

DEVS 기반 모델링을 적용한 잠수함의 어뢰회피 성능 분석 시뮬레이션[†]

강정호^{*}, 이성준^{**}, 차주환^{**}, 유성진^{**}, 이효광^{***}, 이규열^{****}, 김태완^{****}, 고용석^{*****}

A simulation for the analysis of the evasive capability of submarine against a torpedo using DEVS modeling

Jung-Ho Kang, Sung-Jun Lee, Ju-Hwan Cha, Seong-Jin Yoo, Hyo-Kwang Lee,
Kyu-Yeul Lee, Tae-Wan Kim, Yong-Seog Ko

Abstract

A simulation for the analysis of the evasive capability of a conventional costal submarine against a light Anti-Submarine Warfare (ASW) torpedo has been studied. The Torpedo, Submarine Controller, Devoy, and Jammer models of this simulation are analysed and designed using Unified Modeling Language (UML) and in addition they are modeled Discrete Event System Specification (DEVS). We examine maximum speed, acceleration, countermeasure systems capabilities of a submarine, and sonar range of a torpedo as the factors which affect the evasive capability of the submarine. This paper shows the relationships between those various factors and the submarine's evasive capability as the outcome of the simulation. The simulation models can be applied for simulation based acquisition (SBA) of a submarine system.

Key Words: Simulation Based Acquisition(SBA), Discrete Event Simulation, DEVS, Submarine, Evasive Capacity, Torpedo, Unified Modeling Language(UML)

† 본 연구는 수중운동체 특화연구센터 SM-11 수중운동체의 체계/부체계 기능 및 성능 시뮬레이션을 위한 네트워크 기반의 가상 복합운동체 모델구조 연구과제의 일부로 수행 되었음.

* 서울대학교 대학원 협동과정 계산과학전공

** 서울대학교 대학원 조선해양공학과

*** 한국과학기술원 대학원 기계공학과

**** 서울대학교 대학원 조선해양공학과 및 해양시스템공학 연구소

***** 국방과학연구소 해양기술부

1. 서론

1.1 연구배경

최근 선진국에서는 잠수함 등과 같은 수중운동체 개발에 소요되는 기간, 예산 및 개발 위험성 등을 감소시키기 위해 모델링 및 시뮬레이션 기법을 적용하고 있다. 현재 서울대학교에서도 수중 운동체의 체계 / 부체계 기능 및 성능 시뮬레이션을 위한 네트워크 기반의 가상 복합 운동체 모델구조를 연구하고 있다. 각기 다른 시스템 및 다른 개발자에 의해 개발된 시뮬레이션 모델을 네트워크상에서 통합 운용할 수 있는 시뮬레이션 모델 구조를 연구하는 것이다.

이를 위해서는 객체 지향적으로 시뮬레이션 모델을 개발하고 이를 운용할 수 있는 프레임워크를 세우는 것이 중요하다. 이에 대한 기초 연구로서 모델링 및 시뮬레이션 개발 지원을 위한 여러 도구를 분석하고 있으며, 이를 토대로 수중 운동체 모델링 방안을 구체화 시켜가고 있다. 모델 체계 구성의 기반을 마련하기 위해 모델링 방법에서 DEVS(Discrete Event System Specification)[9] 형식론 및 이를 이용한 시뮬레이션 엔진인 DEVSim++[5]을 연구하고 있다.

본 논문에서는 현재 연구하고 있는 모델 구조를 검증하기 위한 프로토타입으로서 잠수함 어뢰 회피 시뮬레이션을 UML기반으로 분석, 설계 단계에서 부터 DEVS를 적용하여 모델링 및 시뮬레이션을 구현한 내용으로 구성되어 있다.

1.2 본 논문의 연구 내용 개괄

본 논문에서 개발한 잠수함 어뢰 회피 시뮬레이션은 재래식 연안 잠수함이 최신 대잠 경어뢰의 공격을 받을 때, 잠수함의 추진 체계 성능 및 어뢰 대항 체계 성능, 그리고 어뢰의 탐지 체계 성능에 따라 어떠한 결과가 나오는지에 대한 논의를 하고자 하며, 이는 잠수함의

시뮬레이션 기반 획득(SBA)에 관련된 타당한 근거를 제시할 수 있는 방법의 하나로 활용될 수 있을 것이다. 이에 대한 어뢰, 잠수함의 주요 제원 및 시나리오는 관련 논문[2] 및 공개자료[6]를 참고하였다.

