Journal of the KIMST, Vol. 24, No. 5, pp. 558-568, 2021
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2021.24.5.558

Research Paper 제어·구동 부문

수중운동체의 롤 제어를 위한 Deep Deterministic Policy Gradient 기반 강화학습

김수용*,1) · 황연걸1) · 문성웅1)

1) 국방과학연구소 해양기술연구원

Reinforcement Learning based on Deep Deterministic Policy Gradient for Roll Control of Underwater Vehicle

Su Yong Kim*,1) · Yeon Geol Hwang1) · Sung Woong Moon1)

1) *Maritime Technology Research Institute, Agency for Defense Development, Korea*

(Received 29 March 2021 / Revised 17 May 2021 / Accepted 16 July 2021)

Abstract

The existing underwater vehicle controller design is applied by linearizing the nonlinear dynamics model to a specific motion section. Since the linear controller has unstable control performance in a transient state, various studies have been conducted to overcome this problem. Recently, there have been studies to improve the control performance in the transient state by using reinforcement learning. Reinforcement learning can be largely divided into value-based reinforcement learning and policy-based reinforcement learning. In this paper, we propose the roll controller of underwater vehicle based on Deep Deterministic Policy Gradient(DDPG) that learns the control policy and can show stable control performance in various situations and environments. The performance of the proposed DDPG based roll controller was verified through simulation and compared with the existing PID and DQN with Normalized Advantage Functions based roll controllers.

Key Words : Underwater Vehicle(수중운동체), Roll Control(롤 제어), Actor-Critic(행동자-비평자), Deep Deterministic Policy Gradient(심층 확정적 정책 그래디언트)

1. 서 론

최근 주요 국가에서 기뢰 위협에 대응하기 위하여 무인 수중운동체를 활용한 기뢰 탐색이나 기뢰 소해 등의 기뢰대항 능력 향상에 대한 연구가 활발히 진행되고 있다[1]. 이러한 다양한 임무를 수행하기 위하여 기본적으로 수중운동체는 자세를 측정 및 제어되어야 한다. 수중운동체의 자세 측정은 관성항법장치를 주로 사용하며 수중운동체의 심도(depth), 경로(course), 롤

* Corresponding author, E-mail: rocksy@add.re.kr

(roll) 3축 자세 제어는 자동조종장치(autopilot)에서 수행한다. 3축의 자세 제어를 위한 기존 제어기의 설계 방법은 수중운동체의 운동 모델을 특정 운동 구간으로 한정하여 선형화하기 때문에 정상 상태의 제어 성능은 안정적이나 과도 상태의 제어 성능은 불안정한 문제가 있다.

기존 선형 제어기의 단점을 개선하기 위하여 강인 제어[2] 및 적응제어[3]와 같은 다양한 연구가 진행되어 왔으며 특히 최근에는 인공지능을 이용한 연구가 활발히 진행되고 있다[4-6]. 최근 딥마인드의 알파고를 통해 주목받고 있는 강화학습은 자가 학습을 통해 문제를 해결할 수 있으며 강화학습이 문제를 해결하기 위한 방법에는 가치함수를 추정하는 방법과 직접 정책을 추정하는 방법으로 나눌 수 있다. 가치함수를 추정하여 최대의 보상을 계산하는 대표적인 강화학습으로 DQN(Deep Q-Network)[7-9]이 있으며 이를 기반으로 기존 제어기의 정상 상태 및 과도 상태의 제어 성능을 개선할 수 있는 방법이 제안되었다[10,11]. 가치함수를 추정하는 방법은 환경과의 상호작용을 통해 주어진 정책에 대한 가치함수를 학습할 수 있으나 연속 행동 공간(continuous action space)에서 적용하기 어렵고 다양한 경험이 필요하므로 경험하지 못한 상태 및 환경에 대한 결과는 정확히 예측하기 힘들다.

직접 정책을 유도하는 강화학습 방법 중에서 on-policy로 연속 행동 공간(continuous action space)에서 적용 가능한 Actor-Critic 방법[11-13]이 제안되었으나 학습에 사용되는 샘플 데이터가 시간적으로 상관되어 있어 학습 성능이 불안정하며 신경망의 출력을 특정 구조를 갖도록 제한하게 된다. 그러므로 본 논문에서는 직접 정책을 유도하는 강화학습 방법 중에서 off-policy로 안정된 신경망 학습이 가능한 Deep Deterministic Policy Gradient(DDPG)[14,15] 기반 수중운동체의 롤 제어기를 제안하고 Critic 신경망에서 추정된 행동가치함수를 이용하여 Actor 신경망이 최적의 롤 제어 정책을 찾을 수 있도록 시스템을 구성하고 선형 제어기의 단점을 개선하기 위하여 비선형 운동방정식을 이용하여 제어 정책 학습을 수행한다. 학습된 DDPG 기반 수중운동체의 롤 제어기의 성능 검증을 위하여 PID 및 가치기반 강화학습 중에서 연속 행동 공간에 적용 가능한 DQN with Normalized Advantage Functions[16] 기반의 롤 제어기와 동일한 조건에서 시뮬레이션을 수행하고 각각의 성능을 비교 분석한다.

2. 수중운동체의 운동방정식

수중운동체의 6자유도 운동에 대한 비선형 운동방정식은 식 (1)과 같이 나타낼 수 있다.

$$
\begin{aligned}
&m(\dot{u}+qw-rv)=X_H+X_P+X_R+X_\delta+X_S\\
&m(\dot{v}-pw+ru)=Y_H+Y_\delta+Y_S\\
&m(\dot{w}+pv-qu)=Z_H+Z_\delta+Z_S\\
&I_x\dot{p}=K_H+K_P+K_\delta+K_S\\
&I_y\dot{q}=M_H+M_\delta+M_S\\
&I_z\dot{r}=N_H+N_\delta+N_S
\end{aligned}
\tag{1}
$$

여기서 m, I_x, I_y, I_z, u, v, w, p, q, r, $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, $\dot{r}$은 수중운동체의 질량, 몸체고정 좌표계에서의 관성모멘트, 전후, 수평, 수직 방향에 대한 속도, roll rate, pitch rate, yaw rate 및 각각의 가속도를 나타낸다. X, Y, Z, K, M, N는 수중운동체가 외부에서 받는 힘과 모멘트이며 하첨자 H, P, R, δ, S는 동유체력, 추진력, 저항, 제어력, 정역학적 힘과 모멘트이다. 식 (1)로부터 종방향 및 횡방향 운동방정식으로 선형화할 수 있으며 롤 제어 시스템의 설계를 위한 횡방향 선형 상태공간 모델식은 다음 식 (2)와 같다[5,11,17].

