Check for updates

Trans. Korean Soc. Noise Vib. Eng., 36(5) : 539~551, 2026
https://doi.org/10.5050/KSNVE.2026.36.5.539

한국소음진동공학회논문집 제 36 권 제 5호, pp. 539~551, 2026
ISSN 1598-2785(Print), ISSN 2287-5476(Online)

강화학습 기반 단일 프로펠러 모듈형 UAV의 비행 및 결합 제어

Reinforcement Learning-based Flight and Coupling Control of a Single-propeller Modular UAV

김 현 민[*] · 손 정 우[†]
Hyeonmin Kim[*] and Jung Woo Sohn[†]

(Received September 29, 2026 ; Revised October 7, 2026 ; Accepted October 8, 2026)

Key Words : UAV(무인항공기), Modular Vehicle(모듈형 이동체), Reinforcement Learning(강화학습), Proximal Policy Optimization(근위 정책 최적화), Coupling(결합)

ABSTRACT

Unmanned vehicles are mobile platforms capable of autonomously performing missions across diverse domains, including ground, maritime, and aerial environments, by leveraging artificial intelligence without direct human intervention. This study proposes a cube-shaped modular unmanned aerial vehicle in which each unit consists of a single propeller and four vanes and can couple with other units in mid-air. Control policies were trained through reinforcement learning in NVIDIA Isaac Lab, a physics-based simulation environment. Hovering and point-to-point navigation of a single module were trained using proximal policy optimization (PPO), and the sequential mid-air coupling of three modules was trained using multi-agent PPO (MAPPO) under centralized training with a decentralized execution scheme. In the hovering scenario, the mean yaw rate decreased to approximately 0.02 rad/s, indicating that the reaction torque of the single propeller was compensated solely by vane deflection. In the navigation scenario, the module reached approximately 23 goals per episode, and in the coupling scenario, a docking success rate of approximately 90 % was achieved. These results demonstrate that attitude stabilization and mid-air coupling of the proposed platform can be achieved solely through reinforcement learning, without relying on conventional controllers such as the proportional–integral–derivative controller.

1. 서 론

무인이동체는 지상, 해양, 상공 등 여러 영역에서 감시 및 정찰, 운송, 재난 대응과 같은 다양한 임무를 수행할 수 있는 지능형 로봇이다. 무인이동체는 단순한 원격 조종 수준에 그치지 않고 인공지능을 기반으로 환경을 인식하고 상황을 판단하여 임무를 수행할 수 있다. 그중 상공에서 활용하는 무인항공기(unmanned aerial vehicle, UAV)는 멀티로터 구조를 채택하고 있으며, 멀티로터는 다수의 고출력 프로펠러가 산출하는 차동 추력을 통해 자세를 제어함으로

† Corresponding Author ; Member, School of Mechanical Engineering, Kumoh National Institute of Technology
E-mail : jwsohn@kumoh.ac.kr
* Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Graduate School, Kumoh National Institute of Technology

A part of this paper was presented at the KSNVE 2026 Annual Spring Conference
‡ Recommended by Editor Seung Hyun Jeong
ⓒ The Korean Society for Noise and Vibration Engineering

써 우수한 기동성과 정지 비행 안정성을 구현한다[1]. 그러나 단일 기체의 크기와 탑재 중량이 수행 가능한 임무 범위의 상한으로 작용하므로, 대형 페이로드나 광범위한 임무를 수행하기 위해서는 기체 자체를 대형화해야 한다는 확장성의 제약이 뒤따른다. 이러한 확장성 문제를 해결하기 위한 시도로, 다수의 비행 모듈을 공중에서 결합하여 하나의 구조체를 구성하는 모듈형 UAV 플랫폼 연구가 이미 다수 진행되어 왔다. Saldana et al.이 제안한 ModQuad는 쿼드로터 모듈들이 큐빅 프레임의 수직면을 맞대어 영구자석으로 결합함으로써 공중에서 자율적으로 결합하는 최초의 모듈형 UAV 구조체를 구현하였으며, 결합된 구조 전체를 안정화하기 위한 분산형 자세 제어기를 함께 제시하였다[2]. 이후 Li et al.의 ModQuad-Vi는 비전 기반 도킹 방법을 통해 외부 모션 캡처 인프라 의존성을 제거하였고[3], Sugihara et al.이 제안한 TRADY는 틸트로터 기반 모듈에 자율적인 결합 및 분리 기능을 부여하여 임무 중 구조 변경이 가능하도록 확장하였다[4]. 한편 Oung and D'Andrea의 distributed flight array는 단일 프로펠러로 구성된 모듈들이 지상에서 결합한 뒤 협조 비행하는 구조를 제안하였으며[5], 최근 Garanger et al.은 단일 고정 피치 프로펠러를 갖는 정십이면체 모듈들을 3차원 형상으로 결합하여 다양한 비행 구성을 구현하였다[6]. 그러나 ModQuad 계열과 TRADY의 각 모듈은 쿼드로터 혹은 틸트로터 기반이며, distributed flight array의 모듈은 단일 로터의 반작용 토크를 상쇄할 수 없어 단독으로는 비행하지 못하고 결합된 상태에서만 비행한다. 또한 이들 연구는 결합된 구조체의 동역학 모델을 바탕으로 한 제어기와 추력 분배 기법을 사용한다. 이에 비해 이 연구의 모듈은 베인을 이용해 반작용 토크를 상쇄함으로써 단일 프로펠러만으로 단독 비행이 가능하고, 독립적으로 비행하던 모듈들이 공중에서 결합하며, 단독 비행부터 결합까지의 전 과정을 모델 기반 제어기 대신 강화학습 정책으로 제어한다는 점에서 기존 연구와 차별성을 갖는다. 이 연구는 추력 생성과 자세 제어의 기능을 물리적으로 분리한 단일 프로펠러 기반 모듈형 UAV 플랫폼을 제안한다. 제안된 플랫폼은 하나의 고출력 프로펠러가 추력을 전담하고, 4개의 저출력 제어 베인이 프로펠러 후류를 편향시켜 독립적으로 자세를 제어하는 구조로 구성된다. 이는 다수의 프로펠러 간 추력 크기 차이로 모멘트를 만드는 차동 추력 방식과 달리, 단일 기류의 방향을 편향시켜 모멘트를 만드는 추력 벡터 제어 방식에 해당한다. 베인에 의한 모멘트는 베인의 편향각 뿐만 아니라 프로펠러 후류의 속도 분포, 로터-베인 간 공력 간섭 등 다수의 비선형 요인에 의해 결정되며[7,8], 이러한 관계를 명시적인 해석적 모델로 정확히 유도하는 것은 매우 복잡하다. 전통적인 제어 기법을 적용하기 위해서는 이러한 비선형 공력 관계를 사전에 정밀하게 모델링하거나 광범위한 시스템 식별 절차를 거쳐야 하는데, 모델링 오차는 곧 제어 성능 저하로 이어질 수 있다.

