드론 제어를 위한 심층 강화 학습의 최신 연구 사례

드론과 같은 로봇 시스템은 대개 비선형 동역학 방정식에 의해 움직임이 기술된다. 또한, 동역학 모델을 구성하는 많은 파라미터는 일반적으로 참값을 측정하는 것이 힘들며 시간에 따라 변하므로 고전적인 제어기 설계 기법을 적용하는 것이 어렵다. 이러한 한계를 극복하기 위해 최근 심층 강화 학습을 로봇 제어에 적용하는 연구가 시작되었다. 심층 강화 학습은 일반적으로 모델을 필요로 하지 않고, 시스템에 입력을 가하여 획득한 보상의 누적합을 최대화하는 방향으로 제어기가 스스로 학습하며, 이로 인해 로봇 시스템의 비선형 동역학이나 파라미터 불확실성 등에 강인한 장점을 지닌다. 이러한 이점 때문에 학계에서 활발한 연구가 수행되고 있으며, 본 논문에서는 이에 대한 최신 연구 동향 및 사례를 요약하여 소개한다.

변 영 훈, 김 한 솔 (국립한국해양대학교 제어계측공학과)

I. 서론

강화 학습(reinforcement learning: RL)은 행동 심리학에서 영감을 받은 기계 학습 알고리즘으로, 에이전트가 환경으로부터 받는 보상의 누적합의 기대치를 최대화하도록 행동을 결정하는 방법이다[1]. RL은 환경이 마르코브 결정 과정(Markov decision process: MDP)으로 주어진 경우에 적용한다. MDP 문제를 해결할 수 있는 기존 방법과의 가장 큰 차이점은 환경에 대한 모델을 요구하지 않는다는 것이다.

이와 같은 장점에도 불구하고 고전적인 RL 기법은 환경의 상태 공간 크기로 인한 차원의 저주 등으로 인해 실제 많은 분야에 적용하기 어렵다. 로봇 시스템과 같이 상태 공간이 연속적인 시스템일 경우에 적용하는 것이 사실상 불가능하여 활발한 연구가 이루어지지 못하였다.

최근에는 신경망의 universal approximator 속성을 이용하여 심층 신경망(deep neural network: DNN)을 이용해 정책(policy)이나 가치 함수(value function)를 근사하는 심층 강화 학습 (deep reinforcement learning: DRL) 연구가 진행되었다. [2]에서는 기존 Q-learning을 DRL로 확장한 Deep Q-network (DQN) 기법을 개발하여 Atari 게임에서 우수한 점수를 얻도록 에이전트를 학습시켰고, 그 결과 에이전트가 일반인들의 평균 점수보다 높은 점수를 획득했다. 또한 [3]에서 소개된 AlphaGo가 인간 바둑 챔피언을 압도하는 모습을 보인 후 대중과 학계 모두가 DRL을 주목하게 되었다.

RL 연구는 크게 가치 함수 기반(value-based) 방법과 정책 경사 기반(policy gradient-based) 방법으로 나뉜다. 가치 함수 기반 방법의 대표적인 기법인 Q-learning [1]은 어떤 상태에서 특정 행동을 취했을 때 보상의 누적합의 기대값을 의미하는 상태-행동 가치 함수(state-action value function)를 갱신하고, 이로부터 가장 가치 있는 행동을 결정하는 기법이다. 이러한 Q 함수를 DNN을 이용하여 근사한 기법으로 DQN이 연구되었고, 이를 개선하여 Double DQN [4], Dueling DDQN [5], Prioritized DDQN [6], Noisy DQN [7] 등이 연구되었으며, 최근에는 이를 모두 결합한 Rainbow [8] 알고리즘이 등장하였다.

한편, 정책 경사 기반의 DRL은 DNN이 최적의 정책을 근사하도록 학습하는 방법으로, DNN의 가중치 (weight)와 편향 (bias)을 갱신하기 위해 경사 상승법 (gradient ascent)을 사용하기 때문에 정책 경사라고 한다. Q-learning 기반의 DRL에서는 DNN에 상태가 입력되면 현재 상태에서 모든 행동에 대한 Q 함수의 값이 출력되는 반면, 정책 경사 기반 방법에서는 현재 상태가 DNN

에 입력되어 에이전트가 수행할 행동이 출력된다. REINFORCE 알고리즘의 성공적인 연구 이후 [9], 이를 개선한 SAC (Soft Actor-Critic) [14], A2C (Advantage Actor-Critic) [10][11], A3C (Asynchronous Advantage Actor-Critic) [10][11], DDPG (Deep Deterministic Policy Gradient) [12], TRPO (Trust Region Policy Optimization) [15], PPO (Proximal Policy Optimization) [16] 등이 연구되고 있고, 이들은 주로 로봇 시스템의 제어 문제에 효과적으로 적용되고 있다.

앞서 소개한 DRL 기법들의 연구를 토대로, 고전적인 RL이나 제어 기법으로는 해결하기 어려웠던 게임 인공지능 분야 [27][28] 뿐만 아니라 로봇의 제어[24][25][26], 자율 주행 [21][22], 드론 제어 [29][30][31][32] 등 다양한 분야에 DRL을 결합하는 연구가 활발하게 진행되고 있다.