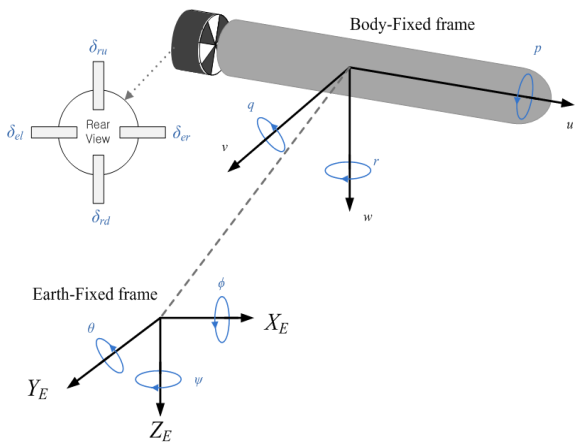


Fig. 1. Body-fixed and earth-fixed coordinate of underwater vehicle

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

$$\begin{bmatrix} \frac{\rho AUY_v}{2} & \frac{\rho ALUY_p}{2} & \frac{\rho ALUY_r}{2} & (W-B) & 0 \\ \frac{\rho ALUK_v}{2} & \frac{\rho AL^2 UK_p}{2} & \frac{\rho AL^2 UK_r}{2} & z_B B & 0 \\ \frac{\rho ALUN_v}{2} & \frac{\rho AL^2 UN_p}{2} & \frac{\rho AL^2 UN_r}{2} & -x_B B & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix}$$

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

여기서, ρ, A, L, W, B, U, x_B, z_B, v, p, r, ϕ, ψ는 각각 물의 밀도, 수중운동체의 단면적, 길이, 중량, 부력 ,속도, 무게중심과 부력중심의 x, z방향 편차, sway velocity, roll rate, yaw rate, roll, yaw를 가리키며, δ_ϕ, δ_r는 롤 제어 입력, 방향 제어 입력을 나타내며 롤 제어 입력은 우승강타(δ_{er}) 및 좌승강타(δ_{el})에 $0.5\delta_\phi$, $-0.5\delta_\phi$로 할당되어 제어된다. 모델링된 수중운동체의 유체계수는 수조 및 해상 시험을 통하여 선정하였으며 Fig. 1은 수중운동체의 몸체고정 및 지구고정 좌표계이다.

3. Deep Deterministic Poilcy Gradient 기반 강화학습

3.1 Dueling Deep Q-Network 강화학습[10,11]

강화학습은 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. DQN은 행동가치 함수를 추정한 후에 각 상태에서 행동가치 함수를 최대화하는 행동을 선택해 최적 정책을 찾아내는 대표적인 가치 기반 강화학습으로 큐러닝(Q-Learning)에 가치신경망(Value Neural Network)을 함께 사용하며 Q함수의 업데이트 식은 아래 식 (3)과 같다.

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1},a') - Q(S_t,A_t)) \quad (3)$$

DQN 알고리즘은 동작 정책을 정의하는 신경망($Q(s,a,\theta)$), DQN 오차함수에 대한 타깃 Q 값을 생성하는데 사용되는 타깃 신경망($Q(s,a,\theta^-)$), 에이전트가 신경망 학습을 위해 무작위로 샘플링하는데 사용하는 리플레이 메모리의 세 가지 주요 구성 요소로 Fig. 2와 같이 구성된다[18].

DQN은 매 타임 스텝마다 메모리에서 샘플을 배치로 추출하여 학습에 사용하며 타깃신경망에서 정답에 해당하는 값을 구하고 심층신경망을 계속 학습시키며 타깃신경망은 일정한 시간 간격마다 학습된 신경망으로 업데이트한다. DQN 학습에 사용하는 오차함수(loss function)는 아래 식 (4)와 같으며 θ^-는 타깃신경망의 매개변수, θ는 신경망의 매개변수이다[13].

$$Loss_{DQN} = \frac{1}{2}(R_{t+1} + \gamma \max_{a'} Q(S_{t+1},a',\theta^-) - Q(S_t,A_t,\theta))^2 \quad (4)$$

Dueling Q-Network[9]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 (5)와 같이 계산하며 Dueling Q-Network을 결합한 Dueling DQN의 구조는 Fig. 3과 같다.

$$Q(s,a) = A(s,a) + V(s) \quad (5)$$

기존의 DQN과 비교했을 때 V(s)로 이어지는 결합가중치를 행동 a와 상관없이 매 단계마다 학습할 수 있어서 DQN보다 적은 수의 에피소드 만으로 학습을 마칠 수 있다. 그러나 Dueling DQN을 적용한 가치함수 추정 방법은 이산 행동 공간(discrete action space)에 적합한 방식으로 연속 행동 공간에 적용하기에는 어려움이 있다.

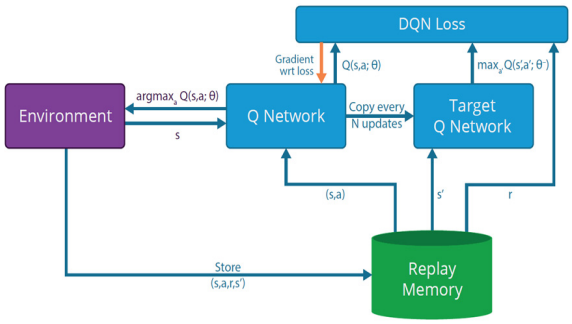


Fig. 2. Configuration of deep Q-Network algorithm[18]

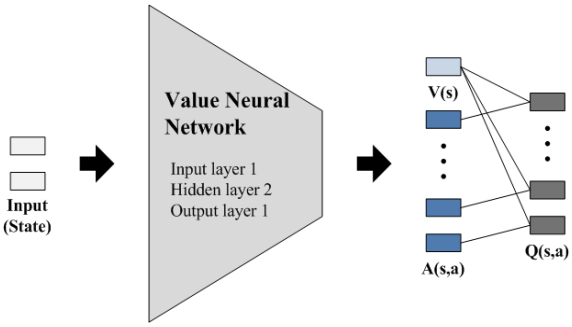


Fig. 3. Structure of dueling deep Q-Network

3.2 Deep Q-Network with Normalized Advantage Functions 기반 강화학습

가치함수를 추정하는 강화학습 방법은 이산 행동 공간에 적합한 방식이나 연속 행동 공간에서 큐러닝에 심층 신경망을 적용 가능하게 하는 DQN with Normalized Advantage Functions(NAF)[16]이 제안되었다. NAF의 기본 개념은 큐러닝에 가치 신경망을 함께 적용하여 Q함수를 업데이트 하며 Dueling Q-Network[9]와 같이 상태 s만으로 결정되는 가치 함수 $V(s|\theta^V)$와 행동에 따라 결정되는 Advantage $A(s,a|\theta^A)$를 별도로 출력하는 신경망을 기반으로 하며 Q함수는 식 (6)과 같다.

