

이산사건 시뮬레이션을 이용한 수중 선체 탐색 시간 예측 모델 개발

이주필¹ · 함승호^{2,†}

국립창원대학교 첨단방위공학과¹

국립창원대학교 조선해양공학과²

Development of Underwater Hull Search Time Prediction Model with Discrete Event Simulation

Joopil Lee¹ · Seung-Ho Ham[†]

Department of Advanced Defense Engineering, Changwon National University¹

Department of Naval Architecture and Marine Engineering, Changwon National University²

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the event of a maritime accident, search plans have traditionally been planned using experiential methods. However, these approaches cannot guarantee safety when the scale of a maritime accident increases. Therefore, this study proposes a model utilizing discrete event simulation (DES) to predict the diving time for compartment searches of a ship located on the seabed. The discrete event simulation model was created by applying the DEVS formalism. The M/V Sewol sinking was used as an example to simulate how to effectively navigate compartments of different sizes. The simulation results showed the optimal dive time with the number of decompression chambers needed to navigate the compartment as a variable. Based on this, we propose a methodology for efficient navigation planning while ensuring diver safety.

Keywords : Modeling&Simulation(모델링 및 시뮬레이션), Discrete Event Simulation(이산사건 시뮬레이션), Underwater Hull Search (수중 선체 탐색), Diving(잠수), Decompression Chamber(감압 챔버)

1. 서론

1.1 연구 배경 및 필요성

선박의 해난 사고는 역사적으로 큰 인명 및 재산 피해를 초래해 왔다. 타이타닉호 침몰 사고(1912)부터 서해 페리호 침몰 사고(1993), 천안함 피격 사건(2010) 및 세월호 침몰 사고(2014)에 이르기까지 해난 사고는 꾸준히 발생해 왔다.

선박이 침몰하는 사고가 발생할 경우, 해저에 가라앉은 선체 내부의 격실을 잠수사가 탐색 작업을 실시하게 된다. 특히 최근 들어 선박의 규모가 대형화되고 구조가 복잡해지면서, 침몰된 선체 내부를 탐색하는 잠수 작업은 매우 위험성이 높아지고 있다. 또한, 잠수 시간에 따라 다시 수면 위로 올라가기 전에 감압을 해야 하는데 감압 시간은 잠수사가 머물렀던 최대 수심과 시간에 따라 달라진다. 이러한 복잡한 감압 절차 외에도 잠수 작업은 다양한 변수에 의해 영향을 받으며, 단순한 계산으로는 결

과를 예측하기 어렵다.

잠수 작업의 복잡성과 위험성 때문에 침몰 선체의 격실 탐색 계획은 종종 과거 잠수 경험을 바탕으로 추정되었다. 그러나 이러한 추정은 고려해야 할 사항이 많아질수록 어려워지며, 단순화 과정을 거치면서 실제와 계획 사이에 큰 차이가 발생할 수 있다.

세월호 사고와 같이 초대형 여객선이 침몰한 경우, 다양한 크기의 격실이 존재함에 따라, 탐색 우선순위, 다이빙 사이트의 수, 필요한 챔버 수 등을 추정하는 것은 사실상 불가능에 가까웠다. 따라서 본 연구는 이산사건 시뮬레이션(DES)을 활용하여 선박 격실 탐색 계획을 시뮬레이션하고, 이를 통해 최적의 탐색 계획을 수립할 수 있는 예측 모델을 제안하고자 한다.

1.2 관련 연구 현황

이산사건 시뮬레이션(DES)을 활용한 연구는 다음과 같다. Hong (2018)의 연구에서는 철근 가공 공장의 생산성 향상을 위한

Table 1 Related research on applications

Author(Year)	Key points(application)
Hong (2018)	Production planning in a rebar factory
Cha and Roh (2010)	Developing a hybrid simulation kernel and framework to improve efficiency in shipbuilding process planning
Son et al. (2010)	Simulation with DEVS-HLA-based modeling methods
Cho et al. (2012)	Using discrete-event simulation techniques to predict hoisting times for tower cranes
Chung et al. (2008)	Developing simulation methods to effectively hit artillery
This study	Development of underwater hull search time prediction model

최적화 방안을 DES를 사용하여 시뮬레이션하였다. 이 연구는 시뮬레이션 결과를 이용하여 실제 공장 운영 시 손실을 감소에 기여할 수 있는 방법을 제시하였다. Cha and Roh (2010)의 연구에서는 조선 산업의 공정 계획을 개선하기 위해 DES와 이산 시간 시뮬레이션(DTS)을 혼합한 새로운 접근 방식을 제안하였다. 이 방법은 조선 공정의 복잡성을 효과적으로 관리하고 정확도를 높이는 방법을 제시하였다. Son et al. (2010)의 연구는 수중운 동체 시뮬레이션을 위한 DEVS-HLA 프레임워크를 개발하였으며, 이는 다양한 시스템 간 상호운용성 향상 방법을 제시하였다. Cho et al. (2012)의 연구는 고층건축공사에서 타워크레인의 양중시간 예측 모델을 개발하여 공사 프로젝트의 효율성을 개선하는 방안을 제시하였다. 마지막으로, Chung et al. (2008)의 연구는 DES를 이용한 갯도 포병 타격 분석을 수행하여 군사 작전의 포 병 활동 이해와 대응 전략 개발에 중요한 정보를 제공하였다.

이들 연구에서 볼 수 있듯이 이산사건 시뮬레이션의 경우 적용할 수 있는 분야는 무궁무진하다. 또한 시뮬레이션 결과를 바탕으로 복잡한 시스템을 설계하고 최적화할 수 있다. 하지만 잠수/탐색 분야에서 적용된 사례나 연구는 찾아볼 수 없었다. 따라서 본 연구에서는 이산사건 시뮬레이션을 기반으로 선박이 침몰 시 선체 내부 격실에 대한 탐색 계획 수립 방안을 예측해 보았다.

제2장에서 이산사건 모델링 및 시뮬레이션(M&S)의 기초를 설명하고, 제3장에서 본 연구에서 적용한 DEVS(Discrete Event System Specification) 형식론(formalism)에 대해 상세히 기술하였다. 제4장에서는 세월호 침몰 사고를 예제로 격실 탐색 시뮬레이션을 어떻게 구성할 것인지 기술하였다. 이어지는 제5장에서는 시뮬레이션 결과를 기반으로 최적의 격실 탐색 계획 수립 방안을 제시하였으며, 최종적으로 제6장에서는 본 논문의 결론과 향후 연구 방향에 대해 서술하였다.

