Journal of the KIMST, Vol. 23, No. 4, pp. 399-406, 2020
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2020.23.4.399

Research Paper 제어·구동 부문

심층 강화학습을 이용한 시변 비례 항법 유도 기법

채혁주[1] · 이단일[1] · 박수정[1] · 최한림[*,1] · 박한솔[2] · 안경수[2]

[1] 한국과학기술원 항공우주공학과
[2] 한화시스템㈜ 항공연구센터

Time-varying Proportional Navigation Guidance using Deep Reinforcement Learning

Hyeok-Joo Chae[1] · Daniel Lee[1] · Su-Jeong Park[1] · Han-Lim Choi[*,1] · Han-Sol Park[2] · Kyeong-Soo An[2]

[1] *Department of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Korea*
[2] *Avionics R&D Center, Hanwha Systems, Korea*

(Received 11 April 2020 / Revised 29 May 2020 / Accepted 26 June 2020)

Abstract

In this paper, we propose a time-varying proportional navigation guidance law that determines the proportional navigation gain in real-time according to the operating situation. When intercepting a target, an unidentified evasion strategy causes a loss of optimality. To compensate for this problem, proper proportional navigation gain is derived at every time step by solving an optimal control problem with the inferred evader's strategy. Recently, deep reinforcement learning algorithms are introduced to deal with complex optimal control problem efficiently. We adapt the actor-critic method to build a proportional navigation gain network and the network is trained by the Proximal Policy Optimization(PPO) algorithm to learn an evasion strategy of the target. Numerical experiments show the effectiveness and optimality of the proposed method.

Key Words : Pursuit-Evasion Game(추격-회피 게임), Proportional Navigation Guidance(비례 항법 유도), Reinforcement Learning(강화학습)

1. 서 론

비례 항법 유도 법칙(PNG : Proportional Navigation Guidance)은 각종 전술 유도탄의 종말 유도에 가장 널리 사용되고 있는 유도 법칙이다. 유도탄과 표적이 이루는 충돌 삼각형 상에서 시선각(LOS : Line of Sight)이 일정한 값으로 수렴하면 유도탄이 표적에 충돌하게 된다는 원리로부터 시선각 변화율에 비례한 유도 명령을 발생시킨다. 비례 항법 유도 법칙에 사용되는 비례 항법 이득의 경우 유도탄의 성능을 결정하는 주요 인자로, 운용 상황과 목적에 따라 적절한 상수를

* Corresponding author, E-mail: hanlimc@kaist.ac.kr

사용하는 것이 보편적이다. 정적인 상황에서는 목적 함수에 따라 최적 비례 항법 이득을 계산할 수 있다. 비행체의 운동이 선형 방정식으로 표현되고 속력이 일정하다는 가정하에 에너지 최소화 문제에 대한 최적 비례 항법 이득은 3으로 알려져 있으며, 비례 항법 이득이 3이 아닌 경우의 비례 항법 유도 법칙 또한 비행 상태와 제어 에너지로 구성되는 문제의 최적해가 될 수 있음이 알려져 있다[1]. 동적인 상황에서는 동적 요소를 유도입력에 보상하여 최적 비례 항법 이득을 산출할 수 있다. 표적이 기동성을 가지는 경우 비례 항법 유도 법칙에 표적의 가속도와 관련된 추가 유도 입력을 가지는 부가 비례 항법 유도 법칙(APNG : Augmented PNG)이 최적해 임이 증명되었으며[2], 비행체의 속력 변화를 고려하는 경우에도 2 이상의 비례 항법 이득이 표적 거리와 제어 에너지를 고려한 최적화 문제의 최적해 임이 증명되었다[3].

하지만 위에서 언급된 방법론들은 표적의 기동정보를 모르는 경우 최적성을 보장할 수 없다. 이에 대한 손실을 보상하기 위해서는 매 순간 최적 제어 문제를 통해 비례 항법 이득을 도출해야 하므로 실제적인 운용 상황에서 활용하기 어렵다. 이러한 한계점을 해결하기 위해 최근 강화학습(RL : Reinforcement Learning)을 통해 환경에 대한 정보를 학습하여 최적 정책을 근사하는 연구들이 제안되고 있다[4,5]. 특히 심층 신경망(deep neural network)을 이용한 심층 강화학습(deep RL) 기법은 기존에 다루기 어려웠던 연속 상태 공간 및 연속 행동 공간에 대해서도 최적 행동 정책을 효과적으로 도출할 수 있다는 것이 알려지고 있다[6,7].

