

파이썬을 이용한 최적 퇴선 경로 산정에 대한 연구

김원옥 · 홍정혁†

한국해양수산연수원(교수)

A Study on Optimum Abandon ship Route Calculation Using Python

Won-Ouk KIM · Jung-Hyeok HONG†

Korea Institute of Maritime and Fisheries Technology(professor)

Abstract

Marine accidents can occur due to various factors such as collisions, sinking, strandings and fires. When faced with such emergencies, the crew's safety becomes the top priority, often necessitating the abandon ship to ensure their survival. However, abandon ship is challenging due to the narrow and complex structure of ships, especially when considering the unique environment of the sea, where professional rescue organizations may not be readily available. To facilitate efficient abandon ship, it's crucial to develop abandon ship scenarios for different situations. This requires the use of an analysis program capable of systematically evaluating and determining the optimal evacuation route. In this paper, researchers utilized Python, a widely-used programming language known for its versatility, to calculate the optimal route for abandon ship. By leveraging Python, they developed an analysis program that could quickly and systematically analyze various abandonment scenarios. The study compared the optimal routes calculated using two different programs, SEA-Pro and Python, across four scenarios. Remarkably, the results showed a high degree of similarity between the optimal paths generated by both programs in all scenarios. Moreover, the proposed method utilizes the open jupyter notebook program, eliminating any financial burden associated with adopting this approach. Additionally, the time required to calculate the optimal path is minimal, further enhancing its utility from an operational perspective. Overall, this method offers a practical and effective solution for optimizing evacuation routes, thereby improving preparedness and response capabilities in maritime emergencies. By leveraging technology and computational analysis, officer can enhance safety measures and minimize the risks associated with marine accidents.

Key words : Marine accidents, Abandon ship, Sea-Pro, Python, Jupyter notebook, Optimizing evacuation routes

I. 서론

선박에 의한 해양사고는 충돌, 침몰, 좌초 및 화재 등 매우 다양한 종류로 발생한다. 이러한 해양사고를 미연에 방지하거나 확실한 완결 조치를 하지 못한 경우 승무원들은 생명을 지키기 위해 퇴선을 감행해야한다. 하지만, 선박은 육상 건물 혹은 타 교통수단에 비해 구조가 상대적으로

협소하고 복잡하여 퇴선이 쉽지 않고 특히, 해상이라는 특수한 공간에서 운항하기 때문에 전문적인 구조 기관의 도움을 거의 받을 수 없다. 또한, 유람선, 도선, 여객선 등 경우 불특정 다수의 승객들은 승선하는 선박의 내외부 구조를 확실하게 파악하기 어렵기 때문에 더욱 퇴선이 어렵다. 따라서, 선박 안전 책임자는 원활하고 신속한 퇴선을 위해서 미리 다양한 상황에 따른 퇴선 시나

† Corresponding author : 051-620-5801, hongjh825@hanmail.net

리오를 구성하고 그 결과를 분석할 필요가 있다. 이에 최적의 퇴선 경로를 체계적이고 효율적으로 검토해야하며 특히, 생명과 관련된 분야라 골든 타임을 놓치지 않도록 신속한 분석이 필요하다. 현재 우리나라에서는 선박 퇴선에 관련한 분석 프로그램에 관해서는 거의 연구되어 있지 않다 (Park, et al., 2004). 외국의 경우는 Vassalos, et al.(2001)이 개발한 선박 침수를 고려한 보행속도 감소 적용이 가능한 Evi, 그리고 선박의 경사각을 임의적으로 적용하여 보행속도를 구하는 Maritime EXODUS 정도만 IMO Test를 거쳤다(Ha et al., 2013). 이에 이 연구에서는 머신러닝(Machine learning)의 기법 중에 하나인 강화학습(Reinforcement Learning)을 이용하여 최적의 퇴선 경로를 산출하고자 한다(Google, 2018; Yamashiita, 2016). 이때, 사용된 개발도구는 전 세계적으로 가장 많이 사용하는 프로그래밍 언어 중 하나인 파이썬을 사용하였다. 파이썬은 데이터 분석, 머신러닝, 그래픽, 학술 연구 등 여러 분야에서 활용되고 있다. 이 연구에서는 파이썬 프로그래밍 기법을 이용하여 다양한 원인으로 보다 원활하고 신속하게 퇴선 할 수 있는 최적 경로를 산출하는 방법에 대해 고찰하고자 한다. 이때, python을 이용한 최적경로의 신뢰성을 확보하기 위해 먼저 개발된 SEA-Pro와 비교분석하였다.