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

Advantage $A(s,a|\theta^A)$는 상태의 비선형 2차 함수로 식 (7)과 같이 나타낼 수 있다.

$$A(s,a|\theta^A) = -\frac{1}{2}(a-\mu(s|\theta^\mu))^T P(s|\theta^P)(a-\mu(s|\theta^\mu)) \tag{7}$$

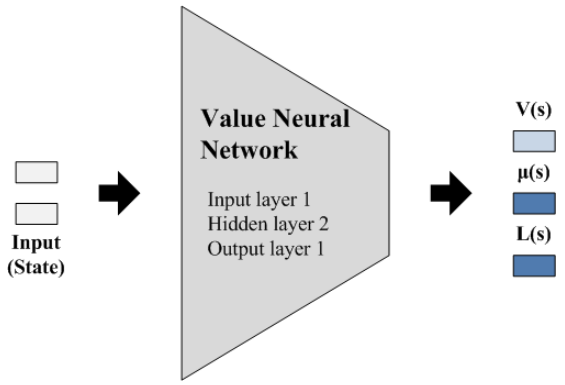


Fig. 4. Structure of deep Q-Network with normalized advantage functions

$P(s|\theta^P)$는 state-dependent, positive-definite square matrix로, $P(s|\theta^P)$는 $L(s|\theta^P)L(s|\theta^P)^T$이다. 여기서 $L(s|\theta^P)$는 대각 항이 지수화 된 신경망의 선형 출력 계층 항목이 나오는 lower-triangular matrix 이다. 일반적인 신경망 함수보다 더 제한적이지만 Q함수는 a에서 2차이므로 Q함수를 최대화하는 동작은 항상 $\mu(s|\theta^\mu)$로 제공되며 DQN에서는 ϵ-탐욕 정책을 통해 샘플을 획득 하였으나 NAF에서는 식 (8)로부터 행동을 결정하며 N은 가우시안 화이트 노이즈(Gaussian white noise)이다.

$$a_t = \mu(s_t|\theta^\mu) + N_t \tag{8}$$

또한 NAF는 앞에서 설명 된 DQN의 리플레이 메모리와 타깃신경망를 사용하며 매 타임 스텝마다 메모리에서 샘플을 배치로 추출하여 식 (9)의 오차함수를 최소화 하도록 신경망을 업데이트하며 타깃신경망 $(\theta^{Q'})$은 학습된 신경망(θ^Q)으로 갑자기 변경되지 않도록 식 (10)과 같이 업데이트한다. DQN with NAF의 구조는 Fig. 4와 같다.

$$Loss_{NAF} = \frac{1}{2}(r_t + \gamma V'(s_{t+1}|\theta^{V'}) - Q(s_t,a_t|\theta^Q))^2 \tag{9}$$

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \tag{10}$$

3.3 Deep Deterministic Policy Gradient 강화학습

강화학습의 최종 목표는 환경으로부터 받는 누적

보상을 최대로 하는 최적 정책을 찾는 것으로 정책 기반 강화학습에서 정책을 근사하는 방법 중 하나로 정책신경망을 적용 할 수 있다. 누적 보상은 정책신경망의 가중치에 따라 달라질 것이며 이는 최적화하고자 하는 목표함수 $J(\theta)$가 된다. $J(\theta)$의 최적화는 경사상승법에 따라 정책신경망을 업데이트하는 것으로 식 (11)과 같이 나타낼 수 있다.

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta) \tag{11}$$

연속 행동 공간에서 탐욕스러운 정책을 찾기 위해서는 매 시간마다 행동의 최적화가 필요하기 때문에 가치 기반 강화학습을 적용하는 것에는 어려움이 있다. 연속 행동 공간에서 목표함수의 경사상승법을 따라서 근사된 정책을 업데이트할 수 있는 방법으로 Deterministic Policy Gradient(DPG)[14]가 제안되었으며 최종적인 목표함수의 Policy Gradient는 식 (12)와 같이 나타낸다.

$$\nabla_\theta J(\theta) \approx E_{s_t \sim \rho^\beta}\left[\nabla_a Q(s,a|\theta^Q) \nabla_{\theta^\mu} \mu(s|\theta^\mu)\right] \tag{12}$$

식 (12)에서 $\nabla_a Q(s,a|\theta^Q)$는 행동 a에 대한 행동 가치 함수의 기울기이며 $\nabla_{\theta^\mu}\mu(s|\theta^\mu)$는 θ^μ에 대한 Actor의 기울기이다. 목표함수의 gradient를 계산하기 위해서는 행동가치 함수 $\nabla_a Q(s,a|\theta^Q)$의 근사가 필요하며 이를 Critic이라는 가치신경망으로 추정하여 정책의 성과를 평가하고 행동의 계산은 Actor라는 정책신경망을 이용한다.

DPG을 위해 신경망을 학습할 때 문제는 샘플이 독립적이고 동일하게 분포되어 있다고 가정한다는 것이다. 그러나 일반적인 Actor-Critic 구조에서는 학습시킬 때 사용하는 샘플이 환경에서 순차적으로 탐색하여 생성되어 시간적으로 상관되어 있으며 이로 인하여 학습을 불안정하게 만들게 된다. 이러한 문제점을 개선하기 위하여 제안된 방법이 Deep Deterministic Policy Gradient(DDPG)[15]이다.

이러한 문제를 해결하기 위한 방법으로 DDPG는 앞에서 설명한 DQN의 에이전트가 신경망 학습을 위해 무작위로 샘플링하는데 사용하는 리플레이 메모리와 DQN 오차함수에 대한 타깃 Q 값을 생성하는데 사용되는 타깃 신경망을 적용하였다.