본 논문에서는 유도탄의 최적 비례 항법 이득을 심층 강화학습을 통해 학습시키고 운용 상황에 따라 실시간으로 비례 항법 이득을 결정하여 추격을 수행하는 기법을 제안한다. 2장에서는 유도탄과 표적의 교전을 추격-회피 문제로 구성하고 각 개체의 행동 전략을 정의한다. 3장에서는 제안하고자 하는 강화학습 기법을 설명하며 4장에서는 시뮬레이션 결과를 통해 알고리즘의 성능을 분석한다. 5장에서는 결론을 기술한다.

2. 문제 정의

2.1 개체 운동 방정식

2D 평면에서 기동을 수행하는 추격자와 회피자는 점질량 운동체로 표현할 수 있다. XY 평면상에서의 운동 방정식은 아래와 같이 표현된다.

$$\begin{aligned} \dot{x}_i &= V_i \cos\psi_i \\ \dot{y}_i &= V_i \sin\psi_i \\ \dot{V}_i &= 0 \\ \dot{\psi}_i &= a_i \end{aligned} \tag{1}$$

여기서 x_i, y_i, ψ_i, V_i, a_i는 각각 i-개체의 평면상 X 좌표, Y 좌표, 기수각, 속력, 제어 입력을 나타낸다. 하첨자 $i \in \{p, e\}$는 추격자(pursuer), 회피자(evader)를 의미한다. 각 개체의 제어 입력은 측방향 가속도를 자신의 속력(V_i)으로 나눈 값이며 제한된 성능을 가진다($|a_i| \leq a_{i_{\max}}$).

각 개체의 속력이 일정한 추격-회피 문제에서 추격자가 회피자보다 빠른 속력($V_e < V_p$)과 좋은 기동성($a_{e_{\max}} < a_{p_{\max}}$)을 가지고 있는 경우 두 개체의 초기 조건과 상관없이 항상 포획에 성공할 수 있음이 알려져 있다[8,9]. 본 논문에서는 추격자와 회피자의 속력은 각각 V_p = 500 m/s, V_e = 200 m/s로 가정하였다. 추격자의 최대 제어 입력($a_{p_{\max}}$)은 6, 회피자의 최대 제어 입력($a_{e_{\max}}$)은 3으로 가정하였다.

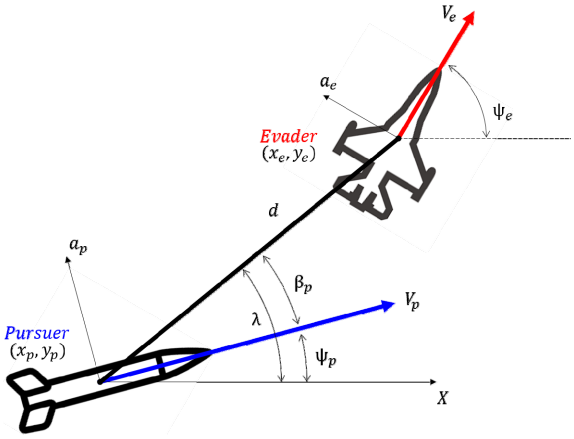

Fig. 1. Pursuit-evasion geometry

2.2 교전 운동 방정식

회피자와 추격자의 교전 상황은 Fig. 1과 같이 표현된다. 추격자와 회피자의 기동 의사결정은 각 개체의

절대적인 상태가 아닌 상대방과의 상대적인 상태를 기반으로 이루어진다. 따라서 추격-회피 게임의 운동은 식 (2)와 같이 추격자의 동체 좌표계를 기준으로 표현할 수 있다.

$$\begin{aligned} x &= x_e - x_p \\ y &= y_e - y_p \\ \psi &= \psi_e - \psi_p \end{aligned} \tag{2}$$

여기서 x, y, ψ는 각각 회피자의 상대 X 좌표, 상대 Y 좌표, 상대 기수각을 의미한다.

2.3 추격자 및 회피자 행동 전략

추격-회피 문제에서 추격자의 요격영역(capture set)을 벗어나기 위한 회피자의 최적 행동은 아래 식 (3)과 같이 알려져 있다[10].

$$a_e^* = \begin{cases} -a_{e_{\max}} \cdot \mathrm{sign}[\sin(\psi_e - \lambda_f)] & when\ \psi_e \neq \lambda_f \\ 0 & when\ \psi_e = \lambda_f \end{cases} \tag{3}$$

이때, ψ_e는 회피자의 기수각, λ_f는 교전 종료 시의 시선각을 나타낸다. sign은 부호함수로 입력이 양수일 경우 +1, 음수일 경우 –1을 나타낸다.