II. 연구 방법

먼저, 피난관련 프로그램 개발에 기본 데이터가 되는 국내외 수식을 아래와 같이 정리한다.

1. 국내외 피난관련 수식

가. IMO

선박에 대한 인명의 안전에 대한 규정은 SOLAS 협약을 기본으로 각 국의 사정을 고려하여 국내법으로 제정하여 시행하고 있다(KR, 2006a). 피난 관련 내용은 제II-2에서 규정하고

있으며 화재안전코드(Fire Safety System Code, 이하 FSS Code) 제13장 탈출 수단 배치편에 상세하게 기술하고 있다(KR, 2006b).

실제 피난에 소요되는 시간인 필요피난시간을 IMO MSC.1/Circ.1238의 피난지침식에 의하여 분석하였다. IMO 피난지침식은 식(1)과 같다.

$$1.25(A+T)+2/3(E+L) \dots\dots\dots (1)$$

여기서, A는 피난반응시간(awareness time), T는 필요피난시간(travel time), E는 승정시간(embarkation time), L은 진수시간(launching time)을 나타낸다(Choi et al., 2010).

나. SFPE

SFPE(Society of Fire Protection Engineers)의 피난계산방법은 다음과 같다(SFPE, 2005). 군중밀도가 0.54/m² 이하일 경우는 식(2), 0.54/m² 이상일 경우의 계산방법인 식(3)을 사용한다.

$$T_p = \frac{P}{[k - 0.266kD(\leq \text{ss than } 0.54)]W_e} \dots\dots\dots (2)$$

$$T_p = \frac{P}{[1 - 0.266D(\text{more than } 0.54)]kDW_e} \dots\dots\dots (3)$$

여기서, T_p : 특정지점을 통과하는데 걸리는 시간(s), P : 인원수(명), k : 피난 속도 상수(복도, 비상구는 1.4 적용), D : 군중밀도[인원수(명)/유효면적(m²)], W_e : 비상구 폭(m)이다.

다. 일본

일본의 경우 건축방재지침에 의해 식(4)와 같이 한 집단의 인원(마지막 인원)이 피난구를 통과하는 시간과 식(5)와 같이 최대이동거리에 대한 피난이동속도를 계산한다.

$$e_1 = \frac{\sqrt{P}}{0.25 \sum W_1} \dots\dots\dots (4)$$

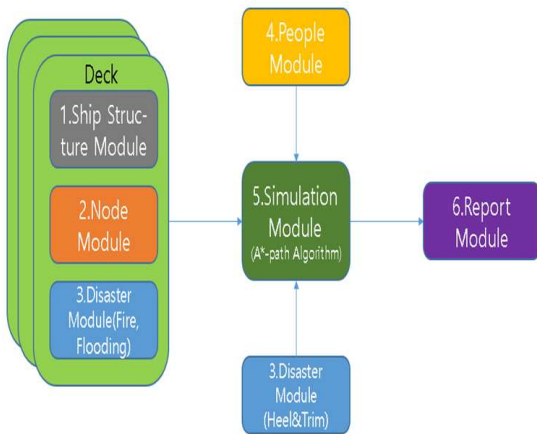
여기서, P는 인원수, W₁은 피난구폭(m)이며 0.25는 상수로 피난구가 통로일 경우 사용된다.

$$e_2 = \frac{L}{V} \dots\dots\dots (5)$$

여기서, L 은 최대이동거리(m), V 는 보행속도 (m/s)이다.