이 연구에서는 이러한 한계를 극복하기 위해 단일 프로펠러와 4개의 베인으로 구성된 모듈형 UAV에 강화학습 기반 제어를 적용하여, 단일 모듈의 비행 제어와 다중 모듈의 공중 결합을 수행한다. 강화학습은 명시적인 동역학 모델 없이 시뮬레이션 환경과의 상호작용을 통해 제어 정책을 학습할 수 있어 쿼드로터의 자세 및 위치 제어에서 그 유효성이 확인되었으며[9,10], 비선형 공력 관계를 직접 학습해야 하는 이 플랫폼에 적합하다. 최근에는 강화학습으로 학습한 비행 정책이 드론 레이싱에서 인간 챔피언 수준의 성능을 달성하였고[11], Isaac Sim 기반의 GPU 병렬 시뮬레이션을 활용한 드론 강화학습 플랫폼이 제안되는 등 관련 연구가 활발히 진행되고 있다[12,13]. NVIDIA Isaac Lab 기반의 물리 시뮬레이션 환경에서 단일 모듈의 비행 제어는 PPO(proximal policy optimization) 알고리즘으로, 다중 모듈의 결합은 MAPPO(multi-agent PPO) 알고리즘으로 학습한다. 학습된 정책을 통해 별도의 고전적 제어기 없이 베인의 편향각 조절만으로 자세 안정화와 모듈 간 공중 결합이 가능함을 확인한다.

2. 시뮬레이션 구성

2.1 PPO 알고리즘

강화학습(reinforcement learning, RL)은 에이전트가 환경과의 반복적인 상호작용을 통해 시행착오로부터 스스로 최적의 행동 전략을 학습하는 기계학습 방법론이다[14]. 지도학습(supervised learning)이 정답 레이블이 주어진 데이터로부터 입력-출력 매핑을 학습하는 것과 달리, 강화학습은 에이전트가 취한 행동에 대한 보상 신호만을 피드백으로 받으며, 이 보상을 누적적으로 최대화하는 행동 정책을 스스로 탐색해야 한다는 점에서 차이가 있다. 강화학습 문제는

일반적으로 마르코프 결정과정(markov decision process, MDP)으로 모델링된다. MDP는 상태공간 S, 행동공간 A, 상태 전이 확률 $P(s_{t+1} \mid s_t, a_t)$, 보상함수 $r(s_t, a_t)$, 할인율 $\gamma \in [0, 1]$로 구성된다. 에이전트는 매 시간 스텝 t마다 현재 상태 s_t를 관측하고, 정책 $\pi_\theta(a_t \mid s_t)$에 따라 행동 a_t를 확률적으로 선택한다. 환경은 이 행동에 대해 다음 상태 s_{t+1}과 보상 r_t를 반환하며, 이러한 상호작용이 에피소드 종료 시점까지 반복된다. 학습의 목표는 기대 누적 할인 보상을 최대화하는 정책 파라미터 θ를 찾는 것이다. 목적함수 $J(\theta)$는 식 (1)과 같이 정의된다.

$$J(\theta) = E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{1}$$

여기서, $\tau = (s_0, a_0, s_1, a_1, \cdots)$는 정책 π_θ를 따라 생성된 궤적(trajectory)을 의미한다. 이러한 목적함수를 직접 최대화하는 대표적인 방법이 정책 경사(policy gradient) 방법으로, 정책 파라미터에 대한 목적함수의 그래디언트를 추정하여 경사 상승법으로 정책을 직접 갱신한다. 정책 경사 방법은 연속적인 행동 공간에도 자연스럽게 적용할 수 있어, 이 연구와 같이 추력 스케일이나 베인 편향각처럼 연속값으로 표현되는 제어 문제에 적합하다. 하지만 순수한 정책 경사 방법은 그래디언트 추정의 분산이 크고, 한 번의 갱신 폭이 과도할 경우 정책 성능이 급격히 저하될 수 있다는 학습 불안정성 문제를 갖는다. 이러한 한계를 보완하기 위해 제안된 알고리즘이 PPO이며[15], 이 연구는 이를 정책 학습 알고리즘으로 채택한다.

한 번의 정책 갱신 폭이 지나치게 크면 성능이 급격히 저하되는 정책 붕괴 문제가 발생할 수 있다. PPO는 이를 완화하기 위해, 이전 정책 $\pi_{\theta\text{old}}$과 새 정책 π_θ간의 확률비(probability ratio)를 이용한 대리 목적 함수(surrogate objective)를 사용한다. 확률비는 식 (2)와 같이 정의된다.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{2}$$

$\theta = \theta_{\text{old}}$일 때 $r_t(\theta_{\text{old}})$=1이 되며, 이 값이 1로부터 멀어질수록 새 정책이 이전 정책으로부터 크게 벗어났음을 의미한다. 이 확률비를 이용한 가장 단순한 대리 목적함수는 식 (3)과 같이 정의된다.

$$L^{\text{CPI}}(\theta) = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] \tag{3}$$

여기서, $\hat{A}_t$는 시점 t에서의 어드밴티지(advantage) 추정값으로, 해당 상태에서 취한 행동이 평균적인 행동보다 얼마나 더 좋았는지를 나타낸다. 그러나 이 목적함수를 아무 제약 없이 최대화할 경우 $r_t(\theta)$가 1로부터 지나치게 멀어지는 방향으로 정책이 갱신될 수 있으며, 이는 앞서 언급한 정책 붕괴 문제로 이어질 수 있다. PPO는 이를 방지하기 위해 확률비를 목적함수 내에서 직접 클리핑(clipping)하는 식 (4)와 같은 목적함수를 제안한다.

$$\begin{aligned} &L^{\text{CLIP}}(\theta) \\ &= \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon)\hat{A}_t\right)\right] \end{aligned} \tag{4}$$

이 식에서 $\text{clip}(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon)$은 확률비 $r_t(\theta)$를 $[1-\epsilon,\ 1+\epsilon]$ 구간으로 제한하는 연산이며, ϵ은 이 허용 범위의 폭을 결정하는 하이퍼파라미터이다. 이러한 클리핑 메커니즘은 별도의 제약 최적화 절차 없이도 1차 미분만으로 구현 가능하며, 다양한 문제에 적용 가능하다는 실용적 장점 덕분에 연속 제어 강화학습 문제에서 가장 널리 사용되는 알고리즘 중 하나이다. PPO 알고리즘의 학습 구조와 흐름을 Fig. 1에 나타내었다. Actor-critic 구조를 바탕으로 하는 정책 경사 알고리즘이며, 온폴리시 방식으로 정책을 업데이트 하게 된다.

2.2 학습 시나리오

이 연구는 우선 단일 모듈 수준에서 두 가지 기초 비행 정책을 학습한다. 하나는 제자리에서 자세와 고도를 유지하는 호버링(hovering) 정책이며, 다른 하나는 임의로 부여되는 목표 지점까지 안정적으로 도달하는 포인트-투-포인트 내비게이션(point-to-point navigation) 정책이다. 두 정책은 호버링 정책이 먼저 안정적인 제자리 비행 성능을 확보한 이후에 내비게이션 정책을 학습하는 순서로 진행하였다. 이때 내비게이션 정책은 호버링 정책의 학습된 가중치를 초기값으로 사용하지 않고, 별도로 처음부터 학습하였다. 일반적인 쿼드로터는 4개의 로터 중 2개씩 서로 반대 방향으로 회전함으로써 각 로터의 반작용 토크가 상쇄되어, 기체가 요(yaw) 방향으로 회전하지 않도록 설계된다. 반면 이 연구가 제안하는 플랫폼은 단일

프로펠러만으로 추력을 발생시키므로, 반작용 토크를 상쇄할 별도의 로터가 존재하지 않아 기체가 요 방향으로 회전하려는 경향을 갖는다. Fig. 2에 나타난 바와 같이, 이 연구에서는 이러한 요 방향 회전을 억제하기 위해, 베인의 편향각을 조절하여 프로펠러 후류의 방향을 국소적으로 편향시킴으로써 반작용 토크에 대응하는 모멘트를 발생시키는 것을 목표로 한다.