한편 객체지향 분석, 설계를 위해 UML을 사용하였으며, Discrete Event 기반의 시뮬레이션을 수행하기 위해 DEVS 형식론에 입각한 DEVSim++을 이용하였다. 본 논문에서는 어뢰 회피 시뮬레이션의 개발 과정 및 이에 대한 결과를 언급하고자 한다.

1.3 본 논문의 구성

2장에서는 효과도 측정 항목 지수(Measure of Effectiveness MOE)와 시나리오 및 주요 자원 등 시뮬레이션의 개요에 대해 언급하였고, 3장에서는 UML을 이용한 시뮬레이션 분석 및 설계를 보여주고, 4장에서는 DEVS 형식론을 토대로 하여 객체 지향적으로 각 체계 모델을 개발하는 내용에 대해 소개하였다. 5장에서는 시뮬레이션 결과를 분석하였으며, 6장에서는 결론 및 향후 연구 계획에 대해 언급하고자 한다.

2. 시뮬레이션 개요

2.1 효과도 측정 항목 지수(MOE)

본 논문에서는 재래식 연안 잠수함이 최신 대잠 경어뢰의 공격을 받을 때 잠수함의 추진 체계 성능 및 어뢰 대항 체계 성능, 어뢰의 탐지 체계 성능에 따라 어떠한 생존 확률을 가지는지에 대해 분석하였다. 본 논문에서 대상으로 삼은 잠수함은 어뢰를 회피하기만 하며 한 개의 디코이¹와 한 쌍(4개)의 재머²로 이루어

1 디코이(decoy): 잠수함과 같은 음향 신호를 내는 장치로서 어뢰로 하여금 잠수함을 혼돈하게 하는 자체 추진장치
2 재머(jammer): 일정 지속 시간동안 소음 클러스터를 발생하여 잠수함의 음향 신호를 차단하는 효과를 내는 장치

진 어뢰 대항 체계를 탑재하고 있다. 관련 논문[2]은 최대 속도만을 효과도 측정 지수로 설정하였으나, 본 논문에서는 이를 확장하여 다각도에서 바라보고 있다.

2.1.1 잠수함의 추진 체계 성능

잠수함의 추진 체계 성능에는 여러 가지가 있겠지만, 본 논문에서는 잠수함의 최대 속도 및 가속도에 대해서 살펴 보고자 한다. 잠수함의 최대 속력을 12 kts에서부터 24 kts까지 변화시켜 가면서 생존 확률을 계산한다. 또한 잠수함의 가속도를 0.04g에서부터 0.06g까지 변화시켜 가면서 생존 확률을 계산한다.

2.1.2 잠수함의 어뢰 대항 체계 성능

본 논문에서 대상으로 삼고 있는 잠수함은 어뢰 대항 체계로서 디코이 한 개와 재머 한 쌍(4개)을 탑재하고 있다. 디코이의 속력을 15 kts에서부터 19kts로 변화시켜 가면서 생존 확률을 계산하고, 재머의 음향 신호 교란 지속 시간은 20초에서부터 40초까지 변화시켜 가면서 생존 확률을 계산한다.

2.1.3 어뢰의 탐지 체계 성능

어떠한 성능의 어뢰가 공격해 오는지에 따라서 잠수함의 생존 확률은 큰 차이를 보일 수 있다. 본 논문에서는 어뢰의 탐지 체계 성능 중 능동 소나³의 탐지 거리를 1,400m에서부터 1,600m까지 변화시켜 가면서 생존 확률을 계산한다.

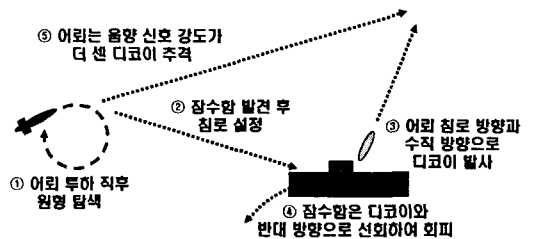
2.2 시나리오 구성

본 논문에서의 잠수함이 어뢰를 회피하는 시나리오는 다음과 같이 어뢰의 초기 원형 탐색, 디코이 충돌 이후 재탐색, 잠수함 최종 추격의 세 단계로 구성이 되어 있으며 관련 논문[2]을 참고한다.