2. 이산사건 모델링 및 시뮬레이션

2.1 모델링 및 시뮬레이션(M&S)

모델링은 현실 세계의 시스템이나 과정을 간소화하여 수학적, 물리적, 또는 컴퓨터 기반의 형태로 표현하는 과정이다. 이

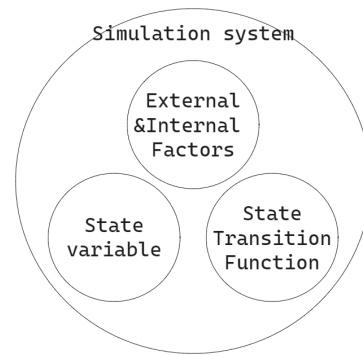


Fig. 1 Simulation system components

과정은 복잡한 현실을 이해하기 쉽고, 분석 가능한 형태로 변환하는 것을 목표로 한다. 모델링을 통해 생성된 모델은 실제 시스템의 중요한 특성을 잘 살리면서도 불필요한 복잡성은 제거하여 시스템의 동작이나 반응을 예측할 수 있는 기반이 된다.

모델은 다음 Fig. 1의 세 가지 구성 요소로 이루어진다. 상태 변수(state variables)는 시스템의 현재 상태를 기술하는 변수이고, 외적 및 내적 요인(external and internal factors)은 시스템의 상태를 변경시킬 수 있는 요인들이다. 마지막으로 상태 변경 함수(state transition function)는 외부 및 내부 요인에 의해 상태 변수가 어떻게 변화하는지 정의하는 규칙이나 함수이다. (Ahn et al., 1994)

모델링 및 시뮬레이션의 개념을 이해하기 위해, A 마트가 마을에 새로운 지점을 개설하는 상황을 예로 들어보겠다. A 마트는 이 새 지점을 방문하는 고객들이 계산할 때 평균 대기시간을 5분 이하로 유지하는 데 필요한 계산원 수를 알고 싶다. 이를 알아내는 가장 직접적인 방법은 실제로 지점을 열고 고객들의 대기시간을 측정 후 적절한 직원 수를 배치하는 것이다. 이 방법은 정확하지만, 많은 시간과 비용이 들어간다.

다른 방법으로 실제로 개설될 지점과 유사한 가상의 지점을 만들어 그곳의 고객 평균 대기시간을 확인하고, 이를 통해 적절한 직원 수를 결정하는 방법이다. 이 방법은 비용과 시간을 크게 절약할 수 있지만, 가상 지점을 실제 지점과 유사하게 잘 구현해야만 현실적인 결과를 얻을 수 있다. 여기서 모델을 구성하는 상태 변수는 대기 중인 고객 수, 현재 업무를 처리 중인 직원 수, 직원의 업무 처리 가능 여부가 된다. 그리고 이 상태 변수를 변화시키는 외적 및 내적 요인으로 고객 도착, 업무 처리 완료, 대기 상태 변경이 있으며, 이러한 요인에 의해 상태 변수가 어떻게 변화하는지를 정의하는 '상태 변경 함수'로 구성된다. 여기서 시간에 따라 이러한 모델의 상태 변수에 어떤 변화를 불러오는지를 계산하는 과정을 '시뮬레이션'이라고 한다 (Bang et al., 2007).

2.2 이산사건 시뮬레이션(DES) (Discrete Event Simulation)

이산사건 시뮬레이션은 모델의 상태를 변화시키는 특정 사건에 초점을 맞춘 시뮬레이션 방식이다. 위에서 예를 든 A 마트가 마을에 새로운 지점을 개설하는 상황에서 대기 중인 고객 수와

같은 상태 변수는 외부 사건(예: 고객 도착)이나 내부 사건(예: 고객 업무 처리 완료)에 의해 변한다. 이러한 변화는 모델의 상태를 시간에 따라 업데이트하며, 사건의 발생 순서에 따라 순차적으로 처리한다. 이산사건 시뮬레이션은 사건 기반(event driven) 방식으로, 시뮬레이션 모델을 현실에 가깝게 재현하고, 사건 발생 시간에 따라 모델 상태를 변경하며 시뮬레이션을 진행한다. 이 과정은 시뮬레이션 엔진(또는 시뮬레이터)에 의해 수행되며, 실제 상황을 모사하여 다양한 시나리오를 예측하고 분석하는 데 유용하다. 이 방법은 특히 대기열 관리, 네트워크 트래픽 분석, 제조 공정 최적화 등 다양한 분야에서 활용될 수 있다 (Bang, 2006).

3. DEVS 형식론

DEVS(Discrete Event System Specification) 형식론은 시스템의 모델링과 시뮬레이션을 위한 계층적이고 수학적인 프레임워크를 제공하는 방법론이다. 이 형식론은 1970년대에 Bernard P. Zeigler에 의해 개발되었으며, 이산사건 시스템의 구조와 동작을 정의하는 데 사용된다. DEVS는 복잡한 시스템을 모듈화하고 계층적으로 구성할 수 있는 강력한 도구로, 시스템 공학, 컴퓨터 과학, 운영 연구 등 다양한 분야에서 활용된다. DEVS 모델은 실제 시스템의 상태를 표현하는 시뮬레이션 모델과 시뮬레이션 시간을 관리하고 이벤트를 스케줄링하는 시뮬레이션 엔진으로 이루어진다.

3.1 시뮬레이션 모델

시뮬레이션 모델은 원자모델(atomic model)과 결합모델(coupled model)로 나뉜다.