포인트-투-포인트 내비게이션 정책의 목표는 매 에피소드 동안 연속적으로 주어지는 목표 지점들을 순차적으로 추종하는 것이다. 에피소드는 하나의 목표 지점으로 종료되지 않으며, 기체가 현재 목표 지점에 도달하면 새로운 목표 지점이 재부여되므로, 정책은 하나의 목표 지점까지 도달하는 능력뿐 아니라 도달 직후 다음 목표를 향해 방향을 재설정하고 다시 가속하는 연속적인 목표 추종 능력을 학습해야 한다. 이 연구가 제안하는 플랫폼은 베인을 이용해 롤 및 피치 방향의 모멘트를 발생시켜 기체를 원하는 방향으로 기울이고, 목표 지점에 가까워짐에 따라 자세를 조정해 감속하되, 다음 목표가 새로 주어지는 시점에 맞추어 다음 방향으로 다시 기울여야 한다. 이 과정에서 베인은 호버링 시나리오와 동일하게 요구되는 요 방향 반작용 토크 상쇄를, 목표가 바뀔 때마다 반복되는 방향 전환을 위한 롤 및 피치 모멘트 생성과 동시에 수행해야 한다.

단일 모듈 수준에서 기초 비행 정책 학습이 완료되면 3대의 모듈이 결합하는 시나리오를 학습한다. 결합 시나리오의 목표는 모듈 3대가 각각 독립된 위치에서 출발하여, 하나의 일자형 복합 비행체로 결합하는 것이며, Fig. 3에 나타내었다. 하나의 중심 모듈이 호버링을 진행하면 두 개의 추종 모듈이 중심 모듈에 접근한다. 최종적으로 상대 거리, 상대 속도, 자세, 각속도가 모두 허용 범위 이내로 안정되었을 때에만 결

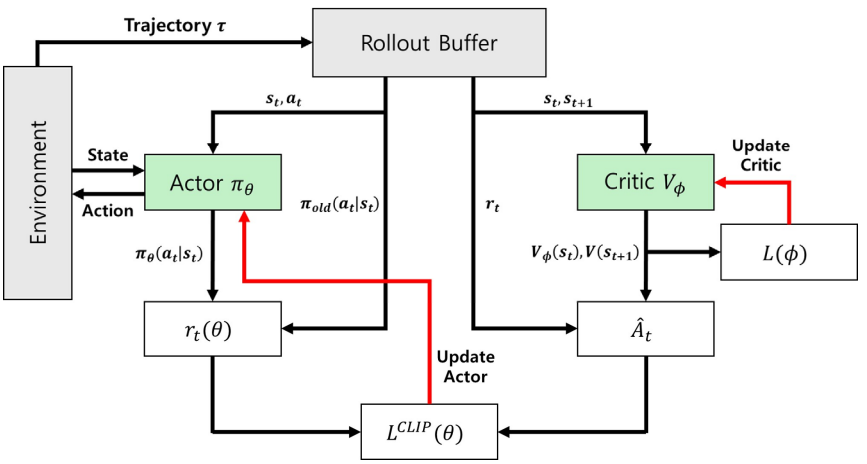


Fig. 1 Architecture for PPO algorithm

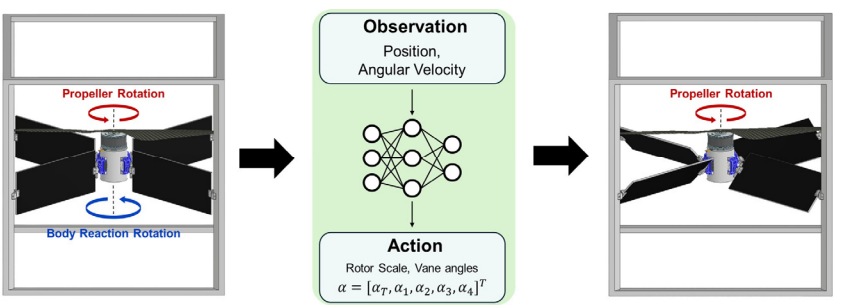


Fig. 2 Single module scenario

합이 성립하므로, 정책은 단순히 목표 방향으로 다가가는 것을 넘어 접근 속도와 상대 자세를 정밀하게 제어해야 한다. 두 추종 모듈은 서로 다른 시점에 독립적으로 결합할 수 있으며, 결합이 일어날 때마다 복합체의 질량과 관성이 불연속적으로 증가한다. 중심 모듈 및 이미 결합된 모듈은 이러한 변화가 발생할 때마다 달라진 동역학에 맞추어 복합체의 자세를 다시 안정화해야 하며, 아직 결합하지 않은 추종 모듈은 계속 목표 지점으로 접근을 이어가야 한다. 따라서 결합 시나리오의 정책은 개별 모듈의 접근 제어와, 결합 시점마다 반복되는 복합체의 재안정화라는 두 가지 목표를 학습한다.

2.3 학습 환경

이 연구의 전체 학습은 NVIDIA Isaac Sim 기반의 물리 시뮬레이션 환경과 Isaac Sim을 기반으로 강화학습 파이프라인을 제공하는 프레임워크인 Isaac Lab을 통해 수행된다[16]. Isaac Sim은 PhysX 물리 엔진을 기반으로 동역학 및 충돌 계산을 GPU를 활용하여 수행하는 시뮬레이터이며, 수천 개의 환경을 복제하여 동시에 병렬 학습이 가능하다는 점에서 강화학습과 같이 방대한 양의 데이터를 요구하는 학습 방식에 적합하다.

Isaac Sim 상의 모듈 모델은 질량 1.47 kg의 본체와 회전 관절로 연결된 4개의 베인으로 이루어진 다물체 모델이다. 중력, 강체 운동, 관절 구동 및 모듈 간 충돌은 PhysX 물리 엔진이 계산하며, 프로펠러 추력과 베인 공력은 매 시뮬레이션 스텝마다 계산되어 본체에 외력과 외부 모멘트로 부가된다. 프로펠러 추력은 정책의 추력 명령에 따라 호버링 추력 T_{hover}의 0.1배에서 2.5배 범위에서 결정되며, 이는 사용 예정인 모터(T-motor MN4110 KV400, 0.406 m 프로펠러 기준 최대 추력 약 37.3 N)와 모듈 질량으로부터 얻은 추력 대 중량비 약 2.59 이내의 범위이다. 프로펠러 회전에 의한 반작용 토크는 추력에 비례하는 것으로 모델링하였으며, 반작용 토크 계수비는 k_q/k_f= 0.02로 설정하였다. 베인에 작용하는 공력은 프로펠러 후류 속도 v를 이용하여 $F=0.5\rho v^2 S C_L \sin\delta$로 계산하였으며, 공기 밀도 ρ는 1.225 kg/m^3, 베인 면적 S는 0.005 m^2, 양력 계수 C_L은 2π로 설정하였다. 여기서 δ는 베인의 편향각이며, 베인 공력에 베인의 모멘트 암(0.112 m)을 곱하여 기체에 작용하는 자세 제어 모멘트를 산출하였다.

