$

Policy Gradient 업데이트 식 (8)에서 $Q_\pi(s,a)$를 advantage 함수로 수정하여 사용한 Actor 신경망의 업데이트 식은 다음과 같다.

$$\theta_{t+1}\approx\theta_t+\alpha[\nabla_\theta \log\pi_\theta(a|s)A(s_t,a_t)] \tag{10}$$

이제 목표함수의 gradient는 advantage 함수에 영향을 받으며 Critic 신경망의 학습은 시간차 오차(Temporal Difference error, TD error)를 통해 진행되며 Critic 신경망의 업데이트를 위한 오차함수는 식 (11)과 같다.

$$Loss_{critic} = \frac{1}{2}(R_{t+1} + \gamma V(s_{t+1}) - V(s_t))^2 \quad (11)$$

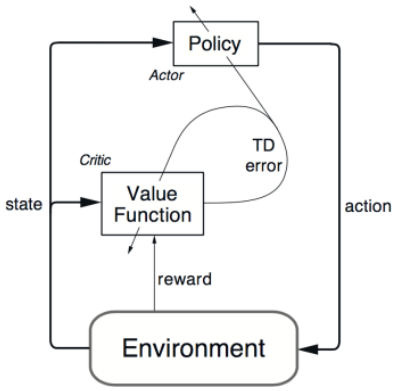
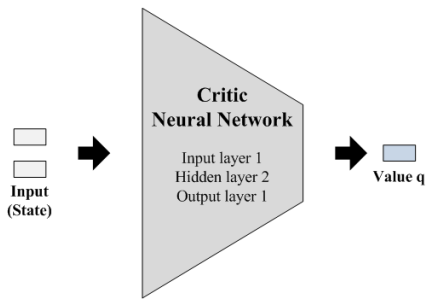
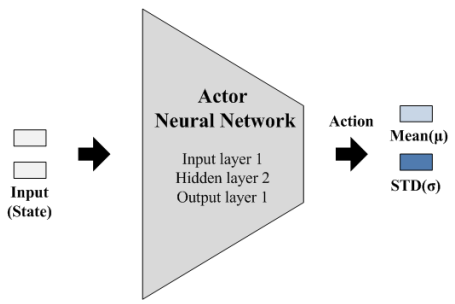


Fig. 7. Configuration of Advantage Actor-Critic algorithm



(a) Layout of the Critic network



(b) Layout of the Actor network

Fig. 8. Structure of Advantage Actor-Critic

이렇게 Actor-Critic이 advantage 함수를 사용하기 때문에 Advantage Actor-Critic(A2C)[8,9]라고 하며 A2C 강화학습의 주요 구성은 Fig. 7과 같다. Critic에서 생성된 시간차 오차를 기반으로 Actor 신경망을 조정하며 A2C의 Critic 및 Actor의 신경망 구조는 Fig. 8과 같으며 연속 공간(continuous space)에 적용 가능하도록 Actor 신경망의 출력은 행동의 평균(mean)과 표준편차(STD)로 나타낸다.

3.4 Dueling DDQN 및 A2C 기반 롤 제어시스템 설계

수중운동체의 PID 롤 제어기의 구조는 Fig. 9와 같으며 롤 제어 명령(ϕ_c)을 입력으로 하고 롤변화율(rollrate) 제어기를 내부 루프로 하는 다중 루프 제어기로 구성되어 있다. 롤 제어 명령은 항상 0을 유지하도록 롤 및 롤변화율 제어 게인을 설정한다[12,13].

DDQN 기반으로 학습된 심층신경망 롤 제어시스템의 구성은 Fig. 10과 같으며 Dueling DDQN 기반으로 학습된 심층신경망 롤 제어시스템의 구성은 Fig. 11과 같다. 본 논문에서 제안한 A2C 기반으로 학습된 Actor 신경망 롤 제어시스템의 구성은 Fig. 12와 같다. DDQN 및 Dueling DDQN 기반 제어시스템은 동일한 심층신경망으로 구성되어 있으나 학습 방법에서 차이가 발생하며 A2C 기반 제어시스템은 학습 방법의 차이 및 연속 공간에 적용 가능한 출력 형태로 인하여 가치기반 심층신경망과 다르게 신경망이 구성된다.