원자모델은 이산사건 시스템의 계층적 구조에서 기본적인 구성 요소로서, 시스템의 행동을 기술한다. 원자모델(atomic model)의 표현식은 Fig. 2와 같다 (Son et al., 2010).

$$AM = (X, Y, S, \delta_{ext}, \delta_{int}, ta, \lambda) \quad (1)$$

- X : 이산사건 입력 집합
- Y : 이산사건 출력 집합
- S : 일련의 이산사건 상태의 집합
- $\delta_{ext} : Q \times X \rightarrow S$: 외부 상태 변경 함수
 $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$: 모든 상태
- $\delta_{int} : Q \rightarrow Q$: 내부 상태 변경 함수
- $\lambda : ta : S \rightarrow R(0, \infty +)$: 외부 상태 변경 함수
- $ta : S \rightarrow R(0, \infty +)$: 시간 진행 함수

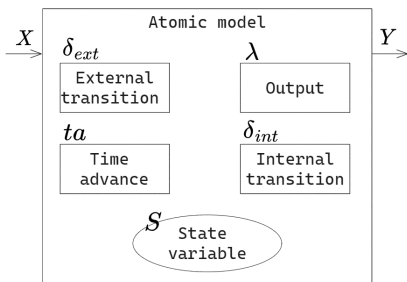


Fig. 2 Atomic model

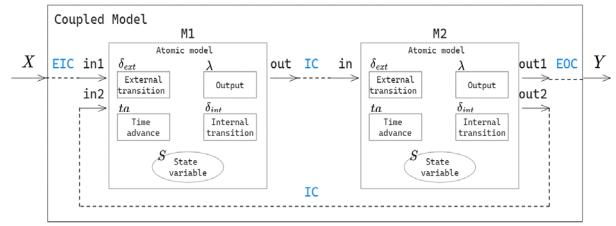


Fig. 3 Coupled model

결합모델은 여러 원자모델 또는 하위 결합모델을 포함하여, 복잡한 시스템을 계층적이고 체계적으로 표현할 수 있게 한다. 이러한 모델들은 이벤트, 외부 입력, 외부 출력 등을 통해 연결되며, 복잡한 시스템의 다양한 구성 요소 간의 상호 작용을 모델링한다. 결합모델(coupled model)의 표현식은 Fig. 3과 같다 (Son et al., 2010).

$$CM = (X, Y, \{M_i\}, EIC, EOC, IC, SELECT) \quad (2)$$

- X : 이산사건 입력 집합
- Y : 이산사건 출력 집합
- $\{M_i\}$: 모든 이산사건 컴포넌트 모델들의 집합
- EIC : 외부 입력 연결 관계
- EOC : 외부 출력 연결 관계
- IC : 내부 연결 관계
- $SELECT: 2\{M_i\} - \emptyset \rightarrow M_i$: 동일 사건의 사건 발생 모델들에 대한 선택 함수

DEVS 형식론의 중요한 장점 중 하나는 계층적이고 모듈러한 특성이다. 이는 시스템의 변경이 손쉽게 가능하다는 장점이 있으며, 특히 사건에 기반하여 진행되는 시뮬레이션에 매우 적합하다. 본 논문에서는 DEVS 형식론의 기본 구조와 주요 요소를 C# 언어로 작성하고, 이를 통해 복잡한 이산사건 시스템의 모델링과 시뮬레이션 구현법을 제시하였다.

3.2 시뮬레이션 엔진

DEVS(Discrete Event System Specification) 형식론의 시뮬레이션 엔진은 복잡한 이산사건 시스템의 모델링과 시뮬레이션을 실행하기 위한 핵심 요소다. 이 시뮬레이션 엔진은 효율적인 이벤트 처리와 시뮬레이션 관리를 위해 세 가지 주요 구성 요소인 시뮬레이터(simulator), 코디네이터(coordinator), 루트 코디네이터(root coordinator)로 나뉘며 그 구조는 Fig. 4와 같다 (Bang, 2006).

시뮬레이터는 단위 시뮬레이션 모델을 관리하며, 관리하는 모델의 동적 행동과 상태 변화를 추적한다. 코디네이터는 하위 시뮬레이터 중 가장 최근에 처리된 이벤트 시간과 다음에 처리해야 하는 이벤트 시간을 저장하여, 시뮬레이션 내 이벤트의 순서와 타이밍을 관리한다. 이를 통해 결합모델 내에서 각각의 원자모델이나 다른 결합모델들이 서로 영향을 미치는 방식을 조정하고, 시뮬레이션의 일관성을 유지한다. 루트 코디네이터는 시뮬레이션의 전체적인 시간 관리와 이벤트 처리 순서를 책임진다.

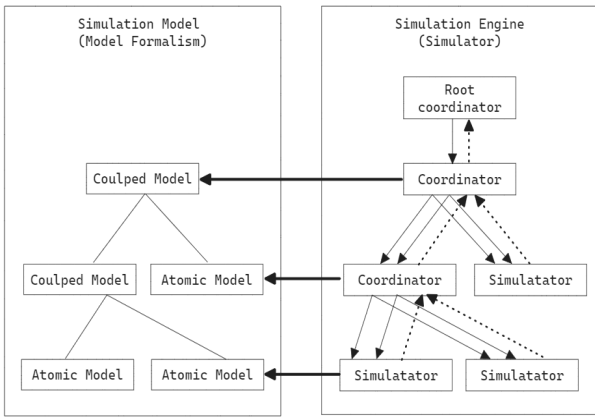


Fig. 4 Simulation Engine

하위 코디네이터의 다음 이벤트 시간을 기준으로 전체 시뮬레이션의 시간을 진행시키고, 해당 이벤트의 처리를 위한 메시지를 전달한다. 루트 코디네이터는 시뮬레이션 엔진 내에서 유일한 구성 요소로, 모든 시뮬레이션 활동의 최상위 관리자 역할을 수행한다.

3.3 시뮬레이션 예제

DEVS 형식론에 따라 간단한 시뮬레이션 예제를 살펴보겠다. 알기 쉽게 사람이 배가 고파지는 것과 그에 따라 요리를 해서 음식을 먹는 과정을 시뮬레이션하면 다음과 같다. 먼저 원자모델로 인간 모델을 만든다. 인간 모델은 다양한 속성을 가지고 있지만 본 시뮬레이션에서는 목적에 따라 음식을 먹으면 포만감이 생기고, 소화됨에 따라 배고파지는 특성만 추상화하여 모델을 만들었다. DEVS 형식론에 따라 구성 요소별로 정리하면 다음과 같다. 입력값으로 음식을 먹는다. 그러면 외부상태변경함수에서 상태변수인 음식 섭취 상태를 '포만감'으로 변경한다. 이후 시간진행함수에서 5시간을 진행한다. 이후 내부상태변경함수에서 음식 섭취 상태를 '배고픔'으로 변경한다. 이후 외부 출력함수에서 음식을 달라는 출력값을 내보낸다. 이러한 구조는 Fig. 5과 같다.