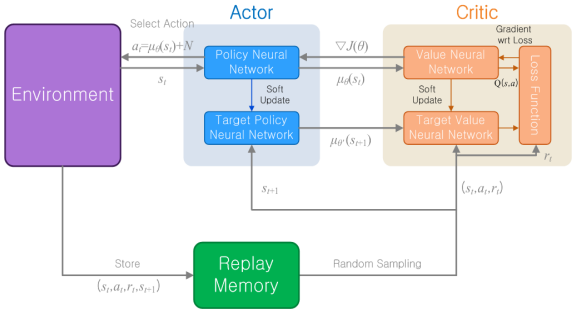


Fig. 5. Configuration of deep deterministic policy gradient algorithm

목표함수의 gradient는 식 (12)에서 보듯이 Critic 신경망의 상태가치함수를 잘 학습시키는 것이 중요하다. DQN에서는 시간차 오차를 계산할 때 발생하는 문제를 해결하기 위하여 타깃 계산용 신경망을 별도로 적용하였다. DDPG에서도 타깃 Actor 신경망 μ'과 타깃 Critic 신경망 Q'을 적용하였고 Critic 신경망의 학습에 사용될 시간차 타깃 y_i과 Critic 신경망의 업데이트를 위한 오차함수는 식 (13)과 같다.

$$\begin{aligned} y_i &= r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1})) \\ Loss_{critic} &= \frac{1}{2N}\sum_{i=1}^{N}(y_i - Q(s_i,a_i))^2 \end{aligned} \tag{13}$$

또한 DQN에서는 타깃 신경망을 일정 시간마다 주기적으로 업데이트 했지만, DDPG에서는 타깃 신경망이 학습된 신경망을 느린 속도로 따라가도록 식 (14)와 같이 구성한다. 이를 통하여 목표 값이 천천히 변경되도록 제한되어 학습 안정성이 크게 개선될 수 있다.

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \end{aligned} \tag{14}$$

마지막으로 DQN은 ϵ-탐욕 정책을 통해 샘플을 획득하였으나 DDPG는 확정적 정책이므로 행동에 노이즈 N(가우시안 화이트 노이즈)을 추가하여 행동에 무작위적인 성질을 추가하였으며 아래 식 (15)와 같이 행동을 결정한다.

$$a_t = \mu(s_t|\theta^\mu) + N_t \tag{15}$$

DDPG 강화학습의 주요 구성은 Fig. 5와 같으며 DDPG는 Actor-Critic 구조를 가지고 있으며 상태변수를 입력받아 확정적 행동을 계산하는 정책의 발전은 Actor 신경망의 업데이트로 Actor 신경망이 계산한 행동과 상태변수를 입력으로 받아서 행동가치를 계산하는 정책의 평가는 Critic 신경망을 사용한다. Actor는 보상을 직접 받을 수 없고, Critic은 행동에 직접 접근할 수 없다.

4. Deep Deterministic Policy Gradient 기반 롤 제어기 설계

4.1 Deep Deterministic Policy Gradient 기반 롤 제어기 설계

수중운동체의 PID 롤 제어기는 롤 제어 명령(ϕ_c)과 롤 오차를 입력 받아 PI 제어 명령을 계산하고 롤변화율(roll rate)을 내부 루프로 D 제어 명령을 계산하는 다중 루프 제어기로 구성되어 있으며 구조는 Fig. 6과 같다. PID 롤 제어 게인은 횡방향 선형 상태 공간 모델식인 식 (2)를 기반으로 롤 제어 명령 0°을 항상 일정하게 유지할 수 있도록 설정한다[4,11,19].

DQN with NAF 기반으로 학습된 가치신경망 롤 제어시스템의 구성은 Fig. 7과 같으며 본 논문에서 제안한 DDPG 기반으로 학습된 Actor 신경망 롤 제어시스템의 구성은 Fig. 8과 같다. 두 제어시스템의 신경망은 학습 방법의 차이로 인하여 다르게 구성된다.

DQN with NAF 신경망은 2개의 은닉층으로 Fig. 9와 같이 구성되며 입력은 상태변수인 수중운동체의 롤과 롤변화율이며 출력은 수중운동체의 구동 롤 제어 입력과 가치 함수이다. 은닉층의 노드수는 각각 100개 이며 은닉층의 활성화 함수는 ReLU 함수이고 출력단의 활성화함수는 linear와 Tanh 함수이며 경사하강법은 adam optimizer를 사용하였다. Learning rate는 0.001, τ는 0.0001, γ는 0.99, 경험 리플레이 메모리 버퍼(buffer) 크기는 5000이며 미니배치(minibatch) 크기는 64로 하이퍼파라미터(hyperparameter)를 설정하였다.

DDPG 신경망의 구성은 크게 Actor 신경망과 Critic 신경망으로 나누어진다. 먼저 Critic 신경망은 2개의 은닉층으로 Fig. 10(a)와 같이 구성되며 입력은 상태변수인 수중운동체의 롤과 롤변화율 및 롤 제어 입력이며 출력은 추정된 상태가치이다. 은닉층의 노드수는 각각 100개 이며 은닉층의 활성화 함수는 ReLU 함수이고 출력단의 활성화함수는 linear 함수이며 경사하강법은 adam optimizer를 사용하였다. 하이퍼파라미터인 learning rate는 0.001, γ는 0.99, τ는 0.001으로 설정하였다. 다음으로 Actor 신경망은 2개의 은닉층으로 Fig. 10(b)와 같이 구성되며 입력은 상태변수인 수중운동체의 롤과 롤변화율이며 출력은 수중운동체의 롤 제어 입력이다. 은닉층의 노드수는 각각 100개 이며 은닉층의 활성화 함수는 ReLU 함수이고 출력단의 활성화함수는 Tanh 함수이며 경사상승법은 adam optimizer를 사용하였다. learning rate는 0.01, τ는 0.001, 경험 리플레이 메모리 버퍼 크기는 5000이며 미니배치 크기는 64로 하이퍼파라미터를 설정하였다.

