Original Article

https://doi.org/10.12985/ksaa.2022.30.2.034
ISSN 1225-9705(print) ISSN 2466-1791(online)

CFIT 자율 회피를 위한 심층강화학습 기반 에이전트 연구

이용원*, 유재림**

Study of Deep Reinforcement Learning-Based Agents for Controlled Flight into Terrain (CFIT) Autonomous Avoidance

Yong Won Lee*, Jae Leame Yoo**

ABSTRACT

In Efforts to prevent CFIT accidents so far, have been emphasizing various education measures to minimize the occurrence of human errors, as well as enforcement measures. However, current engineering measures remain in a system (TAWS) that gives warnings before colliding with ground or obstacles, and even actual automatic avoidance maneuvers are not implemented, which has limitations that cannot prevent accidents caused by human error. Currently, various attempts are being made to apply machine learning-based artificial intelligence agent technologies to the aviation safety field. In this paper, we propose a deep reinforcement learning-based artificial intelligence agent that can recognize CFIT situations and control aircraft to avoid them in the simulation environment. It also describes the composition of the learning environment, process, and results, and finally the experimental results using the learned agent. In the future, if the results of this study are expanded to learn the horizontal and vertical terrain radar detection information and camera image information of radar in addition to the terrain database, it is expected that it will become an agent capable of performing more robust CFIT autonomous avoidance.

Key Words : CFIT(조종상태에서의 지상충돌), TAWS(지상접근경보시스템), Reinforcement Learning (강화학습), Machine Learning(기계학습), Artificial Intelligence Agent(인공지능 에이전트), Collision Avoidance(충돌 회피)

Ⅰ. 서 론

CFIT(controlled flight into terrain) 사고란, 조종사에 의해 완전하게 조종되고 있는 감항성을 가진 항공기가 사고가 발생할 때까지 조종사가 전혀 감지하지 못하거나 경미하게 인지한 상태에서 부주의로 지면(땅), 장애물, 수면 또는 활주로 밖으로 비행함으로써 일어나는 사고를 의미한다.[1] 이러한 CFIT 사고와 관련하여 국제민간항공기구(ICAO)는 GASP(Global Aviation Safety Plan) 비전으로 상업적 항공운항에서 2030년 이후에는 치명적인 사고(fatal accident)가 일어나지 않는 것을 목표로 하고 있다. 이러한 비전을 달성하기 위하여 GASP의 2020~2022년도 발행본에서는 2019년 전체 정기 상업용 항공기 사고의 70.5%,

Received: 01. Jun. 2021, Revised: 14. Sep. 2021,
Accepted: 31. Mar. 2022
* 사업용 조종사
** 청주대학교 항공운항학전공 교수
연락저자 E-mail : jlyoo@cju.ac.kr
연락저자 주소 : 충청북도 청주시 청원구 대성로 298

1) Bateman, D., Honeywell사 소속 연구원

사고로 인한 전체 사망자 수의 18.8%를 차지하고 있는 CFIT, LOC-1(loss of control in-flight), MAC(mid air collision), RE(runway excursion), RI(runway incursion)와 같은 고위험 형태의 사고발생(high risk categories of occurrence) 방지를 강조하고 있으며, 2020년 발행된 IATA의 Safety Report 2019(Edition 56)에서 CFIT 사고는 지난 5년 동안(2015~2019) 치명적 사고의 8%를 차지하며, LOC-1 사고와 RE 사고에 이어 세 번째로 높은 치명적 사고율을 보이고 있다. 이와 같이 CFIT 사고는 치명적 사고의 높은 비중을 차지하고 있음에도 불구하고 지금까지 민간항공분야에서 이러한 CFIT사고 방지를 위한 노력으로 human error 발생을 최소화하기 위한 교육 대책과 관리적 측면의 대책에 치우쳐 있고 CFIT 사고를 획기적으로 방지할 수 있는 공학적 대책[2]은 미흡하다고 할 수 있다. 현재 항공기에 적용되고 있는 대책으로는 크게 세 가지의 유형의 대책이 있다. 첫 번째, 현재 사용 중인 GPWS (ground proximity warning system)나 EGPWS (enhanced ground proximity warning system)과 같은 TAWS(terrain avoidance and warning system)의 사용, 두 번째, GPS에 연관된 최신의 지형지물, 장애물, 활주로에 대한 운영자들의 최신의 Terrain Databases Update, 세 번째, EGPWS와 관련된 새로운 경고 장치에 대한 대응훈련프로그램과 절차의 지원 등이 있다. 이러한 대책의 노력으로 오늘날 CFIT 사고는 많이 감소했지만 앞서 살펴보았듯이 여전히 높은 치명적 사고의 유형으로 남아 있음을 알 수 있다.

따라서, CFIT 사고 방지를 위해 human error 등과 같은 불가항력적인 상황에서도 자동적으로 회피 기동단계까지 수행할 수 있는 기술의 적용이 매우 필요하다고 할 수 있다.

한편, 오늘날 항공기에 적용되고 있는 Autopilot 시스템은 인간의 고도, 속도, 방위 명령에 따라 항공기를 제어하는 시스템으로써 사람의 개입이 전제되어 있는 시스템으로, 비정상적인 긴급한 상황에서 오히려 상황을 악화시키는 경우도 발생하고 있다. 이와 비교하여 Autonomous 시스템은 기계가 현재 상황을 판단하고 인간의 개입 없이 적절하게 항공기를 제어하는 것으로, 인간의 개입이 불가능한 상황에서도 안전한 운행을 가능하게 할 수 있다. 즉, human error 발생 시 인간을 대신해서 지형충돌 방지 기동을 수행할 수 있는 Autonomous 회피 기동의 적용은 CFIT 사고 방지를 위한 획기적인 대책이라 할 것이다.

