공학석사 학위논문

심층강화학습을 이용한 선박의 충돌회피 시뮬레이션

Simulation of Collision Avoidance for Vessel Using Deep Reinforcement Learning

지도교수 이 장 세

2020년 8월

한국해양대학교 대학원

컴 퓨 터 공 학 과

여 동 규

본 논문을 여동규의 공학석사 학위논문으로
인준함

위원장 : 김 재 훈 (인)

위 원 : 박 휴 찬 (인)

위 원 : 이 장 세 (인)

2020년 7월 16일

한국해양대학교 대학원

목 차

List of Tables

List of Figures

심층강화학습을 이용한 선박의 충돌회피 시뮬레이션

여 동 규

한국해양대학교 대학원
컴퓨터공학과

초록

선박 사고는 매년 꾸준히 증가하고 있으며, 인적과실은 선박 사고의 원인 중 높은 비중을 차지하고 있다. 선박 사고는 선박의 구조 손상뿐 아니라 인명피해, 기름 유출로 인한 환경오염 등의 다양한 문제로 발전할 수 있다. 이에 따라 항해사의 의사결정을 돕는 시스템이 요구되고 있으며, IT 기술의 발전 및 신뢰도 향상으로 기존의 시스템을 무인화하기 위한 연구가 활발히 진행되고 있다.

강화학습은 행위자가 스스로 주변환경과 상호작용을 통하여 시행착오를 겪으며 보상을 최대로 받을 수 있는 최적의 행동을 찾는 기계학습의 한 분야이다. 강화학습기반의 충돌회피는 시행착오를 통해 보상을 최대로 받을 수 있는 충돌회피 경로를 탐색한다.

본 논문은 심층강화학습을 이용하여 국제 해상충돌 예방규칙인 COLREGs 규

정을 준수하는 선박과 COLREGs 규정을 준수하지 않는 선박에 대해서 충돌회피 성능을 시뮬레이션하고 비교한다. 심층강화학습 기반의 선박은 센싱된 장애물(선박)의 위치 및 운동 정보를 기반으로 조우 상황을 판단하여 충돌회피에 대한 의사결정을 수행한다. 심층강화학습 기반의 COLREGs 규정을 준수하는 선박과 조우 상황, COLREGs 규정 준수하지 않고 사고를 유발하는 선박과의 조우 상황 등에서의 학습을 통하여 기존의 규칙이나 경험치가 없이 충돌회피를 수행할 수 있음을 보인다.

키워드: 충돌회피, 강화학습, 기계학습

Simulation of a Vessel Collision Avoidance Using Deep Reinforcement Leaning

Yeo, Donggyu

Department of Computer Engineering,
Graduate School of
Korea Maritime and Ocean University

Abstract

Ship accidents are steadily increasing every year, and human error accounts for a high proportion of the causes of ship accidents. Ship accidents can develop into various problems such as damage to the structure of ships as well as human casualties and environmental pollution caused by oil spills. To prevent ship accidents, there is a need for the systems to help sailors' decision-making and recently, research on maritime autonomous surface ships is actively underway as IT technologies develop and their

reliability improves.

Reinforcement learning is one area of machine learning which an agent experiences in trial and error by interacting with the surrounding environment and finds the optimal behavior to receive maximum reward. The collision avoidance based on reinforcement learning explores the collision avoidance path through trial and error that can receive maximum reward.

Using deep reinforcement learning, this paper simulates various scenarios of collision avoidance of ship complying with the COLREGs, which are the international regulations for preventing collisions at sea, and ship not complying with the COLREGs and compares performances of those. The vessel using deep reinforcement learning determines the encounter situation based on the location and movement information on the sensed obstacles and carries out the decision on collision avoidance. It shows that collision avoidance can be carried out without existing rules or heuristics through deep reinforcement learning with encounter situations on ship that complies with COLREGs rules and ship that are able to cause accidents without complying with COLREGs rules.

제 1 장 서 론

최근 컴퓨터 기술의 발전으로 국방, 의료, 교육, 보안과 같이 다양한 분야에서 인간을 대체할 수 있는 무인화 연구가 이루어지고 있다. 이러한 연구는 선박 분야에서도 항해사의 의사결정을 돕거나, 선박의 무인화를 위해 활발하게 이루어지고 있다[1-3]. 해양경찰청의 통계에 의하면 지난 5년간(2014~2018) 연평균 2,718건의 사고가 발생하였고, 2018년에는 3,434건의 사고가 발생하였다. 사고의 원인을 분석해보면 정비불량으로 인한 사고가 40%를 차지하고 운항부주의로 인한 사고가 33%를 차지하고 있다[4]. 이처럼 인적과실로 인한 사고가 높은 비중을 차지하고 있어서 항해사의 업무를 지원하기 위하여 IT 기술 도입에 관심이 높아지고 있다.

4차산업의 영향으로 IT 기술 중에서도 기계학습이 많은 관심을 받고 있다. 기계학습의 한 분야인 강화학습은 1950년대에 동적 프로그래밍으로 시작되었으며, 최적 제어 이론에 뿌리를 두고 있다[5]. 1980년대에 들어서 강화학습에 관한 초기 연구가 이루어졌으며, 1990년대에 알고리즘에 대한 다양한 분석들이 등장하면서 강화학습이라는 단어가 등장하였다[6]. 이후 2015년 몬테카를로 트리 탐색 알고리즘을 기반으로 한 알파고가 인간과의 바둑대회에서 승리한 바 있다[7]. 강화학습은 순차적인 의사결정 문제에서 높은 성능을 보이며, 특히 자동차, 드론, 선박 등의 분야에서 충돌회피 문제를 해결하기 위한 연구가 활발하게 이루어지고 있다.

충돌회피에 대한 연구는 1970년대부터 활발하게 연구되고 있으며[8], 크게 안전영역을 이용한 방법과 충돌위험도를 고려하는 방법으로 구분된다. 안전영역을 이용한 방법은 선박안전영역을 설정하고 안전영역 안에 장애물이 들어오면 충돌을 회피하는 방법으로 상대속도, COLREGs를 고려한

다양한 연구들이 있다[9-10]. 충돌위험도를 고려하는 방법은 TCPA(Time to closest point of approach, 최단접근시간)와 DCPA(Distance at closest point of approach, 최단접근거리)를 종합하여 충돌위험도를 측정하는 방법으로 운항자의 경험, 선박의 운동역학을 고려하는 다양한 연구가 진행된 바 있다[11-12].

자율운항은 다양한 환경적 변수와 다양한 정보를 기반으로 순간마다 의사결정을 요구하기 때문에 높은 수준의 자율운항 기술이 필요하다. 따라서 각국에서는 국가적인 차원에서 자율운항 기술을 연구할 뿐만 아니라, Rolls-Royse, Kongsberg와 같은 대형 운항사에서도 인공지능 기술을 활발히 연구 중이다[13]. Table 1.1은 현재 국가별 자율운항 개발현황을 나타내며, 어느 단계까지 자율운항 또는 무인화로 규정할 것인지에 대한 의견은 차이가 있으나 자율운항기술 개발을 위해 인공지능 기술을 적용하고자 하는 연구는 공통적이다. 특히, Kongsberg사의 자율운항 기술은 10.5 km 떨어진 호튼-모스항구 구간 중 일부 구간에서 제한적으로 자율운항에 성공하여 2019년 2월부터 정기적으로 운항하고 있다.

본 논문에서는 최근 활발하게 연구되고 있는 자율운항 시스템에 필요한 충돌회피 시뮬레이션을 위하여 다양한 수학, 과학 패키지를 제공하는 Anaconda 환경에서 충돌회피 시뮬레이션을 구축하였으며, 심층강화학습 알고리즘 중 하나인 DDPG(Deep Deterministic Policy Gradient) 알고리즘을 적용하여 COLREGs 규정을 준수하는 선박과 COLREGs 규정을 준수하지 않는 선박에 대하여 다양한 선박의 조우상황에서 성능을 검증하고자 한다. DDPG 알고리즘은 연속적인 행동에 대한 의사결정 문제에 높은 성능을 보이며 항해사의 의사결정에도 도움을 줄 수 있을 것으로 기대된다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구로서 충돌회피 규정과 강화학습을 소개한다. 제3장에서는 DDPG 알고리즘을 적용한 충돌회피 시스템 모델을 설명하고, 제4장에서는 제안한 모델의 충돌회피 성능을 분석하기 위한 다양한 선박의 조우상황에서 시뮬레이션을 수행하여 그 결과를 분석 및 검증하고 제5장에서 결론 및 향후 연구로 마무리한다.

Table 1.1 국가별 자율운항 개발현황

국가	프로젝트	특징
노르웨이	SINTEF[1)]	ICT, 해운, 조선, 금융, 보험 정부 산하기관이 협력한 자율운항 개발 사업
핀란드	친환경 자율운항 시스템	해운, 조선, 연구소등이 협력하여 자율운항 기술 개발
덴마크	자율운항선박	해운 분야 기반 조선 클러스트 (빅데이터, 블록체인)
일본	지능형해운 스마트십	일본 고유 기술 개발 분야 추진 인공지능 베이스 무인선박
중국	그린 돌핀	자율운항 벌크선 개발
영국	AAWA[2)]	자율운항선박 기술 개발 인공지능 기반 항해기술
대한민국	자율운항선박 기술개발사업	지능형 항해시스템, 기관 자동화 시스템 개발

1) SINTEF : Stiftelsen for INdustriell og Teknisk Forskning
2) AAWA : Advanced Autonomous Waterborne Applications

제 2 장 관련 연구

2.1 선박의 충돌회피

2.1.1 국제 해상충돌 예방규칙

국제 해상충돌 예방규칙(International regulations for preventing collisions at sea, COLREGs)은 선박 간의 충돌을 방지하기 위해 1972년 국제해사기구(IMO)가 제정한 규칙으로 해상에 있는 모든 선박에 적용되며 선박의 운항자는 충돌사고를 예방하기 위해 필수적으로 준수하여야 한다. COLREGs는 전 세계적으로 통용되는 규칙으로 항해사가 운항하는 유인선박 뿐만 아니라 무인선박도 COLREGs 규정을 고려하여 운항하여야 한다. 선박 항해 중 선박 간의 조우 상황은 마주침(Head-on), 횡단(Crossing), 추월(Overtaking)의 세 가지로 분류하며, 각각의 상황은 다음과 같다[14].

1) 마주치는 상황

① 두 척의 동력선이 마주치거나 거의 마주치게 되어 충돌의 위험이 있을 때는 각 동력선은 서로 다른 선박의 좌현 쪽을 지나갈 수 있도록 침로를 우현(右舷) 쪽으로 변경하여야 한다.

② 선박은 다른 선박을 선수(船首) 방향에서 볼 수 있는 경우로서 다음 각 호의 어느 하나에 해당하면 마주치는 상태에 있다고 보아야 한다

1. 밤에는 두 개의 마스트 등을 일직선으로 또는 거의 일직선으로 볼 수 있거나 양쪽의 현등을 볼 수 있는 경우
2. 낮에는 두 척의 선박의 마스트 등이 선수에서 선미(船尾)까지 일직선이 되거나 거의 일직선이 되는 경우

③ 선박은 마주치는 상태에 있는지가 분명하지 아니한 경우에는 마주치는 상태에 있다고 보고 필요한 조치를 취하여야 한다.