2. SEA-Pro

이 장에서는 파이썬을 이용한 최적퇴선 경로 산정의 신뢰성 확보를 위해 먼저 개발된 선박퇴선분석 프로그램인 SEA-Pro를 간단히 설명하고자 한다(Kim and Kim, 2018). SEA-Pro는 선박에서 퇴선을 수행할 경우 선원 및 승객들의 움직임을 분석하는 것을 목적으로 하고 있으며 프로그램 개발 tool은 Windows 10 Pro 환경에서 C# 6.0 프로그래밍 언어를 사용하였다. 그리고 통합 개발툴(IDE)로는 Visual Studio 2015 community를 사용하였다. SEA-Pro는 Windows 환경에서 GUI프로그램 제작 시 주로 사용되는 Windows Forms 사용자 인터페이스(Leffingwell, 2007)를 이용하여 프로그램을 제작하였다. 퇴선상황 재현에 필요한 요소들로는 선박, 인원, 시뮬레이션(동작모듈)으로 크게 나누어질 수 있다. 선박에 대한 부분은 외형에 해당하는 선박구조, 선박 내부에 사람이 존재할 수 있는 공간인 노드, 사고를 나타내는 재난으로 나누어진다. [Fig. 1]은 모듈별 관계도이다.



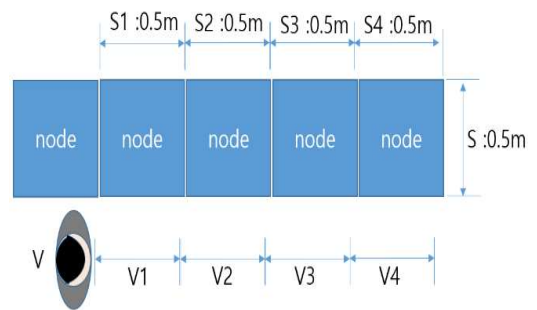
[Fig. 1] Diagram of Module.

퇴선시간 분석에 가장 중요한 요소인 이동 계산방법은 선원 및 승객은 가장 가까운 탈출구의 최단거리로 탈출한다는 cell 모델을 이용하며 cell 모델을 [Fig. 2]에서 보는 바와 같이 세로는 x축, 가로는 y축으로 $\sqrt{x^2+y^2}$ 값이 작을수록 최단거리가 된다. 즉, 왼쪽 제일 위(0.0)에서 출발하여 오른쪽 제일 아래쪽(8.48)으로 탈출하는 경우를 가정할 경우(단, 검은색의 굵은 실선의 네모는 장애물) 최단거리는 굵은 실선과 같다(MST, 2003). 특히, 이미 셀을 점유하고 있는 경우는 재난상황이나 다른 인원들이 있는 것으로 판단하여 주변 최소값을 분석하여 움직인다.

Start	1.00	2.00	3.00	4.00	5.00	6.00
1.00	1.41	2.23	3.16	4.12	5.10	6.08
2.00	2.24	2.83	3.61	4.47	5.39	6.32
3.00	3.16	3.61	4.24	5.00	5.83	6.71
4.00	4.12	4.47	5.00	5.66	6.46	7.21
5.00	5.09	5.39	5.83	6.40	7.07	7.81
6.00	6.08	6.32	6.71	7.21	7.81	8.48 Exit

[Fig. 2] Concept of Cell Model.

이때, 이동거리와 이동시간을 구하는 방법은 [Fig. 3]과 같이 사람의 평균 어깨 너비인 0.5m × 0.5m 크기의 노드를 설정하여 거리[식(6)]을 구하고 상황별 보행속도를 나누어 이동시간[식(7)]을 구한다.