따라서 본 논문에서는 CFIT 사고 방지를 위한 공학적 대책의 하나로, 심층강화학습 기반의 에이전트를 제안하고 학습된 에이전트가 지상충돌이 예상되는 상황에서 항공기를 자율 조종하여 이를 회피할 수 있는 지를 6자유도(6DOF; six degrees of freedom) 비행동역학모델(FDM; flight dynamics model)과 임의의 지형장애물로 구성된 시뮬레이션 환경에서 실험을 통해 그 적용 가능성을 제시하는 것을 목표로 한다. 학습과 실험을 3단계로 구성하여 첫 번째, 심층강화학습을 위한 에이전트와 이를 학습시키기 위한 학습 환경을 구성하고, 두 번째, 학습 환경을 기반으로 에이전트를 학습시킨 후, 마지막으로, 학습된 에이전트를 이용하여 시뮬레이션 환경에서 CFIT 자율 회피 실험을 수행하고 그 성과를 평가한다.

II. 본 론

2.1 심층강화학습 기반 에이전트

에이전트란 Fig. 1과 같이 직면한 환경(environment)에서 자신의 목표(goal)를 달성하기 위해 환경을 인식(percepts)하고 행동(action)함으로써 문제 상황을 해결하는 주체로 정의하고 있다(Russell et al., 2009). 에이전트를 구현하는 방법에는 여러 가지가 존재하나, 본 논문에서는 심층강화학습을 이용하여 이를 구현한다.

심층강화학습(DRL; deep reinforcement learning)은 심층신경망(DNN; deep neural network)을 강화학습에 결합한 것으로서, 기존의 강화학습은 주로 격자세계(grid world) 환경에서 제한된 범위의 불연속적인

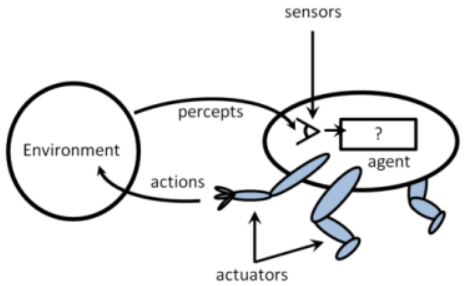

Fig. 1. Intelligent agents (Russell, 2009)

2) 공학적 대책: 안전성을 향상시키기 위한 공학적인 대책(인간의 불확실성을 제거하기 위한 설비 시스템).

상태 공간과 행동 공간을 고려한 학습 방식이지만, 심층강화학습은 자율 주행, 무인기 제어 등과 같은 연속된 상태 공간과 행동 공간을 고려한 학습방식으로, 제어 및 최적화 문제를 해결하는 데 주로 사용한다. 이러한 심층강화학습 알고리즘으로는 DQN(deep Q network), A3C(asynchronous advantage actor critic), DDPG (deep deterministic policy gradient), TRPO(trust region policy optimization), PPO(proximal policy optimization), TD3(twin delayed deep deterministic policy gradients) 등이 있다(신승재 외, 2019).

2.2 심층강화학습 기반 에이전트를 이용한 관련 연구

심층강화학습기반의 에이전트를 운항 분야에 적용한 연구가 국내외에서 다양하게 수행되고 있다. 국내에서는 문일철 외(2020)에 따르면, DDPG를 이용하여 전투기 공중교전이 가능한 에이전트에 대한 연구 사례가 있으나, 항공기 간 또는 지형충돌 회피는 고려되지 않았다. 우주현(2018)에 따르면, DQN을 이용한 무인수상선 간의 충돌회피가 연구된 사례가 있으나, 선박의 운동 특성상 수평 성분에 대한 회피만을 다루고 있다. 국외에서는 Wen et al.(2017)에 따르면 DDPG를 이용한 공중충돌 방지를 위한 ATC(air traffic control) 알고리즘에 대한 연구 사례가 있으나, 이 역시 항공기 간 수평 성분에 대한 충돌회피만을 다루고 있다. Tang et al.(2020)에 따르면, DDPG를 이용한 강한 측풍환경에서 자동 착륙이 가능한 에이전트에 대한 연구 사례가 있으나, 지형장애물에 대한 회피는 고려되지 않았다. 이상에서 살펴보았듯이, 심층강화학습을 자율 운항에 적용한 연구 사례 등은 존재하나, 항공기의 지형장애물에 대한 자율 회피에 대한 연구 사례는 찾아볼 수 없었다.

2.3 심층강화학습 알고리즘

본 논문에서는 에이전트를 학습시키기 위한 심층강화학습 알고리즘으로써 로봇 및 무인항공기 등의 연속 행동공간(continues action space) 학습에 주로 사용하는 Fujimoto(2018) 등이 제시한 DDPG기반의 TD3 알고리즘을 사용한다.

Table 1의 TD3 알고리즘을 이용하여 Fig. 2와 같이 2개의 Actor 심층신경망, 4개의 Critic 심층신경망, 그리고 Replay Buffer로 에이전트를 구성한다.