추격자의 최적 행동 또한 증명되어 있지만, 교전 운동 방정식을 적분하여 해를 도출하기 때문에 실시간 전략으로 사용하는 데 어려움이 있다. 그러므로 유도탄과 같은 추격 상황에서 많은 경우 식 (4)와 같은 유도 순수비례 항법 유도 명령을 이용한다. 본 연구에서도 추격자는 아래와 같은 유도 전략을 사용하여 회피자를 추격한다.

$$a_p = N\dot{\lambda} \tag{4}$$

여기서 N은 비례항법이득, $\dot{\lambda}$는 시선각 회전각속도를 의미한다. 비례 항법 유도는 추격자와 회피자의 시선각 변화를 감소시켜 종말 단계에서는 시선각을 유지한다는 특징을 가지고 있다. 따라서 회피자는 현재 시선각과 종말 시선각이 유사하다는 가정($\lambda_f \approx \lambda$)을 가지고 매 순간 식 (5)의 전략을 취한다고 가정하였다.

$$a_e = \begin{cases} -a_{e_{\max}} \cdot \mathrm{sign}[\sin(\psi_e - \lambda)] & when\ \psi_e \neq \lambda \\ 0 & when\ \psi_e = \lambda \end{cases} \tag{5}$$

3. 심층 강화학습을 이용한 시변 비례 항법 유도

3.1 PPO : Proximal Policy Optimization[11]

PPO 알고리즘은 강화학습 기법 중 정책 경사 기법(policy gradient method)에 해당한다. 정책 경사 기법은 가치 경사 기법(value gradient method)보다 행동의 변화가 점진적으로 발생하여 수렴성에 있어 큰 장점이 있다. 또한, 연속 행동 공간에 대해 효율적이고 확률론적인 정책(stochastic policy)을 학습할 수 있다는 장점이 있다. 일반적으로 정책 경사도는 정책 파라미터(policy parameter, θ)에 대해 아래와 같은 목적 함수 L^{PG}를 최대화하는 방향으로 계산된다.

$$L^{PG} = \hat{E}_t\left[\log \pi_\theta(a_t|s_t)\hat{A}_t\right] \tag{6}$$

이때, a_t와 s_t는 각각 시간 t에서 수행한 행동과 상태를 나타내며 π_θ는 확률론적인 정책, $\hat{A}_t$는 이득 함수의 추정값을 의미한다. $\hat{E}_t$는 샘플링된 데이터에서의 평균값을 의미한다. 앞서 언급했듯이 L^{PG}는 가치 경사에 비해 점진적인 정책 변화를 유도하지만, 여전히 파라미터 공간상에서의 점진적인 변화가 정책 공간상에서 큰 변화를 유발할 수 있다는 한계점이 존재한다.

PPO에서는 정책 공간상에서의 변화를 고려하기 위해 식 (7)과 같이 현재 정책 확률과 이전 정책 확률의 비(r_t)를 정의하고 이를 이용한 대체 목적 함수(surrogate object function)를 구성한다(식 (8)).

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{7}$$

$$L^{CPI}(\theta) = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] \tag{8}$$

$$\begin{aligned} &L^{CLIP}(\theta) \\ &= \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ clip(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \end{aligned} \tag{9}$$

그 후 필요 이상의 정책 갱신을 방지하기 위하여 대체 목적 함수(L^{CPI})에 클리핑(clipping) 기법을 적용한 새로운 목적 함수(L^{CLIP})를 사용한다. 이때, ϵ는 초매개변수를 의미한다.

이득 함수 $\hat{A}_t$는 식 (10)과 같이 상태에 대한 가치 함수 $V(s)$를 통해 추정할 수 있다.

$$\hat{A}_t = -V(s_t) + v_t + \gamma v_{t+1} + \cdots + \gamma^{T-t+1} v_{T-1} + \gamma^{T-t} V(s_T) \quad (10)$$

이때, T는 총 시간 스텝 수, t는 $[0, T]$ 범위의 시간 인덱스, s_t는 t에서의 상태, v_t는 t에서의 보상, γ는 할인요소(discount factor)를 나타낸다. 가치 함수도 네트워크를 이용하여 학습하는 모델의 경우 다음과 같은 목적 함수 L^V를 통해 가치 파라미터(value parameter, μ)를 갱신한다.

$$L^V(\mu) = \hat{E}_t\left[(V_\mu(s_t) - V_t)^2\right] \quad (11)$$

여기서 V_μ는 μ로 구성된 가치 함수 네트워크를 의미하며, $V_t = v_{t+1} + \gamma V_\mu(s_{t+1})$ 이다. 전체적인 알고리즘의 구조는 Fig. 2에서 확인할 수 있다. 액터(actor) 네트워크는 목적 함수 L^{CLIP}을 통해 갱신하며, 크리틱(critic) 네트워크는 목적 함수 L^V를 통해 갱신한다.

