모방학습을 활용한 항공기 충돌회피 정책 네트워크 모델링

(Policy Network for Aircraft Colision Avoidance Modeling via Immitation Learning)

요 약

강화학습을 로보틱스와 제어에 적용한 사례들은 근래에 들어 점점 늘고 있다. 강화학습을 적용하여 제어를 한 경우가 사람이 만든 알고리즘보다 우수한 경우들이 존재하기 때문인데, 다양한 알고리즘들 중 연속적인 제어를 요구하는 로보틱스 영역에서는 Actor Critic을 최적화한 PPO 알고리즘이 주로 사용된다. 본 연구는 이러한 강화학습을 진행하기에 앞선 선행연구이다. 본 연구는 강화학습의 Actor와 Critic의 네트워크로 사용되기 적합한 구조를 찾기위해 모방학습의 방식으로 학습된 여러 다른 구조의 네트워크들을 비교하여 최적의 구조를 찾아낸다.

ABSTRACT

Recently, researches which apply reinforcement learning to robotics and control field are increasing. This is because there are cases in which control by applying reinforcement learning is supereior to that of man-made algorthms. Among various algorithms, the PPO algorithm optimized for Actor Critic is mainly used in the robotics domain that requires continuous control. This paper is a preceding study, prior to such reinforcement learning. In this paper, in order to find a structure that is suitable for using as a network of Actor and Critic of reinforcement learinig, the oprimal structure is found by comparing networks of different structures trained by the method of imitation learinig.

키워드 : 충돌회피, 모방학습, 지도학습, 정책 네트워크

Keywords : Colision Avoidance, Immitation Learning, Supervised Learning, Policy Network

Ⅰ. 서 론

최근 로보틱스와 같은 제어 분야에서 객체간의 충돌 회피를 목적으로, 강화학습을 이용한 기술의 개발이 진행되어 왔다[1]. 대부분의 로보틱스 기술은 연속적인 제어와 고차원적인 동역학 모델을 요구하는데, 이를 강화학습에 적용하기 위해선 최적화된 Actor Critic 방식[2]인 PPO 알고리즘[3]의 사용이 적합하다. 본 연구는 항공기 충돌회피를 강화학습으로 진행하기에 앞서 PPO 알고리즘의 Actor와 Critic에 사용되기에 적합한 뉴럴 네트워크 구조를 찾는 방법으로 모방학습을 이용해 이를 알아보는 선행연구이다. 먼저 모방학습의 데이터를 샘플링 하기 위해 항공기 충돌 시뮬레이션을 설계한다. 여기서, 레이더에서 제공하는 정보와 동일한 5가지의 정보 $(r, v_c, \phi, \frac{d\phi}{dt}, \frac{d\theta}{dt})$를 통해 해당 상태에서 충돌가능성 여부를 판단하고, 회피기동 명령을 계산하여 샘플링 한다. 이렇게 생성된 레이더 정보와 회피기동 명령을 묶어서 네트워크의 입력 Feature와 출력 Label로 제공하여 학습을 진행한다. 마지막으로 학습된 결과를 모델의 파라미터 개수와 상대기와의 최소거리에 대한 표준편차, 평균값들을 사용하여 평가함으로써 가장 적합한 네트워크 구조를 평가한다. 이렇게 결정된 네트워크를 강화학습의 Actor와 Critic의 네트워크로 전이하여 학습시킴으로써 강화학습의 학습 효율을 높일 수 있다.

Ⅱ. 모방학습 데이터 생성

네트워크의 학습은 총 30만개의 학습데이터와 9만개의 검증데이터로 진행된다. 학습 데이터들은 뒤따르는 시뮬레이션과 알고리즘을 사용하여 입력 레이더 정보 5개와 출력 회피 명령 정보 1개 (3종류, 상승명령, 하강명령, 유지명령)가 한 데이더셋으로 구성된다.