Table 1. TD3 algorism

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ
with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta_1' \leftarrow \theta_1$, $\theta_2' \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$, (1)
  $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s' (2)
  Store transition tuple (s, a, r, s') in $\mathcal{B}$ (3)

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$ (4)
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \mathrm{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$ (5)
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \tilde{a})$ (6)
  Update critics $\theta_i \leftarrow \mathrm{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$ (7)
  if t mod d **then**
    Update ϕ by the deterministic policy gradient: (8)
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$ (9)
    Update target networks:
    $\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i'$
    $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$ (10)
  end if
end for

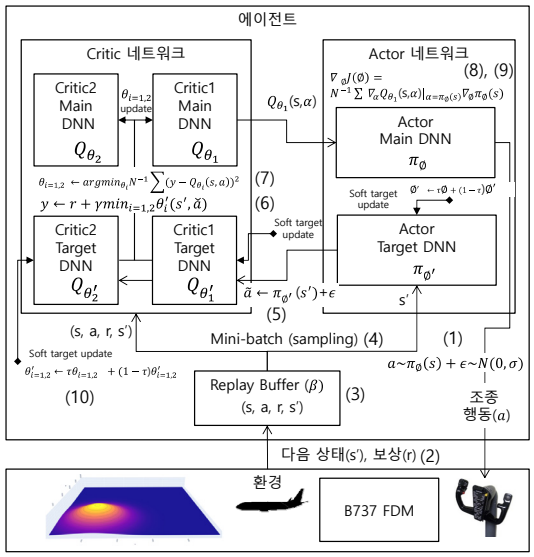

Fig. 2. Diagram of TD3 algorithm

Actor는 주어진 상태에서 최고의 보상을 받을 수 있는 최선의 행동을 선택하는 정책을 심층신경망을 통해 학습하고, Critic은 Actor의 정책에 대한 가치평가를 심층신경망을 통해 학습하게 된다. Critic의 가치평가 결과는 다시 Actor의 정책을 개선하는 데 사용되어, 결국 학습이 진행될수록 Actor의 정책은 최적화된다. Fig. 2에서 학습과정을 살펴보면, Actor와 두 개의 Critic Main 심층신경망 π_ϕ, Q_{θ_1}, Q_{θ_2}는 각각 랜덤

파라미터 ϕ, θ_1, θ_2 로 초기화되고 각각의 Target 심층신경망 $\pi_{\phi'}$, $Q_{\theta_1'}$, $Q_{\theta_2'}$, 의 파라미터 역시 ϕ, θ_1, θ_2 으로 초기화한다.

학습이 진행되면, π_ϕ에 의해 상태 s에서 행동 a를 선택하고 여기에 다양한 행동을 취할 수 있도록 랜덤 노이즈 ϵ를 추가하여 환경에 전달한다(1). 환경에서는 행동 a에 따른 보상 r과 그 다음 상태 s'를 에이전트에 전달한다(2). 일련의 환경과 에이전트가 주고받은 Transition, 즉, (s, a, r, s')들은 Replay Buffer β에 저장되고(3), Buffer가 차면, Mini-Batch 과정을 통해 랜덤하게 N개의 Transition 샘플들을 뽑아 $\pi_{\phi'}$와 $Q_{\theta_1'}$, $Q_{\theta_2'}$에서 사용한다(4). 이는 연속된 Transition들을 사용할 경우 발생할 수 있는 Correlation 문제를 방지하기 위함이다.

계속해서 $\pi_{\phi'}$에 의해 상태 s'에 대한 행동 $\tilde{\alpha}$가 선택되고(5), $Q_{\theta_{i=1,2}}$에서 $\tilde{\alpha}$에 대한 행동가치평가 $Q_{\theta_{i=1,2}'}$ $(s', \tilde{\alpha})$를 수행하고, 둘 중 작은 값을 선택하여 목적함수 y를 계산한다(6). 이는 단일 심층신경망을 사용함으로써 발생할 수 있는 Overestimation Bias 즉, 과평가에 의한 Actor의 잘못된 정책으로의 학습편향을 방지하기 위함이다(Fujimoto et al., 2018). 계속해서 y와 β에서 가져온 (s, a)에 대한 행동가치평가 $Q_{\theta_{i=1,2}}$ (s, a)에 대하여 loss가 최소가 되는 값으로 $\theta_{i=1,2}$를 갱신한다(7).

한편, π_ϕ은 Critic의 $Q_{\theta_1}(s, a)$를 취하여 Policy Gradient Theorem를 이용하여 정책 기울기(policy gradient)가 최대가 되는 값으로 파라미터 ϕ를 갱신한다(8)(9). 파라미터 ϕ'와 $\theta'_{i=1,2}$에 대한 갱신은 학습이 이루어지는 매 d step 마다 τ를 이용한 Soft target update 방식을 사용하여 갱신한다(10). 이를 통해 안정적인 네트워크의 갱신을 통해 안정적인 학습을 수행할 수 있다(Fujimoto et al., 2018).

Table 2와 3은 본 논문에서 구성한 Actor 네트워크와 Critic 네트워크의 심층신경망 설정이며, Table 4는 TD3 알고리즘에서 사용된 파라미터 설정이다.

2.4 학습 환경의 구성

에이전트는 학습 환경을 통해 다양한 지형 충돌 회피에 대한 조종 경험을 쌓는다. 이를 위해 환경은 Fig. 3과 같이 '비행동역학 모델'과 지형 장애물 정보 제공

Table 2. Actor network parameters

Network name	Number of nodes	Activation function
Input layer	8	-
Hidden layer 1	64	relu
Hidden layer 2	128	relu
Hidden layer 3	64	relu
Hidden layer 4	32	relu
Output layer	4	tanh

Table 3. Critic network parameters

Network name	Number of nodes	Activation function
Input layer	12	-
Hidden layer 1	64	relu
Hidden layer 2	128	relu
Hidden layer 3	64	relu
Hidden layer 4	32	relu
Output layer	1	-