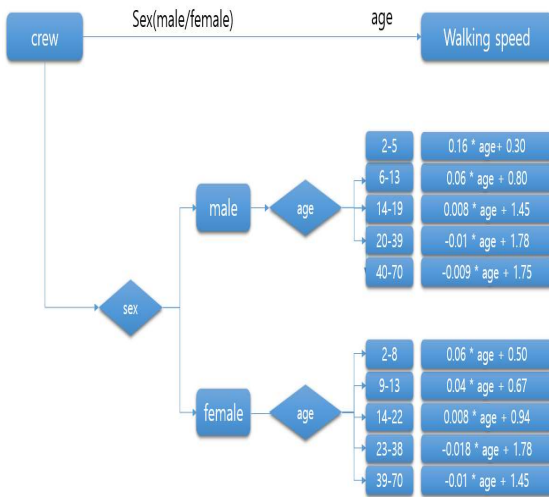


[Fig. 3] Move distance and walking time.

$$Evacuation\ distance = S1 + S2 + S3 + S4 \dots (6)$$

$$Evacuation\ time = \frac{V1}{S1} + \frac{V2}{S2} + \frac{V3}{S3} + \frac{V4}{S4} \dots (7)$$

여기서 S 는 이동거리, V 는 보행속도를 의미한다. 연령별 보행속도는 연령별로 조금씩 달라지는데 [Fig. 4]와 같이 계산된다(Ando and Oki, 1988)

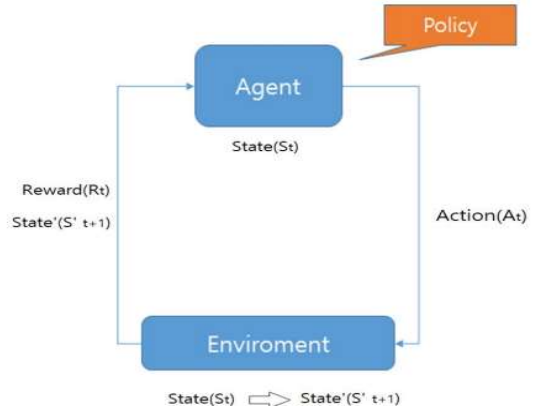


[Fig. 4] Walking speed by age.

3. Python

강화학습이란 컴퓨터를 이용하여 현재 상태에서 어떤 행동을 취하는 것이 최적인가를 학습하여 분석하는 것이다. 이때 하나의 행동을 취할 때마다 보상이 주어지며 이 보상이 최대화하는 방향으로 학습된다. 이 경우 강화학습에 요소인 agent, environment, state, reward에 대한 적절한 설정이 필요하다 이 연구에서 사용된 도구는 필요하다 이 연구에서 사용된 도구는 당 언어는 python을 사용하였다(Yamashiita, 2016).

강화학습에 대한 개요도는 [Fig. 5]와 같다.



[Fig. 5] Concept of Reinforcement Learning.

[Fig. 6]은 jupyter notebook에서 python언어를 이용하여 제작된 프로그램을 폴더에 저장한 화면이다.



[Fig. 6] Main page of Jupyter notebook.

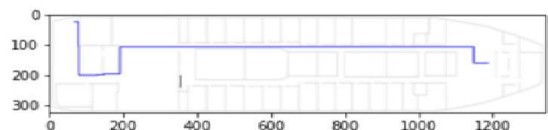
[Fig. 7]은 다양한 변수를 적용하여 강화학습을 적용하여 구해진 결과이다.

```

25 # 최적 경로 찾기
26 shortest_path = find_shortest_path(graph, start_point[:2], end_point[:2])
27
28 # 이미지에 최적 경로 표시
29 image_with_path = image.copy()
30 for i in range(len(shortest_path)-1):
31     cv2.line(image_with_path, shortest_path[i], shortest_path[i+1], (255, 0, 0), 2)
32
33 plt.imshow(cv2.cvtColor(image_with_path, cv2.COLOR_BGR2RGB))
34

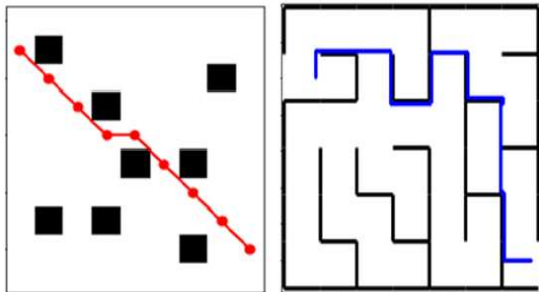
```