2) 횡단하는 상황

두 척의 동력선이 상대의 진로를 횡단하는 경우로서 충돌의 위험이 있을 때에는 다른 선박을 우현 쪽에 두고 있는 선박이 그 다른 선박의 진로를 피하여야 한다. 이 경우 다른 선박의 진로를 피하여야 하는 선박은 부득이한 경우 외에는 그 다른 선박의 선수 방향을 횡단하여서는 아니 된다.

3) 추월 상황

① 추월선은 추월당하고 있는 선박을 완전히 추월하거나 그 선박에서 충분히 멀어질 때까지 그 선박의 진로를 피하여야 한다.

② 다른 선박의 양쪽 현의 정횡(正橫)으로부터 22.5도를 넘는 뒤쪽[3]에서 그 선박을 앞지르는 선박은 추월선으로 보고 필요한 조치를 취하여야 한다.

③ 선박은 스스로 다른 선박을 추월하고 있는지 분명하지 아니한 경우에는 추월선으로 보고 필요한 조치를 취하여야 한다.

④ 추월하는 경우 두 척의 선박 사이의 방위가 어떻게 변경되더라도 추월하는 선박은 추월이 완전히 끝날 때까지 추월당하는 선박의 진로를 피하여야 한다.

2.1.2 피항의 우선순위

COLREGs에 따르면 두 척의 동력선이 서로 진로를 횡단하여 충돌위험이 있을 때 다른 선박을 우현 측에 두고 항해하는 선박이 다른 선박의 진로를 피해야 하는데, 이때 다른 선박을 우현 측에 두고 있어 먼저 피해야

3) 밤에는 다른 선박의 선미등(船尾燈)만을 볼 수 있고 어느 쪽의 현등(舷燈)도 볼 수 없는 위치를 말한다

하는 우선순위를 가진 선박을 피항선(Give-way vessel)이라 한다. 피항선이 유지선(Stand-on vessel)의 진로를 피하여야 할 경우 유지선은 그 침로 및 속력을 유지하여야 한다. 하지만 유지선에게 피항의 의무가 없는 것은 아니다. 이유를 불문하고 유지선은 양선이 아주 가까이 접근하였기 때문에 피항선의 동작만으로 충돌을 피할 수 없다고 판단할 때에는 충돌을 피하기 위한 최선의 협력동작을 취하여야 한다. 충돌위험에 있어서 피항선과 유지선의 피항 동작은 다음과 같다[15].

1) 피항선의 동작

이 법에 따라 다른 선박의 진로를 피하여야 하는 모든 선박은 될 수 있으면 미리 동작을 크게 취하여 다른 선박으로부터 충분히 멀리 떨어져야 한다.

2) 유지선의 동작

① 두 척의 선박 중 한 척의 선박이 다른 선박의 진로를 피하여야 할 경우 다른 선박은 그 침로와 속력을 유지하여야 한다.

② ①에 따라 침로와 속력을 유지하여야 하는 선박[이하 “유지선“(維持船)이라 한다]은 피항선이 이 법에 따른 적절한 조치를 취하고 있지 아니하다고 판단하면 ①에도 불구하고 스스로의 조종만으로 피항선과 충돌하지 아니하도록 조치를 취할 수 있다. 이 경우 유지선은 부득이하다고 판단하는 경우 외에는 자기 선박의 좌현 쪽에 있는 선박을 향하여 침로를 왼쪽으로 변경하여서는 아니 된다.

③ 유지선은 피항선과 매우 가깝게 접근하여 해당 피항선의 동작만으로는 충돌을 피할 수 없다고 판단하는 경우에는 ①에도 불구하고 충돌을 피하기 위하여 충분한 협력을 하여야 한다.

④ ②와 ③은 피항선에게 진로를 피하여야 할 의무를 면제하는 것은 아니다.

Fig. 2.1은 해사안전법에 명시된 충돌위험이 있는 상황에서 대략적인 두 선박 사이의 거리에 따른 유지선의 행동요령을 정리한 것이다. 유지선의 의무를 갖는 선박은 피항선의 동작만으로 충돌을 피할 수 없다고 판단되는 상황이 되기 전까지는 속도와 방향을 유지하며, 피항선의 동작만으로 충돌을 피할 수 없다고 판단되면 우현으로 변침하여 충돌회피에 협력하여야 한다.

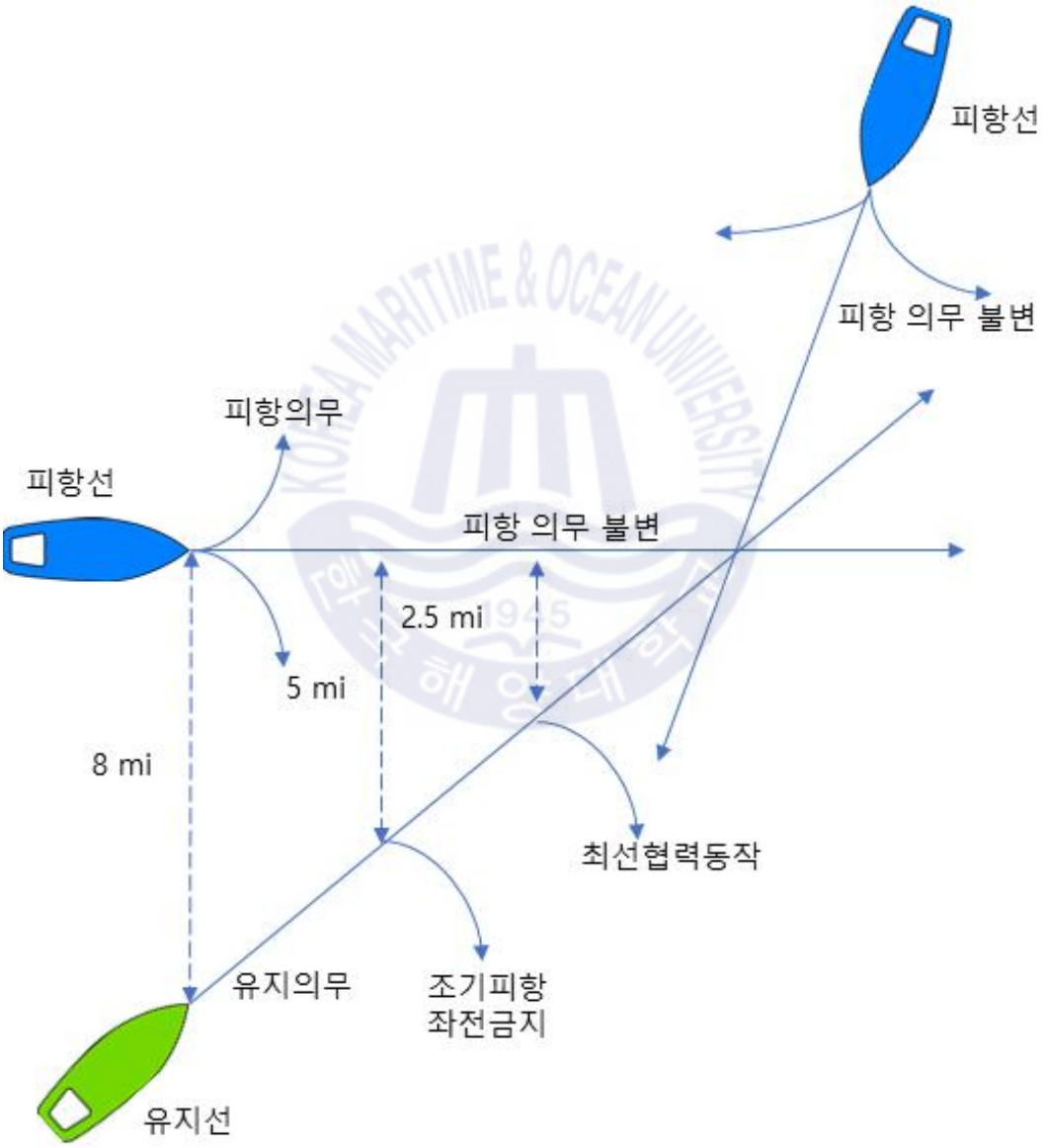


Fig. 2.1 COLREGs 규정에 따른 충돌위험 대처

2.2 강화학습

강화학습은 머신러닝의 한 범주로서 행동심리학의 강화라는 개념을 컴퓨터 학습에 도입한 것이다. 어떤 환경 안에서 정의된 행위자가 현재의 상태를 인식하여, 선택할 수 있는 행동 중 보상을 최대로 받을 수 있는 행동 순서를 선택하는 방법이다. 이 방법은 감독학습(Supervised learning)과는 다르게 입력과 그에 대한 정답을 주지 않아도 목표를 달성하기 위해 목표에 대한 보상을 최대화하는 행동 결정을 선택한다. 로봇, 게임, 제어, 통계 등 다양한 분야에서 포괄적으로 적용되고 있다[16]. 대표적인 연구 사례로는 구글 딥마인드 팀에서 만든 인공지능 바둑 프로그램 알파고가 있으며, 이세돌과의 대국에서 승리하면서 알파고의 핵심 기술인 강화학습이 많은 사람에게 알려졌다.

2.2.1 마르코프 결정 과정

마르코프 결정 과정은 순차적인 행동을 결정하는 문제를 수학적으로 표현한 것으로 강화학습, 로봇 제어, 경제학 등 다양한 분야에서 적용되고 있다. 강화학습은 순차적 행동 결정 문제를 푸는 것이며, 이러한 문제를 수학적으로 표현한 것이 마르코프 결정 과정이다. 마르코프 결정 과정은 $\langle S, A, P, R, \gamma \rangle$라는 요소로 구성되어 있다[17].

S : 상태 집합
A : 행동 집합
P : 상태 변환 확률
R : 보상 함수
γ : 감가율 $(0 \leq \gamma \leq 1)$

상태집합 S는 행위자가 관찰 가능한 상태의 집합을 의미하는 것으로 센서에서 수집되는 데이터들이다. 행동 집합 A는 행위자(Agent)가 시간이

t일 때 상태 S_t에서 할 수 있는 가능한 행동의 집합을 의미한다. 상태 변환 확률 P는 행위자가 상태 S_t에서 행동 a을 취했을 때 다음 상태 S_{t+1}에 도달할 확률을 의미하며, 환경 모델의 일부이다. 보상 함수 R은 행위자가 학습할 수 있는 유일한 정보로 환경(Environment)이 행위자에게 주는 보상을 의미한다. 감가율 γ는 행위자가 현재에 가까운 보상일수록 미래에 받는 보상보다 높은 가치를 부여하기 위한 수학적 표현으로 0과 1 사이의 값이다.

만약 현재의 시간 t부터 시간 k가 지난 후에 보상 R_{t+k}을 받을 것이라고 하면 현재 그 보상의 가치는 식 (2.1)과 같이 나타낸다.

$$\gamma^{k-1} R_{t+k} \tag{2.1}$$

현재로부터 시간이 k만큼 지났기 때문에 미래에 받을 보상 R_{t+k}는 γ^{k-1}만큼 감가된다. 더 먼 미래에 보상을 받게 될수록 행위자가 받는 보상은 줄어든다. Fig. 2.2는 마르코프 결정 과정의 동작을 구조화한 것이다. 학습시키고자 하는 행위자는 s_t에 해당하는 상태에서 a_t에 해당하는 행동을 수행한다. 그러면 환경은 다음 상태에 해당하는 s_{t+1}과 그에 상응하는 보상 r_{t+1}을 행위자에게 반환한다.