이러한 분석을 바탕으로, 본 기술 특집 논문에서는 최근 주목받고 있는 DRL 기법들의 최신 연구 동향을 살펴보고, 이를 드론의 제어 문제에 적용한 연구 사례들을 분석한다.

II. DRL 연구 동향 및 응용 분야

2.1 RL 문제 정의

RL은 MDP로 정의되는 문제를 해결하기 위해 에이전트의 행동 규칙인 정책을 결정하는 기법이며, MDP는 $(S, A, P^a_{SS}, r, \gamma)$의 tuple로 표현된다. 그림 1은 RL의 구성 요소이다. 에이전트는 환경에 행동을 취하고 환경은 이에 대한 새로운 상태 변수와 보상을 내보낸다. 여기서 $S \in \mathbb{R}^n$는 상태 공간으로 에이전트가 환경으로부터 관측하는 값의 모임이며, $A \in \mathbb{R}^m$는 행동 공간으로 에이전트가 선택할 수 있는 행동의 모임이다. $P^a_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a)$는 상태 전이 확률로 현재 상태 s에서 행동 a를 수행했을 때 상태가 s'에 위치할 확률이며, 이는 환경에 의해 결정되는 값이다. 본 논문에서 분석하는 model free 기반의 RL 알고리즘에서 에이전트는 환경에 대해 알지 못하므로 상태 전이 확률은 에이전트의 학습 과정에는 사용되지 않는다. 다음으로 r은 에이전트가 취한 행동으로 인해 환경으로부터 받는 보상이고, γ는 할인율로 향후에 얻게 될 보상보다 현재 받을 수 있는 보상에 더 큰 가중치를 주기 위해 설계하는 파라미터이다.

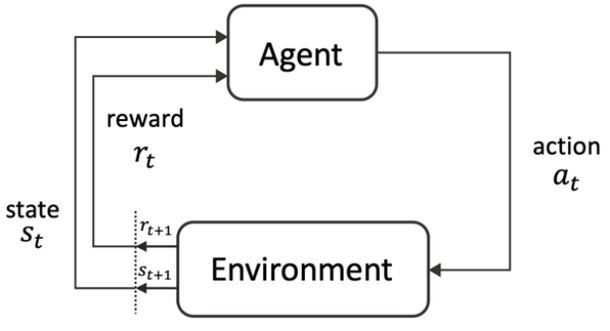

그림 1. RL의 구성요소 [39].

에이전트가 현재 주어진 상태 s에서 특정 행동 a를 선택하도록 정의하는 규칙을 정책 $\pi(a|s)$이라 한다. RL의 학습 목표는 적절한 정책을 수립하여 다음의 식으로 주어지는 누적 보상을 최대화하는 것이다.

$$G_t = r_{t+1} + \gamma r_{t+2} + \cdots = \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}$$

이를 위해서 DNN의 학습 과정을 통해 현재 정책 $\pi(a|s)$를 갱신하며, 최종 목표는 학습된 정책이 최적 정책 $\pi^*(a|s)$에 수렴하는 것이다. 한편, 현재 주어진 정책이 최적 정책에 어느 정도 수렴했는지는 현재 상태의 가치 $V_\pi(s)$나 현재 상태에서 어떤 특정 행동을 할 때의 가치 $Q_\pi(s,a)$의 정보로부터 알 수 있다. 여기서 $V_\pi(s)$는 상태 가치 함수, $Q_\pi(s,a)$는 상태-행동 가치 함수라고 한다.

고전 RL에서는 에이전트가 취할 수 있는 모든 상태와 행동에 대해 상태 가치 함수와 상태-행동 가치 함수를 계산한다. 따라서 기존 RL 기법은 상태의 차원 $\mathbb{R}^n$이 커지거나 연속적인 경우, 혹은 환경에 대한 정보 $P^a_{ss'}$가 주어지지 않는 문제에는 적용할 수 없다. 이러한 한계를 해결하기 위해 가치 함수나 정책을 DNN을 이용해 근사하여 MDP 문제를 해결하는 기법을 DRL이라고 한다. 이어지는 절에서는 DRL의 대표적인 두 가지 접근 방법에 대해 비교 분석과 최신 연구 동향을 알아본다.

2.2 가치 기반 DRL

앞서 살펴 보았듯, 바둑과 같이 상태 공간이 매우 크거나, 로봇 제어와 같이 연속된 값을 가지는 경우 모든 상태에 대해 가치 함수를 계산하는 것이 불가능에 가깝다. 이에 착안하여 가치 함수를 DNN으로 근사한 RL 알고리즘이 연구되었고, 이를 가치 기반 DRL이라고 부른다. 가장 대표적인 알고리즘은 DQN [2]으로 기존 Q-learning 기법에서 Q 함수를 DNN으로 근사한 알고리즘이다. 즉, DQN에서 DNN은 현재 상태를 입력으로 받아 모든 가능한 행동에 대한 Q 값을 출력으로 내보낸다. 탐욕 정책을 따라 에이전트는 출력으로 얻어진 값 중에서 가장 큰 값을 내는 행동을 선택하여 행동한다. 따라서 DQN의 학습 목표는 DNN을 학습시켜 최적의 Q 함수를 근사하도록 하는 것이다.