두 번째 원자모델은 요리 모델이다. 요리 모델은 입력값으로 음식을 달라는 신호를 받으면 외부상태변경함수에서 음식 상태를 '준비 중'으로 변경한다. 이후 시간진행함수에서 1시간을 진행한다. 이후 내부상태변경함수는 음식 준비가 완료되었으므로 상태

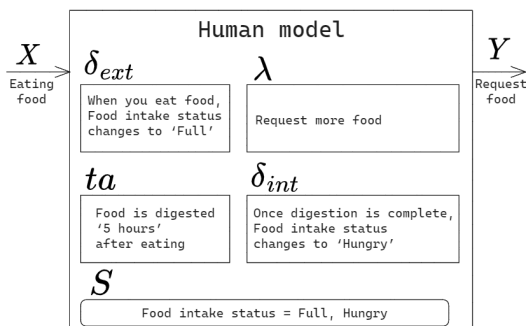


Fig. 5 DEVS example : Human model

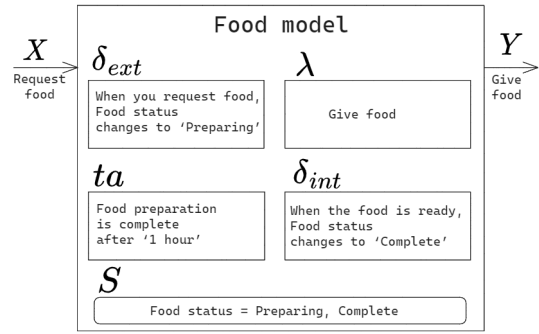


Fig. 6 DEVS example : Food model

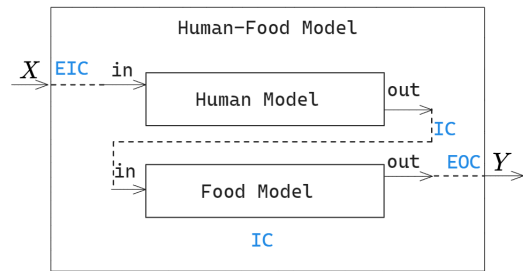


Fig. 7 DEVS example : Coupled model

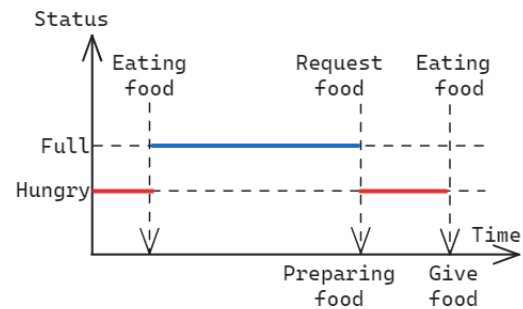


Fig. 8 DEVS example : Simulation Engine

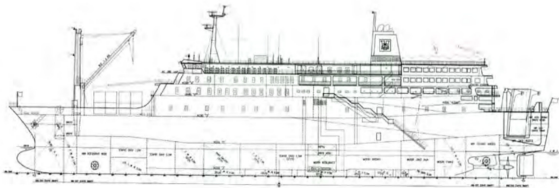
변수인 음식을 '준비 완료'로 바꾼다. 이후 외부 출력함수에서 음식 제공을 출력값으로 내보낸다. 이러한 구조는 Fig. 6과 같다.

두 개의 원자모델을 연결한 결합모델은 Fig. 7과 같다. 이처럼 단순히 원자모델을 연결한 결합모델을 만들 수도 있고 좀 더 복잡하게 연결할 수도 있다.

전체 결합모델의 시뮬레이션 엔진을 살펴보면 Fig. 8과 같다. 먼저 배고픔에 따라 음식을 먹으면 배가 부른 상태가 된다. 그 다음 소화가 되는 시간으로 이동하게 되고 다시 배고픔 상태가 된다. 이에 따라 음식 준비를 시작한다. 음식 준비가 완료되는 시간으로 이동하여 음식이 제공되면 다시 상태는 배부름이 된다.

4. 세월호 격실 탐색 모델링 및 시뮬레이션

세월호 침몰 사고는 2014년 4월 16일에 발생하였다. 이 사고로 인해 승객 476명 중 304명이 사망하거나 실종되었다. 세월호의



Name	Sewol	L.O.A[m]	145.61
Nationality	Korea	Breadth[m]	22
Type	Ro-Ro	Depth[m]	14
Max capacity [person]	956	Displacement [tons]	6,664

Fig. 9 General arrangement of M/V Sewol (Kim et al., 2018)

형상 및 제원은 Fig. 9와 같다.

본 논문에서는 사고 이후 선체 내부를 수중 탐색하는 과정에서 현장에 필요한 최적의 감압 챔버(decompression chamber) 수, 잠수 장소(diving site) 수를 구하고 모든 격실을 탐색하는데 걸리는 시간을 시뮬레이션해 보았다.

4.1 수중탐색 절차

세월호는 침몰 이후에 좌현이 바닥에 닿은 상태로 해저면에 위치하였다. 해당 해역의 최대 수심은 45m로 좌현 격실이 가장 깊은 수심(45m)에 위치하고 우현 격실이 가장 낮은 수심(23m)에 위치하였다.

여객선의 특성상 다양한 크기의 격실이 존재하였는데, 총 146개의 격실이 실종자들이 존재할 가능성이 있는 것으로 판단되어 수색하게 되었다. 격실 크기에 따라 여러 번의 탐색이 필요한 격실이 있는데 그러한 격실은 별도의 색으로 표시해 두었으며, 각 격실의 위치 및 크기는 Fig. 10과 같다.