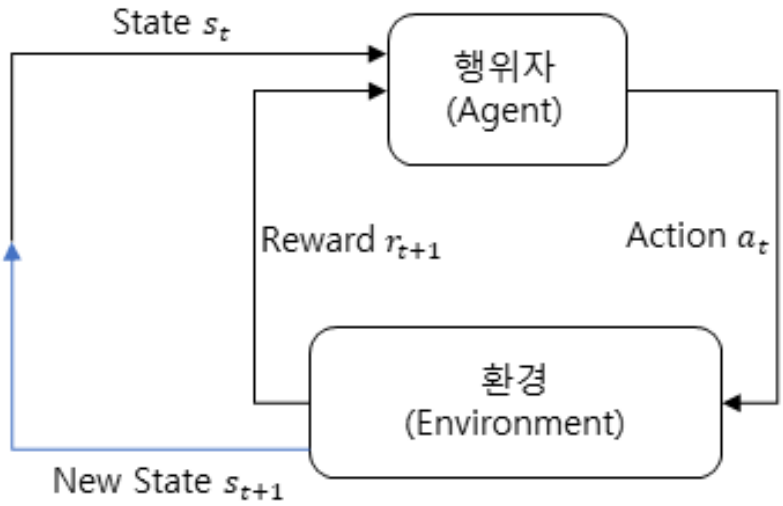


Fig. 2.2 행위자와 환경의 상호작용

마르코프 결정 과정의 목적은 정의된 문제에 대해 각 상태에서 보상을 최대화하는 행동이 무엇인지 결정하는 것이다. 이때 각각의 상태마다 행동이 선택될 확률을 표현하는 함수를 정책 π라 하고, π는 식 (2.2)와 같이 주어진 상태 s에 대한 행동의 분포로 표현된다.

$$\pi(a|s) = \Pr(A_t = a|S_t = s) \tag{2.2}$$

마르코프 결정 과정이 주어진 π를 따를 때, s에서 s'으로 이동할 확률은 식 (2.3)과 같이 계산된다.

$$p_\pi(s'|s) = \sum_{a \in A} \pi(a|s) p(s'|s,a) \tag{2.3}$$

또한, s에서 얻을 수 있는 보상은 식 (2.4)와 같이 계산된다.

$$r_\pi(s) = \sum_{a \in A} \pi(a|s) r(s,a) \tag{2.4}$$

1) 상태 가치함수

상태 가치함수 $v(s)$는 마르코프 결정 과정의 상태 가치함수과 마찬가지로 상태 s에서 시작했을 때 얻을 수 있는 보상의 기댓값을 의미한다. 그러나 마르코프 결정 과정에서의 주어진 정책 π을 따라 행동을 결정하고, 상태를 이동하기 때문에 마르코프 결정 과정에서의 상태 가치함수는 식 (2.5)와 같이 정의된다.

$$\begin{aligned} v_\pi(s) &= E_\pi[G_t | S_t = s] \\ &= E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_{t=s}] \\ &= \sum_{a \in A} \pi(a|s)[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) v_\pi(s')] \end{aligned} \tag{2.5}$$

Fig. 2.3은 상태 가치함수의 예로 특정 정책을 따라 이동했을 때 다음 상태에서 얻게 되는 기대보상을 나타낸다. G(Goal)에 도달하면 100의 보상을 얻는 미로에서 G에 가까울수록 탈출할 확률이 높아지므로 높은 가치를 가지며 멀수록 탈출할 확률이 낮아지므로 낮은 가치를 가지고 있는 것을 볼 수 있다.

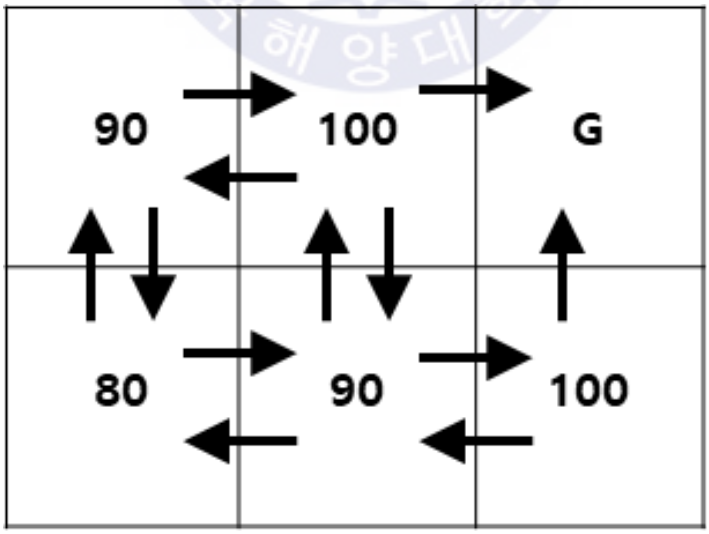


Fig. 2.3 상태 가치함수

2) 행동 가치함수

행동 가치함수 $q_\pi(s,a)$는 상태 s에서 시작하여 a라는 행동을 취했을 때 얻을 수 있는 보상의 기댓값을 의미한다. 행동 가치함수 $q_\pi(s,a)$는 식 (2.6)과 같이 정의된다.

$$\begin{aligned} q_\pi(s,a) &= E_\pi[G_t | S_t = s, A_t = a] \\ &= E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_{t=s}] \\ &= r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) \sum_{a \in A} \pi(a'|s') q_\pi(a'|s') \end{aligned} \tag{2.6}$$

상태 가치함수는 어떠한 상태가 더 많은 보상을 얻을 수 있는지를 알려준다면, 행동 가치함수는 어떠한 상태에서 어떠한 행동을 취해야 더 많은 보상을 얻을 수 있는지 알려준다. 모든 상태에 대해 행동 가치함수를 계산할 수 있다면, 모든 상태에 대해 최적행동을 선택할 수 있게 된다. Fig. 2.4는 미로에서 G로 이동하는 행동은 높은 가치를 가지며 G에서 멀어지는 행동은 낮은 가치를 갖는 것을 알 수 있다.

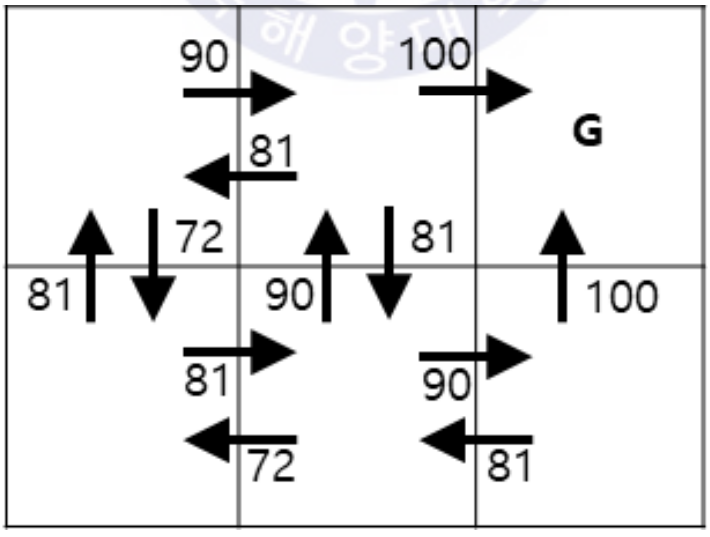


Fig. 2.4 행동 가치함수

2.2.2 강화학습 알고리즘

1) 동적 프로그래밍(Dynamic Programming)

동적 프로그래밍은 효율적인 알고리즘 설계와 분석을 위한 알고리즘으로 데이터와 문제를 나누고 나누어진 부분 문제의 해를 결합해서 최종적인 문제를 해결한다. 각 하위 부분 문제를 계산한 뒤, 그 해를 일정 공간에 저장했다가 이후에 같은 하위 문제가 나왔을 때 이미 계산된 결과를 그대로 사용함으로써 실행에 필요한 자원을 줄일 수 있다. 동적 프로그래밍의 응용으로는 문맥 자유 언어의 인식, 최단 경로 탐색, 행렬 곱의 최적화, 이진 트리 탐색의 최적화 분야 등이 있다[18].

2) Q-러닝 (Q-learning)

Q-러닝은 다양한 강화학습 알고리즘에 응용되는 알고리즘으로 특정 상태에서 취할 수 있는 행동에 기대되는 기댓값을 예측하는 Q-함수를 학습함으로써 최적의 정책을 유도한다. 최적의 정책은 식 (2.7)과 같이 정의된다[19].

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}[r(s,a) + \gamma V^*(s,a)] = \underset{a}{\operatorname{argmax}} Q(s,a) \tag{2.7}$$

식 (2.7)에서 알 수 있듯이 Q-러닝은 상태와 행동에만 영향을 받으며, 구체적인 모델이 없어도 행동의 기댓값을 비교할 수 있다. Q-함수는 초기에 고정된 임의의 값을 가지며 각 시간 t에 특정 상태 s_t에서 행동 a_t을 취하고 새로운 상태 s_{t+1}로 전이되며, 이때 보상 r_t 를 얻는다. Q-값은 식 (2.8)을 통하여 수정된다.

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \quad Q(s_t, a_t) + \alpha \cdot \quad \left(r_t + \gamma \cdot \quad \max_a Q(s_{t+1}, a)\right) \tag{2.8}$$

2.2.3 심층강화학습 알고리즘

1) Deep Q network (DQN)

DQN 알고리즘은 딥마인드에서 고안했으며 심층 신경망을 사용하여 아타리 게임(Atari game)을 인간 수준으로 수행할 수 있다. 리플레이 메모리(Replay memory)와 인공신경망 구조는 Fig. 2.5와 같다. 행위자는 정책에 따라 환경을 탐험하면서 가장 큰 보상을 받을 수 있는 행동을 취하면서 얻는 샘플 (s, a, r, s')을 리플레이 메모리에 저장하며, 에피소드(Episode)를 진행하면서 리플레이 메모리에서 여러 개의 샘플을 무작위로 뽑아 예상한 보상과 실제로 받은 보상의 오차를 계산하여 인공신경망의 매개변수를 수정한다. 리플레이 메모리는 크기가 정해져 있어서 메모리가 가득 차면 오래된 메모리부터 차례로 삭제한다. 행위자가 학습을 반복하면 더 높은 점수를 받으면 더 좋은 샘플을 리플레이 메모리에 저장한다. 이를 경험 리플레이(Experience replay)라고 하며 연속적인 상태 공간에서 발생하는 데이터 간의 연관성(Coupling) 문제를 해결할 수 있다[20].