DNN의 학습은 정답인 Q 함수의 참값과 현재 예측값의 오차를 최소화하는 방향으로 진행된다. 여기서 정답은 Bellman equation을 통해 얻어지는데, 정답을 계산하는 network와 학습이 이루어지는 network가 같으면 학습 과정에서 정답이 지속적으로 변하여 학습이 불안정하다. 이를 해결하기 위해 [4]에서는 정답을 생성하는 target network를 별도로 생성하여 이러한 문제를 해결했다. Target network를 통해 예측된 정답은 다음의 식으로 얻어진다.

$$y = r_{t+1} + \gamma \max_{a} Q^{-}(s_{t+1}, a, \theta^{-}), \tag{1}$$

여기서 θ^{-}는 target network의 가중치를 의미한다. 한편, 현재 상태에서의 Q 함수는 다음의 식으로 예측된다.

$$Q(s_t, a_t, \theta), \tag{2}$$

여기서 θ는 현재 network의 가중치이다. 이제, 식 (1)과 (2)의 제곱 평균 오차(mean squared error: MSE)를 최소화하도록 경사 하강법을 통해 θ를 갱신함으로써, Q 함수의 참값에 도달하며, 정책은 최적 정책이 된다.

또한, DQN은 off policy 알고리즘으로, [2]에서는 experience replay buffer를 통해 학습 데이터의 연관성을 줄여주어 학습을 더욱 안정적으로 진행되도록 하였다. [2]에서 보여준 획기적인 결과에도 불구하고, DQN 알고리즘은 개선의 여지가 많은 초기 연구라고 할 수 있다. 이를 개선하기 위한 연구가 최근까지도 활발히 진행되고 있어 본 논문에서는 이에 대한 대표적인 연구를 소개한다.

DQN에서 Q 함수 값의 과대평가 문제가 발생하여 최적의 행동을 수행하지 못하는 현상이 종종 발생한다. 이에 대해 [4]에서는 두 개의 Q network를 이용한 double DQN을 제안하였다. DQN에서 Q 함수의 과대평가가 이루어지는 이유는 (1)에서의 max 함수 때문이다. 이를 해결하기 위해 [4]에서는 현재 network로부터 Q 값이 최대가 되는 행동을 선택한 후, 이 행동에 대한 target network의 Q값을 정답으로 사용한다. 이를 정리하면 다음 식과 같다.

$$y = r_{t+1} + \gamma Q^{-}\left(s_{t+1}, \operatorname*{argmax}_{a} Q(s_{t+1}, a, \theta), \theta^{-}\right).$$

한편, 기존의 DQN이나 double DQN은 모두 Q 함수를 DNN의 추론 결과로 출력한다. [5]에서 제안한 dueling DQN 역시 최종 출력은 Q 함수의 근사 추론 값이지만, 이를 계산하는 과정에서 그림 2와 같이 상태 가치 함수와 advantage 함수를 중간 추론 결과로 출력한다. 여기서 advantage 함수는 다음의 식으로 표현하며, 특정 상태 s에서 어떤 행동 a가 다른 행동들에 비해 상대적으로 어느 정도의 가치를 가지는지를 나타낸다.

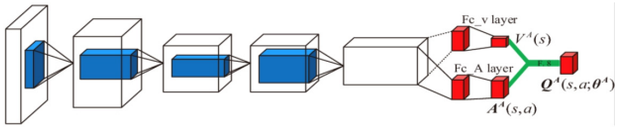

그림 2. Dueling DQN의 network 구조.

$$A(s,a) = Q(s,a) - V(s).$$

Advantage 함수의 정의에 따르면 Q 함수는 중간 추론 결과인 $V(s)$와 $A(s,a)$의 단순 합으로 계산할 수 있으나, 신경망 학습 과정에서 각 출력이 의미하는 바를 유지하기 위해서 다음의 식을 사용하여 최종 출력인 Q 함수를 추론한다.

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a')\right),$$

여기서 $|\mathcal{A}|$는 행동 공간의 크기를 의미한다. 따라서 위의 식은 advantage 함수에 현재 상태에서의 advantage 평균을 빼는 것을 의미한다. 이렇게 함으로써 advantage의 변화량이 줄어들게 되어 안정적인 학습이 가능하도록 하였다.

이렇듯 dueling DQN은 중간 과정에서 상태 가치 함수를 예측하기 때문에 기존 DQN 알고리즘과는 달리 수행하지 않은 행동에 대한 Q 함수도 대략적으로 예측이 가능하다는 장점을 지닌다.

마지막으로 살펴볼 연구는 [6]에서 연구된 prioritized experience replay buffer 기법이다. 기존 DQN 기반 알고리즘에서는 replay buffer에 쌓인 수 많은 학습 데이터 중 일부를 uniform distribution을 따라 random하게 추출하였다. 그러나, 어떤 경험은 RL 에이전트에게 있어 신선하지만 그렇지 않은 경험도 존재한다. 따라서 신선한 경험 위주로 학습을 수행한다면 더욱 빠르고 안정적으로 학습이 진행될 것이다. 신선한 경험이란 현재 Q network에 의한 예측과 경험과의 가치 함수 차이가 큰, 즉 TD (time difference) error가 큰 경험이라고 할 수 있다. 그러나 TD error가 큰 경험만을 선택한다면 소수의 경험만 반복하여 학습하므로 편향 문제가 발생할 수 있다. 이를 해결하기 위해 TD error가 높은 순으로 뽑힐 확률을 크게 하여 확률적으로 경험을 선택하는 stochastic sampling method를 사용한다.