강화학습을 위한 기본 보상 함수는 식 (16)과 같으며 보상식의 계수와 제한 범위는 시뮬레이션을 통한 시행착오로 결정하였다. 롤(ϕ)이나 롤변화율(p)이 제한 범위를 벗어나면 에피소드가 종료되며 식 (16)과 같은 보상이 주어지고 롤(ϕ)과 롤변화율(p)이 제한 범위를 벗어나지 않고 일정 시간스텝을 유지할 경우 에피소드가 자동으로 종료되며 롤과 롤변화율을 0으로 잘 유지할수록 양의 보상을 받을 수 있도록 식 (16)과 같이 구성한다.

$$r=\begin{cases}10-0.5\phi^2-0.2p^2 & |\phi|<\dfrac{\pi}{36},|p|<\dfrac{\pi}{12}\\ -0.5\phi^2-0.2p^2 & |\phi|\geq\dfrac{\pi}{36},|p|\geq\dfrac{\pi}{12}\end{cases} \tag{16}$$

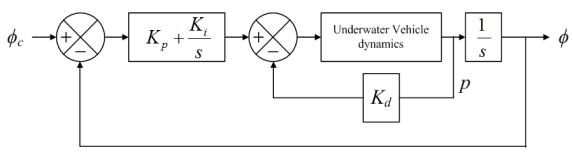


Fig. 6. Configuration of PID roll controller of underwater vehicle

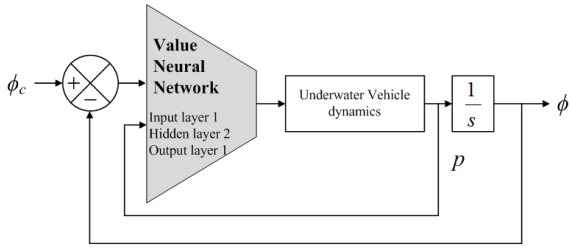


Fig. 7. Configuration of DQN with NAF roll controller of underwater vehicle

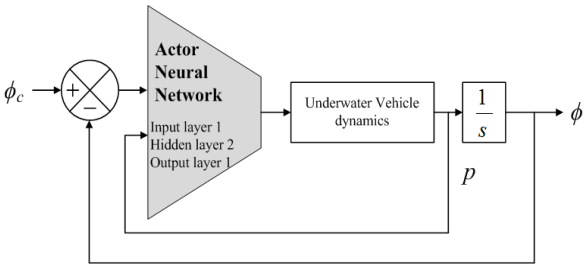


Fig. 8. Configuration of DDPG roll controller of underwater vehicle

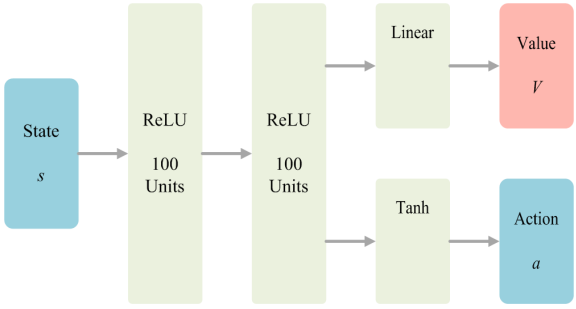
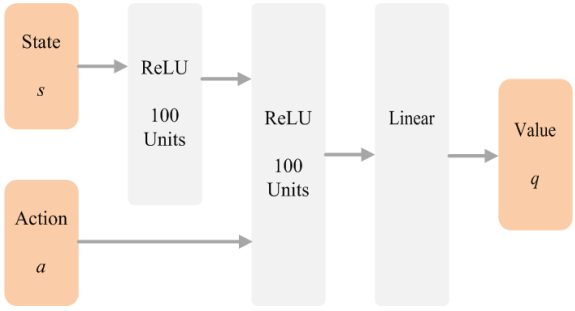
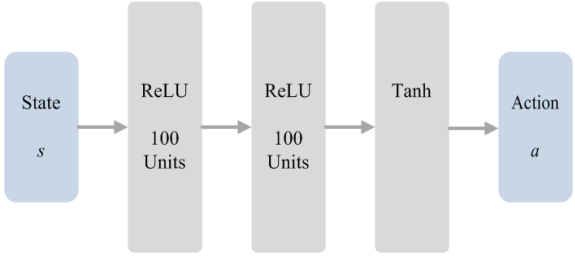


Fig. 9. DQN with NAF networks layout



(a) Layout of the critic network



(b) Layout of the actor network

Fig. 10. DDPG networks layout

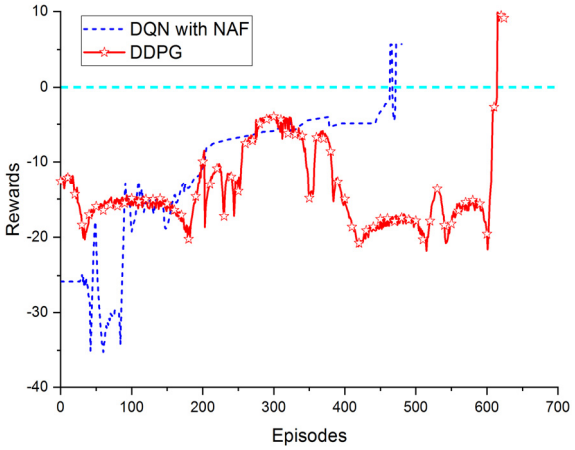


Fig. 11. The learning results of DQN with NAF and DDPG roll controller

4.2 Deep Deterministic Policy Gradient 기반 롤 제어기 학습

수중운동체의 롤 제어에서 롤과 롤변화율을 항상 0으로 일정하게 유지하는 것이 주행 안정성에 중요한 부분이다. 그러므로 앞에서 제안한 DDPG 강화학습 기반 롤 제어기와 수중운동체의 6자유도 비선형 운동 방정식인 식 (1)을 통합하여 롤과 롤변화율을 0으로 잘 유지할 수 있도록 신경망을 학습시킨다. 또한 성능을 비교분석하기 위하여 DQN with NAF 기반 롤 제어기도 동일한 조건으로 신경망을 학습시킨다.

기존의 연구[11]에서는 Dueling DDQN 강화학습 기반의 가치신경망 롤 제어기를 학습시키기 위하여 롤 초기값을 ±5° 사이의 값으로 랜덤하게 주어 다양한 롤 오차의 영향을 학습할 수 있도록 하였다. 그러나 본 논문에서 제안하는 DDPG 강화학습 기반의 정책신경망 롤 제어기는 롤 초기값을 0°로 주어 학습하도록 하여 롤을 항상 0°으로 유지하는 정책을 학습하게 함으로서 다양한 상황에서도 안정적인 제어 성능을 나타낼 수 있도록 구현한다. 성능 비교 분석을 위하여 DQN with NAF 강화학습 기반의 가치신경망 롤 제어기 또한 기존의 연구와 달리 같은 초기 조건에서 학습하여 그 결과를 비교하기로 한다.