DDQN 및 Dueling DDQN 심층신경망의 구성은 입력층의 상태 노드가 수중운동체의 롤과 롤변화율로 2개이고 은닉층은 총 2개이며 각각 100개의 노드수를 가지고 있으며 출력층의 노드는 수중운동체의 롤 제어 입력이며 ±20°를 0.1°간격으로 나눈 이산 공간 출력이다. 은닉층의 활성화 함수는 ReLU함수이고 출력층의 활성화함수는 선형 함수이며 learning rate는 0.003, 경사하강법으로는 adam optimizer를 사용하였으며 경험 리플레이 메모리 버퍼(buffer) 크기는 5000이며 미니배치의 크기는 64이다.

Actor-Critic 신경망의 구성 크게 Actor 신경망과 Critic 신경망으로 나누어진다. 먼저 Critic 신경망의 입력층 상태 노드는 수중운동체의 롤과 롤변화율로 2개이고 은닉층은 총 2개이며 각각 100개의 노드수를 가지고 있으며 출력은 추정된 상태가치이다. 은닉층의 활성화 함수는 ReLU 함수이고 출력층의 활성화함수는 선형 함수이며 learning rate는 0.0001, 경사하강법으

로는 adam optimizer를 사용한다. 다음으로 Actor 신경망의 입력층 상태 노드는 수중운동체의 롤과 롤변화율로 2개이고 은닉층은 총 2개이며 각각 50개의 노드수를 가지고 있으며 출력은 수중운동체의 롤 제어 입력의 평균과 표준편차이며 이를 바탕으로 정규분포 확률로 롤 제어 입력을 결정한다. 은닉층의 활성화 함수는 tanh 함수이고 출력층의 활성화함수는 tanh, sigmoid 함수이며 learning rate는 0.005, 경사하강법으로는 adam optimizer를 사용한다.

학습을 위한 보상식은 식 (12)와 같으며 보상식의 계수는 시뮬레이션을 통한 시행착오로 결정하였다. 롤(ϕ)이나 롤변화율(p)이 제한 범위(±10°)를 벗어나면 에피소드가 종료되고 보상이 주어지며 일정 시간이 경과하면 에피소드가 자동으로 종료되고 롤이 ±0.5° 이내일 때 추가 보상으로 10을 받도록 설계하였다.