2.3 정책 경사 기반 DRL

가치 기반 DRL은 가치 함수를 근사한 후, 이로부터 정책을 선

택하는 방식으로 학습이 이루어진다. 반면 정책 경사 기반 DRL은 DNN이 정책 함수 자체를 근사하는 기법이다. 즉, DNN은 입력으로 상태 변수를 받은 후, 에이전트가 수행해야할 최적의 행동을 결정하여 출력으로 내보낸다. 따라서 신경망이 학습됨에 따라 누적 보상의 기대값이 증가해야 한다. 이를 위해 다음과 같은 목적 함수를 설계하고, DRL의 학습은 목적 함수를 최대화하는 최적화 문제가 된다.

$$J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)] = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} r_t\right],$$

여기서 τ 는 시간 t를 따라 정책 π_θ에 따른 상태와 행동으로 구성된 궤적이고, T는 τ 의 길이다. 이를 구현한 REINFORCE [9] 알고리즘에서 목표 함수의 경사는 다음과 같다.

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[G_t \nabla_\theta \log \pi_\theta(a_t|s_t)].$$

AC [1]는 기본 정책 경사 방법이 타임스텝이 아닌 에피소드 마다 학습할 수 있고 에피소드 길이에 따라 반환값의 분산이 큰 단점을 해결하기 위한 알고리즘이다. 반환값의 분산을 줄이기 위해 목표 함수에서 반환값의 정책을 근사하는 actor 신경망의 출력인 행동의 보상으로 critic 신경망이 TD error를 계산해 actor를 최적화한다. 두 신경망의 최적화 식은 다음과 같다.

$$\nabla_\theta J(\theta) \sim \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, A(s_t, a_t),$$
$$A(s_t, a_t) = r_{t+1} + \gamma V_v(s_{t+1}) - V_v(s_t),$$

여기서 $V_v(\cdot)$는 critic으로 근사되는 상태-가치함수이다. 이때 위의 advantage 함수 $A(s_t, a_t)$를 최적화에 사용하는 알고리즘을 A2C [10]라고 한다.

한편, AC의 한 가지인 A3C [11]는 매 타임스텝마다의 샘플로 학습하는 A2C와 달리, 샘플을 모으는 여러 개의 actor-learner가 하나의 A2C를 글로벌신경망으로 두고 이를 비동기적으로 최적화하는 알고리즘이다. 각 actor-learner가 병렬로 동작하기 때문에 더 빠르게 학습을 진행할 수 있다. 이후 AC 구조를 이용한 알고리즘이 연구되었는데 대표적인 몇 가지 알고리즘을 소개한다.

DDPG [12]는 DPG [13]와 DQN을 결합한 model-free off-policy AC 알고리즘이고, DQN을 이산 행동 공간에서 연속 행동 공간으로 확장하면서 행동 공간이 stochastic하지 않고 deterministic한 정책을 가정한다. 기본적으로 Q 함수를 사용하지만 연속 행동 공간에서 최적의 행동을 찾기 어렵기 때문에 deterministic한 정책을 근사하는 정책 경사 알고리즘인 DPG를 기반으로 한다. 따라서 DDPG의 목표 함수는 DPG의 목표 함수와 유사하다.

$$J(\theta) = \int_S \rho^\pi(s) Q(s, \pi_\theta(s)) ds.$$

여기서 $\rho^\pi(s)$는 정책 π에 따른 상태 분포다. DDPG는 연속 행동 공간에 대한 우수한 확장성으로 인해 다양한 후속 연구가 진행되었다 [19][20].

SAC [14][18]는 DDPG와 유사한 model-free off-policy AC 알고리즘으로 off-policy의 데이터 효율성과 안정적인 수렴성을 목표로 개발되었다. 무작위로 행동하면서 학습이 안정적으로 수렴할 수 있는 정책을 만들기 위해 목표 함수는 정책의 엔트로피와 보상을 결합하는 형태를 가진다.

$$J(\theta) = \sum_{t=1}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_{\pi_\theta}}[r(s_t, a_t) + \alpha\mathcal{H}(\pi_\theta(\cdot|s_t))],$$

여기서 $\mathcal{H}(\cdot)$는 엔트로피 측정치(entropy measure)이고 α는 엔트로피의 중요도를 조절하는 temperature 파라미터이다.