DQN with NAF 기반의 롤 제어기와 DDPG 기반의 롤 제어기의 학습 과정을 그린 그래프는 Fig. 11과 같다. 가로축은 에피소드이고 세로축은 각 에피소드에서 받은 총 보상을 나타내고 있다. 최근 10개의 에피소드가 연속적으로 보상값의 크기가 0 이상일 경우 학습

이 완료된 것으로 판단하였으며 DQN with NAF은 약 480 에피소드가 지난 후 학습이 완료되었으며 DDPG는 약 624 에피소드가 지난 후 학습이 완료되었다. DQN with NAF와 달리 DDPG는 Critic 신경망과 Actor 신경망을 동시에 학습시켜야 되므로 학습 시간이 DQN with NAF에 비해서 오래 걸리는 것을 확인할 수 있다.

5. Deep Deterministic Policy Gradient 강화학습 기반 수중운동체의 롤 제어 시뮬레이션

본 논문에서 제안된 DDPG 기반 롤 제어 성능을 검증하기 위하여 학습이 완료된 DDPG 기반 롤 제어기와 PID 롤 제어기 및 학습이 완료된 DQN with NAF 기반 롤 제어기를 동일한 조건에서 시뮬레이션 하고 제어 성능을 비교 분석하였다. 이 때 기존의 심도 및 경로 제어기를 이용하여 심도 40 m와 경로 0°를 유지하였으며 초기 속도 V1에서 추진기가 작동하여 속도 V1을 유지하도록 시뮬레이션을 수행한다. 또한 수중운동체가 주행 도중 여러 가지 원인(급격한 회전 또는 속도 변화)이 발생하더라도 롤이 ±5° 이하로 유지되게 제어하기 위하여 초기 롤 ±5°에서의 시뮬레이션을 수행한다.

먼저 초기 롤이 0°인 상태에서 주행한 롤 제어 시뮬레이션 결과는 Fig. 12와 같다. Fig. 12에서 보듯이 초기에 추진기 작동에 따른 수중운동체의 속도 변화로 인한 과도 상태에서 PID 롤 제어 성능은 다소 불안정하여 최대 1.2°의 롤 오차가 발생하였으나 정상 상태에서는 롤 오차가 0°로 수렴하여 안정적인 제어 성능을 나타내고 있다. DQN with NAF 기반 롤 제어기는 초기 과도 상태에서는 최대 0.28°의 롤 오차가 발생하였으며 정상 상태 오차는 0.1°로 나타나며 정상 상태 오차가 발생하는 원인으로는 확률 분포기반으로 롤 제어 입력을 선택하기 때문이다. DDPG 기반 롤 제어기는 초기 과도 상태에서 롤 최대 오차가 0.09°이며 정상 상태 오차는 0°로 수렴하고 있다. DQN with NAF 및 DDPG 기반 롤 제어기 모두 과도 상태 및 정상 상태에서 안정적인 제어 성능을 나타내고 있으며 이는 초기 과도 상태를 포함하여 학습하였기 때문이고 DDPG 기반 롤 제어기가 확정 정책으로 롤 제어 입력을 계산하여 DQN with NAF 기반 롤 제어기보다 정상상태에서 개선된 제어 성능을 나타내고 있다. 수중운동체의 롤변화율 주행 결과는 Fig. 13과 같으며 PID 롤 제어기는 초기 과도 상태에서 변동이 크게 발생한 후 정상상태에서 안정적인 상태를 유지하고 있으나 DQN with NAF 및 DDPG 기반 롤 제어기는 전체적으로 롤변화율이 안정적으로 유지되고 있다.

다음으로 초기 롤이 +5°인 상태에서 주행하였을 때의 롤 제어 시뮬레이션 결과는 Fig. 14와 같다. Fig. 14에서 보듯이 수중운동체의 속도 변화에 따른 초기 과도 상태에서 PID 롤 제어 성능은 -0.94°의 오버슈트가 발생하고 정상 상태에서는 오차가 0°으로 수렴하는 제어 성능을 나타내고 있다. DQN with NAF 기반 롤 제어기는 초기 과도 상태에서 오버슈트가 발생하지 않으며 정상 상태 오차는 약 0.1°로 유지하며 안정적인 제어 성능을 나타내고 있으며 DDPG 기반 롤 제어기도 마찬가지로 초기 과도 상태에서 오버슈트가 발생하지 않으며 정상 상태 오차는 0°로 수렴하며 안정적인 제어 성능을 나타내고 있다. 수중운동체의 롤 변화율 주행 결과는 Fig. 15와 같으며 DQN with NAF 및 DDPG 기반 롤 제어기는 과도 상태의 롤변화율이 PID 롤 제어기보다 크게 나타나고 있으나 정상 상태에서는 모두 안정적으로 나타난다.

마지막으로 초기 롤이 -5°인 상태에서 주행하였을 때의 롤 제어 시뮬레이션 결과는 Fig. 16과 같다. Fig. 16에서 보듯이 수중운동체의 속도 변화에 따른 초기 과도 상태에서 PID 롤 제어 성능은 1.44°의 오버슈트가 발생하고 정상 상태에서는 오차가 0°으로 수렴하는 제어 성능을 나타내고 있으며 DQN with NAF 제어 성능은 초기 과도 상태에서 1.65°의 오버슈트로 PID 롤 제어기 보다 큰 오버슈트가 발생하였고 정상 상태에서 오차는 0.1°으로 수렴하였다. 반면에 DDPG 기반 롤 제어기는 초기 과도 상태에서는 오버 슈트가 발생하지 않았으며 정상 상태에서는 오차가 0°로 수렴하는 안정적인 제어 성능을 나타내고 있다. 수중운동체의 롤변화율 주행 결과는 Fig. 17과 같으며 DQN with NAF 기반 롤 제어기의 과도 상태의 롤변화율이 PID 및 DDPG 기반 롤 제어기보다 크게 변화하는 것을 확인 할 수 있다.

Fig. 18~20은 초기 롤 위치에 따른 PID, DQN with NAF 및 DDPG 기반 롤 제어 입력의 결과를 나타낸다. Fig. 18~20에서 보듯이 DQN with NAF 및 DDPG 기반 롤 제어 입력이 연속 행동 공간으로 설계되어 PID 롤 제어 입력과 비교하여 떨림 없이 연속적인 롤 제어 입력을 나타내는 것을 확인 할 수 있다.