제안 플랫폼의 주요 물리 조건과 시나리오별 학습 조건을 Table 1에 정리하였다. 에피소드는 지면 추락, 과도한 상승, 허용 범위를 초과하는 기울기 및 수평 이탈이 발생하면 조기 종료되며, 호버링 시나리오는 제자리 유지가 목적이므로 기울기와 수평 이탈의 허용 범위를 다른 시나리오보다 엄격하게 설정하였다.

단일 모듈 수준에서 학습한 두 정책의 행동 공간과 관측 공간은 Table 2와 Table 3과 같이 설정하였다.

두 정책은 동일한 관측 및 행동 공간 구조를 갖고 있지만 목표 지점을 나타내는 상대 위치는 시나리오별로 차이가 있다. 호버링 시나리오의 경우 호버링 지점에 대한 상대 위치를, 내비게이션 정책에서는 매 에피소드 부여되는 목표 지점에 대한 상대 위치를 나타낸다. 두 정책의 보상 함수는 공통적으로 생존, 자세 안정성이라는 요소를 가지되, 시나리오의 성격에 따라 세부 항목과 함수 형태를 다르게 설계하였다. 두 시나리오의 보상 함수는 Table 4와 Table 5와 같이 설계하였다.

호버링 보상의 각 항목은 모두 tanh 또는 exp 함수

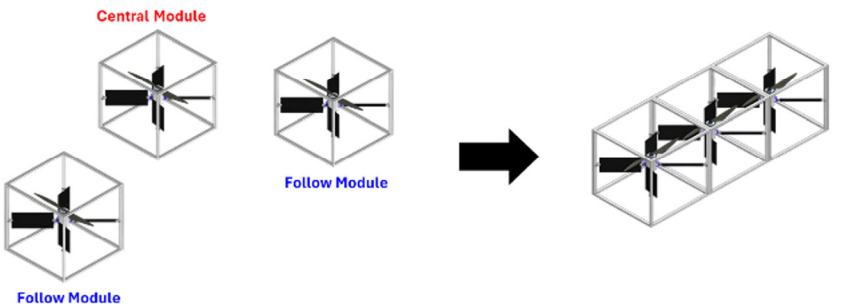


Fig. 3 Multiple module scenario

를 통해 [0, 1] 범위로 유계되도록 설계하였다. 이는 특정 보상 항목이 큰 오차 상황에서 비정상적으로 큰 값을 가져 다른 항목의 기여를 압도하는 것을 방지하기 위함이다. 자세 관련 보상은 attitude와 attitude penalty의 두 항으로 구성되며, 서로 다른 함수 형태를 사용한다. Attitude는 자세 오차가 0에 가까울수록 급격히 커지는 지수함수 형태로, 수평 자세를 정밀하게 유지하도록 유도하는 역할을 한다. 반면 attitude penalty는 오차가 커질수록 완만하게 포화되는 tanh 함수 형태로, 일시적으로 자세가 크게 흐트러지는 상황에서도 벌점이 과도하게 커져 학습이 불안정해지는 것을 방지하는 역할을 한다.

내비게이션 보상은 호버링 보상 구조를 기반으로 하되, 목표 지점으로의 접근을 유도하는 goal progress와 목표 도달 시 부여되는 goal reached가 추가된 형태이다. Goal progress는 이전 스텝 대비 목표까지의 거리 감소량에 비례한 보상을 매 스텝 제공하여 목표를 향한 지속적인 이동을 유도하며, goal reached는 목표 지점(반경 0.2 m 이내) 도달 시 이산적인 추가 점수를 제공한다. 목표에 도달하더라도 에피소드는 종료되지 않고 새로운 목표 지점이 즉시 재부여되므로, goal reached는 에피소드당 여러 차례 반복적으로 획득할 수 있다. 자세 관련 보상과 행동 변화 페널티는 호버링과 동일한 함수 형태를 사용하는데, 이는 내비게이션 정책 역시 이동 중 요 방향 반작용 토크 상쇄와 급격한 행동 변화 억제라는 동일한 저수준 제어 목표를 공유하기 때문이다. 두 정책의 학습에는 PPO 알고리즘을 사용하였으며, 정책 및 가치 네트워크는 모두

Table 1 Platform and training conditions

Parameter	Hovering	Navigation	Coupling
Module mass	1.47 kg		
Module size	410 × 410 × 410 mm		
Vane deflection range	±45°		
Control period	20 ms	20 ms	20 ms
Episode length	10 s	30 s	30 s
Parallel environments	8192	4096	512
Termination: altitude	< 0.05 m or > 5.0 m		
Termination: tilt	> 60°	> 75°	> 75°
Termination: lateral	> 0.3 m	> 3.0 m	> 5.0 m

Table 2 Action space (hovering, navigation)

Component	Dimension	Description
Rotor scale	1	Propeller thrust control (α_T)
Vane angle	4	Vane tilt angle control ($\alpha_1 \sim \alpha_4$)

Table 3 Observation space (hovering, navigation)

Component	Dimension	Description
Relative position	3	Target position in body frame
Linear velocity	3	Linear velocity in body frame
Projected gravity	3	Gravity vector in body frame
Angular velocity	3	Angular velocity in body frame
Vane position	4	Deflection angle of each vane
Vane velocity	4	Angular velocity of each vane

[256, 128, 64] 크기의 완전연결 은닉층으로 구성하였다. Table 6은 두 시나리오에 사용된 하이퍼파라미터를 정리한 것이다.

결합 시나리오는 3개의 모듈이 각각 독립적인 정책에 따라 동작하며 서로 상호작용해야 하므로 단일 에이전트 환경과는 다른 학습 구조를 필요로 한다. 이 연구는 이를 다중 에이전트 강화학습 문제로 정식화하고, 중심화 학습 및 분산화 실행(centralized training with decentralized execution, CTDE) 구조를 갖는 MAPPO를 적용하였다[17,18]. CTDE 구조에서 각 에이전트의 정책 네트워크는 자신의 국소 관측만을 입력받아 독립적으로 행동을 결정하는 반면, 가치 네트워크는 학습 과정에서 전체 시스템의 상태 정보를 입력받아 가치를 추정한다. 이는 학습 시에는 전역 정보를 활용하여 안정적인 가치 추정을 수행하되, 실행 시에는 각 모듈이 자신의 관측만으로 동작할 수 있도록 하기 위함이다. 각 에이전트의 관측 공간은 단일 모듈의 20차원 관측에 다른 모듈에 대한 상대 위치 정보가 추가된 23차원으로 구성되며, Table 7에 관측 공간을 정리하여 나타내었다.

단일 모듈 시나리오에서 목표 지점에 대한 상대 위치를 나타내던 항목은 결합 시나리오에서는 해당 모듈이 결합해야 할 목표 슬롯에 대한 상대 위치를 나타낸다. 여기에 가장 가까운 다른 모듈에 대한 상대 위치가 추가되어, 각 모듈이 근접한 다른 모듈의 존재를 인지할 수 있도록 하였다. 관측 공간에는 중심 모듈과의 상대 자세가 명시적으로 포함되지 않으나, 목표 슬롯의 상대 위치가 기체 좌표계로 표현되므로 모듈이 요 방향으로 회전하면 그 값도 함께 변하여 요 방향 자세에 대한 정보가 간접적으로 반영된다. 또한 학습 시 가치 네트워크에는 모든 모듈의 자세를 포함한 전역 상태가 입력되며, 결합 면의 정렬은 align 보상과 결합 조건의 자세 오차 기준을 통해 유도된다. 행동 공간은 단일 모듈과 동일하다.