2.2.1 어뢰의 초기 원형 탐색

우선 어뢰가 잠수함의 주위에 투하가 된다. 물론 잠수함으로부터 어뢰의 탐지 반경보다 먼 곳에 투하가 된다면, 어뢰는 처음부터 잠수함을 탐지하지 못 하게 될 것이다. 이는 우리가 원하는 결과가 아니기 때문에 어뢰는 잠수함으로부터 자신의 탐지 반경 내인 1,400m에 투하를 시키도록 한다. 어뢰는 투하된 직후 능동 소나를 이용하여 원형으로 돌면서 탐색을 시작한다(<그림 1>의 ①번 설명). 잠수함을 탐지하면 그 방향을 향해 돌진하게 된다(<그림 1>의 ②번 설명).

이 때 잠수함은 수동 소나를 가지고 있으므로 어뢰가 자신을 탐지하여 돌진해 온다는 것을 알게 되고, 약간의 판단 지연 시간 이후에 회피 기동을 수행하게 된다. 우선 디코이를 어뢰의 침로 방향과 수직 방향으로 발사(<그림 1>의 ③번 설명)하여 어뢰가 디코이를 따라가는 동안 디코이와 반대 방향으로 선회하여 회피를 시작한다(<그림 1>의 ④번 설명). 여기서 어뢰의 침로 방향과 수직 방향으로 발사하는 이유는 어뢰가 디코이와 충돌할 때까지 최대한의 시간을 벌기 위함이다. 초기에 어뢰는 잠수함과 디코이를 둘 다 탐지하지만, 음향 신호 강도가 더 센 디코이를 추격하게 된다(<그림 1>의 ⑤번 설명).



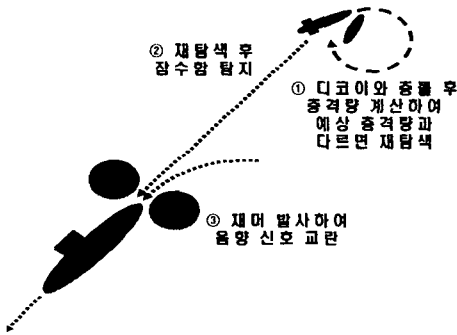
<그림 1> Torpedo Evasive Process of a Submarine 1/3 : Submarine Detection of a Torpedo and Decoy Fire of the Submarin

2.2.2 디코이 충돌 이후 재탐색

어뢰가 디코이를 추격하여 디코이와 충돌하

3 소나(sonar):수중 음향 신호를 탐지하기 위한 장치

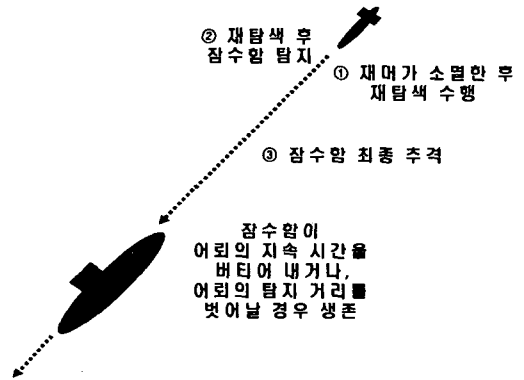
게 되면, 우선 충격량을 계산하게 된다. 충격량을 계산하여 잠수함과 충돌했을 경우의 예상 충격량과 다르다면 디코이인 것을 판별하게 되고, 원형으로 돌면서 잠수함을 재탐색하게 된다(<그림 2>의 ①번 설명). 그 동안 잠수함은 디코이의 진행 방향과 반대 방향으로 회피를 계속 하고 있었고, 다시 어뢰가 잠수함은 탐지(<그림 2>의 ②번 설명)하였을 때에도 수동 소나를 통해 그 사실을 알게 된다. 이 때 잠수함은 재머를 발사하여 교란 신호를 발생시키고, 이 때문에 어뢰는 잠수함의 음파 신호를 잃게 된다(<그림 2>의 ③번 설명). 재머의 일정 지속 시간 이후에 교란 신호가 사라지면 어뢰는 잠수함을 다시 탐색하게 된다.



<그림 2> Torpedo Evasive Process of a Submarine 2/3 : Redetection of the Torpedo after Colliding with the Decoy

2.2.3 잠수함 최종 추격

잠수함은 어뢰 대항 체계를 소진하였고, 이제 어뢰는 곧장 잠수함을 추격한다. 이때 어뢰의 배터리에 의한 지속 시간을 버티어 내거나, 어뢰의 탐지 반경 밖으로 벗어날 수 있다면 잠수함은 생존 하는 것이고, 그렇지 못 하면 잠수함은 어뢰에 의해 격침되는 것이다(<그림 2>의 ①번~③번 설명).