Table 4. TD3 algorithm parameters

파라미터	설명	값
γ	Discount factor	0.99
N	Mini batch size	512
σ	Noise std dev (Main)	0.2
c	Noise clip	0.1
$\tilde{\sigma}$	Noise std dev (Target)	0.2
τ	Interpolation factor	0.005
d	Policy update delay	2

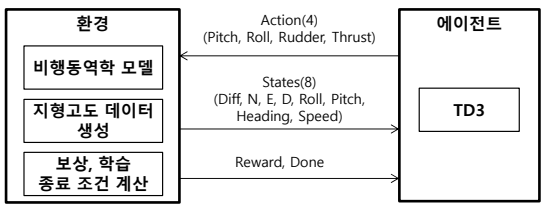

Fig. 3. Interaction between learning agent and environment in reinforcement learning

을 위한 '지형고도 데이터 생성' 그리고 에이전트가의 행동(action)에 대한 평가를 위한 '보상(reward) 및 학습종료 조건 계산'으로 구성한다. 여기서, 환경과 에이전트가 주고받는 상태 값과 행동 값은 각각 Table 5, 6과 같다. 상태 값은 항공기의 위치, 자세 값과 20초 후 위치에 대한 표고 값 등이 되고 행동 값은 항공기 조종명령 값이 된다. 여기서 20초 후의 위치에 대한 표고 값은, 항공기의 방향 벡터 D, 현재 항공기 위치 P, 현재 속력 v라고 할 때 t초 후 항공기의 위치 P'는 $\overrightarrow{P'} = \overrightarrow{D} \times v \times t + \overrightarrow{P}$를 통해 구해진다. 여기서 위치 P'에서의 지형고도 값은 '지형고도 데이터 생성'을 통해 P'의 North, East 좌표에 따른 지형고도 값을 얻을 수 있다. 이때 위치 P'의 지형고도 값이 P'의 고도 값보다 같거나 높으면 충돌로 처리한다.

조종명령 'Action[0]'의 값 '-1'은 조종간의 좌우 동작 범위의 왼쪽 최대치를 의미하며 '1'은 오른쪽 최대치를 의미한다. 마찬가지로 'Action[1]'의 값 '-1'은 조종간의 전후 동작 범위의 앞쪽(forward) 최대치, '1'은 뒤쪽(backward) 최대치를 의미한다. 'Action[2]'의 값 '-1'은 왼쪽 러더(rudder) 최대치, '1'은 오른쪽 러더 최대치를 의미한다. 마지막으로 'Action[3]'의 값 '0'은 Min Thrust 상태, '1'은 Max Thrust 상태를 의미한다.

Table 5. Agent action definition

Actions	값 설명	범위
Action[0]	조종간 Aileron command	-1(좌)~1(우)
Action[1]	조종간 Elevator command	-1(전)~1(후)
Action[2]	조종간 Rudder command	-1(좌)~1(우)
Action[3]	Thrust command	0(Min)~1(Max)

Table 6. Reinforcement learning state definition

States	값 설명	단위
State[0]	Track 기준 20초 후 위치에 대한 지형고도 값	meter
State[1]	NED 좌표의 기준 North 방향 항공기 위치	meter
State[2]	NED 좌표의 East 방향 항공기 위치	meter
State[3]	NED 좌표의 Down 방향 항공기 위치	meter
State[4]	항공기 Bank 각도	degree
State[5]	항공기 Pitch 각도	degree
State[6]	항공기 Heading 각도	degree
State[7]	항공기 속력	meter/sec

2.4.1 비행동역학모델

비행동역학모델은 오픈소스 기반 6DOF FDM 라이브러리, JSBSim을 사용한다. JSBSim은 1996년도에 개발되어 현재까지 C-172, A320, B737 등과 같은 상업용 항공기 및 F-16, F-15 등의 군용기를 포함한 다양한 기종에 대한 6DOF FDM들이 공개 및 사용되어지고 있다. 본 논문에서는 JSBSim B737 FDM을 사용한다.

2.4.2 지형고도 데이터 생성

CFIT 자율 회피 기동을 에이전트가 학습하기 위해서는 다양한 지형장애물에 대한 회피 경험을 에이전트에게 제공할 수 있어야 한다. 따라서 산악 지형과 같이 거리가 가까울수록 지형 장애물의 표고 값이 증가되는 지형고도 데이터 생성 모델을 이용하여 다양한 지형장애물 환경을 생성할 수 있어야 한다.

본 논문에서는 이러한 다양한 높이와 경사를 갖는 지형장애물의 고도 데이터를 생성하기 위해 실제 지형의 고도 정보를 이용하기보다는 2차원 정규분포를 이용한 지형고도 데이터 생성 모델, $h = 400 * N(\mu, \sigma^2)$을 사용한다. 여기서 N은 가우시안 정규분포 함수이며, 이때 표고 h는 평균 μ와 표준편차 σ 값을 통해 정의되며, 400은 표고 높이를 조절하기 위한 상수이다.

학습을 위한 비행영역은 N(north) 방향으로 0~10km, E(east) 방향으로 0~10km이며 중심좌표는 N 방향 5km, E 방향 5km 지점이 되도록 설정한다. 이 비행영역에 대하여 2차원 10,000×10,000 배열에 지형고도 데이터 생성 모델을 적용하여 1미터 단위의 2차원 고도 데이터를 μ, σ값 변경을 통해 학습 시 동적으로 생성할 수 있게 한다. Table 7은 학습 수행 시

Table 7. Terrain obstacle center location and elevation range of μ, σ

	평균(μ)	표준편차(σ)
설정 범위	-0.5<μ<0.5	0.07<σ<0.1
생성 범위	N, E:2.5km~7.5km	636m~909m

랜덤하게 선택될 μ, σ 값의 변경 범위이며 그에 따른 지형장애물의 중심위치 이동 범위와 가장 높은 지형고도의 범위이다. μ는 0.05 단위로, σ 값은 0.001 단위로 변경한다.