Table 4 Hovering reward function

Component	Weight	Function	
Alive	+1.0	$\begin{cases}1 & \text{Survive}\\ 0 & \text{Terminate}\end{cases}$	(5)
Height	+15.0	$1-\tanh\left(\frac{\lvert z-1.0\rvert}{0.5}\right)$	(6)
Position	+10.0	$1-\tanh\left(\frac{\lVert p-p_{\text{goal}}\rVert}{0.8}\right)$	(7)
Yaw stability	+15.0	$1-\tanh\left(\frac{\lvert\dot{\psi}\rvert}{10}\right)$	(8)
Yaw rate	-10.0	$\tanh\left(\frac{\lvert\dot{\psi}\rvert}{5}\right)$	(9)
Attitude	+3.0	$\exp\left(-\frac{\phi_r^2+\theta_p^2}{0.04}\right)$	(10)
Attitude penalty	-2.0	$\tanh\left(\frac{\sqrt{\varphi_r^2+\varphi_p^2}}{0.5}\right)$	(11)
Action rate	-0.01	$\tanh(\lVert\Delta a_t\rVert)$	(12)

Table 5 Navigation reward function

Component	Weight	Function	
Alive	+5.0	$\begin{cases}1 & \text{Survive}\\ 0 & \text{Terminate}\end{cases}$	(13)
Goal progress	+30.0	$\tanh\left(\frac{d_{t-1}-d_t}{0.1}\right)$	(14)
Goal reached	+30.0	$\begin{cases}1 & d_t<0.2\\ 0 & \text{otherwise}\end{cases}$	(15)
Yaw stability	+15.0	$1-\tanh\left(\frac{\lvert\dot{\psi}\rvert}{10}\right)$	(16)
Yaw rate	-10.0	$\tanh\left(\frac{\lvert\dot{\psi}\rvert}{5}\right)$	(17)
Attitude	+2.0	$\exp\left(-\frac{\phi_r^2+\theta_p^2}{0.04}\right)$	(18)
Attitude penalty	-0.5	$\tanh\left(\frac{\sqrt{\phi_r^2+\theta_p^2}}{0.5}\right)$	(19)
Action rate	-0.01	$\tanh(\lVert\Delta a_t\rVert)$	(20)

Table 6 PPO hyperparameters

Component	Hovering	Navigation
Network architecture	[256, 128, 64]	[256, 128, 64]
Learning rate	0.0001	0.00001
Entropy coefficient	0.01	0.005
Gradient norm clip	0.5	1.0
Rollouts	64	64
Discount factor	0.99	0.99
GAE	0.95	0.95
Ratio clip	0.2	0.2
Total training steps	500 000	2 000 000

Table 8에 결합 시나리오의 보상을 정리하여 나타내었다. 결합 시나리오의 보상은 모듈별로 개별 계산되며, 목표 슬롯으로의 접근 및 정렬을 유도하는 항목과 결합 성립 시의 보너스, 그리고 근접 상황에서의 안전성을 위한 페널티 항목으로 구성된다. Approach와 align은 각각 목표 슬롯까지의 거리 감소량과 자세 오차 감소량에 비례한 보상을 제공하여, 위치와 자세를 동시에 목표 상태로 수렴시키도록 유도한다. 여기서 식 (22)의 d_t는 시점 t에서 모듈과 가장 가까운 미점유 목표 슬롯 사이의 거리이며, 식 (23)의 e_t는 중심 모듈의 자세를 기준으로 정의된 목표 자세와 모듈 자세 사이의 회전 각도 오차이다. Dock bonus는 결합이 최초로 성립하는 시점에 한 번만 부여되는 이산적 보상이다. Linear velocity와 mutual approach는 근접 상황에서의 안전성을 위한 페널티 항목이다. Linear velocity는 목표 슬롯 3.0 m 이내에서 속도에 비례한 페널티를 부여하여 감속된 접근을 유도하며, mutual approach는 다른 모듈로부터 0.6 m 이내에 있을 때 해당 모듈을 향하는 속도 성분에 비례한 페널티를 부여하여 과도하게 빠른 상호 접근을 억제한다. 결합 시나리오의 학습 하이퍼파라미터는 Table 9와 같이 설정하였다.

결합은 목표 슬롯까지의 상대 거리 0.15 m 미만, 상대 속도 0.15 m/s 미만, 목표 자세 대비 각도 오차 0.35 rad(약 20°) 미만, 상대 각속도 1.0 rad/s 미만의 네 가지 조건을 동시에 만족할 때 성립하는 것으로 정의하였다. 목표 슬롯은 중심 모듈로부터 좌우 방향으로 모듈 한 변의 길이인 0.41 m만큼 떨어진 위치이며, 중심 모듈의 위치와 자세를 따라 함께 이동한다. 결합이 성립하면 해당 추종 모듈은 물리 엔진의 고정 조인트를 생성하는 대신, 매 물리 시뮬레이션 스텝마다 중심 모듈 기준의 목표 위치와 자세로 보간하여

Table 7 Observation space (coupling)

Component	Dimension	Description
Target slot relative position	3	Target slot position in body frame
Linear velocity	3	Linear velocity in body frame
Projected gravity	3	Gravity vector in body frame
Angular velocity	3	Angular velocity in body frame
Vane position	4	Deflection angle of each vane
Vane velocity	4	Angular velocity of each vane
Nearest module relative position	3	Position of the nearest module in body frame

Table 8 Coupling reward function

Component	Weight	Function	
Alive	+1.0	$\begin{cases}1 & \text{Survive}\\ 0 & \text{Terminate}\end{cases}$	(21)
Approach	+20.0	$\tanh\left(\frac{d_{t-1}-d_t}{0.1}\right)$	(22)
Align	+20.0	$\tanh\left(\frac{e_{t-1}-e_t}{0.1}\right)$	(23)
Dock bonus	+50.0	$\begin{cases}1 & \text{Success}\\ 0 & \text{Otherwise}\end{cases}$	(24)
Yaw stability	+8.0	$1-\tanh\left(\frac{\lvert\dot{\psi}\rvert}{10}\right)$	(25)
Yaw rate	-5.0	$\tanh\left(\frac{\lvert\dot{\psi}\rvert}{5}\right)$	(26)
Attitude	+1.0	$\exp\left(-\frac{\phi_r^2+\theta_p^2}{0.04}\right)$	(27)
Attitude penalty	-0.2	$\phi_r^2+\theta_p^2$	(28)
Linear velocity	-0.5	$\min(\lVert v_{\text{body}}\rVert,\ 15.0)\cdot 1[d<3.0]$	(29)
Mutual approach	-3.0	$\text{clamp}\left(v_{\text{own}}\cdot\hat{r},\ 0,\ 15.0\right)\cdot 1[d_{\text{other}}<0.6]$	(30)
Action rate	-0.01	$\lVert\Delta a_t\rVert^2$	(31)

Table 9 MAPPO hyperparameters

Component	Coupling
Network architecture	[256, 128, 64]
Learning rate	0.00005
Entropy coefficient	0.005
Gradient norm clip	1.0
Rollouts	64
Discount factor	0.99
GAE	0.95
Ratio clip	0.2
Total training steps	2 000 000

고정하는 방식으로 중심 모듈에 연결된다. 또한 결합 전 접근 과정에서는 목표 슬롯으로부터 0.05 m 이내에서 작용하는 인력을 부여하여 최종 정렬을 보조하였다.