DQN 알고리즘의 다른 특징은 목표신경망(Target network)을 사용한다는 것인데, 경험 리플레이를 사용하는 행위자는 매 단계 리플레이 메모리에서 여러 개의 샘플(Batch)로 추출해서 학습에 사용한다. Batch란 모델을 학습할 때 사용되는 예제들의 집합을 말하며, Batch 크기를 설정하여 몇 개의 예제를 가져올지 설정할 수 있다. 식 (2.9)는 Q-러닝에서 Q-함수를 수정하는 수식을 나타낸다.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)) \quad (2.9)$$

DQN은 오류함수로 평균 제곱 오차(MSE)를 사용하는데 DQN 행위자가 학습에 사용하는 오류함수는 식 (2.10)과 같다. 이 오류함수를 최소화하는 방향으로 인공신경망이 수정된다.

$$MSE = (\text{정답} - \text{예측})^2$$
$$= (R_{t+1} + \gamma \max_{a'} Q(s', a', \theta) - Q(S, A, \theta))^2 \quad (2.10)$$

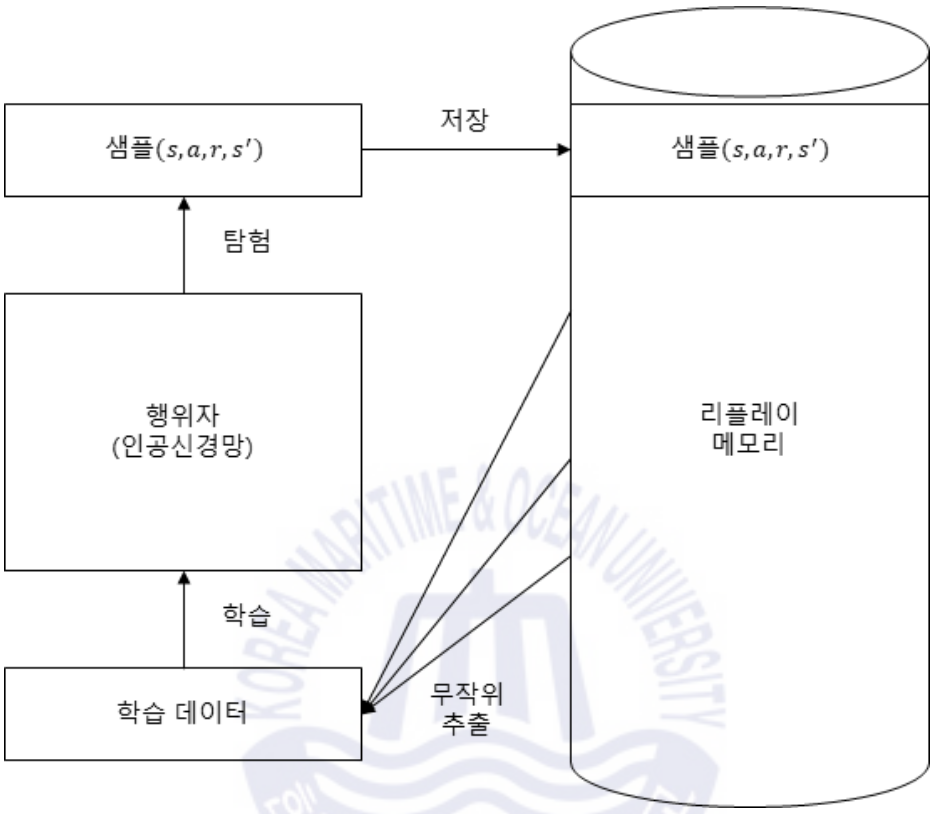


Fig. 2.5 리플레이 메모리와 인공신경망 구조

목표신경망을 사용하지 않으면 정답을 만들어 내는 신경망이 계속 수정되어 정답이 변하는 문제가 발생할 수 있는데, 이를 해결하기 위한 방법으로 목표신경망을 따로 두어서 정답을 만들어 내는 인공신경망을 일정 시간 동안 수정하지 않고 충분히 학습할 수 있게 유지하는 방법이다. 목표신경망을 따로 만들어서 목표신경망에서 정답에 해당하는 값을 구하고 구한 정답을 통해 다른 인공신경망을 계속 학습시키면서 목표신경망은 일정한 시간마다 그 인공신경망으로 수정하는 방법으로 두 가지 학습의 간섭을 차단하는 방법이다. 오류함수 수식에서 이를 구분하기 위해 목표신경망의 매개변수는 θ^-로 표현하고, 인공신경망의 매개변수는 θ로 표현한다. 식 (2.11)은 목표신경망을 이용하는 DQN 오류함수를 나타낸다.

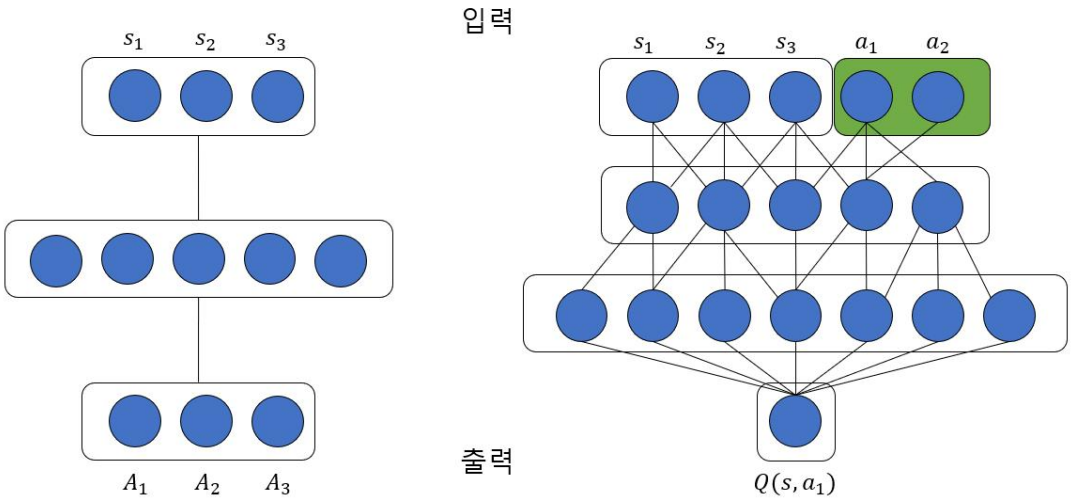
$$MSE = (\text{정답} - \text{예측})^2$$
$$= (R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a', \theta^-) - Q(S_t, A_t, \theta))^2 \quad (2.11)$$

2) Deep Deterministic Policy Gradient (DDPG)

DQN 알고리즘의 신경망은 행위자의 행동 집합의 크기에 따라 출력 값의 개수가 결정된다. 각각의 출력 값은 하나의 행동에 대한 Q-Value 값을 나타내기 때문에 신경망의 출력 개수는 행동 집합의 크기와 비례한다. 따라서 행동 집합의 크기가 크거나 무한할 경우 신경망의 출력 개수 또한 기하급수적으로 증가하게 되며, 행동 집합에서 가장 높은 Q-Value를 갖는 행동을 구하기 불가능하여 연속적인 행동을 갖는 행위자에 대해서 학습이 어렵다.

DDPG 알고리즘은 액터-크리틱 신경망(Actor-Critic Network)을 이용하여 Q-Value를 기준으로 학습하지 않고 정책을 학습한다. 강화학습에서 정책이란 행위자가 특정 상태에서 어떤 행동을 취해야 하는가를 말한다. 따라서 DDPG 알고리즘에서 각각의 행동에 대한 모든 Q-Value를 알 필요가 없으며, 학습된 정책으로 행위자의 상태에 맞는 행동을 결정할 수 있다. DDPG 알고리즘은 DQN 알고리즘과 액터-크리틱 신경망을 결합한 알고리즘이다. Fig. 2.6은 DDPG 알고리즘에서 사용하는 액터-크리틱 신경망의 구조를 나타낸다[21]. 액터-크리틱 신경망은 두 개의 신경망 즉, 액터 신경망(Actor network)과 크리틱 신경망(Critic network)을 사용하며, 액터 신경망은 상태가 주어질 때 행위자가 취해야 할 행동을 결정하고, 크리틱 신경망은 현재 상태에서 최적의 행동을 결정할 수 있는 정책을 결정한다.

Fig. 2.7은 액터-크리틱 신경망의 매개변수를 수정하는 과정을 나타낸다. 액터 신경망은 정책(Policy)에 따라서 행동을 결정하고 크리틱 신경망은 Q-Value를 구하여 최적의 정책을 결정한다. 이때 액터 신경망과 크리틱 신경망은 학습 도중에 새로운 입력값이 들어오게 되면 학습이 정상적으로 이루어지지 않게 되므로 목표신경망을 이용하여 학습의 간섭을 차단한다.



(a) DDPG-액터 신경망 (b) DDPG-크리틱 신경망

Fig. 2.6 액터-크리틱 신경망 구조의 예

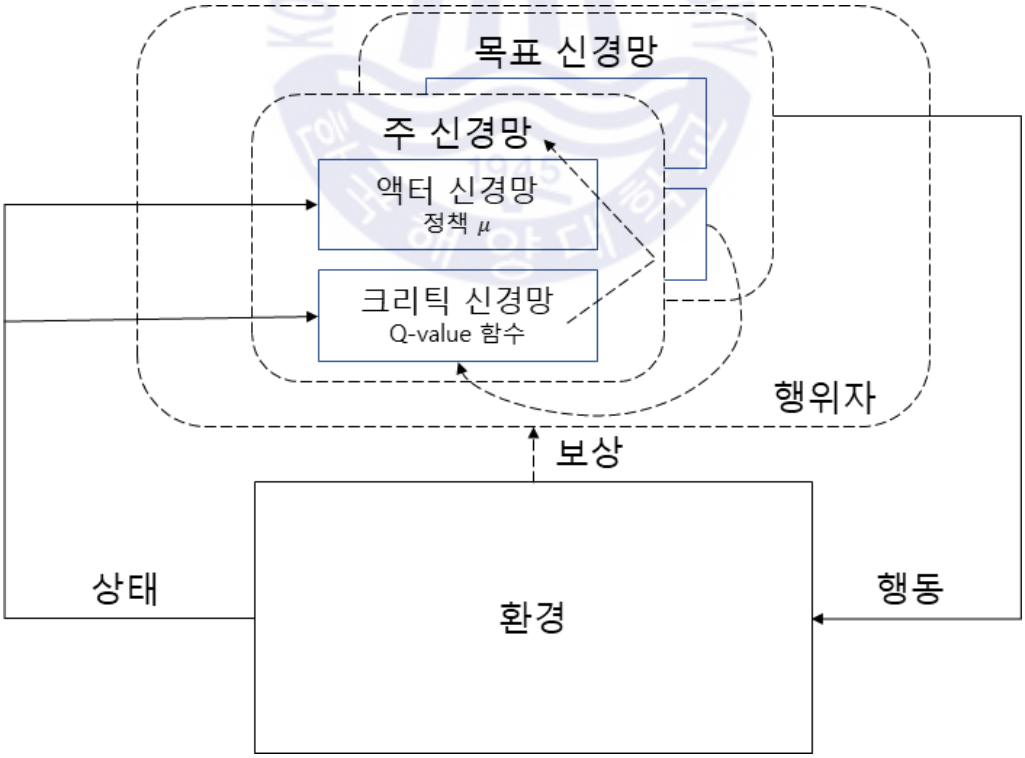


Fig. 2.7 액터-크리틱 신경망 수정과정

제 3 장 심층강화학습 기반 선박 충돌회피 시스템

Fig. 3.1은 본 논문에서 제안하는 DDPG 알고리즘을 이용한 충돌회피 시스템의 구조를 나타낸다. Fig. 3.1과 같이 충돌회피 시스템은 크게 행위자와 환경으로 구성된다. 행위자는 크리틱 모듈(Critic Module), 리플레이 메모리, 액터 모듈(Actor Module)로 구성되며 환경은 보상 모듈, 타선 모듈, 뷰어 모듈로 구성된다. 시스템은 다음과 같은 흐름으로 동작한다. 환경이 액터 모듈에게 현재 상태를 알려주고 액터 모듈은 정책에 따라 행동 집합 중에서 가장 높은 보상이 예상되는 행동을 결정한다. 크리틱 모듈은 리플레이 메모리에 저장된 데이터를 무작위로 16개 추출하여 학습에 사용하며, 액터 모듈의 행동에 따른 보상을 평가하여 액터 모듈의 정책을 수정하면서 학습을 반복하게 된다. 이 과정에서 액터 모듈과 크리틱 모듈은 각각의 목표신경망에 의하여 최적화된다. Fig. 3.2와 같이 주 신경망(Main Network)과 목표 신경망(Target Network)의 수정과정은 주 신경망이 학습하는 과정에 수정되어 충분히 학습하지 못하는 문제를 해결하여 신경망의 안정성을 높이고, 오차함수를 통해서 주 신경망의 크리틱 모듈을 수정한다. 정해진 에피소드가 끝나거나 크리틱 모듈의 기울기가 정해진 값 이하로 수렴하면 학습이 종료된다. 뷰어 모듈은 자선과 타선의 움직임 변화를 보여주며, 타선 모듈은 COLREGs 규정을 지키는 선박과 지키지 않는 선박 두 가지로 구성되어 있고 속도는 일정하다. 보상 모듈은 행위자의 행동을 판단하여 정해진 기준으로 보상을 부여하는 함수로써 선박이 충돌의 위험이 있을 때 충돌을 회피하여 가장 빠르게 목적지로 도착할 수 있게 보상을 설정한다.