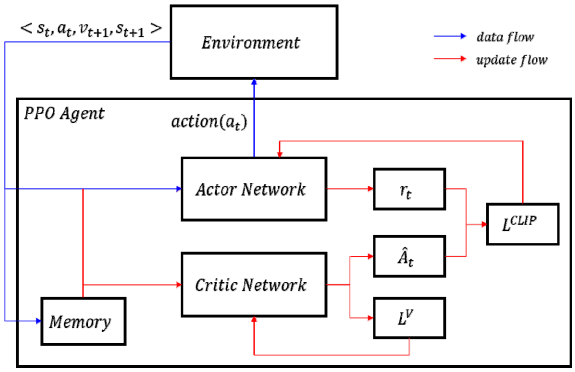

Fig. 2. Actor-critic PPO agent

3.2 비례 항법 이득 네트워크

본 연구에서 비례 항법 이득을 학습시키기 위한 네트워크로 액터-크리틱 구조를 사용하였다[12,13]. 비례 항법 이득 네트워크는 교전 운동 방정식의 상태 변수(s)를 입력으로 받는다. 액터 네트워크는 상태 변수를 입력받아 행동 정책을 도출하며 크리틱 네트워크는 입력 받은 상태의 가치를 판단한다. 본 논문에서는 각 네트워크를 다층 퍼셉트론(multi layer perceptron)을 이용하여 구성하였다.

$$s = [x,\ y,\ \cos(\psi),\ \sin(\psi)] \quad (12)$$

유도탄의 비례 항법 이득은 일반적으로 $N \in [2,6]$인 범위에 최적 이득이 존재하므로 액터 네트워크의 출력은 아래와 같은 형태로 정의한다.

$$N_{learned} = 2 + 4 \cdot sigmoid(z) \quad (13)$$

이때, z는 액터 네트워크의 마지막 층(layer)의 출력을 의미한다($z \sim \pi_\theta(a_t \mid s_t)$). 크리틱 네트워크는 액터 네트워크와 같은 입력을 받아 상태에 대한 가치 $V(s_t)$를 도출한다.

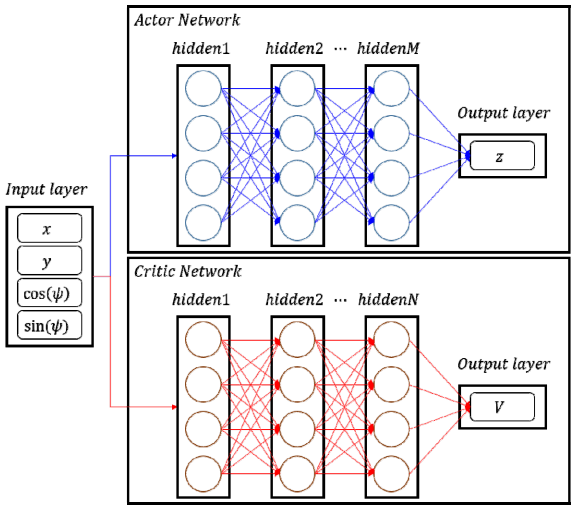

Fig. 3. Actor-critic networks structure

3.3 보상함수

강화학습 학습자는 자신의 행동에 대한 보상 값을 환경으로부터 관측하여 학습을 진행한다. 본 연구에서는 회피자 추격 문제에 대해 다음과 같이 보상함수를 설정하였다.

먼저 두 개체의 상대 거리 d와 시선각과 추격자의 기수각의 차이(방위각) ϕ를 정의하였다.

$$d = \sqrt{x^2 + y^2}$$
$$\phi = \lambda - \psi_p \quad (14)$$

상대 거리와 방위각은 추격의 성공을 결정하는 중요한 요소이다. 특히 방위각이 작을수록 추격자의 요격 영역이 커지기 때문에 작은 방위각을 유지하는 것이 추격에 유리하다. 이를 반영한 보상함수는 아래와 같

이 표현된다.

$$v = \begin{cases} 60 & d < 100m, \cos(\phi) > 0.866 \\ -1+\cos(\phi) & otherwise \end{cases} \quad (15)$$

추격을 수행 중일 때는 시간에 대한 페널티(–1)와 시야 유지에 대한 보상($\cos(\phi)$)을 지속해서 받게 된다. 에피소드는 시뮬레이션 시간으로 최대 60초 동안 진행되며 식 (15)에서 명시한 추격 성공 조건을 만족할 경우 60의 보상을 받으며 에피소드가 종료된다. 이는 최대 60초 동안 시뮬레이션이 진행되며 추격에 성공하였을 경우 양수의 보상을 얻도록 정의되었다. 60초 후에도 추격에 성공하지 못하면 0의 보상을 받으며 에피소드가 종료된다.