$$r = -\frac{1}{2}\phi^2 - 0.1p^2 \tag{12}$$

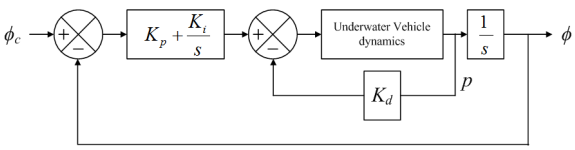


Fig. 9. Configuration of PID roll controller of Underwater Vehicle

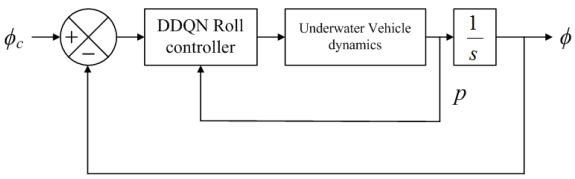


Fig. 10. Configuration of DDQN roll controller of Underwater Vehicle

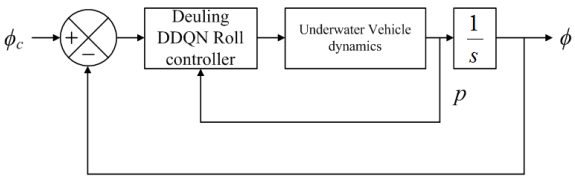


Fig. 11. Configuration of Dueling DDQN roll controller of Underwater Vehicle

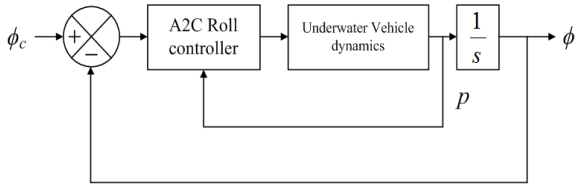


Fig. 12. Configuration of A2C roll controller of Underwater Vehicle

4. 강화학습 기반 수중운동체의 롤 제어 시뮬레이션

4.1 Dueling DDQN 및 A2C 기반 롤 제어기 학습

본 논문에서 제안한 강화학습 기반 수중운동체의 롤 제어 성능을 검증하기 위해서 DDQN, Dueling DDQN 및 본 논문에서 제안한 A2C 강화학습 기반의 심층신경망 롤 제어기를 학습시키도록 한다. 본 제어의 목적은 수중운동체가 주행 중 롤을 0으로 유지하는 것으로 심층신경망을 학습하는 것이다. 다양한 롤 오차의 영향을 학습시키기 위하여 롤 초기값을 ±5°사이의 값으로 랜덤하게 주어진다. 또한 심층신경망이 목표를 이루기 위해서 500 타임 스텝을 유지하지 못하고 에피소드가 종료되면, 즉 롤 및 롤변화율이 제한범위를 벗어나면 식 (12)와 같은 보상값을 주어 더 나은 학습을 유도하도록 구성하였다.

DDQN과 Dueling DDQN 기반의 롤 제어기의 학습 과정을 그린 그래프는 Fig. 13과 같다. 가로축은 에피소드이고 세로축은 각 에피소드에서 받은 총 보상을 나타내고 있다. 최근 10개의 에피소드가 연속적으로 보상값의 크기가 0 이상일 경우 학습이 완료된 것으로 판단하였으며 DDQN의 경우 약 127 에피소드 정도가 지난 후 학습이 완료되었으며 Dueling DDQN의 경우 약 48 에피소드 정도가 지난 후 학습이 완료되는 것을 확인 할 수 있다. DDQN 기반으로 학습했을 때와 비교하여 Dueling DDQN 기반으로 학습했을 경우 상태와 행동을 나누어 학습함으로써 보다 빠른 학습이 가능한 것을 확인 할 수 있다.

A2C 기반의 롤 제어기의 학습 과정을 그린 그래프는 Fig. 14와 같다. 최근 10개의 에피소드가 연속적으로 보상값의 크기가 0 이상일 경우 학습이 완료된 것으로 판단하였으며 A2C는 약 977 에피소드가 지난 후 학습이 완료되었다. 가치 기반의 Dueling DDQN과

달리 A2C는 Critic 신경망과 Actor 신경망을 동시에 학습시켜야 되므로 학습 시간이 Dueling DDQN 비해서 오래 걸리는 것을 확인할 수 있다.

4.2 Dueling DDQN 및 A2C 기반 롤 제어 시뮬레이션 결과

본 논문에서 제안된 A2C 기반 롤 제어기의 학습이 완료된 후 성능을 검증하기 위하여 심도 제어와 경로 제어는 설계된 PID 제어기를 이용하여 수중운동체가 40m의 심도에서 일정한 경로와 속도로 주행되도록 하였다. 이 때 롤 제어가 정상적으로 수행되는지를 시뮬레이션을 통하여 검증하고 PID 롤 제어기와 Dueling DDQN 기반 롤 제어기의 성능을 같이 비교 분석하였다. 여기서 DDQN과 Dueling DDQN의 제어 성능에는 큰 차이가 없어 Dueling DDQN 결과를 기준으로 비교 분석하기로 한다.

먼저 주행 초기 롤이 0°인 상태에서 주행하였을 때 롤 제어가 정상적으로 이루어지는지 시뮬레이션을 통하여 확인하였고 그 결과가 Fig. 15와 같다. Fig. 15에서 보듯이 PID 롤 제어기는 초기에 롤 오차가 최대 1.2° 발생하였으나 Dueling DDQN 및 A2C 기반 롤 제어기의 롤 오차는 0.2°로 차이가 나는 것을 확인할 수 있다. 그 원인으로는 Fig. 16에서 보듯이 수중운동체의 초기 주행에서 추진기 작동에 따른 속도 안정화 과정에서 과도 상태가 발생하고 있다. 이런 초기 과도 상태로 인하여 초기에 PID 롤 제어 성능은 다소 불안정 하나 Dueling DDQN 및 A2C 기반 롤 제어기는 초기 과도 상태를 포함하여 학습이 진행되어 과도 상태에서도 안정적인 제어 성능을 나타내고 있다. 초기 과도 상태에서의 성능은 차이가 나지만 정상 상태에서는 PID 롤 제어기, Dueling DDQN 및 A2C 기반 롤 제어기 모두 오차가 0°로 수렴하는 것을 확인 할 수 있다. Fig. 17의 수중운동체의 롤변화율 주행 결과를 살펴보면 PID 롤 제어기는 초기 과도 상태에서는 거동이 크게 발생하나 정상상태에서는 안정적이며 Dueling DDQN 기반 롤 제어기는 롤 제어 입력이 이산 제어(discrete control)가 되어 ±1° 정도의 떨림이 발생하여 불안정하였으나 A2C 기반 롤 제어기는 과도 상태 및 정상 상태 모두 안정적인 것을 확인 할 수 있으며 이는 Dueling DDQN 기반 롤 제어기와 달리 롤 제어 입력을 연속 제어(continuous control)로 설계하여 학습하였기 때문이다.