정책 경사 알고리즘의 학습 안정성을 높이는 연구 중 하나로 TRPO [15]가 있다. TRPO는 정책을 최적화할 때 정책이 빈번하게 바뀌어 학습이 어려운 것을 줄이기 위해 정책의 변화 정도를 줄이는 방법을 사용한다. 이 변화 정도를 측정하는데 Kullback-Leibler (KL) divergence를 사용했고, 아래 목표 함수를 최대화하도록 최적화한다.

$$J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta_{old}}},\alpha\sim\beta}\left[\frac{\pi_\theta(a|s)}{\beta(a|s)} \hat{A}_{\theta_{old}}(s, a)\right].$$

여기서 θ_{old}는 업데이트 전의 정책 신경망 파라미터, $\rho^{\pi_{\theta_{old}}}$는 $\pi_{\theta old}$에 따른 상태 분포, $\beta(\alpha|s)$는 샘플을 모으는 행동 정책이고, $\hat{A}(\bullet)$은 추정한 advantage 함수이다. PPO [16][17]는 TRPO의 목표 함수를 clip함으로써 단순하게 만드는 동시에 TRPO의 장점을 승화시킨 알고리즘이며 목표 함수는 아래와 같다.

$$J^{CLIP}(\theta) = \mathbb{E}\left[\min\left(r_\pi(\theta)\hat{A}_{\theta_{old}}(s,a), \text{clip}(r_\pi(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{old}}(s,a)\right)\right],$$
$$r_\pi(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)},$$

여기서 $\text{clip}(r_\pi(\theta), 1-\epsilon, 1+\epsilon)$은 $r_\pi(\theta)$가 $[1-\epsilon, 1+\epsilon]$을 넘지 않도록 하는 함수이며 ϵ은 하이퍼파라미터이다.

2.4 DRL 응용 연구

DRL의 성공적인 연구로 인하여 최근 많은 후속 연구가 수행되고 있다. [24]에서는 로봇 매니퓰레이터의 제어 연구에 DRL 기법을 적용하였는데, DDPG와 D4PG (Distributed Distributional

Deterministic Policy Gradient) [23] 방식을 실험을 통해 비교하여 D4PG의 우수성을 입증하였다. [25]는 모델 불확실성과 시변 외란을 포함하는 n-링크 로봇 매니퓰레이터의 DDPG 기반 PI 제어기를 설계하여 시뮬레이션을 통해 검증하였다. [26]에서는 드론과 이에 장착된 매니퓰레이터를 동시에 제어하는 DDPG 기반의 제어 기법을 제안하여 실험을 통해 우수한 성능을 검증하였다. 이렇듯 최근 로봇 제어 연구에는 정책 경사 기반 알고리즘이 주로 사용되는데, 이는 정책 경사 기반 알고리즘의 연속 행동 결정 능력이 주요하게 작용한다.

한편, [27]에서 제안한 DQN 기반의 알고리즘이 Atari 게임에 적용되어 2017년에 DQN 기반 알고리즘의 SOTA를 달성하였다. 이 알고리즘은 DQN의 여러 개선 아이디어를 종합한 것으로 2017년 당시 DQN 기반 알고리즘 중 가장 우수한 성능을 달성하였다. [28]에서는 Atari 게임을 실행하는 RNN 기반의 RL 에이전트를 제안하였고, 특히 학습 과정에서 distributed prioritized experience replay를 새롭게 제안하여 이로부터 우수한 성능을 달성하였음을 증명하였다.

III. 드론 제어를 위한 DRL 연구

본 고에서는 쿼드로터 형태의 회전익 무인항공기를 드론으로 칭한다. 드론 제어 연구는 크게 드론의 안정적인 비행을 위한 안정화 제어 연구와 드론이 주어진 임무를 수행하기 위해 목적지까지 비행하고 장애물과 충돌을 방지하기 위한 경로를 생성하는 경로 계획 연구로 이루어진다. 본 장에서는 드론의 제어 문제에 DRL이 적용된 사례를 살펴본다.

3.1 드론의 안정화 제어

드론의 안정화 제어기 설계 문제는 높은 제어 주기의 요구, 비선형 동역학 방정식, 복잡한 모델 파라미터, 그리고 매우 불안정한 비선형 동역학 특성 등으로 인해 매우 도전적인 연구 분야이다. 이러한 어려움에도 불구하고 [29]에서는 actor-critic 기반의 DRL 기법을 새롭게 제안하였고, 이를 통해 드론 안정화 제어기를 설계하였다. Actor-critic 구조를 차용하여, 그림 3과 같이 상태 가치 함수를 출력하는 actor network와 네 개의 rotor의 추력값을 출력하는 policy network가 설계되었다. 각 network는 공통으로 자세각도, 3차원 위치, 그리고 각속도와 선형 가속도를 입력으로 취하였다. 또한, 일반적으로 드론의 안정화 제어에는 200Hz 이상의 높은 제어 주기를 요구하는데, DRL 기반 제어기를 저가의 임베디드 보드에서 200Hz 이상의 갱신 주기를 가지도록 구현하는 것은 매우 어려운 일이다. [29]에서는 Intel Atom

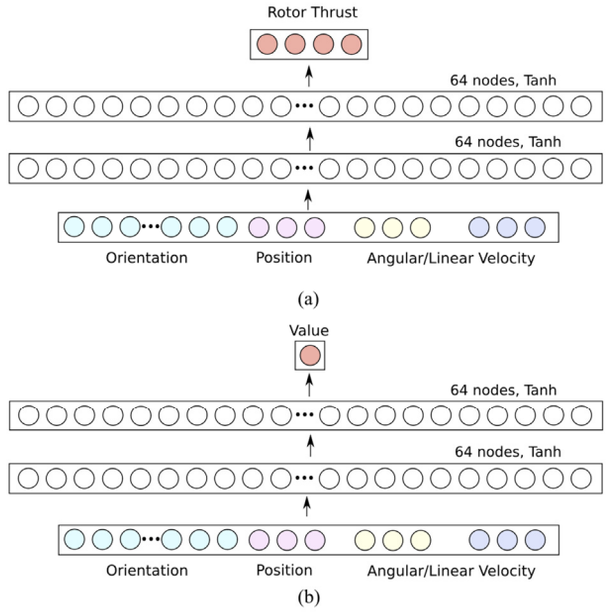

그림 3. [29]의 network 구조. (a) policy network. (b) value network.