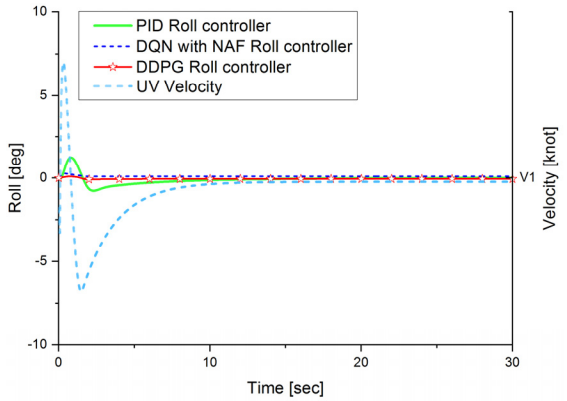


Fig. 12. The simulation results of PID, DQN with NAF and DDPG roll control(initial roll: 0°)

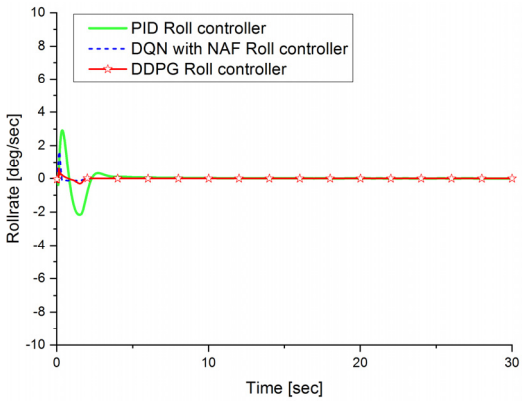


Fig. 13. The roll rate trajectory results of PID, DQN with NAF and DDPG roll controller(initial roll: 0°)

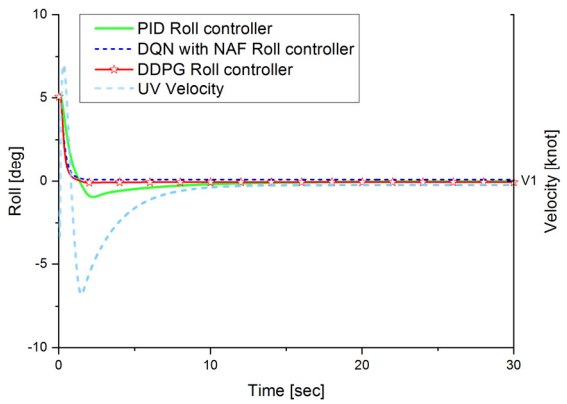


Fig. 14. The simulation results of PID, DQN with NAF and DDPG roll control(initial roll: +5°)

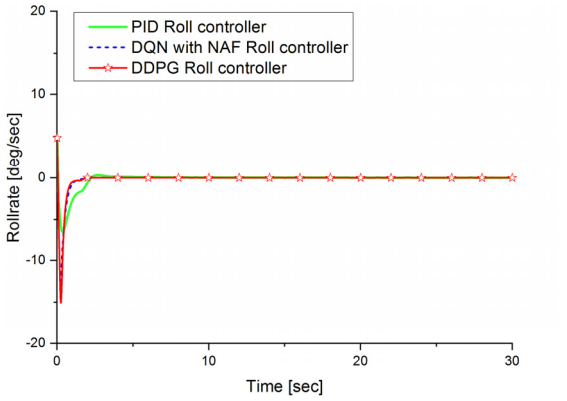


Fig. 15. The roll rate trajectory results of PID, DQN with NAF and DDPG roll controller(initial roll: +5°)

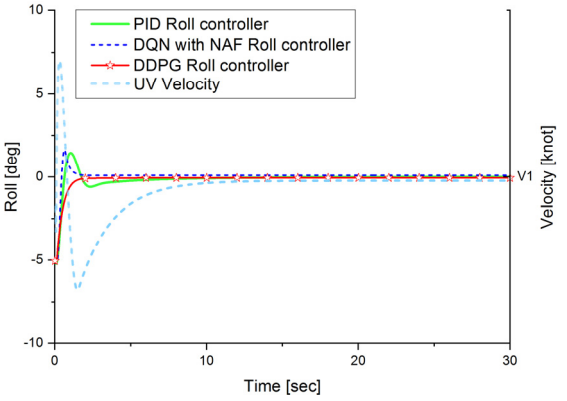


Fig. 16. The simulation results of PID, DQN with NAF and DDPG roll control(initial roll: −5°)

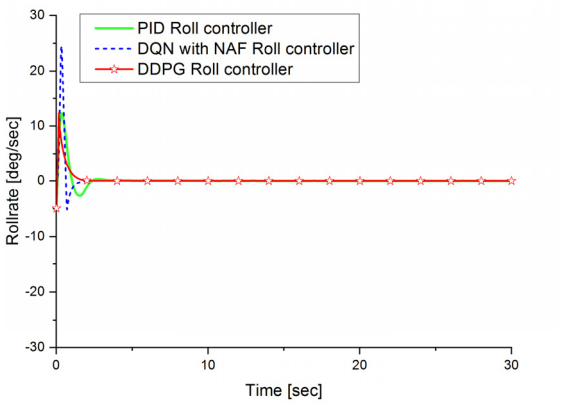


Fig. 17. The roll rate trajectory results of PID, DQN with NAF and DDPG roll controller(initial roll: −5°)