<그림 3> Torpedo Evasive Process of a Submarine 3/3 : The Final Submarine Chase of the Torpedo

2.3 각 대상의 주요 자원

2.3.1 대잠 경어뢰

본 논문에서 대상으로 하고 있는 어뢰는 대잠 경어뢰이다[2][4]. 전장이 2.6m, 지름이 0.32m이고, 최대 속력은 40kts이며, 가속 구간 없이 곧바로 최대 속력을 내는 것으로 가정하고 있다. 어뢰의 선회 반경은 65m이며, 배터리에 의한 지속 시간은 8분이다. 소나는 능동 소나로서 유효 탐지 거리는 1,400m에서 1,600m까지 변화시켜 가면서 시물레이션을 수행한다. 투하 직후 원형 탐색을 수행하고, 잠수함 또는 디코이가 발견되면 음향 신호 강도가 센 물체를 추격하게 된다. 충돌시 예상 충격량과 다르면 재탐색을 수행한다.

2.3.2 소형 재래식 연안 잠수함

본 논문에서는 배수량 1,000ton 급의 소형 재래식 연안 잠수함을 대상으로 하였다[2]. 선회 반경은 120m이고, 최대 속력은 12 kts에서부터 24 kts까지, 가속도는 0.04%에서부터 0.06%까지 변화시켜 가면서 시물레이션을 수행한다. 수동 소나를 탑재하고 있기 때문에 상대방의 소나 신호를 탐지하여 방향과 거리를 알아낼 수 있다.

어뢰에 의해 추격 당할 경우, 한 개의 디코

이와 한 쌍(4개)의 재머를 발사하여 추격을 지연시키며, 오로지 수동적 회피 기동만 할 뿐 대항 어뢰를 발사하는 등의 능동적인 회피 기동은 하지 않는다.

2.3.3 어뢰 대항 체계

본 논문에서 대상으로 하는 잠수함이 탑재하고 있는 어뢰 대항 체계는 한 개의 디코이와 한 쌍(4개)의 재머이다. 디코이는 자체 추진하며, 속력을 15 kts에서부터 19kts로 변화시켜가면서 시물레이션을 수행한다[2]. 어뢰 침로 방향과 수직 방향으로 발사되어 어뢰에 의해 격추될 때까지 계속 직진 운동한다. 잠수함과 똑 같은 음향 신호를 발생시켜서 어뢰로 하여금 잠수함으로 착각하도록 한다.

재머는 소음을 발생시켜 어뢰의 추격을 지연시키는 역할을 하며, 잠수함의 양현에서 2개씩 발사된다. 40m 전진 후 정지하여 소음 클러스터를 형성하게 되는데, 그 지속 시간을 20초에서 40초까지 변화시켜 가면서 시물레이션을 수행하게 된다.

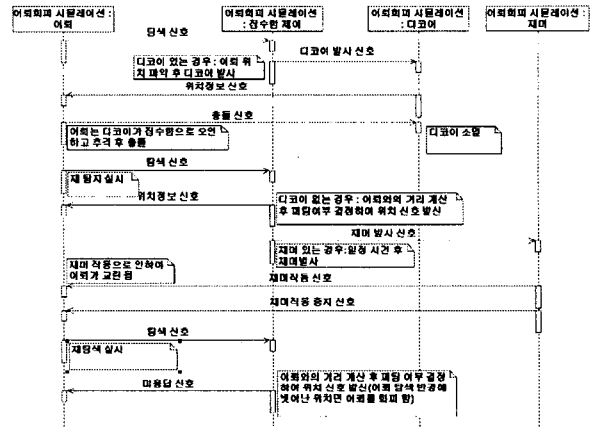
3. UML을 이용한 시물레이션 분석 및 설계

본 논문에서 구현한 시물레이션은 이산 사건 시물레이션(Discrete Event Simulation)으로, 발생하는 이벤트에 따라 각 모델의 상태(state)가 변화하고 다음 이벤트를 스케줄링 하는 방식으로 진행된다. 이에 각 시물레이션 모델들은 그 모델이 가지는 상태와 이벤트를 중심으로 설계될 필요가 있다. 따라서 본 논문에서는 각 모델의 상태도와 전체 시물레이션의 메시지 흐름을 UML을 사용하여 개념적으로 표현한다.