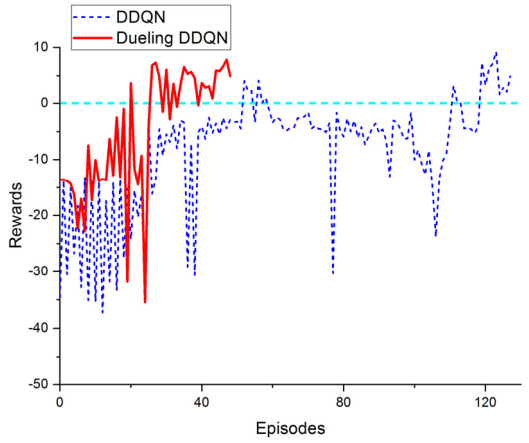


Fig. 13. The learning results of DDQN and Dueling DDQN Roll controller

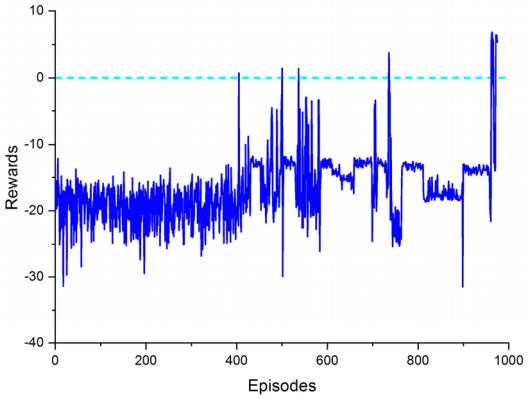


Fig. 14. The learning results of A2C Roll controller

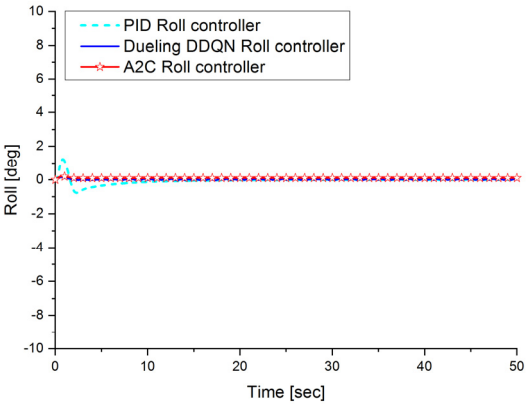


Fig. 15. The simulation results of PID, Dueling DDQN and A2C roll control(roll error : 0°)

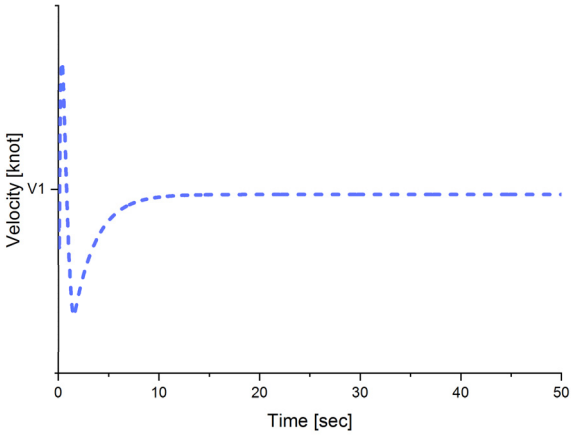


Fig. 16. The velocity trajectory result of Underwater Vehicle

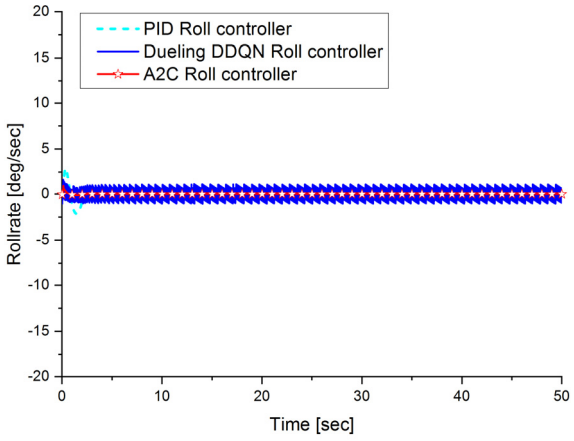


Fig. 17. The rollrate trajectory results of PID, Deuling DDQN and A2C roll controller(roll error : 0°)