4. 학습 및 시뮬레이션

제안된 강화학습 알고리즘을 검증하기 위해 시뮬레이션에서 얻어진 에피소드 데이터를 이용하여 학습을 수행하였다. 이후 학습된 액터 네트워크를 이용하여 성능을 분석하였다. 시뮬레이션과 학습은 Table 1에 명시된 환경에서 수행되었다.

Table 1. Simulation environment

항목	내용
운영체제	Windows 10
CPU/RAM	Intel i7-8700K@3.7GHz/32GB
프로그래밍 언어	Python3.7, Pytorch-CPU1.1

4.1 학습 과정

학습에 사용된 액터 네트워크와 크리틱 네트워크는 2개의 잠재층을 가진 다층 퍼셉트론을 이용하였다. 두 잠재층의 노드(node) 수는 각각 64, 32개로 설정하였다. 최적화 알고리즘(optimizer)은 ADAM 알고리즘을 사용하였다. 학습에 사용된 설계 파라미터는 Table 2와 같다. 총 2만 번의 학습 과정을 거쳤으며 학습 과정에 따른 보상의 변화는 Fig. 4에 나타나 있다. 그래프에 청색 실선으로 표현된 보상은 학습 과정에 따른 이동 평균 보상을 나타낸 것이며 분홍색 영역은 이에 따른 표준 편차 영역을 나타내고 있다. 학습 과정을 통해 비례 항법 이득 네트워크가 얻는 보상이 일정 수준(≈ 43)으로 수렴하는 것을 확인할 수 있다.

Table 2. Learning parameters

Parameters	Name	Value
ϵ	clip parameter	0.2
γ	discount factor	0.99
α	learning rate	0.0001

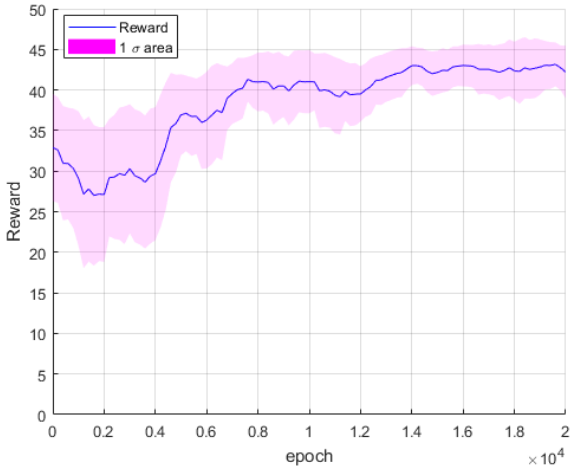

Fig. 4. Learning process

4.2 추격 성능 분석

학습된 액터 네트워크를 이용하여 세 가지 Case에서의 시뮬레이션을 수행하여 추격 기동 형태를 분석하였다. Case 1에서는 무기동 표적에 대한 추격 성능을 분석하였으며, Case 2, 3에서는 최적 회피 기동을 수행하는 표적에 대한 추격 성능 분석을 수행하였다. 모든 Case에서 추격자의 초기 상태는 $[x_p,\ y_p,\ \psi_p]$ = [0 km, 0 km, 0 rad]로 고정하였으며 회피자의 초기 상태는 Table 3과 같이 설정하였다.

Table 3. Initial states

Case	Initial State of Evader			Evasive maneuver
	x[km]	y[km]	ψ[rad]	
Case 1	4.0000	0.0000	-1.5708	X
Case 2	3.2438	2.7690	-1.5805	O
Case 3	3.7606	-0.0430	0.0153	O

4.2.1 Case 1: 무기동 표적

표적은 기동 없이 추격자의 전방에서 $-y$축 방향으로 진행한다. 이러한 상황에서는 비례 항법 이득이 상수 3인 경우가 유도 오차 및 에너지 최소화 문제의 최적해로 알려져 있다.