3.1 전체 시물레이션 Sequence Diagram

전체 시물레이션의 전반적인 분석과 설계를 위해서 UML의 Sequence Diagram을 이용하

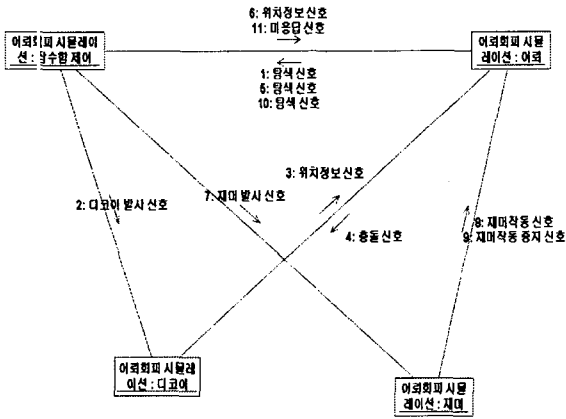
였다. Sequence Diagram은 UML의 Interaction Diagram의 한 종류로 객체와 객체간의 상호관계, 그리고 객체 간에 보낼 수 있는 Message들로 구성되어 있다. 특히 Sequence Diagram은 시간적 순서를 강조하는 시스템의 Dynamic View를 보여줌으로써 전체 시물레이션에 대한 시간적 순서에 따른 이해를 향상시킬 수 있는 Diagram이다. 이를 이용하여 전반적인 어뢰 회피 시물레이션을 분석 및 설계하는 것을 보여준다.



<그림 4> Sequence Diagram of the Submarine's evasive capability Simulation

3.2 전체 시물레이션 Collaboration Diagram

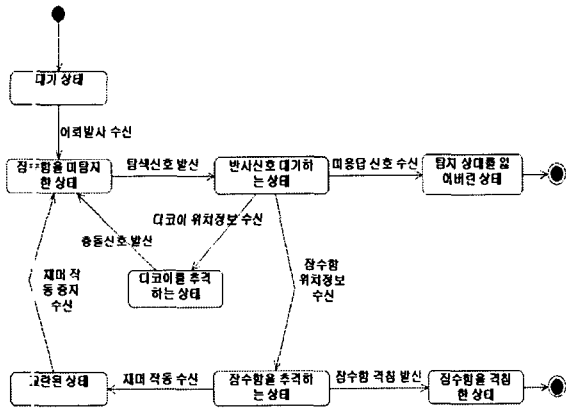
앞에서 언급한 Sequence Diagram과 같은 Interaction Diagram의 한 종류인 Collaboration Diagram을 사용하여 분석 및 설계를 시작하였다. 이는 Sequence Diagram이 시간적 순서를 강조하는 Diagram이라면 Collaboration Diagram은 객체와 객체간의 관계를 구조적 구성을 중심으로 구성하는 Diagram이다. 이를 이용하여 어뢰 회피 시물레이션의 구조적 구성 관계를 분석 및 설계하는 것을 보여준다.



<그림 5> Collaboration Diagram of the Submarine's evasive capability Simulation

3.3 어뢰 모델 Statechart Diagram

Statechart Diagram은 상태(State), 전이(Transition), 사건(Event), 활동(Activity)로 구성하여 사건에 따라 순차적으로 발생하는 객체 간의 행동에 중점을 두고 작성하는 다이어그램이다.



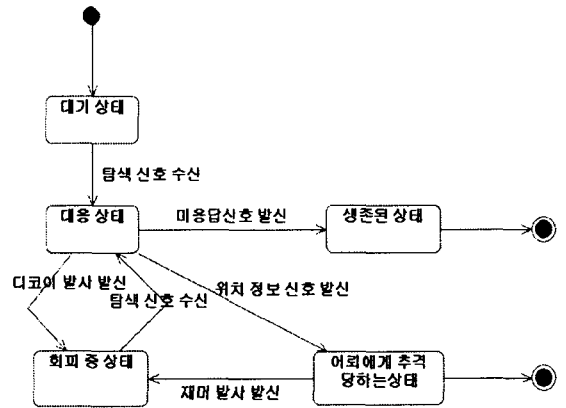
<그림 6> Statechart Diagram of the Torpedo Model

어뢰 모델은 대기 상태에서 출발하여 소나를 이용해 탐지할 경우와 못 할 경우에 대하여 잠수함을 미탐지 한 상태와 반사 신호 대기 상태를 오가게 된다. 디코이를 탐지할 경우 디코이

를 추적하는 상태가 되며, 잠수함을 탐지할 경우 잠수함을 추적하는 상태가 된다. 재머에 의해 음향 신호가 교란 되었을 경우에는 추적 지연 상태가 되고, 탐지 거리를 벗어났을 경우 탐지 상태를 잃어버린 상태, 잠수함을 공격하였을 경우 잠수함을 격침한 상태가 되며, 이때 해당되는 메시지를 주고 받게 된다.