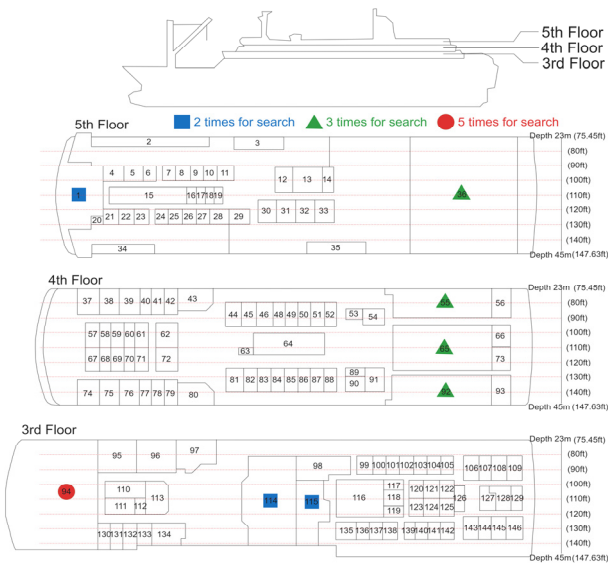


Fig. 10 Compartment configuration of M/V Sewol (Kim et al., 2018)



Fig. 11 Configuring SSSD diving equipment

침몰 선체 내부에 들어가 탐색 작업을 하는 경우 일반적인 SCUBA 잠수 장비를 사용하지 않는다. 탐색 환경이 매우 위험하기 때문에 잠수사와 지속적으로 통신이 가능하고 호흡 공기를 제한 없이 지속적으로 공급할 수 있는 SSSD(Surface Supplied Diving System) 잠수 방법을 사용하게 되는데 잠수 장비 형상은 Fig. 11과 같다 (Ryan and Michael, 2018).

깊은 수심에서 오랜 시간 잠수를 하게 되면 몸속에 고압의 기체가 용해된다. 이러한 기체를 완전히 배출하지 않고 수면(surface)으로 상승하게 되면 압력이 낮아짐에 따라 혈액 속에 녹아 있던 기체들이 다시 기포로 바뀌면서 몸의 주요 혈관을 막게 된다. 이 경우, 해당 조직이나 장기가 산소와 영양분을 공급받지 못하게 되어 괴사가 발생하는데 이것이 감압병이다.

감압병 발생을 예방하기 위해서 잠수사는 해당 수심에서 잠수를 마치게 되면 감압이라는 절차를 진행하여 몸속에 녹아 있는 기체를 안전하게 배출하게 된다. 감압 시 사용하는 규칙은 다양한 방법이 있으나 본 논문에서는 한국 해군이 사용하고 있는 미 해군 감압표(US Navy decompression table)를 적용하였다.

Fig. 12에는 수중 작업 후 감압 절차를 시간에 따른 그래프로 표현하였다. 미 해군 감압표에서는 감압 시 수중 감압(inwater decompression)을 실시하고 해면으로 상승한 이후에 다시 감압 챔버로 이동하여 감압(chamber decompression)을 실시하는 방법이다. 이러한 방법을 사용하는 이유는 수중에서 감압하는 것보다 감압 챔버에서 감압을 실시하는 것이 더 빠르게 몸속에 녹아 있는 고압의 기체를 배출할 수 있도록 도와주고, 잠수사가 감압병이 발생한 경우에도 치료 및 처치가 수중보다 챔버 안이 훨씬 용이하기 때문이다.

일정 부분 수중에서 감압을 실시한 이후에 해면으로 상승하는 것은 고압의 기체를 일부 배출한 이후에 수면으로 상승해야 챔버 안으로 들어가기 전까지 잠수병 발병 확률이 줄어들기 때문이다. 잠수사가 해면으로 상승한 이후에 챔버 안으로 들어갈 때까지가 가장 위험한 순간으로 이 시간을 전환시간(transition time)이라고 부르고 5분 이내 실시하는 것을 원칙으로 한다.

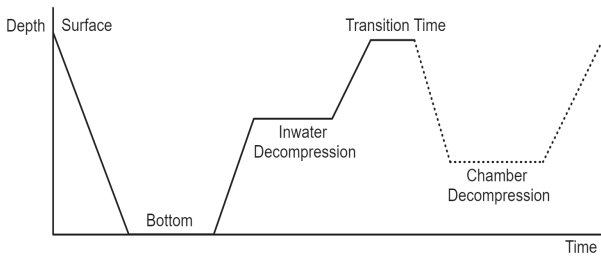


Fig. 12 Decompression procedure (Ryan and Michael, 2018)

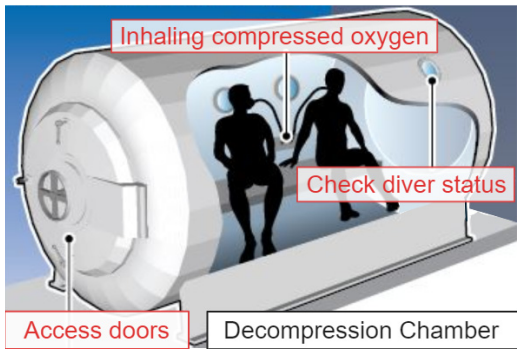


Fig. 13 Decompression Chamber

감압 챔버는 내부를 물이 아닌 공기를 넣어 압축하는 방법으로 신체를 압력에 노출시킨다. 수중에서 20m에 위치하는 것은 챔버 내부에 압축 공기를 넣어 내부 압력을 3bar로 맞추는 것과 같은 것으로, 이렇게 압축 공기를 이용하여 챔버 내부를 잠수사가 감압해야 하는 수중과 같은 압력으로 만든 뒤 천천히 챔버 내부 압력을 줄여 나가면서 신체에 녹아 있는 고압 기체를 배출시키는 것이 챔버 감압의 원리이다. 챔버의 형상은 Fig. 13과 같다.

격실별로 크기가 다르기 때문에 탐색을 완료하는데 걸리는 시간이 다르고 또한 격실별 수심이 다르기 때문에 격실별로 적용하는 감압표가 다르게 된다. 이를 모두 정리하여 탐색 격실의 수심에 따른 잠수 계획은 Table 2와 같다.

Table 2 Decompression table of M/V Sewol Diving

Depth		Bottom time (min)	Decompression time (min)	
(ft)	(m)		In water	Chamber
80	24.4	60, 90	0	30, 45
90	27.4	60	0	30
100	30.5	60, 90	0, 2	45, 75
110	33.5	60	0	60
120	36.6	60, 90	0, 33	60, 105
130	39.6	60, 90	12, 47	75, 120
140	42.7	60	25	90
150	45.7	60	37	90

4.2 수중탐색 시뮬레이션 원자모델 (Atomic model)

세월호 격실 탐색 시뮬레이션을 위한 모델 구성은 다음과 같다. 원자모델로 generator, queue, diving site, transition, chamber, transducer가 있으며 시뮬레이션에 필요한 입력 정보와 이벤트를 통해 생성된 정보를 저장하는 diving info로 구성된다.