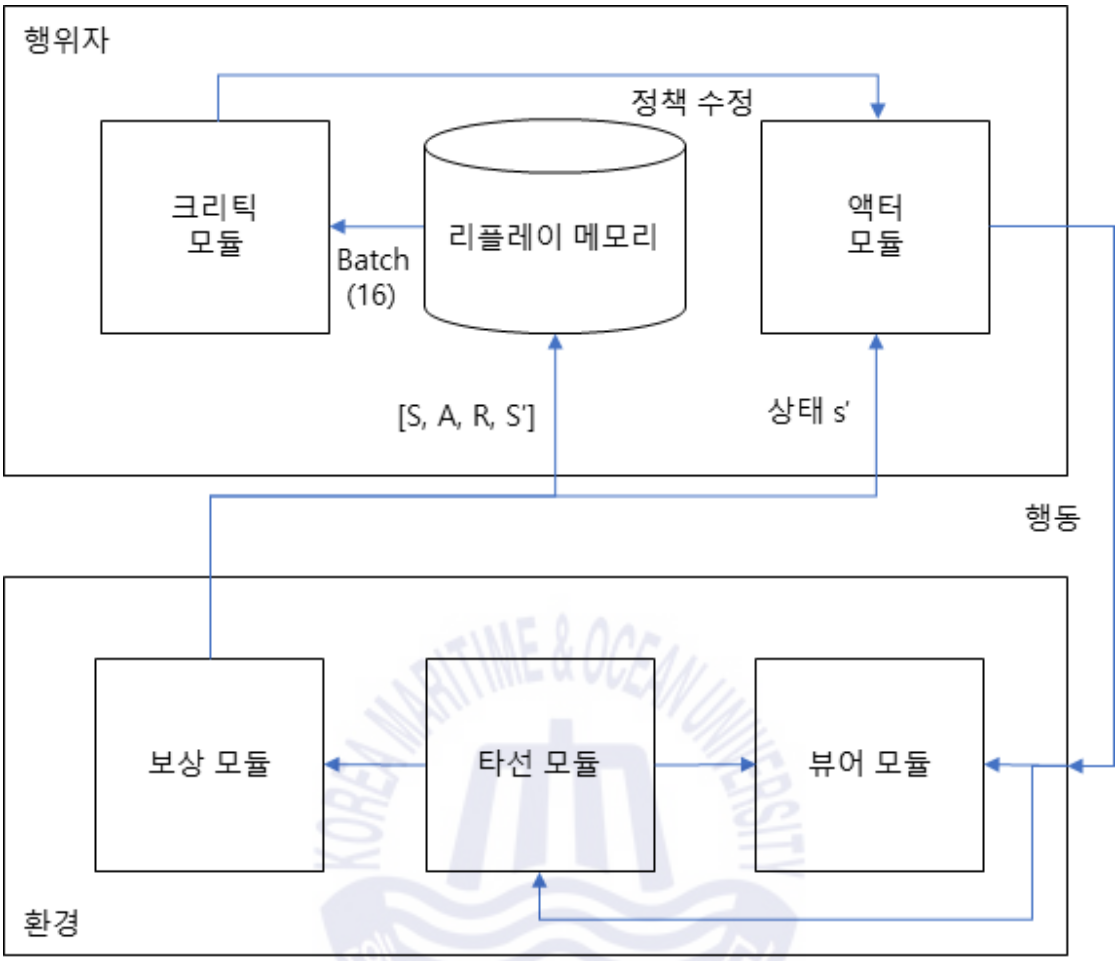


Fig. 3.1 DDPG 알고리즘을 적용한 충돌회피 시스템

3.1 행위자

3.1.1 액터 모듈

액터 모듈은 주 신경망의 액터 신경망과 목표 신경망의 액터 신경망으로 구성된다. 자율운항 선박이 현재 상태에서 수행할 행동을 출력하고 크리틱 모듈로부터 행동을 평가받고 그에 대한 기울기를 전달받아 액터 신경망을 수정한다. 식 (3.1)은 액터 신경망의 수정을 나타내며, θ는 크리틱 모듈이 제안하는 방향으로 수정된다.

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)(Q_v(s_t, a_t)) \tag{3.1}$$

액터 신경망은 입력층, 은닉층과 출력층 세 개의 네트워크로 구성되어 있으며, 입력층은 환경으로부터 상태를 입력받는다. Fig. 3.2는 액터 모듈과 크리틱 모듈 간의 정책 수정 과정을 주 신경망과 목표 신경망의 관점으로 나타낸 것이다. 액터 신경망은 주 신경망과 목표 신경망에서 현재 상태에 대한 최적의 행동을 결정하며, 크리틱 신경망으로부터 최신 정책으로 수정받은 정책 기울기 함수에서 액터 신경망의 정책을 수정받으면서 현재 상태에 더욱 정확한 행동을 예측한다. 은닉층은 단층으로 20개의 노드로 구성되어 있으며 가중치를 연산한다. 출력층에서는 -45° 부터 $+45^{\circ}$ 에 대한 출력값이 -1~1로 나타난다.

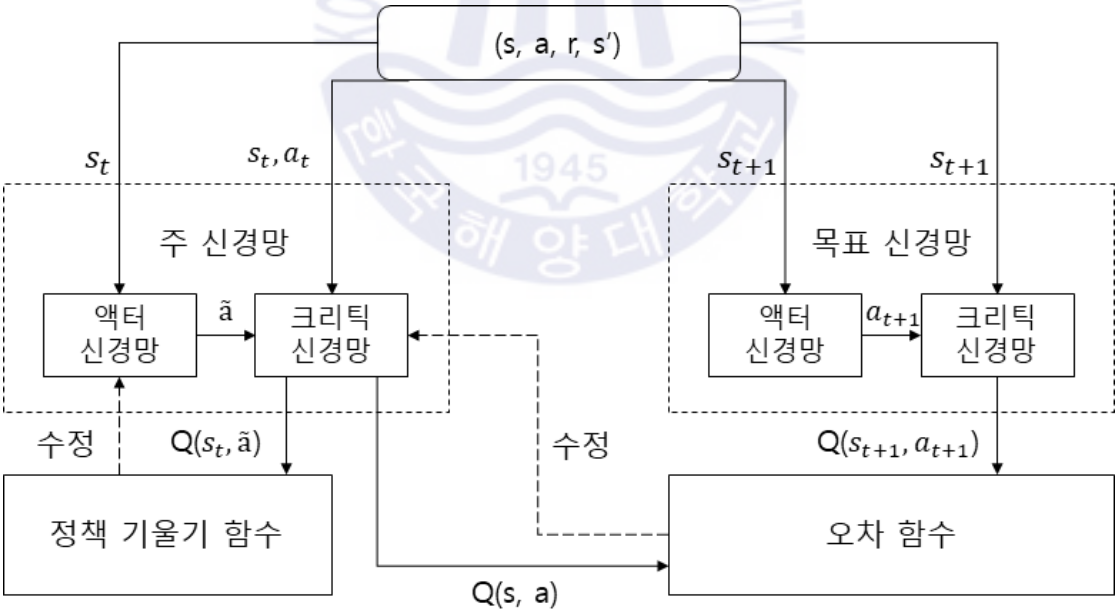


Fig. 3.2 주 신경망과 목표 신경망의 수정과정

3.1.2 크리틱 모듈

크리틱 모듈은 주 신경망의 크리틱 신경망과 목표 신경망의 크리틱 신경망으로 구성된다. 크리틱 신경망은 입력층, 은닉층과 출력층 세 개의 네트워크로 구성되어 있으며, 입력층은 액터 신경망의 입력층에 입력과 같은 상태와 액터 신경망의 출력값을 같이 입력한다. 크리틱 신경망의 입력층과 은닉층은 액터 신경망의 입력층과 은닉층과 유사하게 구성된다.

크리틱 모듈은 액터 신경망의 행동에 대한 행동 가치함수를 출력하여 액터 모듈에 기울기를 전달하여 정책을 수정하며 크리틱 신경망은 식 (3.2)와 같이 수정된다.

$$v \leftarrow v + \beta(r_{t+1} + \gamma Q_v(s_{t+1}, a_{t+1}) - Q_v(s_t, a_t)) \nabla_v Q_v(s_t, a_t) \tag{3.2}$$

식 (3.3)은 앞서 구한 보상을 평가하여 크리틱 모듈이 액터 모듈에 수정을 제안하는 θ를 계산한다.

$$\theta \leftarrow \theta + \nabla_\theta \log \pi_\theta(a_t | s_t)(r_{t+1} + \gamma Q_v(s_{t+1}, a_{t+1}) - Q_v(s_t, a_t)) \tag{3.3}$$

Fig. 3.2에서 주 신경망의 크리틱 신경망은 액터 신경망의 행동을 평가하여 정책 기울기 함수를 수정해준다. 목표 신경망의 크리틱 신경망은 주 신경망과 같은 방법으로 행동 가치함수를 구하여, 오차 함수를 통해서 주 신경망의 크리틱 신경망을 수정해 주 신경망의 크리틱 신경망이 더욱 정확한 정책을 찾을 수 있도록 도와준다.

3.1.3 리플레이 메모리

리플레이 메모리는 상태 s, 행동 a, 보상 r, 다음 상태 s'를 시간 단위마다 튜플의 형태로 저장한다. 리플레이 메모리 크기는 2000으로 고정되어

있고 메모리가 가득 차면 오래된 데이터부터 삭제하고 저장한다. 액터 신경망과 크리틱 신경망이 수정될 때마다 mini-batch로 표본을 16개 추출하여 구성하고 크리틱 모듈에 전달한다. mini-batch가 순차적인 데이터로 구성되지 않으므로 데이터 사이의 상관관계가 줄어 지역 최저점에 빠지는 문제를 해결할 수 있다.

3.2 환경

3.2.1 타선 모듈

타선 모듈은 환경에서 동적 장애물 역할을 하며 COLREGs 규정을 따르는 모델과 규정을 따르지 않는 모델 두 가지로 구성되며, 자선의 모델과 같이 출발지, 목적지와 속도를 설정할 수 있으며 센서를 통해서 장애물의 위치와 거리를 수집하여 동작한다.