그림 4. [29]의 실험 결과.

processor가 탑재된 약 50g 무게의 Intel compute stick을 이용하여 이를 해결하였다. 설계된 제어기는 드론을 효과적으로 안정화하였으며, 그림 4에서 볼 수 있듯이 심지어 드론을 뒤집은 채로 공중에 던졌을 때에도 안정적으로 제어가 가능하였다.

한편, 드론의 불안정한 동역학으로 인해 DRL의 초기 학습 시에는 드론이 바로 추락하여 충분한 time step 만큼 학습 데이터를 얻기 어렵다. 이는 초기 학습 수렴 시간을 지연시키는 요인으로 작용한다. 이에 [30]에서는 드론을 안정화하는 기본적인 제어기가 사전에 설계가 되어 있는 환경을 가정하였다. 이 상태에서 설계된 DRL 기반 제어기는 기본 제어기와 결합되어 모델 불확실성과 외란에 강인하게 드론을 제어하도록 학습된다. 전체 제어 시스템 구조는 그림 5와 같고, 그림에서 알 수 있듯이 사전에 설계된 기본 제어기와 제안하는 actor-critic 기반의 DRL 제어기가 결합되어 드론을 제어한다.

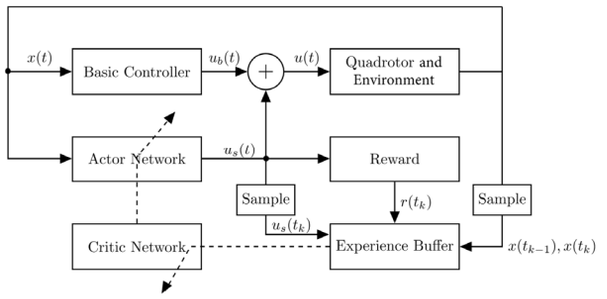

그림 5. [30]의 전체 제어 시스템 구조.

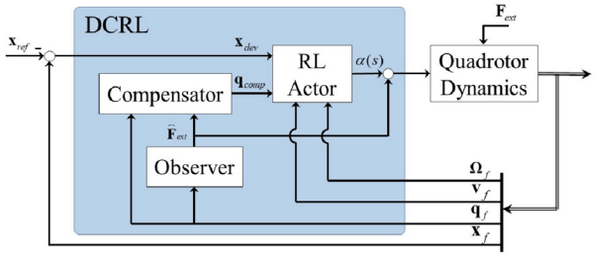

그림 6. [31]의 제어 시스템 구조.

최근에는 기존에 연구된 제어 기법을 DRL 기법에 적용하여 드론의 제어 성능을 강화하는 연구도 진행되고 있다. [31]에서는 기존에 연구된 고전적인 외란 관측기 설계 기법을 적용하여 드론에 인가되는 외란을 추정하는 외란 관측기를 설계하였다. 전체 제어 시스템 구조는 그림 6과 같고, 그림에 나타나 있듯이 추정된 외란은 보상기로 전달된다. 보상기로부터 보상된 자세각도 정보와 위치 추종 오차가 DRL의 에이전트로 전달되어 각 모터의 추력이 계산된다. DRL의 출력에 외란을 감쇠하기 위한 추력 보상값이 더해진다. 이렇게 함으로써 DRL 단독 운용에 비해 외란에 강인한 제어 성능을 확보하였음을 실험을 통해 검증하였다. [32]에서는 드론의 전체 동역학을 자세와 위치 제어 시스템으로 분할한 후, 각 부분 시스템에 대한 추종 제어기를 PPO 기반의 DRL 기법을 통해 설계하는 연구를 수행하였다. 또한, 상태 변수가 원하는 값을 추종하도록 선형 참조 모델을 설계하였다. 이를 통해 기존의 제어기 설계 기법에서 제어기만을 DRL로 수정하여 기존보다 체계적인 제어기 설계 및 성능 분석이 가능하도록 하였다.

3.2 드론의 경로 계획

드론의 무인 임무 수행을 위해 드론의 경로 계획 기법이 중요한 역할을 한다. 드론의 경로 계획 연구에서 드론의 동역학 시스템은 별도의 제어기에 의해 안정화되어 있다고 가정한다. 따라서 평가 지표에는 드론의 제어 성능은 포함되지 않으며, 드론이 계획된 경로를 따라 비행한다고 가정했을 때 얼마나 효율적으로 장애물과 충돌 없이 목표 위치에 도달할 수 있는가를 평가한다.