Diving info는 이벤트 발생시 생성되는 변수값을 저장하고 최종적으로 시뮬레이션이 종료될 때 저장한 변수값을 모두 출력한다. Diving info가 저장하는 변수의 종류는 Table 3와 같다.

Generator는 146개의 격실에 대한 잠수 계획을 만들어 낸다. 격실 크기에 따라 2~3번 탐색하는 격실도 있어 총 161번의 잠수 계획이 만들어진다.

잠수 계획 수립 시 최대 작업 수심 및 해저 체류 시간에 따라 감압표가 결정되는데 본 연구에서는 해저 체류 시간을 기준값에서 최대 10분까지 임의로 조정하여 생성하였다. 이 기준에 의하여 generator에서는 각 격실에 대한 잠수 순서를 결정한다. Generator 원자 모델의 구성은 Fig. 14과 같다.

Queue에서는 생성된 161개의 잠수 계획을 주어진 조건에 따라 diving site로 내보내는 역할을 한다. Queue에서 diving site로 출력값을 내보내는 경우 현재 사용 가능한 diving site와 chamber의 개수를 확인해서 내보내게 된다. 이러한 조건을 만족하는지 확인하고 diving info를 전달하는 것이 queue의 역할이다. Queue 원자 모델의 구성은 Fig. 15과 같다.

Table 3 Diving Info variables

Type	Variable name
Search sector	Sector ID
Diving time	Diving site ID
	Diving time start, End
	Bottom time
	In water decompression time
Transition time	Transition time start, end
Chamber time	Chamber ID
	Chamber time start, end
Total	Diving time + Transition time + Chamber time

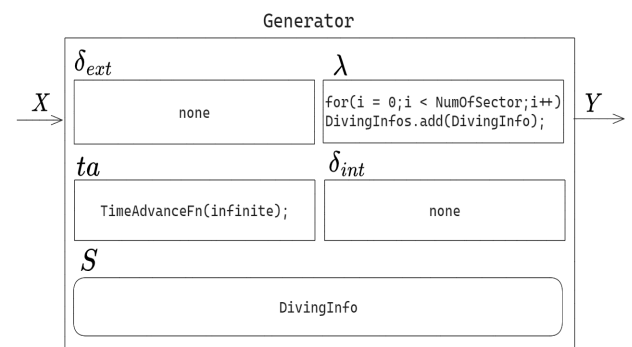


Fig. 14 Generator atomic model

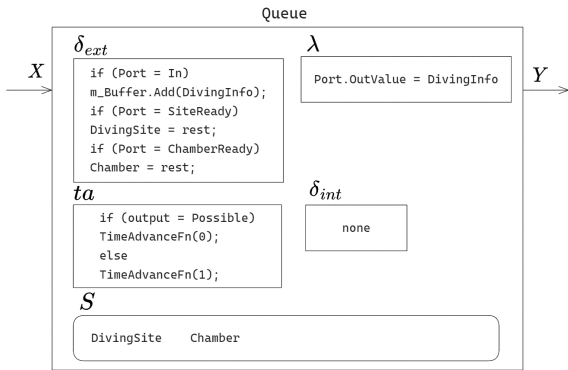


Fig. 15 Queue atomic model

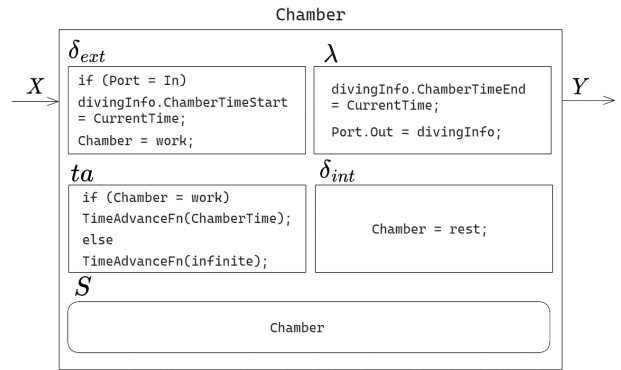


Fig. 18 Chamber atomic model

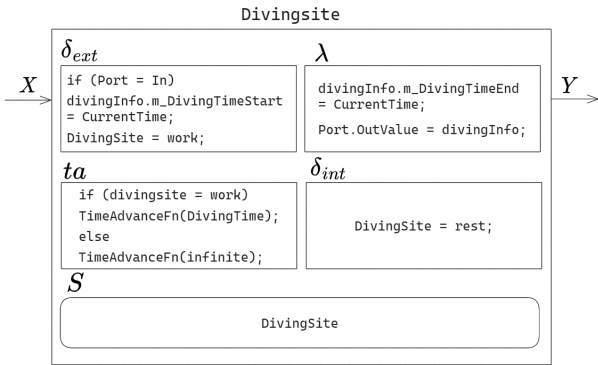


Fig. 16 Diving site atomic model

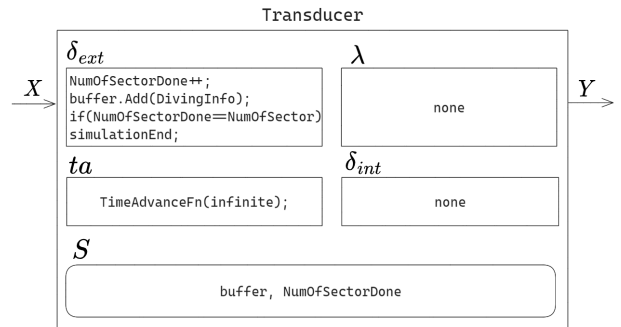


Fig. 19 Transducer atomic model

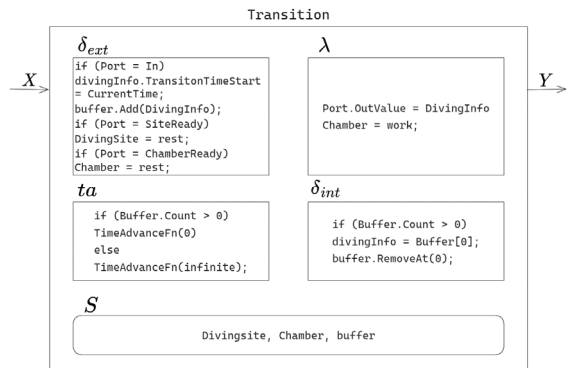


Fig. 17 Transition atomic model

Diving site에서는 diving info에 저장된 diving time 시간이 지나면 출력값을 transition으로 전달한다. Diving site 원자 모델의 구성은 Fig. 16과 같다.