3.4 잠수함 제어 모델 Statechart Diagram

잠수함 제어모델은 대기 상태에서 시작하다가 어뢰로부터 소나 탐지 메시지가 들어오면 대응 상태가 되어 탐지 여부를 전송하게 된다. 디코이나 재머를 발사하거나 탐지가 안 되었다는 메시지가 들어오면 어뢰 회피 상태로서 계속 회피를 수행하게 되고, 어뢰의 탐지 거리를 벗어났을 경우 생존된 상태로서 잠수함은 회피에 성공하게 된다.

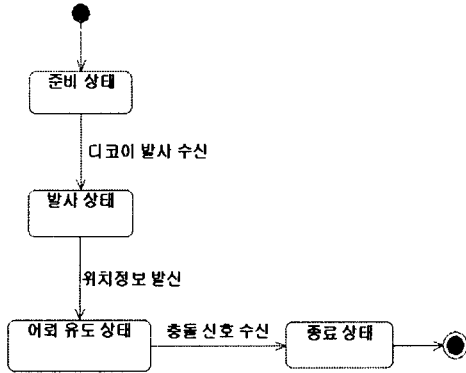


<그림 7> Statechart Diagram of the Submarine Controller Model

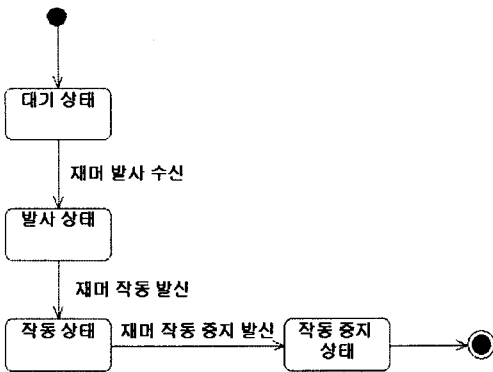
3.5 디코이, 재머 모델 Statechart Diagram

어뢰 대항 체계에는 디코이 모델과 재머 모델이 있다. 디코이 모델은 준비 상태에서 디코이를 발사하는 발사 상태, 디코이가 어뢰를 유도하는 어뢰 유도 상태, 어뢰와 충돌하여 소멸되는 종료 상태로 구성되어 있고, 재머 모델은

대기 상태에서 발사 상태인 발사 상태, 재머가 음향 신호를 교란하기 시작하는 작동 상태, 끝나는 작동 중지 상태로 구성되어 있다.



<그림 8> Statechart Diagram of the Decoy Model



<그림 9> Statechart Diagram of the Jammer Model

4. DEVS 모델링을 적용한 잠수함 어뢰회피 모델

4.1 DEVS 형식론

DEVS(Discrete Event System Specification) 형식론은 계층적이고 모듈화된 이산사건(Discrete Event) 모델을 위해 정의된 이론이다. 시스템은 시간의 흐름에 따라 입력, 출력, 상태변화(state transition) 함수들을 가진다.

DEVS 형식론은 시스템이 일반적으로 갖는 특성들을 정의하여 시스템을 모델링 할 수 있는 기반을 제공한다. DEVS 형식론에서 정의하는 모델은 Coupled 모델과 Atomic 모델의 두 가지로 분류된다.

아래는 Atomic 모델을 정의하는 항목이다.

$$M = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle$$

여기서,

X : 입력 사건의 집합

Y : 출력 사건의 집합

S : 순차적 상태의 집합

$\delta_{int} : S \rightarrow S$: 내부 전이 함수

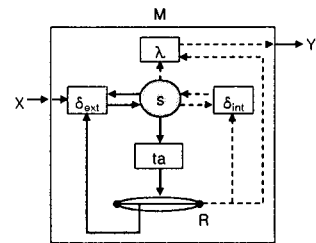
$\delta_{ext} : Q \times X \rightarrow S$: 외부 전이 함수

$\lambda : S \rightarrow Y$: 출력 함수

$ta : S \rightarrow Real$: 시간 진행 함수

단, $Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$

e : 최근의 상태 전이 이후로 흐른 시간



- Characteristic Function
- Set (State/Real Number)
- Internal transition
- External transition

<그림 10> Atomic Model Representation of the DEVS Formalism

Coupled 모델은 여러 하위 모델들로 이루어진 모델이다.