Fig. 5에서 청색 실선은 비례 항법 이득 네트워크를 활용한 기동 궤적이며 흑색 점선과 분홍색 점선은 각각 고정 비례 항법 이득 3 또는 4를 이용한 기동 궤적을 보여준다. 모든 경우 성공적으로 표적을 추격함을 확인할 수 있다.

Fig. 6은 추격 과정에 대한 상태 및 명령 정보를 나타내며 시간이 흐름에 따라 시선각이 수렴하고 기동 명령 또한 0으로 수렴하는 것을 확인할 수 있다. 학습된 시변 이득을 사용한 경우와 고정 이득 4(N = 4)를 사용한 경우 고정 이득 3(N = 3)을 사용한 것보다 큰 기동 명령을 사용하여 먼저 추격에 성공하였다.

학습된 네트워크를 활용하여 제어 명령을 생성하는 시간은 평균 0.00619초로 평균 0.00615초인 고정 이득 기법과 비슷한 수준을 나타내었다.

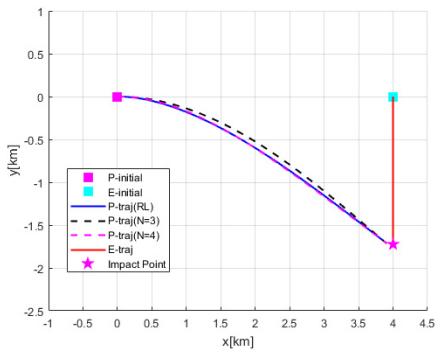

Fig. 5. Planar trajectory (Case 1)

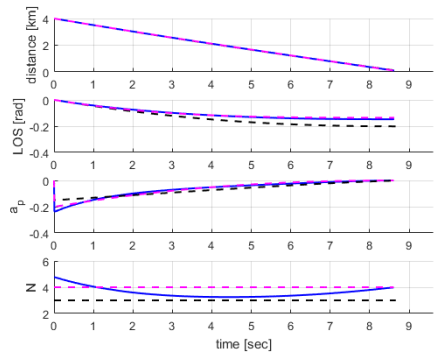

Fig. 6. Pursuit history (Case 1)

4.2.2 Case 2: 기동 표적

회피자가 최적 회피 기동을 수행하는 상황으로, 초기 조건은 회피자가 $-y$축 방향으로 진행하며 추격자는 회피자의 기수를 기준으로 전방 약 50도에서 $+x$축 방향으로 진행하도록 설정하였다.

Fig. 7에서 나타난 바와 같이 Case 2에서도 추격자는 학습된 네트워크를 사용하여 성공적으로 회피자를 추격하는 것을 확인할 수 있다. 회피자의 경우 $-y$축 방향으로 출발하였지만 시간이 지남에 따라 회피 기동을 통해 추격자에서 멀어지는 방향으로 기수를 돌리는 행동을 보였으며 추격자의 경우 tail-chase 형태로 시선각을 유지해 추격하는 행동을 취하였다.

Fig. 8은 추격 과정에 대한 정보를 나타내며 시간이 흐름에 따라 시선각이 수렴하는 것을 볼 수 있다. 약 8초 이후로 수렴된 시선각을 유지하며 약 13.5초 후 추격에 성공하였다. 학습된 네트워크가 도출한 비례 항법 이득은 초기 N = 3.7에서 시작하여 점진적으로 증가하며 시간이 지난 후 N = 4.5 수준을 유지하였다.

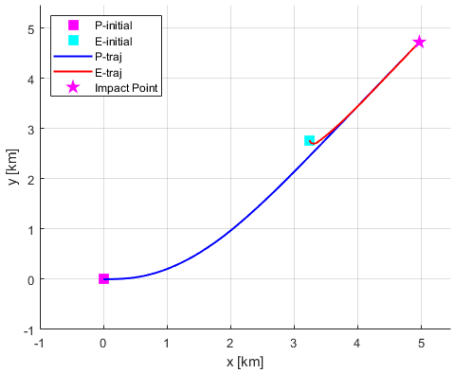

Fig. 7. Planar trajectory (Case 2)

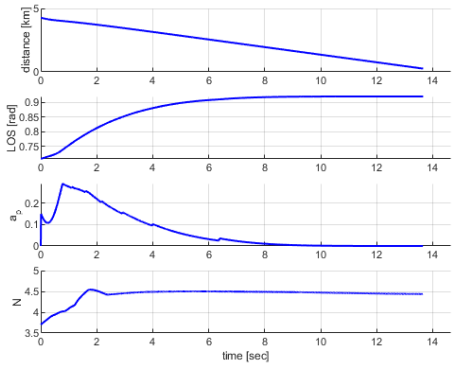

Fig. 8. Pursuit history (Case 2)

4.2.3 Case 3: 기동 표적

최적 회피 기동을 수행하는 표적으로, 초기 조건은 회피자가 $+x$축 방향으로 진행하며 추격자는 회피자의 후미 방향에서 $+x$축 방향으로 진행하는 경우이다.