Fig. 4는 본 논문에서 사용한 지형고도 데이터 생성 모델을 이용하여 생성한 일부 지형고도 데이터를 가시화시킨 결과이다.

2.4.3 보상 및 종료 조건 설정

환경은 에이전트의 조종 행동에 대한 CFIT 회피 결과를 매 학습 시점(step)마다 보상으로 제공한다. 이러한 보상을 통해 에이전트는 자신의 학습 정책(policy)을 수정하여 보다 많은 양(positive)의 보상을 받을 수 있는 행동을 수행하게 된다. 또는 추락, 지형 충돌, 목표지점과 너무 멀어질 경우 등에 대해서는 음(negative)의 보상도 제공한다. 또한 더 이상 학습을 진행할 수 없는 경우 등에 대해서 학습을 종료하고 새로운 학습을 진행할 수 있도록 한다. 본 논문에서 설정한 이러한 보상 및 종료 조건은 Table 8과 같다.

2.5 학습 수행

2.5.1 학습 시나리오

학습을 위한 시나리오는 Fig. 5와 같이 가로 10km, 세로 10km의 비행영역으로 설정한다. 비행영역에는 지형고도 데이터 생성 모델을 이용하여 지형 생성 영역에 랜덤하게 지형 장애물의 고도 데이터가 생성된다. B737 FDM의 초기 설정 값은 Table 9와 같다. 항공기는 초기위치(E: 5km, N: 0km)에서 출발하며, 초기 속력은 100m/s, 고도는 400m로 설정한다.

2.5.2 학습 진행

학습에 사용한 컴퓨터는 Intel CPU i7 3.0Ghz, RAM

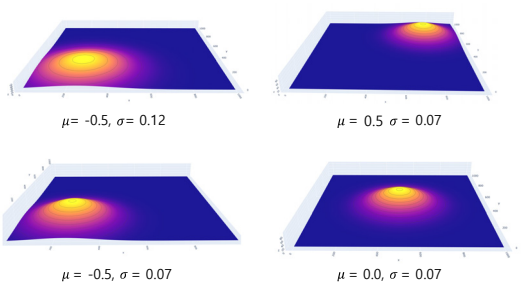

Fig. 4. Results of terrain obstacles generation

Table 8. Conditions for getting rewards

조건	Reward	Done	비고
충돌 감지 이후 Pitch 값이 전 Step보다 크지 않을 경우	-1	Episode 진행	충돌 경고 후 CFIT 회피 기동을 하지 않을 경우, 실속 속도에 가까워질수록 음의 보상 부여
80m/s 이하로 속력이 내려갈 경우	-1	Episode 진행	
목표지점으로 최단거리로 비행 시	+1	Episode 진행	항공기가 CFIT 회피 기동이 아닐 경로점 비행유지를 위한 보상 부여
목표지점과 Heading이 5도 이내로 유지할 경우	+1	Episode 진행	
목표지점 500미터 안으로 도착	+200	Episode 종료	목표성공 보상 부여
추락 또는 장애물 충돌 시	-200	Episode 종료	목표실패 보상 부여

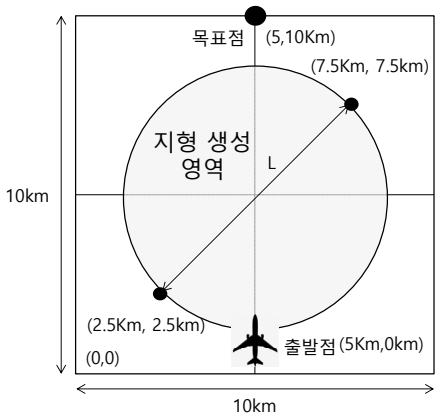

Fig. 5. Initial position setup for learning scenario

Table 9. Initial B737 FDM setup for learning scenario

Actions	값 설명	비고
Weight	48,534	kg
초기위치	N: 0km E: 5km	NED 좌표
초기고도	400	meter
초기방위	0	degree
속력	100	meter/sec
Configuration	Clean	Landing Gear Up, No Flap

32GB, 비디오카드는 GeForce RTX 2070, 학습도구는 Tensorflow를 사용하였다.

학습 진행은 Fig. 6과 같이 진행한다. 학습이 시작되면 B737 FDM을 초기화하고 최초 에이전트의 조종행동 값을 이용하여 비행을 시작하게 된다. 고도가 '0'이 되거나 현재 위치에서의 지형고도보다 낮으면 충돌로 처리되고 음의 보상을 받는다. 만약 에이전트가 충돌을 회피하게 되고 목표지점에 도착하게 되면 목표달성에 대한 양의 보상을 받게 되고 학습목표달성 여부에 따라 지형고도 데이터를 교체하고 다시 학습을 수행할지 기존 지형고도 데이터를 가지고 재학습을 수행할지 결정하게 된다. 여기서 학습목표는 '목표 지점 n회 이상 도착' 등의 조건이 된다. 이 과정을 여러 번 반복하여 학습종료조건 달성 여부에 따라 재학습 또는 학습을 종료한다. 학습종료 조건은 'm회 이상 학습목표 달성' 등이 된다. 본 논문에서 설정한 학습목표와 학습종료조건은 Table 10과 같다.