Out [2]: <matplotlib.image.AxesImage at 0x21ef730a070>

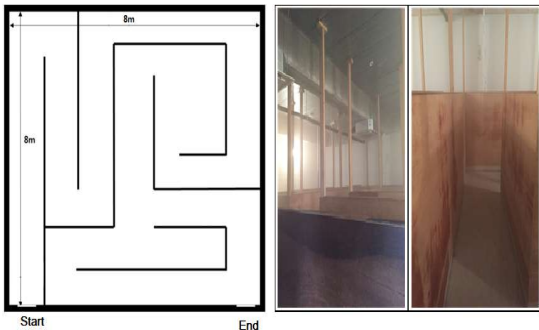


[Fig. 7] Result of Reinforcement Learning.

최적경로는 재난상황이 발생한 구역을 회피하여 최단거리를 구하는 것으로 인명생존율 향상을 위해 이를 확인할 필요가 있다. 이 연구에서는 파이썬을 사용하여 재난구역을 임의 지정하여 컴퓨터가 이를 인식하고 최적 퇴선 경로를 예측하는 시뮬레이션을 수행하고자 한다. 특히, 최적 경로 산정에 주로 사용되는 길 찾기 알고리즘(A* pathfinding)을 사용하여 최단거리를 탐색, 처음부터 시작지점에서 도착지점을 확실히 알고 있다는 가정 하에 분석된다(Kim and Kim, 2018). [Fig. 8]은 이러한 기능을 가진 python을 이용하여 최적 경로를 다양한 환경에서 구한 예시이다. 위쪽 2개의 그림은 미로에서 길을 찾는 방법으로 선박을 처음 승선하는 승객들에게는 선내 구조는 미로와 같이 느껴진다. 아래쪽 2개는 특정 공간에 장애물이 있을 경우 이를 회피하여 최적경로를 찾는 결과물이다.



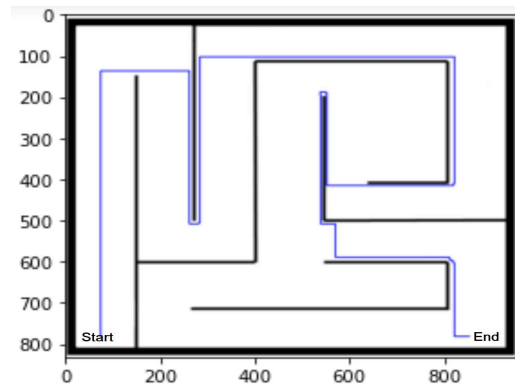
[Fig. 8] Example of optimum evacuation route.



[Fig. 9] Structure & Actual darkroom.

다음은 한국해양수산연수원에서 실시하는 안전 교육 중 실제 선박에 화재가 발생하여 연기로 인해 주변을 확인하지 못할 경우 퇴선을 대비한 암실 훈련장을 대상으로 최적 경로를 확인하고자 한다.

Kim(2016)의 연구에 의하면 총 474명을 대상으로 탈출훈련을 수행한 결과 선원들에게 암실훈련장의 구조를 설명하지 않은 경우, 대피허용한계 거리인 5 m일 경우에 비해 앞이 보이지 않는 가시거리 0 m인 경우 최대 약 2.5배 증가함을 알 수 있었다. 특히, 전체 대상인원 중 약 10%에 해당되는 훈련생은 도착지점을 찾지 못하고 훈련장에 갇혔다. [Fig. 10]은 python을 이용하여 최적 탈출 경로를 구한 결과이다.