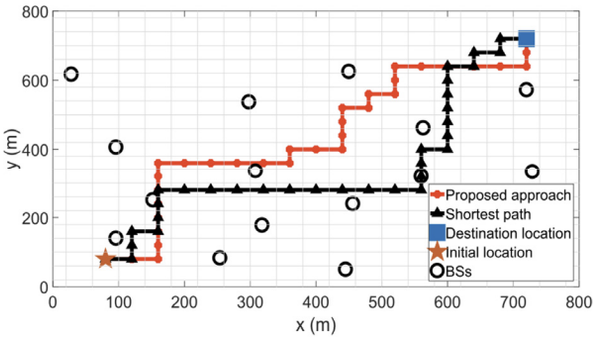

그림 7. [35]의 실험 결과.

[33]에서는 UAV의 도심지에서의 3차원 경로 계획을 위한 DDPG 기반의 알고리즘이 연구되었다. 이 연구에서 에이전트는 목적지까지의 거리 오차가 작아질 수록 큰 보상을 얻고, 장애물과 충돌했을 때 보상을 잃도록 환경을 구성하였다. [34]에서는 DQN 기반의 UAV의 충돌 회피 경로 계획 기법을 연구하였다. 기존 연구가 다양한 센서로부터 획득한 주변 장애물의 위치와 목적지까지의 위치 오차 등 환경에 대한 정보를 입력 받은 것과 달리, [34]에서는 드론이 주행 중 획득한 영상 이미지만이 입력으로 이용된다.

일반적으로 드론은 무선 통신을 통해 지상 통제 시스템과 연결된다. 무선 통신은 주변 지형과 장애물에 의해 통신 지연, 노이즈, 손실 등의 결함이 포함될 수 있는데, 이는 제어 성능의 하락 또는 기체의 충돌과 같은 심각한 문제를 야기한다. [35]에서는 이에 착안하여 경로 계획 과정에서 주변 지형 지물을 고려하여 통신 지연이 최소화되는 경로를 계획하는 연구를 수행하였고, 알고리즘은 ESN (Echo State Network) 기반의 DRL 기법을 적용하였다. 그림 7에서 볼 수 있듯이 단순히 최소한의 거리를 비행하는 것이 아니라 통신 지연이 최소화된 경로를 계획했음을 알 수 있다.

목적지까지의 비행 뿐만 아니라 [36]에서는 주어진 공간의 모든 부분을 커버하면서 전력 효율을 최대화하는 경로를 계획하는 CPP (Coverage Path Planning) 기법을 DDQN 기반의 DRL로 구현하였다. Network의 입력으로는 현재 공간의 지도, 드론의 위치, 커버리지가 주어지고, 드론의 이동 방위가 출력된다. 학습 시에는 드론의 제한된 전력량과 목표로 하는 커버리지를 균형 맞추어 최대한의 전력 효율로 목표를 달성하도록 보상이 설계되었다. 마지막으로 [36]에서는 실험을 통해 제안하는 방법의 우수성을 검증하였다.

기존 경로 계획 기법을 DRL에 적용하는 연구도 수행되었다. [37]에서는 지상 목표물을 추적하는 드론의 경로 계획 기법이 연

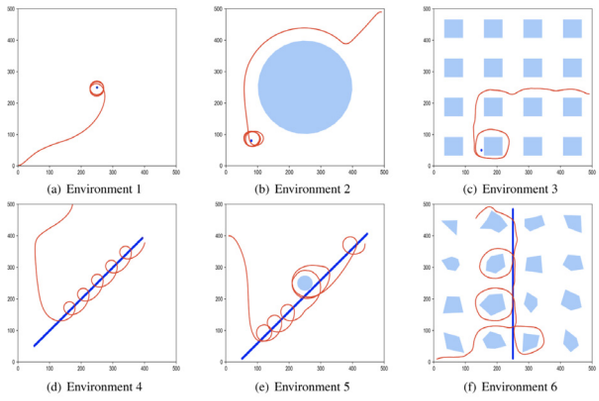

그림 8. [37]의 시뮬레이션 결과.

구되었다. 보상 함수 설계 시에 인공 포텐셜 장 [38] 기법을 적용하여 기존에 널리 연구된 경로 계획 연구를 DRL로 확장하였다. DDPG 기반의 DRL 구조를 차용하였는데, critic과 actor network 설계 시 LSTM (Long Short Term Memory) 구조를 적용하여 현재 학습 데이터와 과거 학습 데이터 모두를 이용하여 효과적인 학습이 이루어지도록 하였다. 시뮬레이션을 통해 제안하는 기법을 검증하였고, 그림 8과 같이 경로 계획 결과가 인공 포텐셜 장의 결과와 비슷함을 알 수 있다.

Ⅳ. 결론

본 기술특집 논문에서는 최근 주목받고 있는 DRL의 최신 연구 동향 및 드론 제어에의 적용 사례를 알아 보았다. 이를 위해 DRL의 두 가지 접근 방법인 가치 기반 기법과 정책 경사 기법의 연구를 비교 분석하였다. 이후, 드론의 안정화 제어와 경로 계획 기법에 DRL을 적용한 연구 사례를 심층적으로 살펴보았다. 본 고에서 살펴본 많은 연구들이 드론의 복잡한 비선형성과 모델 불확실성에 강인한 제어가 가능함을 보였으며, 특히 경로 계획 기법에서 기존 연구에 비해 발전된 성능을 냄을 보였다.