3. 시뮬레이션 결과

3.1 학습 결과 및 분석

학습 결과는 각 시나리오별로 공통적으로 학습 스텝에 따른 평균 총 보상 그래프, Isaac Lab 시뮬레이션 환경에서 학습 전후 비행 모습 비교, 해당 시나리오에 가장 중요한 성능 지표를 함께 나타내었다. 총 보상은 여러 보상 항목의 가중합이므로 학습의 수렴 여부를 판단하는 데에는 유용하지만, 각 시나리오가 실제로 달성하고자 하는 제어 목표가 충족되었는지를 직접적으로 보여주지는 못한다. 이에 따라 호버링 시나리오는 요 방향 각속도의 변화를, 내비게이션 시나리오는 에피소드당 목표 지점 도달 횟수를, 결합 시나리오는 도킹 성공률을 각각 핵심 지표로 함께 제시하였다.

호버링 정책의 학습 결과를 Fig. 4 ~ Fig. 7에 나타내었다. Fig. 4에서 호버링 시나리오의 평균 총 보상은 학습이 진행됨에 따라 꾸준히 상승한 뒤 수렴하였다. Fig. 5와 Fig. 6은 학습 전후 정책의 비행 모습을 비교한 것으로, 학습이 완료되기 전에는 반작용 토크를 상쇄하지 못해 기체가 제자리에서 요 방향으로 회전하며, 반작용 토크 상쇄와 자세 유지에 필요한 베인의 적절한 편향각을 알지 못하므로 기체가 한쪽으로 기울어져 추락한다. 반면 학습이 완료된 후에는 베인의 편향각을 조절하여 요 방향 회전을 억제하고, 목표 고도에서 안정적으로 제자리를 유지한다. 단일 프로펠러 구조는 반작용 토크를 상쇄할 로터가 없어 기체가 지속적으로 회전하려는 경향을 가지며, 이를 4개의 베인만으로 억제할 수 있는지가 이 시나리오의 실질적인 검증 대상이다. 이를 확인하기 위해 학습 중 저장된 체크포인트별로 평가 에피소드를 수행하여 평균 요 각속도를 측정하였고, 그 결과를 Fig. 7에 나타내었다. 평균 요 각속도는 학습이 진행됨에 따라 약

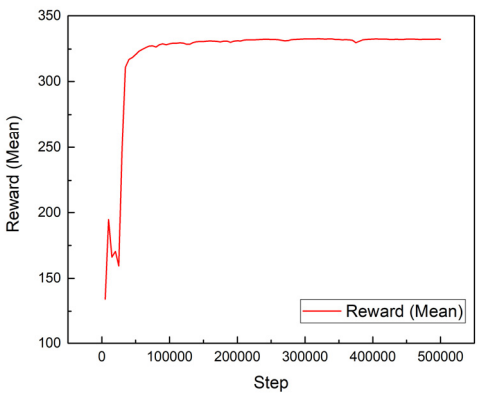


Fig. 4 Hovering total reward (mean)

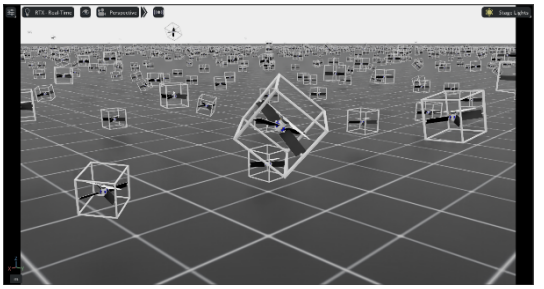


Fig. 5 Hovering before training

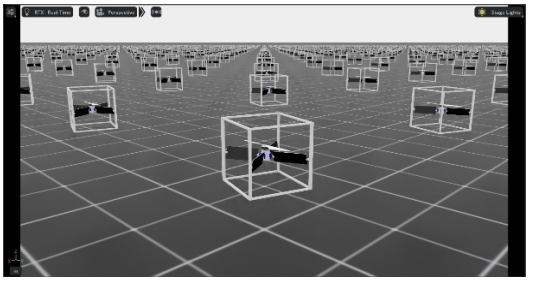


Fig. 6 Hovering after training

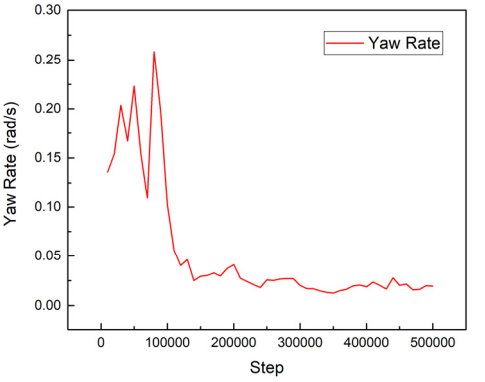


Fig. 7 Hovering yaw rate (mean)

0.02 rad/s까지 감소하였으며, 이는 베인의 편향각 조절만으로 반작용 토크를 상쇄하는 제어가 별도의 고전적 제어기 없이 학습되었음을 보여준다.

이 연구에서는 고전적 제어기와의 정량적 비교를 수행하지 않았으나, 제안 플랫폼에 고전적 제어 기법을 적용하는 경우와 비교하여 강화학습 기반 접근은 다음과 같은 장점을 갖는다. 첫째, 고전적 제어기를 설계하기 위해서는 베인 편향각, 프로펠러 후류, 기체 모멘트 사이의 비선형 관계를 해석적으로 모델링하거나 시스템 식별을 통해 제어 할당 관계를 구해야 하는 반면, 이 연구의 정책은 이러한 관계를 시뮬레이션과의 상호작용을 통해 직접 학습하였다. 둘째, 고전적 제어 기법에서는 호버링, 내비게이션, 결합 시나리오와 결합 상태에 따라 제어기 구조와 게인을 개별적으로 설계하고 조정해야 하지만, 이 연구에서는 동일한 학습 프레임워크에서 시나리오별 보상 함수와 학습 설정을 조정하여 각 정책을 학습하였다. 셋째, 결합 시나리오에서 고전적 접근은 접근 경로 생성, 모듈 간 충돌 회피, 결합 후 안정화를 위한 제어기를 각각 설계하고 조합해야 하는 반면, MAPPO 기반 정책은 이를 하나의 정책 학습 과정으로 통합하여 약 90 %의 도킹 성공률을 달성하였다. 다만 고전적 제어기와의 정량적 성능 비교는 이 연구의 한계로 남아 있으며, 향후 PID 기반 제어기를 기준 모델로 한 비교 연구를 수행할 예정이다.