[Fig. 10] Optimum evacuation route.

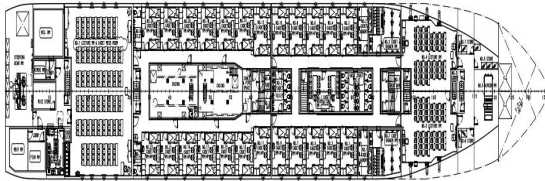
III. 연구 결과

이 연구에서는 파이썬을 이용하여 구해진 최적 경로의 신뢰성을 확보하기 위해 우리 학회에 먼저 게재된 SEA-Pro를 참조 한다(Kim and Kim 2018). 이때 사용된 선박은 한국해양수산연수원의 실습선인 한반도호(전장 103.5m, 선폭 16.0m, 흘수 5.4m, 배수량 4,898M/T)이다([Fig. 11]).



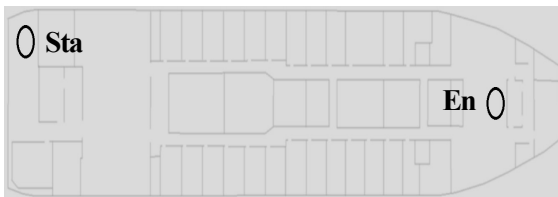
[Fig. 11] Trainingship HANBANDO.

여러 deck 중 실습생이 생활하는 거주구역인 main deck([Fig. 12])의 도면을 이용하여 최적경로를 분석하였다.



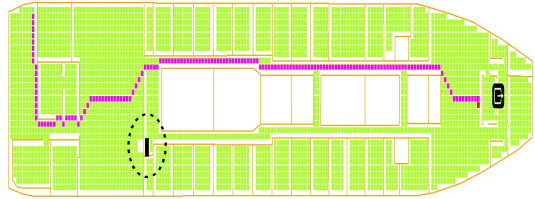
[Fig. 12] Trainingship HANBANDO Main deck.

[Fig. 13]은 한반도호 메인테크 도면으로 출발점과 도착점을 지정한 것이다.



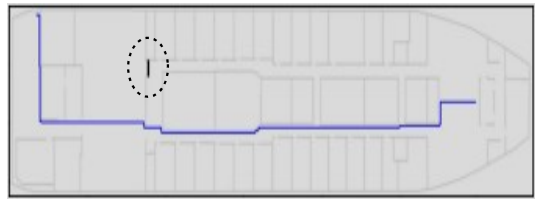
[Fig. 13] Specify the starting and ending points.

[Fig. 13]은 미리 지정된 출발점과 도착점에 선미 우측 통로부분(원 모양)에 화재나 침수 등 재난상황이 발생하여 퇴선 경로로 이용하지 못하는 경우를 산정하여 최적경로를 구한 것이다. 위쪽 그림은 파이썬을 이용하여 구한 최적경로이며 아래 그림은 SEA-Pro를 이용하여 구한 경로이다. 보는 바와 같이 거의 일치함을 확인할 수 있었다.



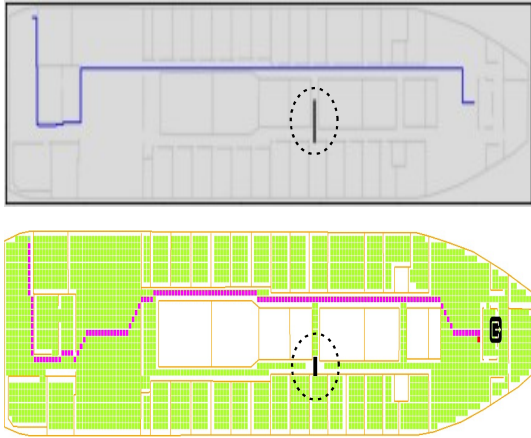
[Fig. 13] scenario ①.