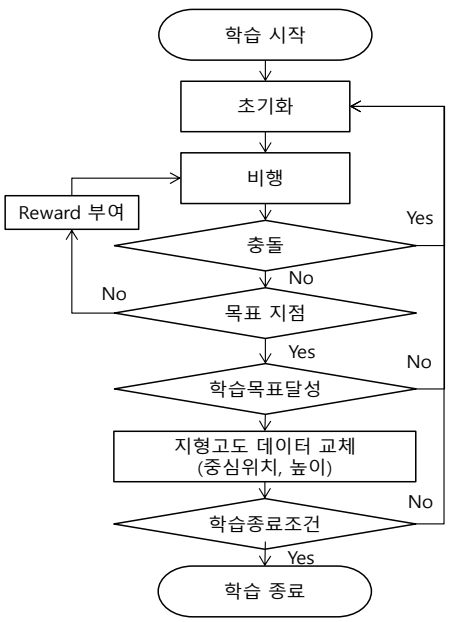

Fig. 6. Chart of learning flow

Table 10. Conditions for learning goal and completion

	조건
학습목표	목표지점 500미터 근접 또는 누적비행거리 10km 이상 연속 3회
학습종료조건	학습목표달성율이 70%이상 시(학습목표 달성 Episode 수/총 Episode 수)

2.5.3 학습 결과

Fig. 6의 과정을 통해 학습을 수행하였다. Fig. 7은 Episode 1회 당 비행거리를 그래프로 표시하였다. Episode란, Fig. 6에서 '초기화'가 수행되고 다음 '초기화'가 수행될 때까지의 과정을 의미한다. Fig. 7을 보면, 학습 초반에는 2km 남짓 비행하다가 1,300 Episode 이후에는 10km 이상을 매 Episode마다 비행하는 것을 볼 수 있다. 충돌 없이 목표지점 근접 또는 10km 이상 비행하였을 경우 학습목표를 달성한 것이기 때문에 2,000 Episode 이후부터는 학습목표 달성률이 70% 이상이 되어 학습종료 조건을 만족하여 학습이 종료된 것을 알 수 있다. Fig. 8은 Episode 1회당 받은 보상 점수를 그래프로 도시하였다. 이 그래프에서도 1,300 Episode 이후에는 일정한 범위의 보상을 지속적으로 받는 것으로 보아, 학습이 완료되었다는 것을 알 수 있다.

2.6 학습 결과를 이용한 CFIT 자율 회피 실험

학습된 에이전트의 학습 성능을 실험하기 위해 Table 11과 같이 3종류의 CFIT 상황을 주고 자율 회피

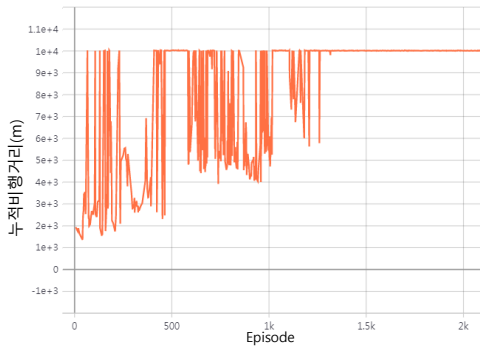

Fig. 7. Distance per episode

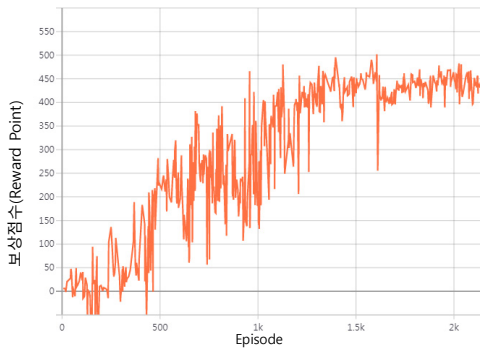

Fig. 8. Rewards per episode

Table 11. Experimental list for agent learning performance

실험종류	조건
실험1	위치가 동일하고 최고 높이가 각각 1,000m, 700m, 500m인 지형장애물에 대한 충돌 회피 실험
실험2	동일한 지형장애물에 대해서 충돌 경고 시간을 10초, 20초로 다르게 설정했을 경우, 충돌 회피 실험
실험3	출발지, 목적지 부근의 동일 고도 지형장애물에 대한 충돌 회피 실험
항공기 초기 조건	학습 시와 동일 조건으로 실험 1,2,3에 적용

실험을 구성하였다.

2.6.1 실험 1

항공기의 초기 조건은 Table 9와 동일하게 설정하였다. 지형장애물은 지형고도 데이터베이스를 이용하여 3개의 지형 장애물 A, B, C를 구성하였다. 지형 장애물의 최고 높이는 각각, 1,000m, 700m, 500m로 설정하였다. 중심 위치는 비행영역 중심인 가로 5km, 세로축 5km 지점에 위치시켰다. 지형 A는 에이전트가 학습 과정에서 학습한 최고 지형 장애물 높이 900m보다 100m가 높고 지형 B는 최저 높이 600m보다 100m가 낮다.

Fig. 9는 에이전트가 항공기를 조종하여 지형 장애물 A, B, C에 대한 자율 회피 기동을 수행한 궤적을 보여준다. 빨간색 점은 시작위치를, 파란색 점은 도착지점을 표시한다. 3개 지형 장애물에 대한 자율 회피를 성공적으로 수행하고 목적지 부근까지 비행한 것을 볼 수 있다. Fig. 10은 지형 장애물 A, B, C 회피 조작 시의 시간에 따른 항공기 고도를 보여준다.