내비게이션 정책의 학습 결과는 Fig. 8 ~ Fig. 11에 나타내었다. Fig. 8에서 내비게이션 시나리오의 평균 총 보상은 호버링과 마찬가지로 상승한 뒤 안정적으로 수렴하는 양상을 보였다. Fig. 9와 Fig. 10은 학습 전후 정책의 비행 모습을 비교한 것으로, 학습 전에는 목표 지점이 아닌 다른 방향으로 이동하다가 추락하는 반면, 학습 후에는 연속적으로 부여되는 목표 지점에 도달한다. 내비게이션 시나리오에서는 목표 지점에 도달하더라도 에피소드가 종료되지 않고 새로운 목표가 즉시 재부여되므로 에피소드당 목표 도달 횟수가 정책의 실질적인 성능을 직접적으로 반영하는 지표가 된다. 도달 횟수가 많다는 것은 목표까지 빠르게 이동했을 뿐 아니라, 도달 직후 다음 목표를 향

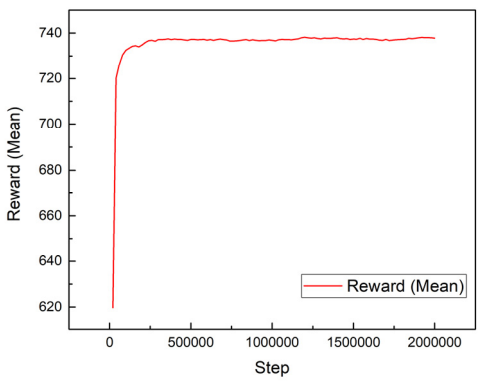


Fig. 8 Navigation total reward (mean)

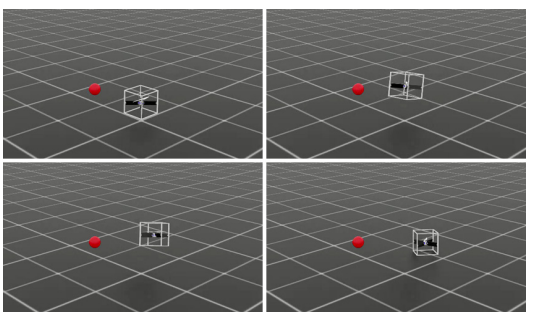


Fig. 9 Navigation before training

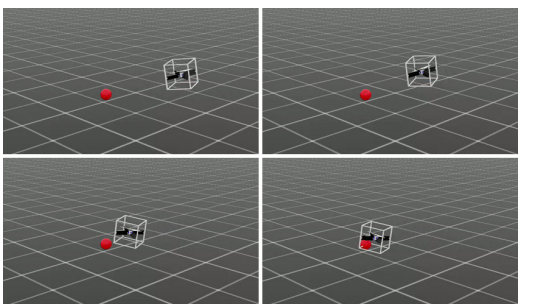


Fig. 10 Navigation after training

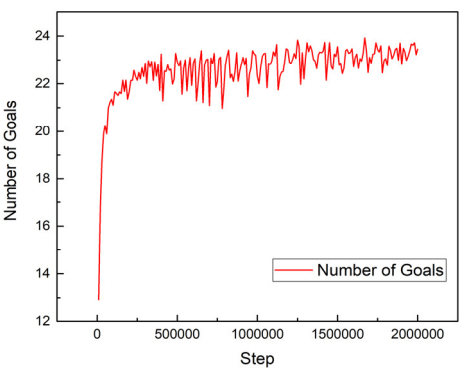


Fig. 11 Navigation number of goals (mean)

해 신속히 방향을 전환하고 그 과정에서 자세를 잃지 않았음을 함께 의미한다. Fig. 11은 체크포인트별 에피소드당 평균 목표 도달 횟수를 나타낸다. 학습이 완료된 시점에는 약 23회 목표 지점에 도달하였다. 제안한 플랫폼은 추력 방향이 기체 좌표계에 고정되어 있어 수평 이동을 하기 위해서는 기체 자체를 목표 방향으로 기울여야 하며, 목표 도달 시에는 다시 자세를 회복해야 한다. 이 결과는 이러한 기울임과 회복의 반복이 요 방향 안정화와 동시에 수행 가능함을 보여준다.

결합 시나리오의 학습 결과는 Fig. 12 ~ Fig. 15에 나타내었다. Fig. 12에서 결합 시나리오의 평균 총 보상은 학습 초기에 상승한 뒤, 호버링 및 내비게이션과 달리 학습 종료 시점까지 일정한 범위 내에서 진동하며 명확한 수렴에 도달하지 않았다. Fig. 13과 Fig. 14는 학습 전후 정책의 비행 모습을 비교한 것으로, 학습 전에는 3개의 모듈이 서로 충돌하며 추락하는 반면 학습 후에는 두 추종 모듈이 중심 모듈의 좌우에 결합하여 일자형 복합 비행체를 구성한다. 총 보상과 달리 Fig. 15에 나타낸 도킹 성공률은 학습이 진행되는 동안 꾸준히 상승하여 최종적으로 약 90 % 수준에 도달하였다.

이와 같이 총 보상과 도킹 성공률이 서로 다른 양상을 보이는 것은 이 환경의 보상 구조와 종료 조건에서 그 원인을 찾을 수 있다. 첫째, 이 결합 환경은 세 모듈 중 한 대라도 종료 조건에 도달하면 나머지 모듈을 포함한 전체 에피소드가 함께 종료되는 구조를 갖는다. 실제로 전체 에피소드의 약 12 % ~ 13 %에서 이러한 조기 종료가 발생하였다. 조기 종료는 대부분 결합이 성립하기 전 추종 모듈이 목표 슬롯으로 접근하는 과정에서 기울기 허용 범위(75°)를 초과하여 발생하였으며, 이 시뮬레이션에서는 모듈 간 물리적 충돌을 활성화하였으므로, 충돌로 인해 자세를 잃고 종료 조건에 도달하는 경우도 발생한다. 조기 종료된 에피소드에서는 생존 보상이 더 이상 누적되지 않을 뿐 아니라 도킹 보너스를 획득할 기회 자체가 사라지므로, 해당 에피소드의 총 보상은 정상적으로 완주한 에피소드에 비해 크게 낮은 값을 가진다.

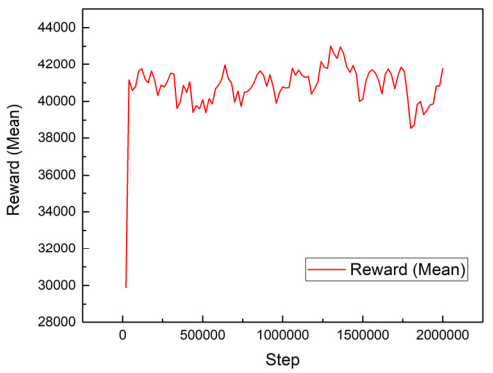


Fig. 12 Coupling total reward (mean)