기호

r : 상대기와 본체간의 상대거리
v_c : 상대기의 접근속도
ϕ : Azimuth
θ : Line of sight
t : 경과 시간
h : 고도

1. 시뮬레이션 설계

본체의 시야 밖에 존재하는 항공기의 회피는 고려하지 않는다는 가정하에, 매 시뮬레이션은 그림. 1.과 같이 상대기가 본체로부터의 충돌지점에 $\pm 50^\circ$의 각도로 접근하도록 한다. 또한 최적의 회피 명령에 대한 데이터를 생성하기 위해, 상대기는 늘 같은 고도가 아닌, 본체의 고도를 기준으로 $\pm 200(m)$에서 본체에 접근한다.

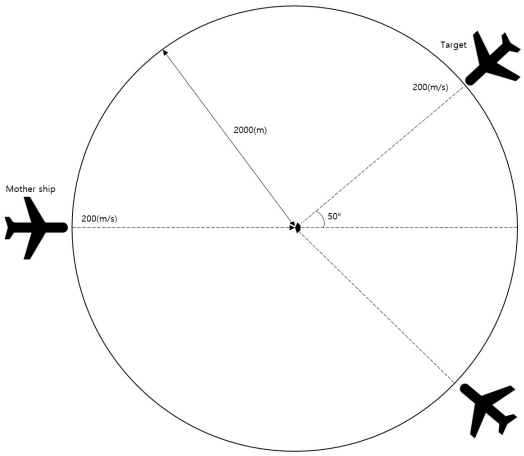

그림. 1. 본체와 상대기의 충돌시뮬레이션 개요

2. 알고리즘 설계

충돌 여부를 감지하는 방식은 입력 정보 5가지를 통해 3가지의 파라미터를 계산하여 결정한다. 각각 수직 최소거리(MDV), 수평 최소거리(MDH), 현재 수직 고도 차이(d_c)이고, 각 파라미터들은 아래와 같은 수식으로 계산된다.

$$MDV = \frac{r^2}{v_c} \times \frac{d\theta}{dt} \qquad (1)$$

$$MDH = \frac{r^2}{v_c} \times \frac{d\phi}{dt} \qquad (2)$$

$$d_c = r \times \theta \qquad (3)$$

이렇게 계산된 파라미터들로 그림. 2.의 알고리즘을 통해 충돌여부를 감지하고, 회피명령을 주게 된다. 여기서 $\dot{h}_{cmd}$는 고도 변화율(회피기동) 명령이고, d_s는 최소 회피거리이다(상대거리가 이 거리보다 가까워지면 충돌로 간주한다).

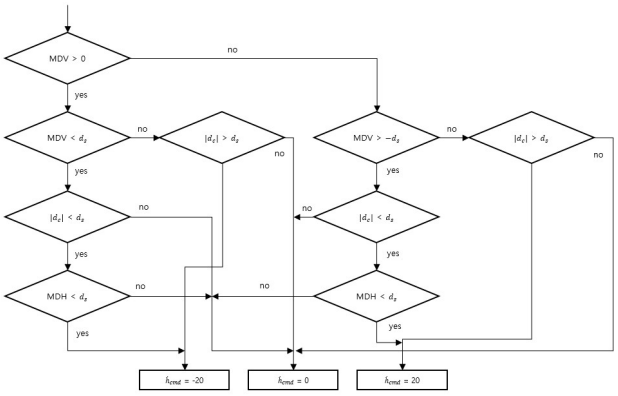

그림. 2. 회피기동 명령 알고리즘 개요

Ⅲ. 네트워크 설계

네트워크의 구조를 변경하며 가장 적합한 구조를 찾기 위해 그림. 3.과 같이 ResNet구조로부터 영감을 받아 세가지 블록을 만들어 변화시킬 수 있도록 설계하였다. 각 블록은 Fully connected layer들로 구성되며, 활성함수로 음수 영역의 기울기를 0.1을 가지는 Leaky ReLU를 사용한다.