$$M = \langle X, Y, M, EIC, EOC, IC, SELECT \rangle$$

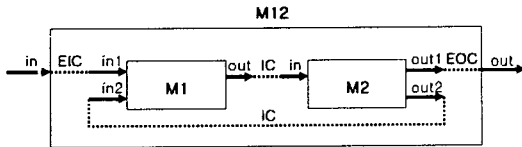
여기서

X : 입력 사건의 집합

Y : 출력 사건의 집합

M : 하위 모델 집합

EIC : 외부 입력 사건 결합 관계
 EOC : 외부 출력 사건 결합 관계
 IC : 내부 사건 결합 관계
 SELECT : 하위 모델의 우선순위



EIC : External Input Coupling relation
 EOC : External Output Coupling relation
 IC : Internal Coupling relation

<그림 11> Coupled Model Representation of the DEVS Formalism

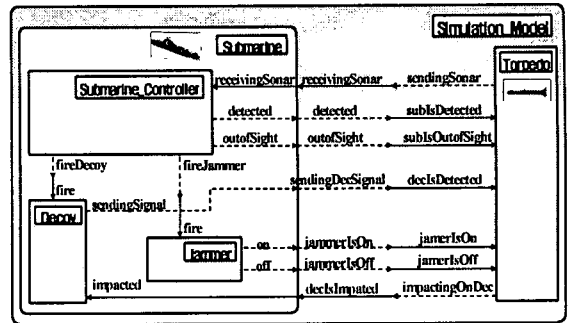
어뢰 모델은 Atomic 모델 하나로 이루어져 있고, 잠수함 모델은 두 개의 Atomic 모델이 통합된 Coupled 모델로 이루어져 있으며, 최종적으로 어뢰 모델과 잠수함 모델이 통합되어 Coupled 모델을 이루어 시뮬레이션을 수행한다.

4.2 어뢰회피 시뮬레이션 전체 모델

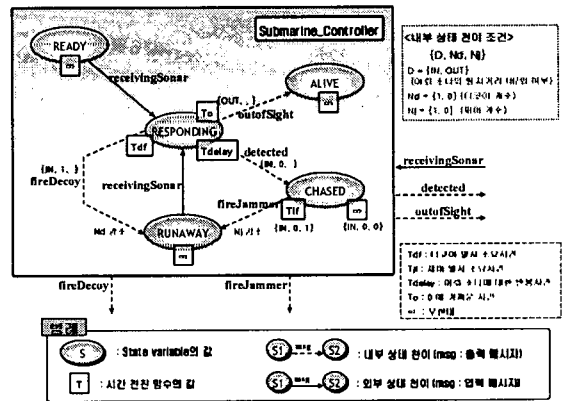
전체 시뮬레이션 모델은 크게 어뢰 모델과 잠수함 모델로 구성되어 있는 coupled 모델이다. 어뢰 모델은 atomic 모델 하나로 이루어져 있고, 잠수함 모델은 잠수함 제어 모델, 디코이 모델, 재머 모델의 세 개의 atomic 모델이 통합된 coupled 모델로 이루어져 있다.

4.3 어뢰 Atomic 모델

어뢰의 Atomic 모델은 READY 상태에서 출발하여 소나를 이용해 탐지할 경우와 못 할 경우에 대하여 DETECTING와 RECEIVING 상태를 오가게 된다. 디코이를 추격할 경우 CHASE_DEC 상태가 되며, 잠수함을 추격할 경우 CHASE_SUB 상태가 된다. 재머에 의해 음향 신호가 교란 되었을 경우에는 CONFUSED 상태가 되고, 탐지 거리를 벗어났

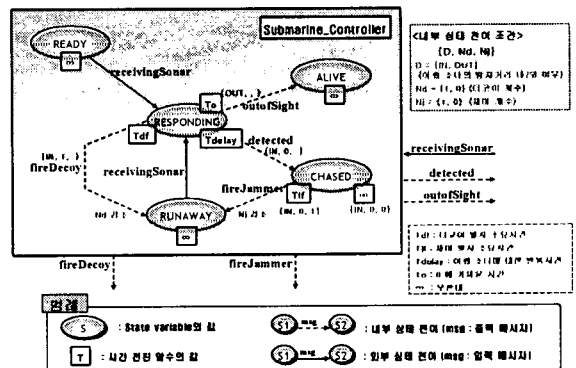


<그림 12> Message Couplings of the Whole Simulation Model and the Submarine Model



<그림 13> Atomic Model of the Submarine Controller

을 경우 LOST, 잠수함을 격침시켰을 경우 KILL 상태가 되며, 이 때 해당되는 메시지를 주고 받게 된다.