2.6.2 실험 2

학습 시 현재 속도 기준으로 20초 후의 위치에 대한 충돌경고를 받게 하였다. 초기 속도 100m/s 기준 대략 2km 전방에 대한 충돌 경고를 받게 된다. 실험에서는 이 시간을 10초와 20초로 설정하였다.

Fig. 11, 12는 지형 B를 이용하여 항공기 예상 충돌 경고 시간을 다르게 하여 학습된 에이전트가 자율 회피를 수행한 결과를 보여준다. Fig. 11에서 (a)는 10초 설정 궤적, (b)는 20초 설정한 궤적이다. Fig. 12는 시

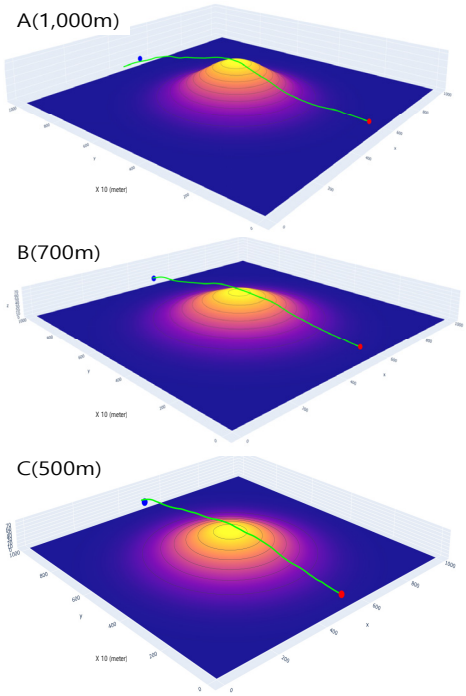

Fig. 9. Results of CFIT autonomous avoidance by terrain obstacles. A(1,000m), B(700m), C(500m)

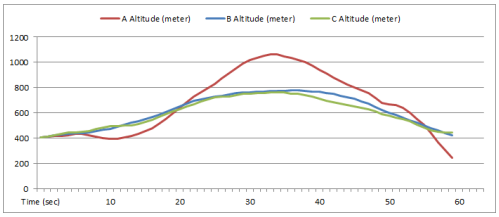

Fig. 10. Altitude comparison by terrain obstacle A, B and C

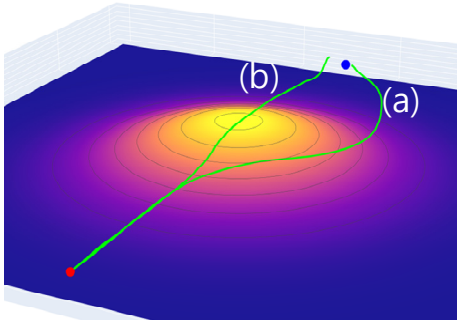

Fig. 11. Path comparison by conflict warning time. (a) 10 sec (b) 20 sec

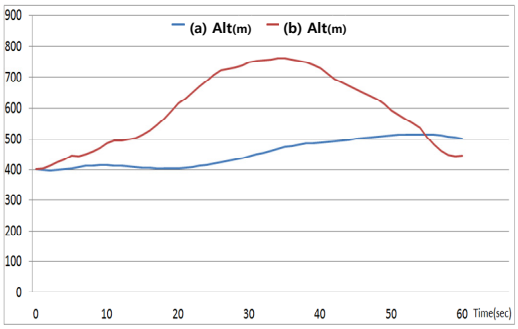

Fig. 12. Altitude comparison by conflict warning time. (a) 10 sec (b) 20 sec

간에 따른 고도를 보여준다. (b) 궤적은 출발 시부터 충돌 경고를 받아 바로 고도를 상승시켜 수직 방향의 회피를 수행하는 반면에 상대적으로 늦은 시간에 경고를 받은 (a) 궤적은 지형 장애물에 대해 충돌 가능성이 높은 수직 방향의 회피가 아닌 수평 방향의 회피를 수행하는 것을 볼 수 있다.

2.6.3 실험 3

마지막으로, 학습된 에이전트를 이용하여 위치와 높이가 상이한 지형에 대한 자율 회피 실험을 하였다. Fig. 13은 높이가 1,300m, 중심위치가 N 방향 4km, E 방향 4km인 지형에 대한 자율 회피 결과를 보여준다. Fig. 14는 높이가 1,100m, 중심위치가 N 방향 7km, E 방향 7km인 지형에 대한 자율 회피 결과를 보여준다. 실험결과에서 볼 수 있듯이 출발지 또는 목적지 근처에 장애물이 있을 경우에도 충돌 가능성이 있는 수직 회피보다는 수평 회피를 수행하는 것을 볼 수 있다.

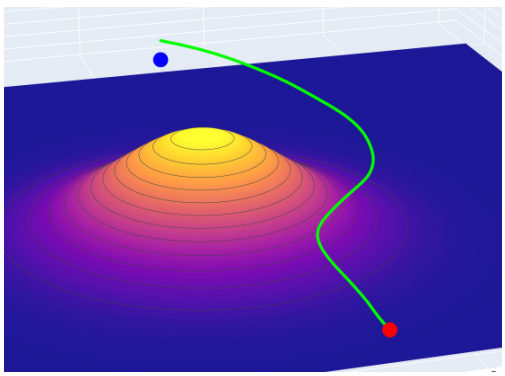

Fig. 13. Results of CFIT autonomous avoidance by terrain obstacles. Height 1,300m. (N: 4km, E: 4km)