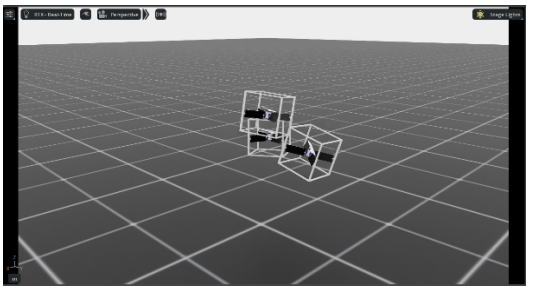


Fig. 13 Coupling before training

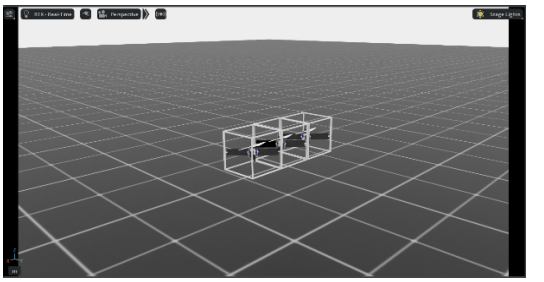


Fig. 14 Coupling after training

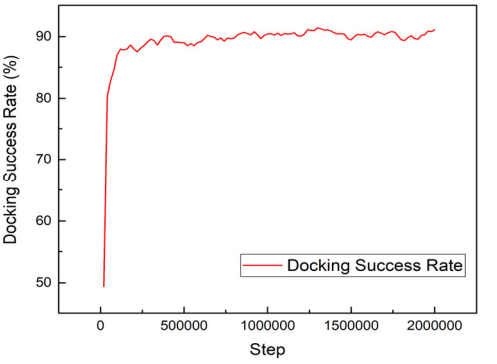


Fig. 15 Docking success rate

둘째, 도킹 성공률은 두 추종 모듈이 모두 결합을 완료했는지 여부만을 판정하는 이진적 지표인 반면, 총 보상은 접근, 정렬 과정에서의 진행량, 자세 안정성, 속도 페널티 등 다수의 항목이 함께 누적된 결과이므로 결합 자체는 안정적으로 수행하게 되더라도, 접근 과정에서의 속도 페널티나 자세 관련 페널티가 잔존하는 경우 총 보상은 이론적 최댓값에 도달하지 않는다. 이러한 점을 종합하면 총 보상의 미수렴은 정책이 결합 자체를 학습하지 못했음을 의미하는 것이 아니라 소수의 조기 종료 에피소드와 보조적 페널티 항목의 잔존에서 비롯된 것으로 분석된다. 따라서 이 시나리오에서는 총 보상보다 도킹 성공률이 정책의 실질적인 성능을 더 직접적으로 반영하는 지표이며 도킹 성공률의 지속적인 상승은 네 가지 결합 조건을 동시에 충족시키는 정밀한 접근 제어가 학습되었음을 보여준다.

4. 결 론

이 연구에서는 하나의 프로펠러와 4개의 베인으로 구성되어 서로 결합 가능한 큐브 형태의 모듈형 UAV 플랫폼을 제안하고, NVIDIA Isaac Sim 및 Isaac lab 기반의 물리 시뮬레이션 환경에서 강화학습을 통해 자세 제어 및 결합 시나리오를 학습하였다. 단일 모듈 수준에서는 PID와 같은 별도의 고전적 제어기 없이 순수 강화학습만으로 반작용 토크를 상쇄하여 요 방향 자세를 안정화하고, 연속적으로 부여되는 목표 지점을 반복 추종할 수 있음을 확인하였다. 다중 모듈 결합 시나리오에서는 CTDE 구조의 MAPPO를 통하여 네 가지 도킹 조건을 동시에 충족하는 정밀한 접근 제어가 가능하며, 결합에 따라 질량과 관성이 변화하는 상황에서도 자세 안정화가 유지됨을 확인하였다. 또한 결합과 같이 여러 개체가 상호작용하는 제어 작업에서는 보상 함수의 설계와 종료 조건이 학습 결과에 큰 영향을 미치는 것을 확인하였다. 향후 연구에서는 결합 모듈 수를 다섯 대로 확장하여 X자형 편대를 구성하고 각 모듈이 결합 위치를 자율적으로 결정하는 역할 할당 문제를 다룰 예정이다. 또한 조기 종료의 원인을 규명하여 도킹 성공률을 개선하고, 실모델 제작을 통해 시뮬레이션에서 학습된 정책의 실환경 적용 가능성을 검증하고자 한다.

후 기

이 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능혁신인재양성사업의 지원(IITP-2026-RS-2026-25549709, 50 %)과 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2026-RS-2024-00437190, 50 %)을 받아 수행된 연구임. 이 과제물은 2026년도 교육부 및 경상북도의 재원으로 경북앵커센터의 지원을 받아 수행된 지역성장 인재양성체계(앵커)-(특성화 대학)의 결과입니다(2026-anchor-15-105).

References

(1) Mahony, R., Kumar, V. and Corke, P., 2012, Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor, IEEE Robotics & Automation Magazine, Vol. 19, No. 3, pp. 20~32.

(2) Saldana, D., Gabrich, B., Li, G., Yim, M. and Kumar, V., 2018, ModQuad: The Flying Modular Structure That Self-assembles in Midair, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 691~698.

(3) Li, G., Gabrich, B., Saldana, D., Das, J., Kumar, V. et al., 2019, ModQuad-Vi: A Vision-based Self-assembling Modular Quadrotor, Proceedings of the International Conference on Robotics and Automation (ICRA), pp. 346~352.

(4) Sugihara, J., Nishio, T., Nagato, K., Nakao, M. and Zhao, M., 2023, Design, Control, and Motion Strategy of TRADY: Tilted-rotor-equipped Aerial Robot with Autonomous In-flight Assembly and Disassembly Ability, Advanced Intelligent Systems, Vol. 5, No. 10, 2300191.

(5) Oung, R. and D'Andrea, R., 2011, The Distributed Flight Array, Mechatronics, Vol. 21, No. 6, pp. 908~917.

(6) Garanger, K., Khamvilai, T., Epps, J. and Feron, E., 2025, The Dodecacopter: A Versatile Multirotor System of Dodecahedron-shaped Modules, arXiv:2504.16475.

(7) Pflimlin, J. M., Binetti, P., Souères, P., Hamel, T.

and Trouchet, D., 2010, Modeling and Attitude Control Analysis of a Ducted-fan Micro Aerial Vehicle, Control Engineering Practice, Vol. 18, No. 3, pp. 209~218.

(8) Lee, H. and Lee, D. J., 2019, Prediction of Aerodynamic Noise Radiated from a Small Multicopter Unmanned Aerial Vehicle using Acoustic Analogy, Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 29, No. 4, pp. 518~526.

(9) Hwangbo, J., Sa, I., Siegwart, R. and Hutter, M., 2017, Control of a Quadrotor with Reinforcement Learning, IEEE Robotics and Automation Letters, Vol. 2, No. 4, pp. 2096~2103.

(10) Koch, W., Mancuso, R., West, R. and Bestavros, A., 2019, Reinforcement Learning for UAV Attitude Control, ACM Transactions on Cyber-physical Systems, Vol. 3, No. 2, pp. 1~21.

(11) Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V. et al., 2023, Champion-level Drone Racing Using Deep Reinforcement Learning, Nature, Vol. 620, pp. 982~987.

(12) Xu, B., Gao, F., Yu, C., Zhang, R., Wu, Y. et al., 2024, OmniDrones: An Efficient and Flexible Platform for Reinforcement Learning in Drone Control, IEEE Robotics and Automation Letters, Vol. 9, No. 3, pp. 2838~2844.

(13) Kim, Y. J., Hong, S. M. and Oh, J. S., 2022, Design of Control Algorithm for Micro Electric Vehicle Suspension System Using Reinforcement Learning Algorithm, Transactions of the Korean Society for Noise and Vibration Engineering, Vol. 32, No. 2, pp. 124~132.

(14) Kaelbling, L. P., Littman, M. L. and Moore, A. W., 1996, Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research, Vol. 4, No. l, pp. 237~285.

(15) Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., 2017, Proximal Policy Optimization Algorithms, arXiv:1707.06347.

(16) Mittal, M., Roth, P., Tigue, J., Richard, A., Zhang, O. et al., 2025, Isaac Lab: A GPU-accelerated Simulation Framework for Multi-modal Robot Learning, arXiv:2511.04831.

(17) Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. et al., 2017, Multi-agent Actor-critic for Mixed Cooperative-competitive Environments, Advances in Neural Information Processing Systems, Vol. 30, pp. 6382~6393.

(18) Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y. et al., 2022, The Surprising Effectiveness of PPO in Cooperative Multi-agent Games, Advances in Neural Information Processing Systems, Vol. 35, pp. 24611~24624.