다음으로 주행 초기 롤이 +5°인 상태에서 주행하였을 때 롤 제어가 정상적으로 이루어지는지 시뮬레이션을 통하여 확인하였고 그 결과가 Fig. 18과 같다. Fig. 18에서 보듯이 PID 롤 제어기는 초기 과도 상태에서 -0.94°의 오버슈트가 발생하였고 정상상태 오차가 0°로 수렴하였으나 Dueling DDQN 및 A2C 기반 롤 제어기는 과도 상태에서 오버슈트가 발생하지 않고 정상상태 오차가 0°으로 수렴하는 것을 확인할 수 있으며 이는 초기 과도 상태를 포함하여 학습하였기 때문이다. Fig. 19의 수중운동체의 롤변화율 주행 결과를 살펴보면 PID 롤 제어기와 Dueling DDQN 및 A2C 기반 롤 제어기 모두 비슷한 경향을 나타내고 있으나 Dueling DDQN 기반 롤 제어기의 경우 앞에서 설명한 롤 제어 입력이 이산 형태이므로 PID 및 A2C 기반 롤 제어기보다 상대적으로 롤변화율이 크게 떨리고 불안정하다.

마지막으로 주행 초기 롤이 -5°인 상태에서 주행하였을 때 롤 제어가 정상적으로 이루어지는지 시뮬레이션을 통하여 확인하였고 그 결과가 Fig. 20과 같다. Fig. 20에서 보듯이 PID 롤 제어기는 초기 과도 상태에서 1.44°의 오버슈트가 발생하였고 정상상태 오차는 0°로 정상적으로 수렴하였으나 Dueling DDQN 및 A2C 기반 롤 제어기는 과도 상태에서 롤 위치 오차가 +5°와 동일하게 오버슈트가 발생하지 않으면서 정상상태 오차가 0°으로 잘 수렴하는 것을 확인할 수 있다. 이는 앞에서 설명한 바와 같이 초기 과도 상태를 포함하여 학습하였기 때문이다. Dueling DDQN보다 A2C 기반 롤 제어 과도 성능이 약간 개선된 것을 확인할 수 있으며 롤 위치 오차가 ±5°일 때의 과도 상태 제어 성능의 차이가 발생하는 원인으로는 추진기 초기 기동에 따른 불균형 회전에 의한 영향과 수중운동체 무게 중심의 오프셋(offset)이 약간 존재하기 때문이다. Fig. 21의 수중운동체의 롤변화율 주행 결과에서 A2C 기반 롤 제어기는 초기 과도 상태의 변화가 PID 및 Dueling DDQN 기반 롤 제어기보다 크게 나타나고 있으나 빠르게 정상상태에 도달하여 안정화되고 있으나 Dueling DDQN 기반 롤 제어기는 PID 및 A2C 기반 롤 제어기와 달리 정상 상태에서 ±1° 정도의 떨림이 발생하여 불안정하다.

Fig. 22는 롤 위치 오차에 따른 PID 롤 제어 입력의 결과를 나타내며 Fig. 23은 롤 위치 오차에 따른 A2C 기반 롤 제어 입력의 결과를 나타낸다. Fig. 22와 Fig. 23에서 보듯이 A2C 기반 롤 제어 입력이 연속 공간으로 설계되어 PID 롤 제어 입력과 비교하여도 떨림 없이 안정적인 롤 제어 입력을 나타내는 것을 확인할 수 있다.