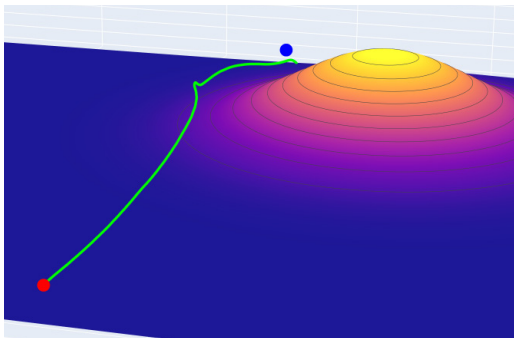

Fig. 14. Results of CFIT autonomous avoidance by terrain obstacles. Height 1,100m. (N: 7km, E: 7km)

Ⅲ. 결 론

오늘날 CFIT 사고에 대하여 살펴보면 ICAO는 고위험 사고유형으로, IATA는 세 번째로 높은 치명적 사고유형으로 분류하고 있으며, FAA를 포함한 국내·외 일반항공(general aviation)과 군용기 분야에서도 높은 사고율을 보이고 있다. 그러나 지금까지 국내·외적으로 CFIT사고 방지를 위한 노력으로 human error를 방지할 수 있는 공학적 대책은 경보시스템(TAWS)에 머물러 있으며, 자동적으로 회피기동까지 수행할 수 있는 공학적 대책은 적용되지 않고 있다.

한편 CFIT 사고의 특성상 사람의 개입이 전제되어 있는 Autopilot 시스템보다는 기계가 현재 상황을 판단하고 인간의 개입 없이 적절하게 항공기를 제어하는 Autonomous Autopilot 시스템이 CFIT 사고 방지를 위한 필요한 대책이라 할 것이다. 이러한 필요성에 따라 본 논문에서는 CFIT 사고 방지를 위한 공학적 대책의 하나로, 심층강화학습 기반의 에이전트 활용을 제시하고, 학습 환경을 구성하여 에이전트를 학습시킨 후, 학습된 에이전트가 여러 CFIT 상황에서 항공기를 조종하여 자율 회피 수행이 가능함을 시뮬레이션 실험을 통해 확인하였다.

다만, 학습된 에이전트의 조종 행동이 사람과는 달리, 필요 없는 러더 조작을 하거나, 수평 비행 상황에서도 지속적으로 조종간을 흔드는 등의 거친 조종 행동에 대한 개선이 필요하며, 본 논문의 실험으로 다루지는 않았지만 학습 시 경험하지 못한 복수의 지형 장애물에 대한 회피 실험에서 회피 성공률이 현저히 낮았는데, 이는 학습 시 보상 설계를 보완하여 해결해야 할 문제로 식별되었다.

이후에는 본 연구를 확장시켜 지형 데이터베이스 외에 레이다의 수평·수직의 지형 탐지정보 및 카메라 영상정보를 통합하여 학습시킨다면, 좀 더 강인한 CFIT 자율 회피를 수행할 수 있는 에이전트가 될 수 있을 것이라 기대한다.

References

1. Russell, S., and Norvig, P., “Artificial Intelligence—A Modern Approach”, Prentice Hall, Hoboken, New Jersey, 2009, pp.30-32.
2. Shin, S. J., Jo, C. R., Jeon, H. S., Yoon, S. H., and Kim, T. Y., "A survey on deep reinforcement learning libraries“, ETRI, 34(6), 2019, pp.87-99.
3. Yun, H. J., Park, N. S., Yoon, J. K., and Son, Y. S., “Research trends on deep reinforcement learning”, ETRI, 34(4), 2019, pp.1-14.
4. Wo, J. H., “Collision avoidance for an unmanned surface vehicle using deep reinforcement learning”, Ph.D. Thesis, Seoul National University, Seoul, Feb 2018.
5. Sharma, T., “Optimum flight trajectories for terrain collision avoidance”, Master’s Thesis, Royal Melbourne Institute of Technology University, Melbourne, Australia, Mar 2006.
6. Baomar, H., and Bentley, P. J., “Autonomous navigation and landing of airliners using artificial neural networks and learning by imitation”, 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, Hawaii, USA, 2017.
7. Kim, J. S., “Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay”, Applied Sciences, 10(2), 575, 2020, pp.5-6.
8. Fujimoto, S., Hoof, H., and Meger, D., “Addressing function approximation error in actor-critic methods”, Proceedings of the 35th International Conference on Machine Learning, PMLR (Proceedings of Machine Learning Research), Stockholmsmässan, Stockholm, Sweden, 2018, pp.1587-1596.
9. Xie, J., Peng, X., Wang, H., Niu, W., and Zheng, X., “UAV autonomous tracking and landing based on deep reinforcement learning strategy”, Sensors, 20(19), 5630, 2020, pp.7-13.
10. Moon, I. C., Kim, J. M, and Kim, D. J., “Modeling and simulation on One-vs-One air combat with deep reinforcement learning”, Journal of the Korea Society for Simulation, 29(1), 2020, pp.39-46.
11. Meyer, E., Heiberg, A., Rasheed A., and San, A. O., “COLREG-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning”, IEEE Access, 8, 2020, pp.165344-165364.
12. Young, C. S., “Warning system concepts to prevent controlled flight into terrain (CFIT)”, AIAA/IEEE Digital Avionics Systems Conference, Fort Worth, TX, USA, 1993, pp.463-474.
13. Zhang, Y., Antonsson, E. K., and Grote, K., “A new threat assessment measure for collision avoiddance system”, 2006 IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 2006, pp.968-975.
14. Källström, J., and Heintz, F., “Reinforcement learning for computer generated forces using open-source software”, Interservice/Industry Training, Simulation, and Education Conference, Orlando, FL, USA, 2019, Paper No. 19197, pp.1-11.