1) COLREGs 규정을 준수하는 선박은 앞서 2.1절에서의 설명과 같이 피항선의 의무 또는 유지선의 의무에 따라서 행동이 다르며 피항선의 의무를 갖는 경우 일정 거리 이상 접근 시 우현으로 변침하였다가 충돌회피에 성공하면 다시 기존의 항로로 돌아간다. 유지선의 의무를 갖는 경우에는 변침 없이 기존 항로를 계속 유지한다.

2) COLREGs 규정을 준수하지 않는 선박은 항해사의 부주의로 인한 사고 유발의 가능성이 있는 선박을 표현하기 위하여 피항선의 의무를 가짐에도 피항하지 않고 항로를 유지한다.

3.2.2 보상 모듈

강화학습은 행위자가 환경과 상호작용하면서 학습을 하게 되는데 이 과정에서 행위자가 수행한 행동에 대한 보상이 주어져야 한다. 심층학습을 이용한 충돌회피 시뮬레이션의 학습을 위해 사용된 보상은 다음과 같다.

1) 시간의 흐름에 따른 음의 보상

시간의 흐름에 따른 음의 보상은 충돌위험이 없을 때는 기존의 항로를 벗어나지 않게 하고, 충돌위험이 발생하여 충돌회피하는 과정에서 불필요한 회피기동을 방지하고 기존의 항로를 크게 벗어나지 않기 위해 제공하는 보상으로 매 스텝마다 음의 보상을 계속 받게 된다.

2) 충돌에 따른 음의 보상

충돌에 따른 음의 보상은 행위자가 동적 장애물로부터 일정 거리 이상 가까워졌을 때 최우선으로 고려하는 보상으로 -1(가장 큰 음의 보상)의 보상값을 갖도록 설정하였다. 충돌에 따른 음의 보상을 받게 되는 경우 행위자는 에피소드 진행도와는 상관없이 에피소드를 종료한다.

3) 충돌회피에 따른 양의 보상

충돌회피에 따른 양의 보상은 행위자가 충돌회피에 성공하였을 때 받게 되는 보상으로 항해 중 장애물과 충돌위험이 있을 때, 기존의 항로를 포기하고 충돌회피를 수행할 수 있게 만든다. 자선의 센서에 장애물이 발견되면 충돌위험이 있는 것으로 판단하고 자선의 센서에서 타선이 완전히 사라지면 보상을 받게 된다.

3.2.3 뷰어 모듈

행위자와 환경으로부터 자선과 타선의 정보(위치, 속도, 방향)를 수신하여 현재 위치를 격자 지도에 나타내며, 시간마다 자선과 타선의 위치 좌표를 격자 지도에 기록한다.

제 4 장 선박의 충돌회피 시뮬레이션

본 장에서는 선박 충돌회피 시스템을 이용하여 다양한 시나리오에 대한 충돌회피 시뮬레이션을 수행하고 그 결과를 분석한다. 선박의 충돌회피 시스템의 개발 및 시뮬레이션 환경은 다음과 같다. 운영체제는 Ubuntu 16.04 LTS, 프로그래밍 언어는 Python 3.6, 개발환경은 Anaconda를 사용하였으며, Anaconda는 수학, 과학 분야에서 사용되는 다양한 패키지들을 모아놓은 환경이다.

4.1 COLREGs 규정 준수 선박에 대한 충돌회피

본 절에서는 COLREGs 규정을 준수하는 선박에 대해 횡단하는 상황, 마주치는 상황의 두 가지로 나누어 실험한다. 또한, 횡단하는 상황의 경우는 자선을 기준으로 타선이 좌현 그리고 우현에 위치하는 두 가지 상황으로 나눌 수 있으며 심층강화학습으로 학습된 자선이 '유지선, 피항선의 의무를 잘 수행하는가' 와 '성공적인 충돌회피 성능을 보이는가' 를 분석한다.

4.1.1 시뮬레이션 시나리오 및 초기조건 설정

자선과 타선의 항로에 따른 항로유지 및 심층강화학습에 의한 충돌회피 판단에 대한 수행 결과를 검증한다. 이를 위해 시뮬레이션 초기조건으로 Fig. 4.1과 같이 1:1 상황에서 COLREGs를 준수하는 선박에 대한 충돌회피

시나리오로 구성하였다. 자세한 설명은 다음과 같다.

1) 횡단하는 상황(자선이 유지선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : 자선과 타선이 횡단하는 시나리오이다. 자선 기준 좌현에서 타선이 접근하는 상황으로 타선이 피항의 의무가 있으므로 우현으로 변침하여야 한다.

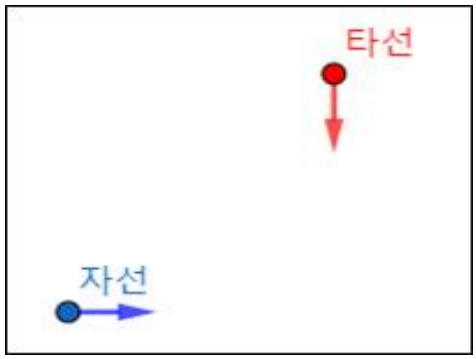
본 실험에서는 타선이 규정에 따라 피항하는 상황에서 심층강화학습으로 학습된 자선의 충돌회피를 분석한다. 또한, 타선이 피항하는 상황에서 타선의 피항 시점을 다르게 하여 비교한다.

2) 횡단하는 상황(자선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : 자선과 타선이 횡단하는 시나리오이다. 자선 기준 우현에서 타선이 접근하는 상황으로 자선이 피항의 의무가 있으므로 우현으로 변침하여야 한다.

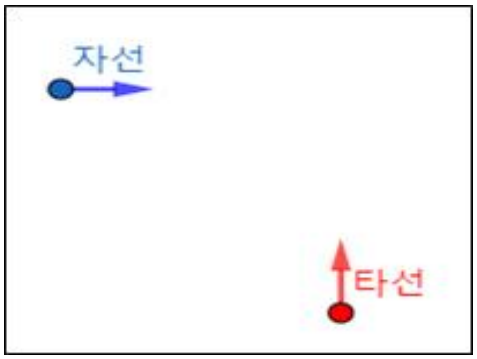
본 실험에서는 타선이 규정에 따라 항로를 유지하는 상황에서 심층강화학습으로 학습된 자선의 충돌회피를 분석한다.

3) 마주치는 상황(양선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : 자선과 타선이 정면으로 마주치는 시나리오이다. 양선이 모두 피항의 의무를 가지므로 양선 모두 우현으로 변침하여야 한다.

본 실험에서는 타선이 규정에 따라 피항하는 상황에서 심층강화학습으로 학습된 자선의 충돌회피를 분석한다. 또한, 타선이 피항하는 상황에서 타선의 피항 각도를 다르게 하여 비교한다.



(a) 횡단하는 상황
(자선이 유지선의 의무)



(b) 횡단하는 상황
(자선이 피항선의 의무)

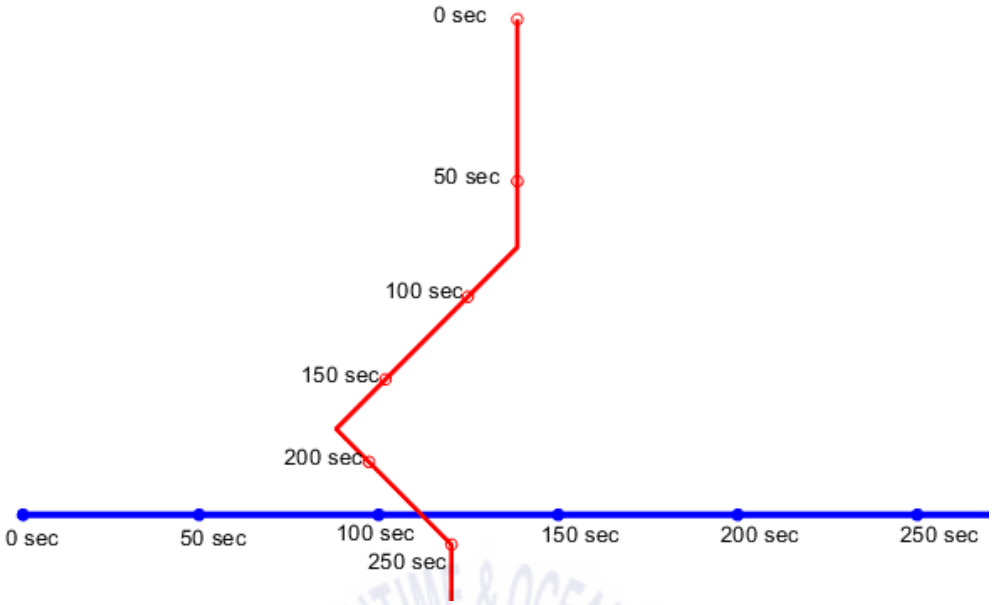
(c) 마주치는 상황
(양선이 피항선의 의무)

Fig. 4.1 충돌회피 시나리오

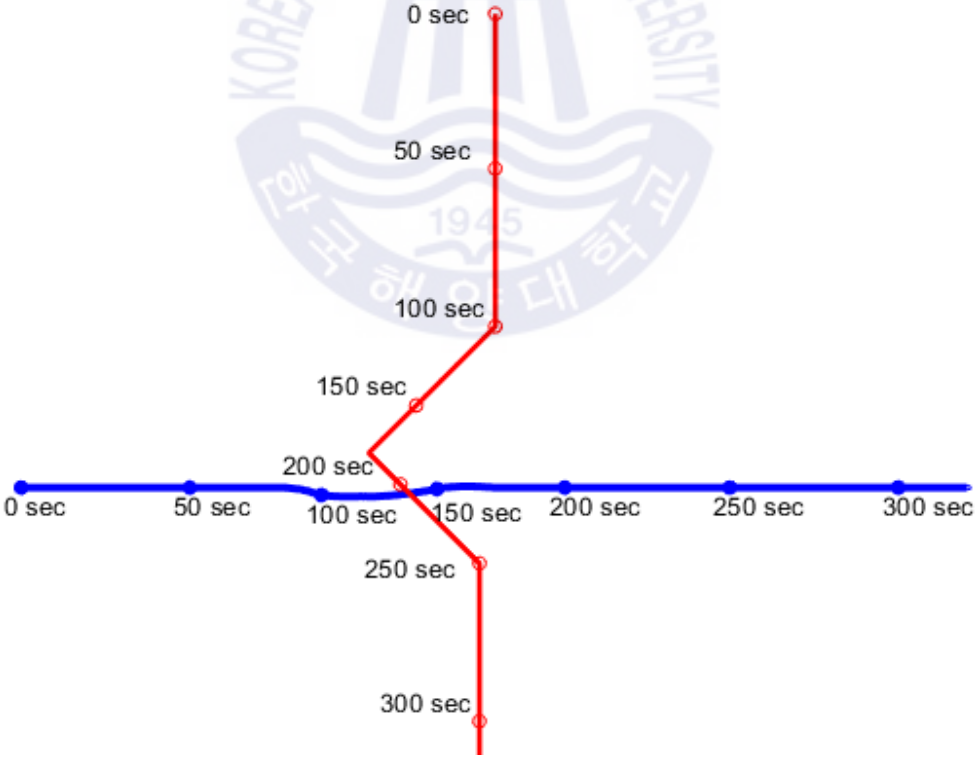
4.1.2 시뮬레이션 분석

1) 횡단하는 상황(자선이 유지선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : Fig. 4.2는 횡단하는 상황에서 자선은 유지선의 의무, 타선은 피항선의 의무를 갖고 타선이 COLREGs 규정에 따라서 피항의 의무를 수행하는 상황에 대하여 시뮬레이션한 결과이다.