Fig. 15에서 Fig. 21의 시뮬레이션 결과에서 보듯이 정상상태에서의 롤 제어 성능은 본 논문에서 제안한 A2C 기반 롤 제어기와 PID 롤 제어기 모두 안정적인 성능을 나타내었으나 DDQN 기반 롤 제어기는 롤변화율에서 불안정한 성능을 나타내는 것을 확인할 수 있었다. 또한 초기 과도 상태에서는 PID 롤 제어기의 제어 성능이 다소 불안정해졌는데 이는 PID 제어 게인을 선정할 때 일반적으로 제한된 속도(V1)에 대하여 선형화 하고 설계를 진행하므로 과도 상태에서는 발생할 수밖에 없는 문제점이었으나 A2C 기반 롤 제

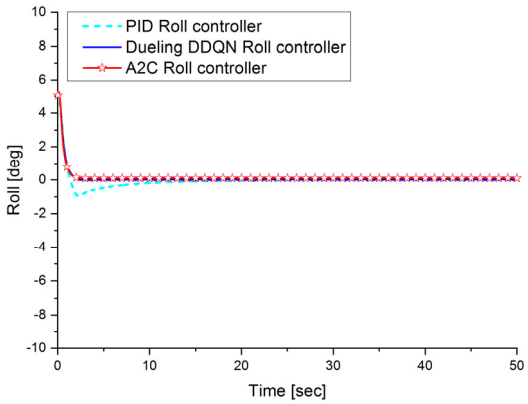


Fig. 18. The simulation results of PID, Dueling DDQN and A2C roll control(roll error : +5°)

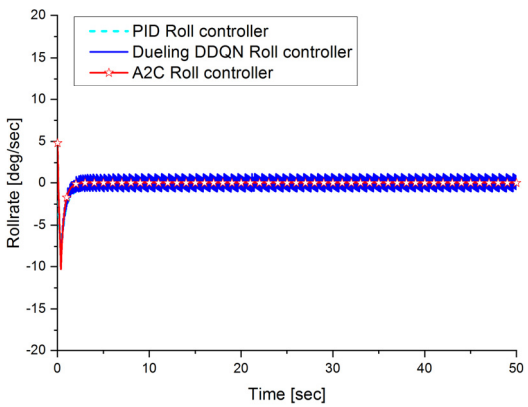


Fig. 19. The rollrate trajectory results of PID, Deuling DDQN and A2C roll controller(roll error : +5°)

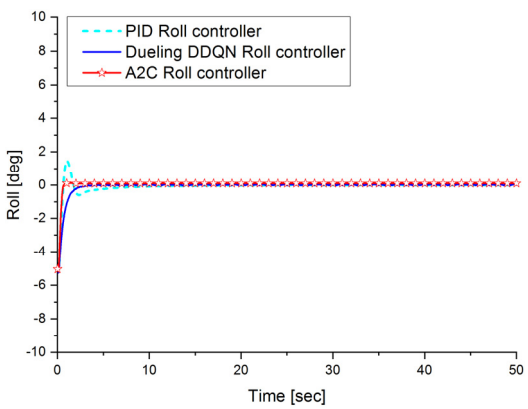


Fig. 20. The simulation results of PID, Deuling DDQN and A2C roll control(roll error : −5°)

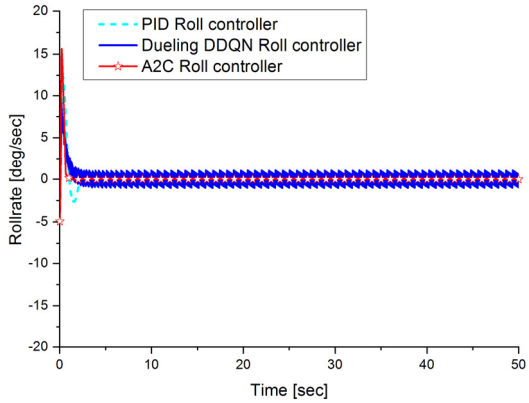


Fig. 21. The rollrate trajectory results of PID, Dueling DDQN and A2C roll controller(roll error : −5°)