[Fig. 14]는 선미 좌측 통로부분(원 모양)에 화재나 침수 등 재난상황이 발생하여 퇴선 경로로 이용하지 못하는 경우를 산정하여 최적경로를 구한 것이다. 이 시나리오의 경우도 보는 바와 같이 거의 일치함을 확인할 수 있었다.



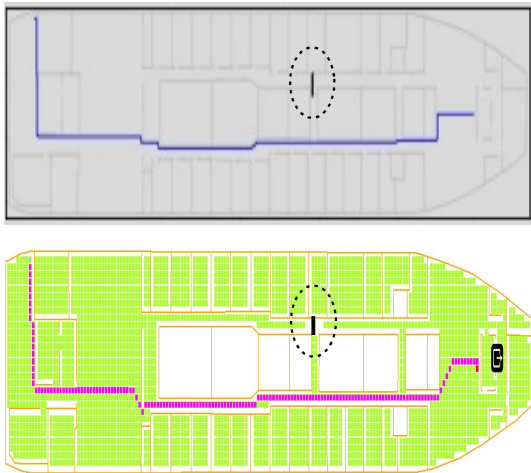
[Fig. 14] scenario ②.

[Fig. 15]는 메인테크 중앙 우측 통로부분(원 모양)에 재난상황이 발생하여 퇴선 경로로 이용하지 못하는 경우를 산정하여 최적경로를 구한 것으로 보는 바와 같이 거의 일치함을 확인할 수 있었다.



[Fig. 15] scenario ③.

마지막으로 [Fig. 16]은 메인데크 중앙 좌측 통로부분(원 모양)에 재난상황이 발생하여 퇴선 경로로 이용하지 못하는 경우를 산정하여 최적경로를 구한 것으로 보는 바와 같이 거의 일치함을 확인할 수 있었다.



[Fig. 16] scenario ④.

총 4가지의 시나리오를 대상으로 SEA-Pro와 파이썬을 이용한 최적경로 산정 시 모두에서 거의 일치함을 알 수 있었다. SEA-Pro의 경우는 프로그램을 구입하여 도면을 업로드하여 사용자가 노드를 지정하여야 그 결과를 알 수 있다. 즉,

분석시간이 이 연구에서 제안한 방법보다 많이 소요된다. 또한, 전용 분석프로그램을 구입하여 사용해야하는 반면에 파이썬을 이용하는 경우 무료로 제공되는 jupyter notebook이라는 툴을 이용하여 간단한 방법에 의해 도면을 그림파일로 캡처하여 업로드하고 출발점과 도착점을 사용자가 마우스를 통해 지정만 해주면 최적경로를 구할 수 있다. 무엇보다 최적경로를 계산하는 시간도 SEA-Pro보다 훨씬 단축되어 경제적, 시간적 효율성이 아주 높아진다.

IV. 결론

국내외 피난 최적 경로 및 시간을 구하는 프로그램은 주로 육상건축물에 대한 것으로 선박을 대상으로 하는 프로그램은 거의 없는 실정이다. 선박에서는 SOLAS에 의거 매월 1회 퇴선훈련을 실시하고 있는데 많은 선박에서 미리 정해진 시나리오로 훈련을 수행하고 있지 않은 실정이다. 즉, 다양한 상황을 고려한 훈련은 잘 이루어지고 있지 않는 실정이다. 특히, 여객선, 실습선 등 많은 인원이 승선하는 선박의 경우 재난으로 피난 동선을 이용하지 못하는 경우를 실시간 확인하여 다른 최적 경로를 예상하여 인원들을 이동시키는 훈련은 쉽지 않다. 더구나, 생명을 보장해야하는 경우 골든타임을 놓치지 않기 위해 신속한 퇴선 동선 파악이 필요한데 이런 경우 프로그램을 통한 분석이 반드시 필요하다. 이 연구에서 제안한 방법의 장점을 정리하면 아래와 같다.