타선이 충돌회피 시점을 빨리하여 자선으로부터 충분한 거리로 피항할 경우 Fig. 4.2 (a)와 같이 자선은 기존의 항로를 유지하는 것을 확인할 수 있었다. 하지만 Fig. 4.2 (b)와 같이 타선이 조금 늦게 충돌회피를 시작하는 경우 자선이 90초에서 먼저 방향을 조금 바꾸는 것을 확인할 수 있다. 이는 타선이 피항하지 않을 것을 대비하는 것으로 분석되며, 타선이 우현으로 피항하기 시작하자 자선은 곧바로 원래의 항로로 복귀하는 것을 확인할 수 있었다.



(a) 타선의 피항 시점이 충분히 빠른 경우



(b) 타선의 피항 시점이 늦는 경우

Fig. 4.2 횡단하는 상황에서 유지선의 의무일 때

2) 횡단하는 상황(자선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : Fig. 4.3는 횡단하는 상황에서 자선이 피항선의 의무, 타선이 유지선의 의무를 갖는 상황에 대하여 충돌회피를 수행한 결과이다.

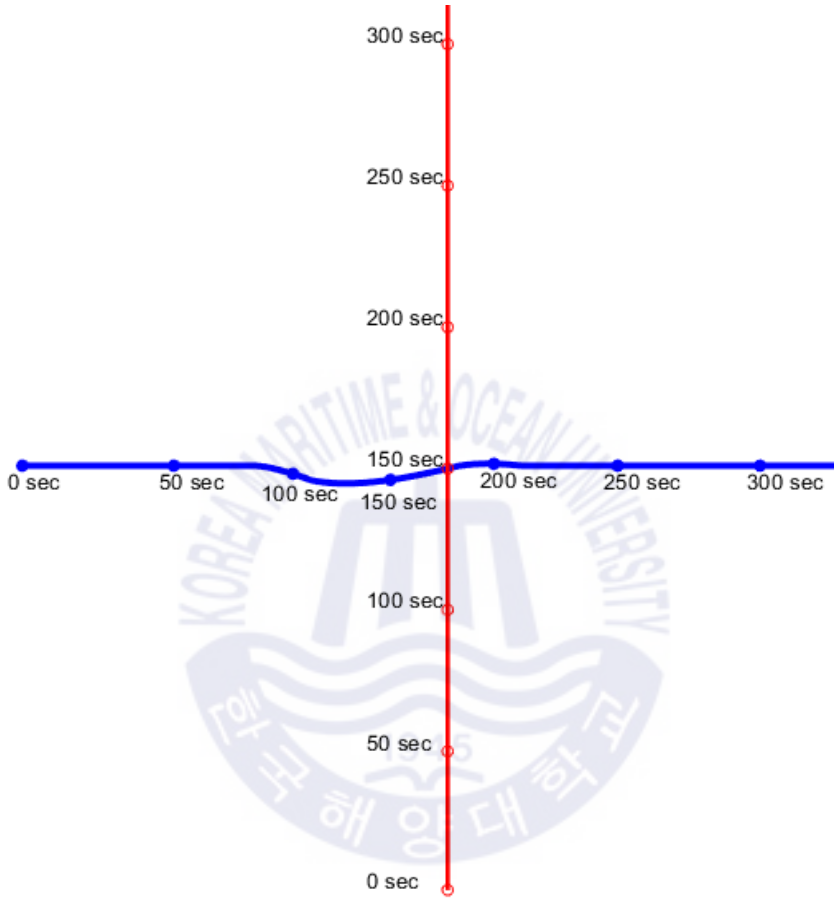
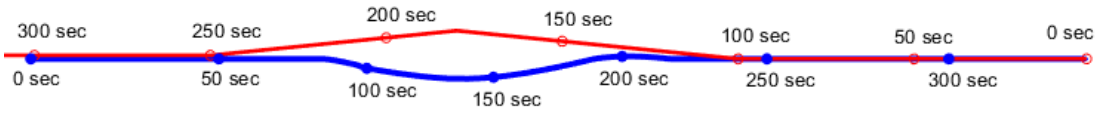


Fig. 4.3 횡단하는 상황에서 피항선의 의무일 때

Fig. 4.3에서 자선이 피항선의 의무를 가지므로 우현으로 변침하여 피항의 의무를 수행하여 충돌회피에 성공하였으나 충돌회피를 수행하는 과정에 있어서 우현으로 충분히 변침하여 회피하는 방법이 아닌 우현으로 조금만 변침하여 충돌의 시점을 늦추고 다시 기존의 항로로 돌아오는 것을 확인할 수 있었다.

3) 마주치는 상황(양선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수할 때 : Fig. 4.4는 자선과 타선이 정면으로 마주치는 상황으로 양선이 모두 피항의 의무를 가지므로 타선이 COLREGs 규정에 따라서 피항의 의무를 수행하는 상황에 대하여 시뮬레이션한 결과이다.



(a) 타선의 피항 각도가 작을 경우

(b) 타선의 피항 각도가 큰 경우

Fig. 4.4 마주치는 상황에서 양선이 피항선의 의무일 때

타선의 피항 각도를 다르게 한 실험에서 두 실험 모두 자선이 피항선의 의무를 지키는 것을 확인할 수 있었다. 본 실험에서는 타선의 피항 각도에 따라서 자선의 충돌회피 반응이 다른 것을 확인할 수 있었다. Fig. 4.4 (a)와 같이 타선의 피항 각도가 작은 경우에 자선은 충분한 각도로 피항을 진행하였고, Fig. 4.4 (b)와 같이 타선의 피항 각도가 큰 경우에 자선은 비교적 작은 각도로 피항하는 것을 확인하였다. 이는 최근접 거리인 150sec 구간을 비교해 볼 의미가 있는데 150sec에서 유사한 거리를 유지하는 것으로 보아 거리에 따른 충돌위험도를 학습하는 것으로 보인다.

4.2 COLREGs 규정 미준수 선박에 대한 충돌회피

본 절에서는 COLREGs 규정을 준수하지 않는 선박에 대해 마주치는 상황, 횡단하는 상황의 두 가지로 나누어 실험한다. 횡단하는 상황의 경우는 자선을 기준으로 타선이 좌현 그리고 우현에 위치하는 두 가지 상황으로 나눌 수 있며, 심층강화학습으로 학습된 자선이 '유지선, 피항선의 의무를 잘 수행하는가' 와 '성공적인 충돌회피 성능을 보이는가' 그리고 '타선이 피항선의 의무를 가짐에도 피항하지 않을 때 자선이 어떻게 피항을 하는가' 를 분석하고자 한다.

4.2.1 시뮬레이션 시나리오 및 초기조건 설정

자선과 타선의 항로에 따른 항로유지 및 심층강화학습에 의한 충돌회피 판단에 대한 수행 결과를 검증한다. 이를 위해 시뮬레이션 초기조건으로 Fig. 4.1과 같이 1:1 상황에서 COLREGs를 준수하지 않는 선박에 대한 충돌회피 시나리오로 구성하였다. 자세한 설명은 다음과 같다.

1) 횡단하는 상황(자선이 유지선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : 자선과 타선이 횡단하는 시나리오이다. 자선 기준 좌현에서 타선이 접근하는 상황으로 타선이 피항의 의무가 있으므로 우현으로 변침하여야 하지만 변침하지 아니하고 항로를 유지하는 경우이다.

2) 횡단하는 상황(자선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : 자선과 타선이 횡단하는 시나리오이다. 자선 기준 우현에서 타선이 접근하는 상황으로 자선이 피항의 의무가 있으므로 우현으로 변침하여야 하며, 타선은 항로를 유지하여야 하지만 좌현으로 피항하는 상황이다.

3) 마주치는 상황(양선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : 자선과 타선이 정면으로 마주치는 시나리오이다. 양선이 모두 피항의 의무를 가지므로 우현으로 변침하여야 한지만 타선이 변침을 하지 않고 항로를 유지하는 경우이다.

4.2.2 시뮬레이션 분석

1) 횡단하는 상황(자선이 유지선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : Fig. 4.5는 횡단하는 상황에서 자선은 유지선의 의무, 타선은 피항선의 의무를 가짐에도 타선이 피항하지 않는 상황에 대하여 충돌회피를 수행한 결과이다.

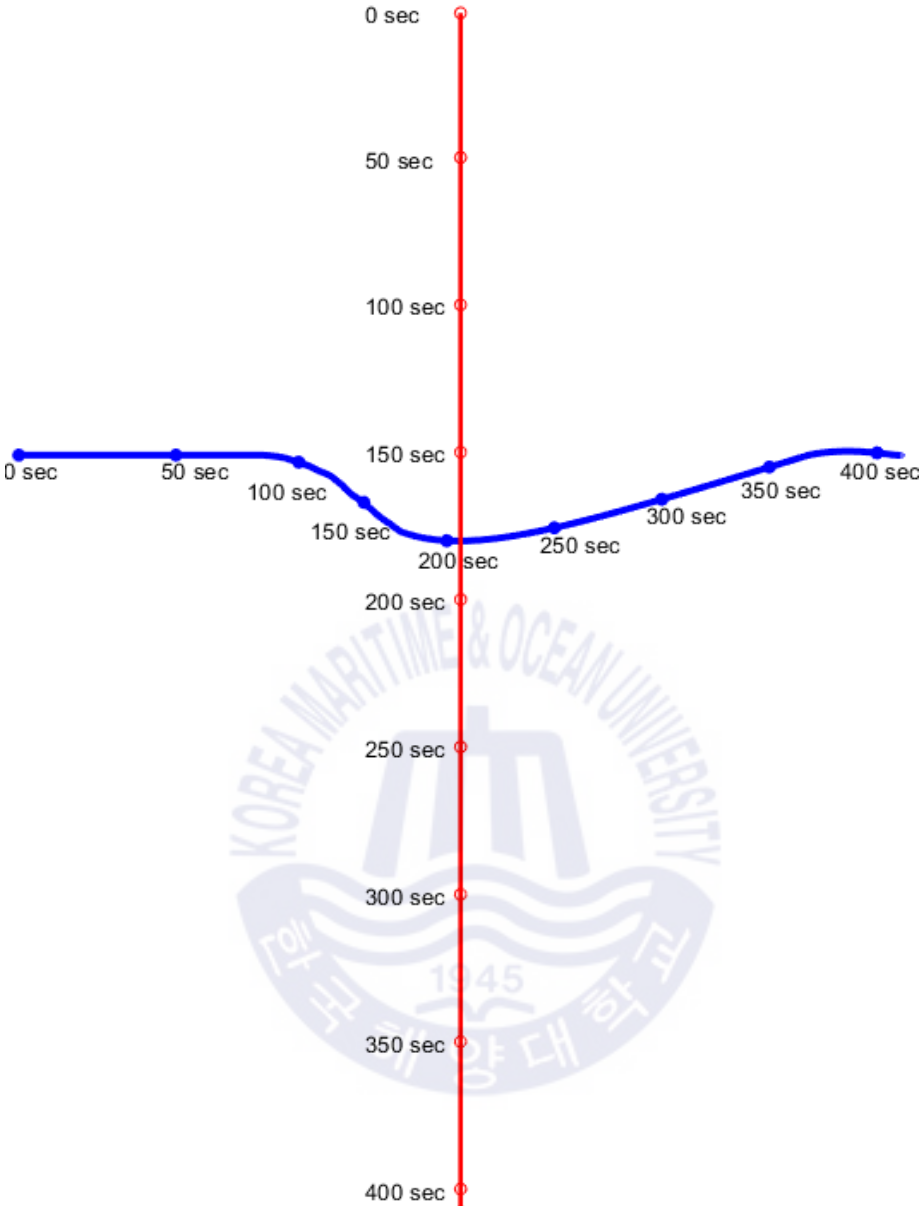


Fig. 4.5 횡단하는 상황(자선이 유지선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때

Fig. 4.5에서 자선이 유지선의 의무임에도 자선이 피항선의 의무일 때와 비슷한 시점에서 피항을 준비하는 모습을 볼 수 있다. 시간이 흐름에도 타선이 피항의 의도를 보이지 않자 우현으로 변침하여 충돌회피에 성공하는 것을 볼 수 있다.