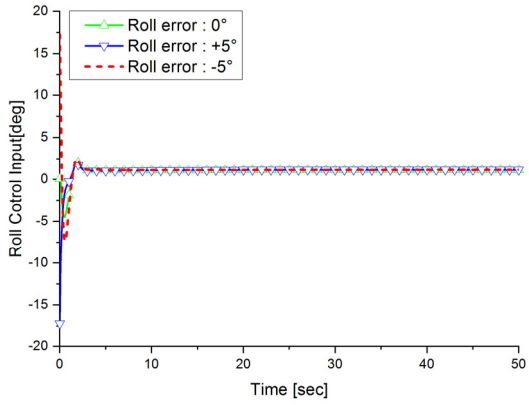


Fig. 22. The control input δ_ϕ results of PID roll controller

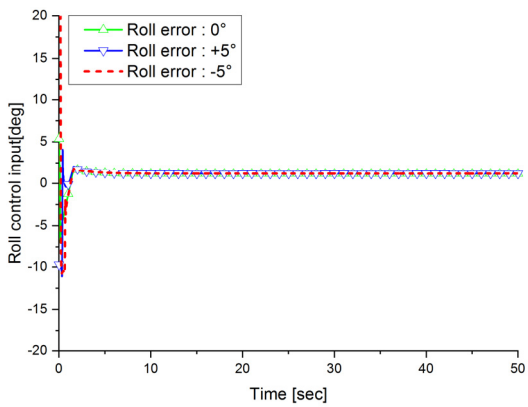


Fig. 23. The control input δ_ϕ results of A2C roll controller

어기는 초기 과도 상태부터 정상 상태까지 학습함으로써 이를 극복하고 초기 과도 상태 및 정상 상태 둘 모두 안정적인 롤 제어 성능을 나타내는 것을 확인할 수 있었다.

5. 결 론

수중운동체의 운동 모델 선형화를 통한 기존 제어기 설계 단점을 극복하기 위하여 본 논문에서는 행동에 대한 보상을 통해 학습 할 수 있는 강화학습 방법 중에서 연속공간에서 안정된 신경망 학습이 가능한 A2C 기반의 수중운동체 롤 제어기를 제안하였다. 제안된 A2C 기반 롤 제어기의 성능은 시뮬레이션을 통하여 검증하였고 PID 및 Dueling DDQN 기반 롤 제어기와 비교 분석하였다. 본 논문에서 제안된 A2C 기반의 수중운동체 롤 제어기는 초기 과도 상태 및 정상 상태의 학습을 통하여 PID 및 Dueling DDQN 기반 롤 제어기와 비교하여 개선된 제어 성능을 나타내는 것을 확인할 수 있었다. 이를 바탕으로 향후 다양한 제어 시스템에 적용한다면 수중운동체의 제어 성능은 개선될 수 있을 것이다.

References

[1] Jongho Shin, and Sanghyun Joo, “NN-based Adaptive Control for a Skid-Type Autonomous Unmanned Ground Vehicle,” Journal of Institute of Control, Robotics and Systems, Vol. 20, No. 12, pp. 1278-1283, 2014.

[2] S. Y. Kim, et. al., “Neural Network for a Roll Control of the Underwater Vehicle,” KIMST Annual Conference Proceedings, pp. 14-15, 2018.

[3] H.-J. Chae, et. al., “Time-varying Proportional Navigation Guidance using Deep Reinforcement Learning,” Journal of the Korea Institute of Military Science and Technology, Vol, 23, No. 4, pp. 399-406, 2020.

[4] S. Y. Kim, et. al., “Reinforcement Learning for a Roll Control of the Unmanned Underwater Vehicle,” Naval Ship Technology & Weapon Systems Seminar Proceedings, pp. 474-477, 2019.

[5] Volodymyr Mnih, et. al., “Playing Atari with Deep Reinforcement Learning,” In NIPS Deep Learning Workshop, 2013.

[6] Hado van Hasselt, et. al., “Deep Reinforcement Learning with Double Q-learning,” AAAI, Vol. 16, 2016.

[7] Ziyu Wang, et. al., “Dueling Network Architectures for Deep Reinforcement Learning,” Proceedings of The 33rd International Conference on Machine Learning, 2016.

[8] R. S. Sutton, and A. G. Barto, “Reinforcement Learning: An Introduction,” The MIT Press, pp. 328-333, 2018.

[9] W. W. Lee, et. al., “Reinforcement Learning with Python and Keras,” Wikibook, pp. 225-277, 2020.

[10] H. J. Cho, et. al., “A Two-Stage Initial Alignment Technique for Underwater Vehicles Dropped from a Mother Ship,” International Journal of Precision Engineering and Manufacturing, Vol. 14, No. 12, pp. 2067-2073, 2013.

[11] Arun Nair, et. al., “Massively Parallel Methods for Deep Reinforcement Learning,” In ICML Deep Learning Workshop, 2015.

[12] S. Y. Kim, et. al., “Robust Depth Control for an Autonomous Navigation of the Underwater Vehicle,” KIMST Annual Conference Proceedings, pp. 14-15, 2013.

[13] Benjamin C. Kuo, “Automatic Control Systems,” Prentice Hall, 1994.