최신 연구 동향을 살펴 본 결과, 저자들은 현재 DRL 관련 연구가 태동기를 넘어 본격적인 실용화 단계에 진입했다고 생각한다. 따라서 향후 관련 연구가 더욱 활발히 이루어질 것이라 기대하며, 본 고의 분석 내용이 관련 연구자들에게 도움이 되기를 바란다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행한 연구임(No. NRF-2022R1I1A3071476).

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv:1312.5602.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwiese, I. Antonglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484-489, 2016.

[4] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," arXive:1509.06461.

[5] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, and N. D. Freitas, "Dueling network architectures for deep reinforcement leaning," arXiv:1511.06581.

[6] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," arXiv:1511.05952.

[7] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, "noisy networks for exploration," arXiv:1706.10295

[8] M. Hessel, J. Modayil, H. V. Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," arXiv:1710.02298.

[9] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems*, vol. 12, pp. 1057-1063, 1999.

[10] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," arXiv:1602.01783.

[11] H. C. Jang, Y. C. Huang, and H. A. Chiu, "A study on the effectiveness of A2C and A3C reinforcement learning in parking space search in urban areas problem," *International*

Conference on Information and Communication Technology Convergence, pp. 567-571, 2020.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv:1509.02971, 2015.

[13] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 1, pp. 387-395, 2014.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1861-1870, 2018.

[15] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889-1897, 2015.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv: 1707.06347, 2017.

[17] C. C.-Y. Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," arXiv: 2009.10897, 2020.

[18] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," arXiv:1812.05905, 2018.

[19] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," arXiv:1804.08617, 2018.

[20] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] S. Wang, D. Jia, and X. Weng, "Deep reinforcement learning for autonomous driving," arXiv:1811.11329v3.

[22] X. Xiong, J. Wang, F. Zhang, and K. Li, "Combining deep reinforcement learning and safety based control for autonomous driving," arXiv:1612.00147.

[23] G. B. Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," arXiv:1804.08617.

[24] C. Chu, K. Takahashi, and, M. Hashimoto, "Comparison of deep reinforcement learning algorithms in a robot manipulator control application," *2020 International Symposium on Computer, Consumer and Control*, pp. 284-287, 2020.

[25] P. Lu, W. Huang, J. Xiao, F. Zhou, and W. Hu, "Adaptive proportional integral robust control of an uncertain robotic manipulator based on deep deterministic policy gradient," *Mathematics*, vol. 9, no. 17, p. 2055, 2021.

[26] Y. C. Liu and C. Y. Huang, "DDPG-based adaptive robust tracking control for aerial manipulators with decoupling approach," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8258-8271, 2022.

[27] M. Hessel, J. Modayil, H. Van Hasselt, T. Schault, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," arXiv preprint arXiv:1710.02298.

[28] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney, "Recurrent experience replay in distributed reinforcement learning," *International Conference on Learning Representations*, 2021.

[29] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096-2103, 2017.

[30] X. Lin, Y. Yu, and C. Y. Sun, "Supplementary reinforcement learning controller designed for quadrotor UAVs," *IEEE Access*, vol. 7, pp. 26422-26431, 2019.

[31] C. H. Pi, W. Y. Ye, and S. Cheng, "Robust quadrotor control through reinforcement learning with disturbance compensation," *Applied Sciences*, vol. 11, p. 3257, 2021.

[32] Y. Byeon and H. S. Kim, "PPO-based model reference tracking control for a quadrotor UAV," *Journal of Institute of Control, Robotics and Systems*, vol. 28, no. 11, pp. 1022-1027, 2022.

[33] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud "Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach," *Proceedings of the 2020 IEEE International Symposium on Circuits and Systems*, 2020.

[34] A. Singla, S. Padakandla, and S. Bhatnagar, "Memory-based

deep reinforcement learning for obstacle avoidance in UAV with limited environment knowledge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 107-118, 2021.

[35] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVS: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125-2140, 2019.

[36] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "UAV coverage path planning under varying power constraints using deep reinforcement learning," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1444-1449, 2021.

[37] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064-29074, 2020.

[38] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 735-740, 2003.

[39] S. Y. Choi, P. T. Le, T. C. Chung, "Autonomous control of bicycle using deep deterministic policy gradient algorithm," *Convergence Security Journal*, vol. 18, no. 3, pp. 3-9, 2018.

저자약력

변 영 훈

- 2021년 국립 한국해양대학교 제어자동화공학부(공학사)
- 2021년~현재 한국해양대학교 제어계측공학과 석사과정 재학 중
- 관심 분야는 딥러닝, 컴퓨터 비전, 객체 인식, 심층 강화 학습

김 한 솔

- 2011년 한양대학교 전자컴퓨터공학부(공학사)
- 2012년 연세대학교 전기전자공학과(공학석사)
- 2018년 동대학교 전기전자공학과(공학박사)
- 2018년~2019년 삼성전자 무선사업부 책임연구원
- 2019년~현재 한국해양대학교 제어자동화공학부 부교수 재직 중
- 2020년~2021년 한국마린엔지니어링 학회 편집위원
- 관심분야는 지능로봇, 무인항공기, 지능제어, 딥러닝 기반 영상 처리, 심층 강화 학습