① 선박의 도면을 그림파일로 간단히 캡처하여 그 파일을 업로드한 후 재난구역과 출·도착점을 사용자가 임의 지정하여 실시간 퇴선동선 산정하기 때문에 누구나 쉽게 사용이 가능하다.

② 사고 장소 추가 발생 등으로 변경된 경로를 신속하게 확인하여 퇴선을 유도할 필요가 있을 경우 빠른 판단이 가능해진다.

이 연구는 강화학습에 의한 최적경로 산정에

대한 것으로 향후, 이렇게 구해진 퇴선경로를 선박 내 피난구 유도등 등과 같이 경로를 표시해주는 장치와 연결한 지능형 퇴선장비에 대한 연구 및 화재 및 침수 등으로 보행속도가 감소되는 상황을 고려한 총 퇴선시간 확인 등의 기능이 추가된 연구가 추가로 필요할 것이다.

References

- Ando K, Ota H and Oki T, "Forecasting the flow of people," *Railway Research Review*, vol. 45, no. 8, 8~14, 1988.
- Choi J, Kim SY, Shin SC, Kang HJ and Park BJ(2010). Development of an Evacuation Time Calculation Program for Passenger Ships Based on IMO Guidelines, MSC.1/Circ.1238, *Journal of the Society of Naval Architects of Korea*, 47(5), 719~724.
<https://doi.org/10.3744/SNAK.2010.47.5.719>
- D Vassalos, Kim HS, G Christiansen and J Majumder(2001). A mesoscopic model for passenger evacuation in a virtual ship-sea environment and performance based evaluation, *Conference on Pedestrian and Evacuation Dynamics*, Duisburg, Germany, 4~5 April 2001.
- Google(2018). <https://ko.wikipedia.org/wiki/>.
- Ha S, Cho YO, Ku NK, Lee KY and Roh MI (2013). Passenger Ship Evacuation Simulation Considering External Forces due to the Inclination of Damaged Ship,50(3), 175~181.
<https://doi.org/10.3744/SNAK.2013.50.3.175>
- Kim WO and Kim DH(2018). Development Ship Evacuation and Analysis Program (SEA-Pro), *Journal of Fishier and Marine Educational Research*, 30(2), 420~430.
<https://doi.org/10.13000/JFMSE.2018.04.30.2.422>
- Kim WO(2016). A Study on Evacuation Time According to Seafarer Visibility, *Journal of the Korean Society of Marine Environment & Safety*, 22(6), 600~606.
<https://doi.org/10.7837/KOSOMES.2016.22.6.600>
- Kim MK, Kim WH and Hagiwara I(2001). New Evaluation Method of Egress Safety in the Japanese Building Standard Law, *Proceedings of 2014 KIFSE Annual Fall Conference*, 198~203.
- Korea Registry(2006a). Ministry of Oceans and Fisheries, SOLAS-Consolidated ed. 2006, Haein press, 359~661.
- Korea Registry(2006b). Ministry of Oceans and Fisheries, International Code for Fire Safety Systems(FSS code), Haein press, 99~138.
- Leffingwell, D.(2007). *Scaling Software Agility*, Part II, 102~114.
- Ministry of Science and Technology(2003). The Development of simulation system available for the analysis of fire behavior and pre-estimate of egress feature in internal space of a building, HOSEO University, 19~21.
- Park JH, Lee DG, Kim HT and Yang Y(2004). Development of Evacuation model for human safety in maritime casualty, *Ocean Engineering* 31, 1537~1547.
- SFPE(2005c). *SFPE Handbook of Fire Protection Engineering*, (3~367) - (3~371).
- Yamashiita, T(2016). *Illustration De Manabu Deep Learning*, jpub, 198~199.

-
- Received : 27 July, 2024
 - Revised : 26 August, 2024
 - Accepted : 03 September, 2024