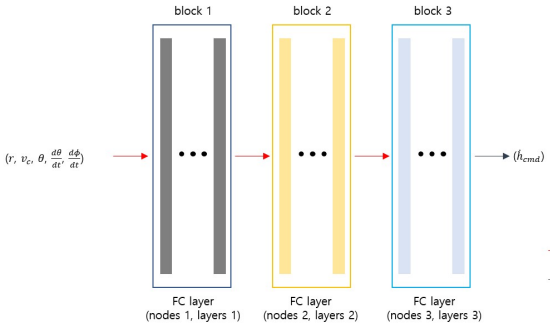

그림. 3. 설계한 네트워크 구조 개요

Ⅳ. 실 험

학습을 사용될 하이퍼 파라미터로 Learning rate는 $1e-3$, Batch size는 300을 사용했고, Cross Entropy loss 함수와 20 epoch의 반복도를 가지는 Step LR scheduler을 사용하였다. 각 블록의 층수는 [1, 1, 1]과 [2, 2, 2], 각 층의 노드수는 20에서 80까지 20간격으로 변화시키며 실험을 진행하였다. 결과의 표는 공간상의 이유로 참고문헌[4]에 표기해두었다. 결과적으로 최종 모델은 노드수 [40, 20, 60], 층수 [2, 2, 2]로 결정되었고 해당 네트워크의 성능은 그림. 4.에 표시된 Confusion matrix로 확인 할 수 있다.

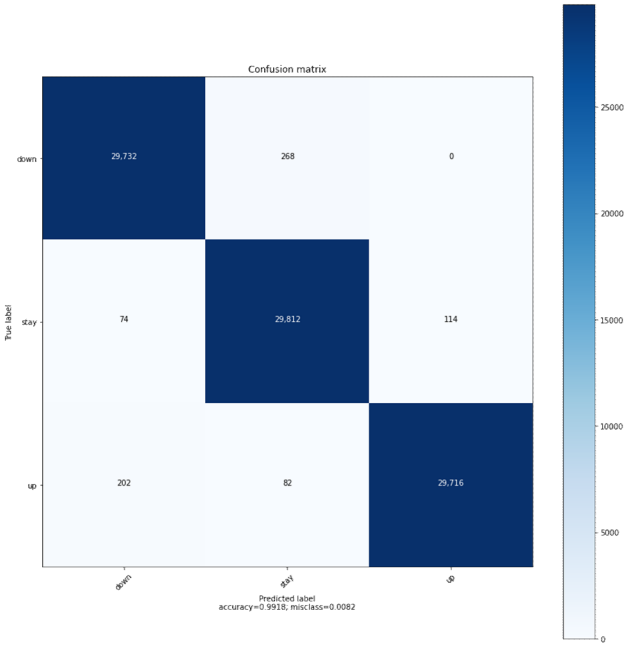

그림. 4. 최종 모델의 Confusion matrix

Ⅴ. 결 론

본 연구에서는 강화학습을 활용한 항공기 회피 연구에 앞서 Actor와 Critic의 모델로 어떤 구조의 네트워크가 가장 적합한지 모방학습을 통해 확인해보았다. 추후에 본 연구에서 확인한 모델을 토대로 PPO 알고리즘을 통해 충돌회피 연구를 진행할 예정이다.

References

[1] Y. F. Chen, M. Liu, M. Everett, J. P.How, “Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning,” Retrieved Jan., 07, 2021, from https://arxiv.org/pdf/1609.07845.pdf.

[2] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicarap, T. Harley, D. silver, K. Kavukcuoglu “Asynchronous Methods for Deep Reinforcement Learning,” ICML, 2016.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, “Proximal Policy Optimization Algorithms,” Retrieved Jan., 07, 2021, from https://arxiv.org/pdf/1707.06347.pdf.

[4] K. W. Park, Immitation Learining(2020), Retrieved Jan., 07, 2021, from https://github.com/kun-woo-park/Imitation_learning.