Transition에서는 잠수사가 잠수를 종료하고 해수면까지 상승한 뒤 복장을 해제하고 챔버에 들어가는 과정을 의미한다. 총제한 시간은 5분으로 주어진다. 5분을 초과하는 경우 감압병 발생 가능성이 커져 별도의 치료표를 적용하여 감압 챔버를 사용하게 된다. 본 시뮬레이션에서는 모든 잠수사가 정상적으로 5분 이내 장비를 해제하고 챔버로 들어가 감압을 실시한 것으로 가정하였다. Transition 원자 모델의 구성은 Fig. 17과 같다.

Chamber에서는 Diving Info에 저장된 chamber time 시간이 지나면 출력값을 최종 단계인 transducer로 전달한다. Chamber 원자 모델의 구성은 Fig. 18과 같다.

Transducer는 잠수 작업이 종료되었음을 나타내는 원자모델로 입력된 diving info를 저장한다. 최종적으로 마지막 잠수사가 Chamber 모델에서 transducer로 넘어오면 시뮬레이션을 종료하는 코드를 실행한다. 그리고 모든 시뮬레이션을 종료할 때 diving info에 저장된 정보를 csv 파일로 저장하여 출력한다. Transducer 원자 모델의 구성은 Fig. 19와 같다.

4.3 수중탐색 시뮬레이션 결합모델 (Coupled Model)

결합모델 구성은 다음과 같다. EF(Experiment Frame)는 DEVS 기반 시뮬레이션에서 필요한 요소들인 generator와 transducer로 구성되어 있다.

Diving procedure는 수중 격실 탐색을 실시하고 미 해군 잠수 매뉴얼 기준 감압을 실시하는 절차를 구현한다. Diving site와 chamber는 개수를 달리하여 연결할 수 있으며, 그에 따라 시뮬레이션 결과가 바뀌게 된다. 결합모델 내에서 queue 모델은 chamber와 transition 모델에서 각각 현재 diving site가 완료되었는지, 어느 챔버가 사용 중인지의 정보를 받는다. 이렇게 받은 정보를 사용하여 diving site로 잠수사를 보내게 되는데 규칙은 다음과 같다. 현재 diving site에 있는 잠수사와 chamber에 들어가 있는 잠수사의 총합이 chamber 개수를 넘지 않아야 한다. 이는 안전을 위한 조치로 현재 잠수 중인 잠수사가 문제가 발생할 경우 해당 잠수사는 잠수를 바로 종료하고 수면으로 상승하여 챔버 치료를 받아야 한다. 따라서 잠수 중인 잠수사가

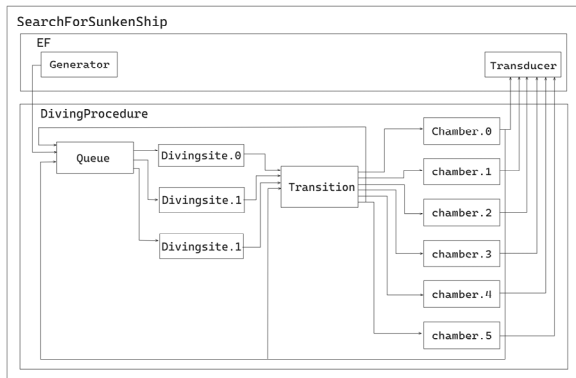


Fig. 20 Configuration of coupled Model

들어갈 수 있는 여분의 챔버는 항상 존재해야 한다.

SearchForSunkenShip은 전체 시뮬레이션 모델 중 가장 큰 단위로 EF와 diving procedure를 연결한다. 결합 모델의 구성은 Fig. 20과 같다.

5. 시뮬레이션 결과 분석

본 논문에서는 선박 수중탐색을 계획하는 단계에서 2가지 사항을 검토하고자 한다. 첫째는 몇 개의 chamber를 사용하는 것이 효과적인지와 두 번째는 격실을 탐색하는 순서를 결정하는 것이다. 이를 위해 총 5가지 케이스를 선정하였다. Diving site는 실제 세월호 상황과 똑같이 세 곳에서 실시하는 것으로 하였고 chamber의 개수를 4~8개까지 증가시키면서 시뮬레이션을 실시하였다.

3개의 diving site에서 잠수사를 투입할 때 격실 탐색의 순서는 소요 시간 기준으로 모두 임의로 투입하는 경우(random-random-random), 가장 오래 걸리는 순서로 투입하는 경우(max-max-max), 마지막으로 가장 시간이 적게 걸리는 순서로 투입하는 경우(min-min-min)의 3가지 방법을 검토하였다.

시뮬레이션 결과를 분석하면 Fig. 21과 같다. 가장 시간이 적게 걸린 방법은 챔버를 7개 사용하고 diving site에서 잠수사를 max-max-max로 투입하는 경우이다. 이 경우 4,465분의 다이빙 시간이 소요되었다. 하지만 잠수 결과를 비교한 그래프를 보면 챔버를 6개 사용하는 순간부터 총시간에 큰 차이가 없는 것을 볼 수 있다.