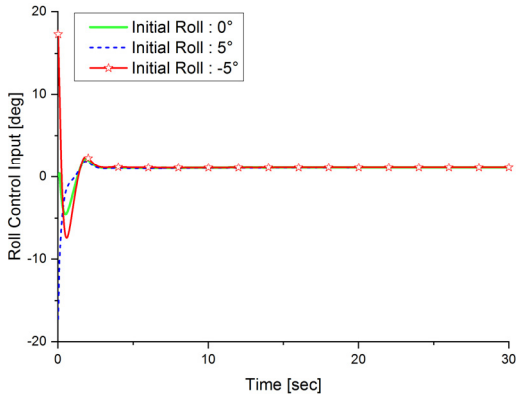


Fig. 18. The control input δ_ϕ results of PID roll controller

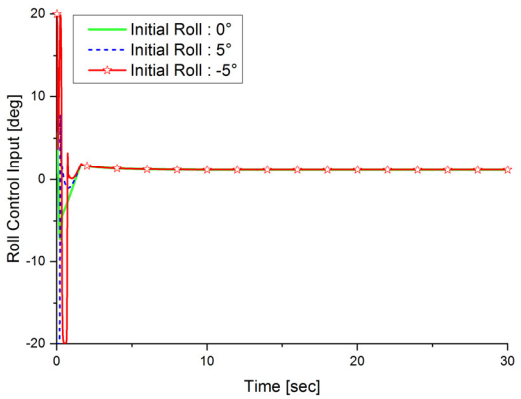


Fig. 19. The control input δ_ϕ results of DQN with NAF roll controller

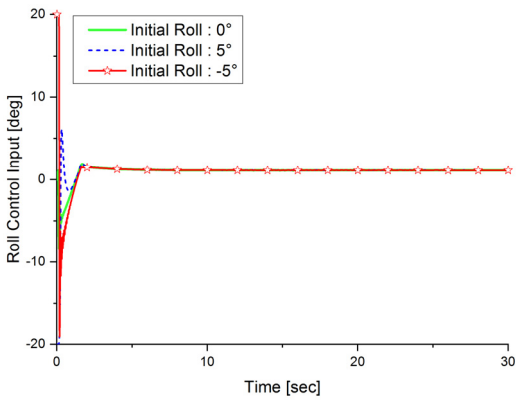


Fig. 20. The control input δ_ϕ results of DDPG roll controller

Fig. 12의 롤 제어 시뮬레이션 결과를 살펴보면 초기 롤 0°인 상태에서 DQN with NAF 및 DDPG 기반 롤 제어 성능은 과도 상태 및 정상 상태 모두 안정적으로 나타났다. 이는 초기 롤이 0°인 상태에서 학습을 수행한 결과로 정상적인 학습이 된 것으로 판단할 수 있다. 또한 초기 롤 +5°인 상태에서의 롤 제어 시뮬레이션 결과는 Fig. 14와 같으며 초기 롤이 0°인 상태에서만 학습을 수행했음에도 불구하고 DQN with NAF 및 DDPG 기반 롤 제어 성능은 과도 상태 및 정상 상태 모두 안정적인 것을 확인할 수 있다.

마지막으로 Fig. 16의 초기 롤 -5°인 상태에서의 롤 제어 시뮬레이션 결과를 살펴보면 DQN with NAF 기반 롤 제어기는 DDPG 기반 롤 제어기와 달리 과도 상태에서의 제어 성능에서 차이가 발생하였다. 이는 DQN with NAF이 가치 기반 강화학습으로 빠른 학습 수렴 성능을 나타내지만 경험하지 못한 상황이나 환경에서는 불안정한 제어 성능이 나타날 수 있다는 것을 보여주고 있다. 그러나 본 논문에서 제안한 정책 기반 강화학습인 DDPG를 적용한 롤 제어기는 롤 제어 정책을 학습함으로써 학습 수렴 성능이 상대적으로 오래 걸리지만 경험하지 못한 상황 및 환경에서도 제어가 가능하다는 것을 확인할 수 있다.

6. 결 론

강화학습의 문제 해결 방법은 가치 기반 방법과 정책 기반 방법으로 나눌 수 있으며 가치 기반 방법은 다양한 경험을 통한 빠른 학습이 가능하지만 경험하지 못한 상황에 대한 결과를 정확히 예측하기 힘드나 정책 기반 방법은 학습에 오랜 시간이 걸리지만 정책을 학습하므로 다양한 상황과 환경에 유연하게 대응이 가능하다. 그러므로 본 논문에서는 수중운동체가 물 속에서 다양한 임무를 안정적으로 수행하기 위해 적합한 DDPG 기반 수중운동체의 롤 제어기를 제안하였다. 제안된 DDPG 기반 롤 제어기의 성능은 동일한 조건에서 PID 및 DQN with NAF 기반 롤 제어기와의 시뮬레이션을 통하여 비교 검증 및 분석하였다. 본 논문에서 제안된 DDPG 기반의 수중운동체 롤 제어기는 롤 제어 정책 학습을 통하여 기존의 제어기와 비교하여 다양한 상황 및 환경에서도 개선된 제어 성능을 나타내는 것을 확인하였다. 이를 바탕으로 수중운동체의 다양한 임무 수행을 위한 제어 성능 향상에 도움

이 될 수 있을 것으로 판단된다.

References

[1] J. Heo, et. al., “Technology Development of Unmanned Underwater Vehicles (UUVs),” Journal of Computer and Communications, Vol. 5, No. 7, pp. 28-35, 2017.

[2] K. Y. Jung, et. al., “Autopilot Design of an Autonomous Underwater Vehicle using Robust Control,” Transaction on Control Automation, and Systems Engineering, Vol. 4, No. 4, pp. 264-269, 2002.

[3] J.-Y. Park, et. al., “Depth Controller Design for Submerged Body Moving near Free Surface Based on Adaptive Control,” Journal of Ocean Engineering and Technology, Vol. 29, No. 3, pp. 270-282, 2015.

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

[5] 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.

[6] 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.

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

[8] H. v. Hasselt, et. al., “Deep Reinforcement Learning with Double Q-learning,” AAAI, Vol. 16, 2016.

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

[10] S. Y. Kim, et. al., “The roll control of Unmanned Underwater Vehicle using Double Deep-Q Network Reinforcement Learning,” KIMST Annual Conference Proceedings, pp. 1601-1620, 2020.

[11] B. Lee, “Roll control of Underwater Vehicle based Reinforcement Learning using Advantage Actor-Critic,” Journal of the Korea Institute of Military Science and Technology, Vol. 24, No. 1, pp. 123-132, 2021.

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

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

[14] Silver, David, et al. “Deterministic Policy Gradient Algorithms,” Proceedings of the 31st International Conference on Machine Learning(ICML-14), 2014.

[15] T. P. Lillicrap, et al. “Continuous Control with Deep Reinforcement Learning,” International Conference on Learning Representations(ICLR), 2016.

[16] S. Gu, et. al., “Continuous Deep Q-Learning with Model-based Acceleration,” International Journal of Precision Engineering and Manufacturing, Vol. 14, No. 12, pp. 2067-2073, 2013.

[17] 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.

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

[19] B. C. Kuo, “Automatic Control Systems,” Prentice Hall, 1994.