초기 시선각은 –0.0114 rad으로 추격자의 기수각과 차이가 작아 Fig. 9에서 확인할 수 있듯이 직선적인 움직임을 통해 회피자를 추격하였다.

앞선 두 경우와는 다르게 학습된 네트워크가 도출한 비례 항법 이득은 추격 과정 동안 N = 4.5 수준을 유지하였다. Case 3의 경우는 시선각의 변화가 거의 없어 항법 이득에 의한 추격 성능의 민감도가 낮은 경우이다. 이 경우에 액터 네트워크의 출력이 N = 4.5로 나오는 것은 강화학습 중 다양한 에피소드에서 직선 움직임을 가지기 전후 해당 이득을 사용하는 것이 평균적으로 좋은 보상을 얻었다는 것을 의미한다.

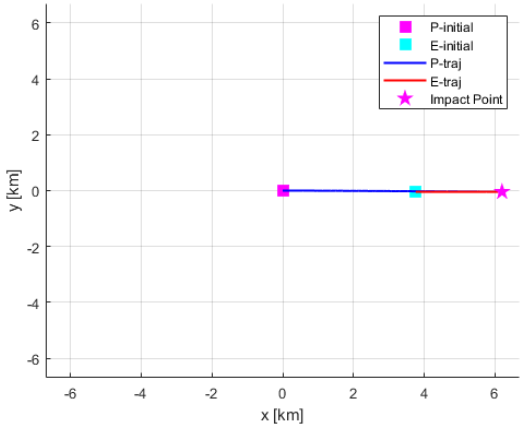

Fig. 9. Planar trajectory (Case 3)

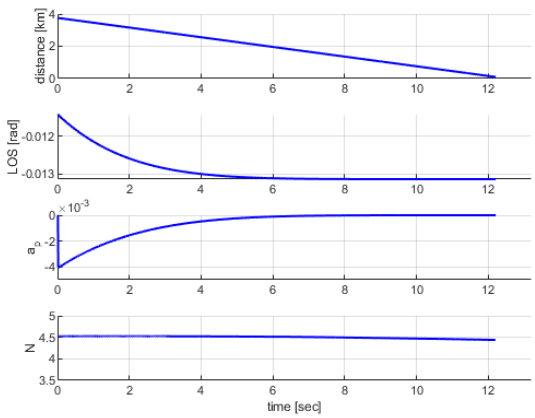

Fig. 10. Pursuit history (Case 3)

4.3 고정 비례 항법 이득과의 비교

제안된 시변 비례 항법 이득 알고리즘의 성능을 검증하기 위해 1000개의 초기 상태를 샘플링하고 시뮬레이션을 수행하였다. $N \in \{2, 3, 4, 4.5, 5, 6\}$인 고정 비례 항법 이득들을 사용하여 결과를 비교하였다. 이때, 회피자는 추격자의 움직임에 대한 최적 회피 기동을 수행하였다.

일반적으로 추격 성능을 평가하기 위해 추격 성공 시간과 추격에 소모된 에너지를 지표로 이용한다. 이에 대한 성능 분석을 수행하기 위해 식 (16)과 같이 소모에너지(J_{energy})와 소모시간(J_{time})의 가중치(w)합으로 구성된 목적함수를 설계하였으며 고정 비례항법 유도 법칙과 시변 비례항법 유도법칙에 대한 전체 시뮬레이션에 결과를 Fig. 11에 도시하였다.

$$\min J = J_{energy} + wJ_{time} \tag{16}$$

N = 66인 경우 최소 시간을 소모하여 추격에 성공하였으며 N = 2인 경우 최소 에너지를 소모하여 추격에 성공하였다. 목적함수의 가중치가 0.0105 ≦ w ≦ 0.1067인 범위에서는 학습된 네트워크를 통해 도출한 시변 이득을 사용하는 것이 다른 고정 이득을 사용하는 것에 비해 좋은 성능을 나타내는 것을 확인할 수 있었다.