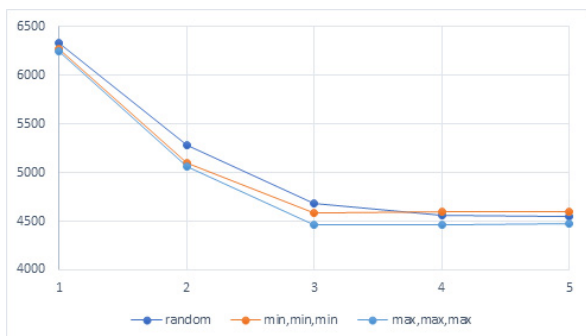


Fig. 21 Simulation optimization results

Table 4 Simulation results by case

Case	Search method	Total time (min)	STBY count	Average STBY time
case1 (divingsite 3 chamber 4)	random	6334	128	33.56
	min,min,min	6278	146	31.15
	max,max,max	6245	139	33.04
case2 (divingsite 3 chamber 5)	random	5287	90	19.13
	min,min,min	5095	119	12.90
	max,max,max	5060	90	17.93
case3 (divingsite 3 chamber 6)	random	4685	34	12.06
	min,min,min	4581	4	1.00
	max,max,max	4468	8	2.88
case4 (divingsite 3 chamber 7)	random	4560	4	7.75
	min,min,min	4597	0	0
	max,max,max	4465	1	1.00
case5 (divingsite 3 chamber 8)	random	4545	0	0
	min,min,min	4592	0	0
	max,max,max	4471	0	0

이것의 원인은 Table 4에서 확인할 수 있는데, 챔버를 6개 사용하면서부터 잠수사가 diving site에 들어가지 않고 대기하는 횟수를 나타내는 STBY(stand-By) count가 급격히 감소한다. 잠수사의 수중 투입 조건 때문에 탐색을 시작하지 못하고 대기하는 경우가 발생하게 되는데, 이러한 STBY count의 횟수를 최소화해야 총 격실 탐색 시간을 최소화할 수 있다. 시뮬레이션 결과 chamber를 6개 사용할 때부터 그 조건을 만족시키기 때문에 그 이상의 챔버를 사용하는 것은 비용 대비 효과가 적은 것을 확인하였다.

또한 투입 방법 측면에서도 임의의 순서로 잠수사가 투입되는 것보다는 최소 시간 또는 최대 시간 순서로 잠수사를 투입하는 것이 더 빠르게 탐색을 끝낼 수 있음을 확인하였다. 최대 시간 순서로 투입하는 경우가 탐색 시간이 가장 짧았는데, 이것은 탐색 초반에 가장 오랜 시간이 걸리는 격실 탐색을 3개의 diving site에서 동시에 완료함으로써 격실 탐색에 소요되는 총 탐색 시간을 줄일 수 있었기 때문이라고 판단된다.

6. 결론

본 연구는 선박 침몰 사건 이후 해저에 위치한 선체의 격실 탐색 작업 시간을 예측하는 데 DEVS를 활용하여 정량적인 접근을 시도하였다. 이는 과거에 주로 경험적 또는 정성적 방법에 의존하던 탐색 계획 수립 방법을 정량적 데이터 확인을 통하여 탐색 작업의 시간, 필요한 자원의 양, 작업 효율성을 개선할 수 있는 방법을 제시하였다. 본 논문의 예제로 적용한 세월호 격실 탐색 시뮬레이션에서 감압 챔버의 개수가 하나씩 늘어남에 따라 6,200여 초에서 5,000여 초, 4,400여 초로 탐색 시간이 감소하였다. 하지만, 특정 개수(6개) 이상부터는 탐색 시간이 약 4,465 ~ 4,471분으로 탐색 시간 감소에 영향을 주지 못하는 것을 확인하였다. 그리고 격실 탐색을 임의(random)의 순서로 탐

색하는 것보다는 가장 오래 걸리는 순서로 탐색하는 것이 가장 작은 시간이 소요되었다. 결론적으로 본 연구에서는 탐색 계획 수립 시 감압 챔버의 최적 수량을 결정하고 총 탐색 시간이 최소가 되는 격실 탐색 방법을 제시하였다.

향후 연구 계획으로 본 연구의 시뮬레이션에서 조류의 변화, 격실 탐색 시 난이도 설정 등 시뮬레이션 상황을 고도화하여 실제 수중 탐색 작업에서 고려해야 할 다양한 요소들을 포함한 보다 심층적인 시뮬레이션 연구를 진행할 예정이다.

후 기

이 논문은 2024학년도 창원대학교 연구년교수 연구비에 의하여 수행된 연구입니다.

References

Ahn, M.S., Park, S.B., and Kim, T.G., 1994. DEVSim++ : A semantics based environment for object-oriented modeling of discrete event systems. *Journal of Computing Science and Engineering*, 21(9), pp.1652-1664.

Bang, K.W., 2006. *Combined discrete event and discrete time simulation framework for shipbuilding process planning*. Master Thesis, Seoul National University.

Bang, K.W., Cha, J.H. and Lee, K.Y., 2007. Proposal of a combined discrete event and discrete time simulation framework and its application to the block erection process in shipbuilding. In: *Proceedings of the Society of CAD/CAM Engineers Conference*, Pyungchang, South Korea, 1-2 February 2007, pp.297-305.

Cha, J.H. and Roh, M.I., 2010. Combined discrete event and discrete time simulation framework and its application to the block erection process in shipbuilding. *Advances in Engineering Software*, 41(4), pp.656-665.

Cho, C.Y., Cho, M.Y. and Shin, Y.S., 2012. Prediction model for hoisting times of tower crane using discrete-eventsimulation in high-rise building construction. *Journal of the Architectural Institute of Korea Structure and Construction*, 28(1), pp.151-158.

Choi, C.B., Jung, J.H., Lyoo, C.H. and Kim, E.Y., 2020. Modeling & simulation environment for solving waste problems of the local community using discrete event system formalism. *Journal of the Korea Society for Simulation*, 29(1), pp.71-79.

Chung, Y.H., Shin, K.T. and Park, J.W., 2008. A study on mine artillery hit using DEVS. *Journal of the Korea Society for Simulation*, 17(3), pp.45-51.

Hong, M.Y., 2018. *A study on the optimal production planning of rebar manufacturing works in plant using discrete event simulation*. Master Thesis, Seoul: Ewha Woman's University.

Kim, C.J., Kim, Y.M., Kwon, Y.B. and Jang, B.S., 2018. *Sewol investigation commission general report*. Sewol Investigation Committee.

Ryan, W. and Michael, D., 2018. U.S. *Navy diving manual rev 7*. Naval sea systems command.

Son, M.J., Cha, J.H., Kim, T.W., Lee, K.Y. and Nah, Y.I., 2010. The DEVS-based detailed implementation method of the command and fire control system for the underwater vehicle DEVS-HLA simulation in the engagement level. *Journal of the Society of Naval Architects of Korea*, 47(4), pp.628-645.



이 주 필



함 승 호