2) 횡단하는 상황(자선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : Fig. 4.6은 횡단하는 상황에서 자선이 피항선의 의무, 타선이 유지선의 의무를 갖는 상황에 대하여 타선이 유지선의 의무를 지키지 않고 좌현으로 변침하는 상황에 대하여 충돌회피를 수행한 결과이다.

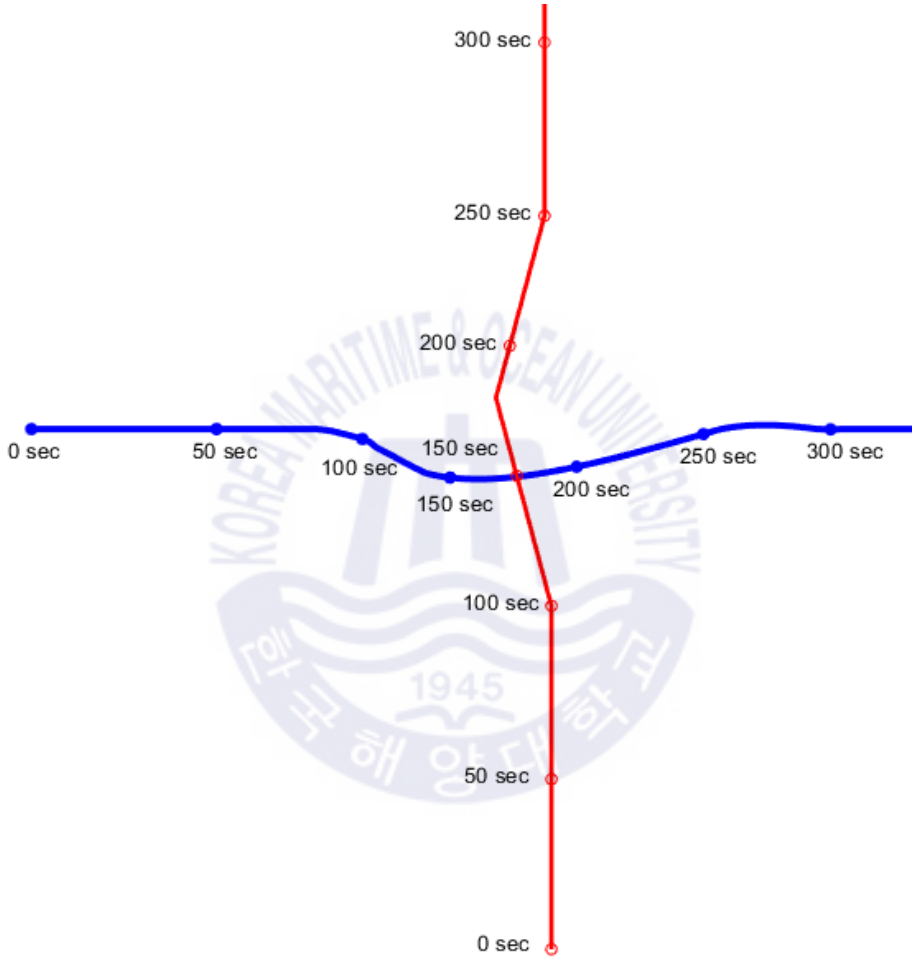


Fig. 4.6 횡단하는 상황(자선이 피항선의 의무를 갖는 경우)에서 타선이 피항 때

Fig. 4.6에서 자선이 피항을 시작하였음에도 타선이 100sec에 좌현(자선 쪽)으로 변침하여 다가오자 우현으로 더 변침하여 피항하는 것을 확인할 수 있었으며, Fig. 4.3과 Fig. 4.6의 결과와 비교하였을 때 Fig. 4.6에서는 타선이 유지선의 의무를 지키지 않고 좌현으로 다가오는 경우 우현으로 크게 변침하여 충돌회피에 성공하는 것을 확인할 수 있었다.

3) 마주치는 상황(양선이 피항선의 의무를 갖는 경우)에서 타선이 규정을 준수하지 않을 때 : Fig. 4.7은 마주치는 상황에서 자선과 타선 모두가 피항의 의무를 갖지만, 타선은 피항하지 않고 항로를 유지하는 상황에 대하여 충돌회피를 수행한 결과이다.

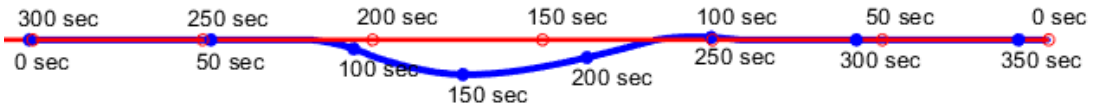


Fig. 4.7 마주치는 상황에서 피항선의 의무일 때

Fig. 4.7은 타선이 피항의 의무를 보이지 않자 Fig. 4.4에 비해 크게 피항하는 것을 확인할 수 있었다. 마주치는 상황에서 타선의 피항 여부와 타선의 피항 각도에 따라서 자선의 피항 경로가 다른 것을 확인할 수 있었으며 이는 심층강화학습을 적용한 자선이 타선의 행동에 대비할 수 있게 일정 거리를 유지하는 것으로 분석된다.

제 5 장 결론 및 향후 연구

항해사의 부주의로 인한 선박 사고 감소를 위해 항해사의 수준으로 의사결정을 할 수 있는 자율운항 시스템 개발은 과거부터 다양한 방법을 통해 제안되었다. 심층강화학습을 이용한 충돌회피 시스템을 실제 선박에 사용하기 위해서는 시뮬레이션 상에서 높은 정확도를 갖는 모델이 필요하다. 본 논문에서는 심층강화학습을 기반으로 충돌회피 시스템을 개발하고 심층강화학습을 이용하여 충돌위험이 있는 여러 가지 상황을 학습하였으며 COLREGs 규정을 준수하는 선박과 COLREGs 규정을 준수하지 않는 선박에 대하여 충돌회피 수행능력을 분석하였다. 본 실험을 바탕으로 세 가지 결론을 도출하였다.

첫째, 심층강화학습을 통하여 학습된 결과를 분석해본 결과, 다양한 시나리오 결과의 공통점으로 충돌회피를 시작하는 시점이 비슷한 것을 볼 수 있다. 이는 스스로 학습을 통하여 COLREGs 규정을 준수하는 선박인지 규정을 준수하지 않는 선박인지 판단하기 이전에 보상을 최대로 받을 수 있는 회피 시점을 학습한 것으로 분석된다.

둘째, 심층강화학습을 통하여 COLREGs 규정을 준수하는 선박과의 조우상황에서 타선이 COLREGs 규정대로 움직이는 상황을 학습하여 COLREGs 규정을 준수하지 않는 선박과의 조우상황에서 COLREGs 규정을 모방하여 충돌회피하는 것을 확인할 수 있으며 이는 보상을 통하여 COLREGs 규정을 학습한 것으로 판단된다.

셋째, Fig. 4.4와 Fig. 4.7의 실험결과를 비교해본 결과 유사한 상황임에도 타선의 피항조치 유무와 피항 각도에 따라서 경로가 다른 것을 확인하였다. 이는 자선과 타선과의 충돌위험도를 스스로 학습하여 반영하는 것

으로 분석된다.

본 논문에서는 선박의 운동역학을 반영하지 않아 실제 선박의 움직임과는 차이가 있을 수 있다. 실제 선박의 제원과 선박의 운동역학을 반영하여 실제 선박의 움직임을 표현할 수 있다면 더욱 정확하고 안전한 선박의 경로를 탐색할 수 있을 것이다. 또한, 심층강화학습은 보상에 따라 학습을 어떤 방향으로 할 것인지 달라진다. 따라서, 선박의 안전영역, CPA에 대한 연구를 보상으로 적용하는 연구도 진행되어야 한다.

참고문헌

[1] 우주현, (2018). “심층강화학습을 이용한 무인수상선의 충돌회피.“ 박사학위논문 서울대학교 대학원

[2] 김동함, (2019). “Velocity Obstacles와 심층강화학습을 이용한 VLCC급 유조선의 충돌회피 방법.“ 박사학위논문 한국해양대학교 일반대학원

[3] 손남선, 윤근항, 황태현, (2014). 무인선 자율운항 시스템 개발. 대한조선학회지, 51(2), pp. 18-22.

[4] 해양경찰청, (2019). 2018년 해상조난사고 통계연보

[5] Howard, R. A. (1960). Dynamic programming and markov processes.

[6] Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine learning, 8(3-4), pp. 279-292.

[7] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), pp. 484-489.

[8] Fujii, Y., & Tanaka, K. (1971). Traffic capacity. The Journal of navigation, 24(4), pp. 543-552.

[9] Colley, B. A., Curtis, R. G., & Stockel, C. T. (1983). Manoeuvring times, domains and arenas. The Journal of Navigation, 36(2), pp. 324-328.

[10] Goodwin, E. M. (1975). A statistical study of ship domains. The

Journal of navigation, 28(3), pp. 328-344.

[11] Hasegawa, K., & Kouzuki, A. (1987). Automation collision avoidance system for ships using fuzzy control.

[12] Ahn, J. H., Rhee, K. P., & You, Y. J. (2012). A study on the collision avoidance of a ship using neural networks and fuzzy logic. Applied Ocean Research, 37, pp. 162-173.

[13] 정회룡, 장형준, 송영은, (2019). 대형 무인 선박의 자율운항 기술 개발동향. 제어로봇시스템학회 논문지, 25(1), pp. 76-87.

[14] IMO Std. COLREGs, (1972). Convention on the International Regulations for Preventing Collisions at Sea, *IMO*.

[15] 해사안전법 [2020. 02. 18. 법률 제17056호]

[16] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[17] Feinberg, E.A. and Shwartz, A., eds, (2002). Handbook of Markov Decision Processes. Boston, MA: Kluwer.

[18] Bertsekas, D. P., Bertsekas, D. P., Bertsekas, D. P., & Bertsekas, D. P. (1995). Dynamic programming and optimal control (Vol. 1, No. 2, p. 4). Belmont, MA: Athena scientific.

[19] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

[20] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

[21] Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In

Advances in neural information processing systems, pp. 1008-1014.