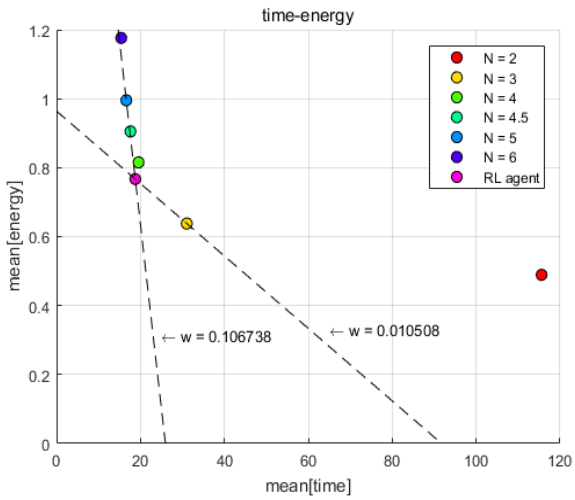

Fig. 11. Time-energy objective

5. 결 론

비례 항법 유도는 각종 전술 유도탄에 가장 널리 사용되는 법칙으로 운용 목적에 따라 최적 비례 항법 이득 상수를 도출하여 사용한다. 하지만 회피자의 기동 전략에 대한 정보가 부족한 경우 매 순간 기동을 추론하고 최적 제어 문제의 해를 도출해야 하므로 실제적인 운용이 불가능하다. 최근 심층 강화학습 기법을 통해 이러한 문제의 최적 정책을 근사하는 연구들이 제안되고 있다. 본 논문에서는 최적 비례 항법 이득을 추론하기 위한 네트워크로 액터-크리틱 구조를 사용하였으며 PPO 알고리즘을 이용하여 강화학습을 진행하였다. 학습된 네트워크는 시뮬레이션을 통해 실시간으로 비례 항법 이득을 결정하여 추격에 성공하는 것을 확인할 수 있었으며 목적 함수에 따라 고정 이득 기법보다 효과적임을 확인할 수 있었다. 제안된 시변 비례 항법 유도 법칙은 운용 상황에 따라 다양한 회피 기동을 학습할 수 있다는 확장성을 가지고 있다. 또한, 새로운 목적함수의 설계를 통해 요격 시간 및 요격 각도 제어 등의 특정 임무를 수행하는 기법으로 활용될 수 있을 것으로 기대한다.

후 기

이 논문은 2018년도 한화시스템(주)의 재원을 지원받아 수행된 연구임

References

[1] Hangju Cho, “Navigation Constants in PNG Law and the Associated Optimal Control Problems,” Proc. Korean Automatic Control Conference, Seoul, Korea, pp. 578-583, 1992.

[2] Vitalij Garber, “Optimum Intercept Laws for Accelerating Targets,” AIAA Journal, Vol. 6, No. 11, pp. 2196-2198, 1968.

[3] In-Soo Jeon, and Jin-Ik Lee, “Analysis on Optimality of Proportional Navigation with Time-varying Velocity,” Journal of the Korean Society for Aeronautical & Space Sciences, Vol. 37, No. 10, pp. 998-1001, 2009.

[4] Christopher JCH Watkins and Peter Dayan, “Q-learning,” Machine Learning, Vol. 8, No. 3-4, pp. 279-292, 1992.

[5] David Silver, et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” Nature, Vol. 529, No. 7587, pp. 484-489, 2016.

[6] Yan Duan, et al., “Benchmarking Deep Reinforcement Learning for Continuous Control,” International Conference on Machine Learning, pp. 1329-1338, 2016.

[7] Tuomas Haarnoja, et al., “Soft Actor-critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” arXiv preprint arXiv:1801.01290, 2018.

[8] Ernest Cockayne, “Plane Pursuit with Curvature Constraints,” SIAM Journal on Applied Mathematics, Vol. 15, No. 6, pp. 1511-1516, 1967.

[9] G. T. Rublein, “On Pursuit with Curvature Constraints,” SIAM Journal on Control, Vol. 10, No. 1, pp. 37-39, 1972.

[10] Josef Shinar, Moshe Guelman, and Alon Green, “An Optimal Guidance Law for a Planar Pursuit-evasion Game of Kind,” Computers & Mathematics with Applications, Vol. 18, No. 1-3, pp. 35-44, 1989.

[11] John Schulman, et al., “Proximal Policy Optimization Algorithms,” arXiv preprint arXiv:1707.06347, 2017.

[12] Vijay R. Konda, and John N. Tsitsiklis, “Actor-critic Algorithms,” Advances in Neural Information Processing Systems, pp. 1008-1014, 2000.

[13] Volodymyr Mnih, et al., “Asynchronous Methods for Deep Reinforcement Learning,” International Conference on Machine Learning, 2016.