<그림 14> Atomic Model of the Submarine Controller

| |
|--|
| $Torpedo = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ |
| $X = \{sublsOutOfSight, sublsDetected, declsDetected, jammerIsOn, jammerIsOff\}$ $Y = \{sendingSonar, impactingOnDec\}$ $S = \{READY, DETECTING, RECEIVING, LOST, KILL, CHASING_DEC, CHASING_SUB, CONFUSED\}$ |
| $\delta_{ext} : Q \times X \rightarrow Q, Q = \{(s,e) s \in S \text{ and } 0 \leq e \leq ta(s)\}$ $\delta_{ext}((RECEIVING, t), declsDetected) = (CHASING_DEC)$ $\delta_{ext}((RECEIVING, t), sublsDetected) = (CHASING_SUB)$ $\delta_{ext}((RECEIVING, t), sublsOutOfSight) = (LOST)$ $\delta_{ext}((CHASING_SUB, t), jammerIsOn) = (CONFUSED)$ $\delta_{ext}((CONFUSED, t), jammerIsOff) = (DETECTING)$ |
| $\delta_{int} : Q \rightarrow Q$ $\delta_{int}(READY) = (DETECTING)$ $\delta_{int}(DETECTING) = (RECEIVING)$ $\delta_{int}(CHASING_DEC) = (DETECTING)$ $\delta_{int}(CHASING_SUB) = (KILL)$ |
| $\lambda : Q \rightarrow Y$ $\lambda(DETECTING) = sendingSonar$ $\lambda(CHASING_DEC) = impactingOnDec$ $\lambda(CHASING_SUB) = (no\ output)$ $\lambda(READY) = (no\ output)$ |
| $ta : S \rightarrow R$ $ta(Model) = \infty, Mode1 = \{RECEIVING, CONFUSED, LOST, KILL\}, ta(READY) = To$ $ta(DETECTING) = TL$ $ta(CHASING_DEC) = Tdd$ $ta(CHASING_SUB) = Tsd$ |

<그림 15> The DEVS Formalism of Torpedo Atomic Model

4.4 잠수함 Coupled 모델

잠수함 모델은 Coupled 모델로서, 잠수함의 전체 계산을 주관하는 잠수함 제어 Atomic 모델과 어뢰 대항 체계로서의 디코이 Atomic 모델, 재머 Atomic 모델로 구성되어 있다.

| |
|---|
| $Submarine = \langle X, Y, M, EIC, EOC, IC, SELECT \rangle$ |
| $EIC = (Submarine.receivingSonar, Submarine_Controller.receivingSonar), (Submarine.declsImpacted, Decoy.impacted)$ |
| $EOC = (Submarine_Controller.detected, Submarine.detected), (Submarine_Controller.outofSight, Submarine.outofSight), (Decoy.sendingSignal, Submarine.sendingDecSignal), (Jammer.on, Submarine.jammerIsOn), (Jammer.off, Submarine.jammerIsOff)$ |
| $IC = (Submarine_Controller.fireDecoy, Decoy.fire), (Submarine_Controller.fireJammer, Jammer.fire),$ |
| $SEL = (Controller \rangle Decoy \rangle Jammer)$ |

<그림 16> The DEVS Formalism of Submarine Coupled Model

잠수함 제어 Atomic 모델은 소나 탐지 정보를 받아 탐지 거리 내에 있는지 밖에 있는지를 판단하여 탐지 여부를 알려주는 역할을 한다. 그리고 상황에 따라 어뢰 대항 체계인 디코이 Atomic 모델과 재머 Atomic 모델에게 발사 명령을 내린다.

4.4.1 잠수함 제어 Atomic 모델

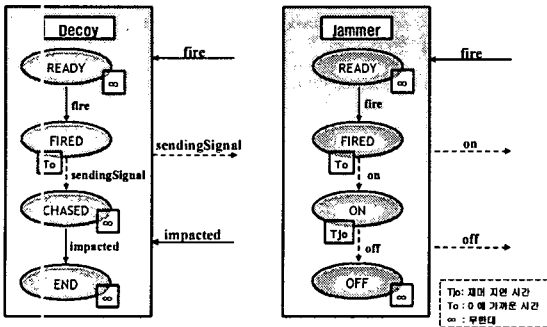
잠수함 제어 모델은 READY 상태에서 시작하다가 어뢰로부터 소나 탐지 메시지가 들어오면 RESPONDING 상태가 되어 탐지 여부를 전송하게 된다. 디코이나 재머를 발사하거나 탐지가 안 되었다는 메시지가 들어오면 RUNAWAY 상태로서 계속 회피를 수행하게 되고, 어뢰의 탐지 거리를 벗어났을 경우 ALIVE 상태로서 잠수함은 회피에 성공하게 된다.

| |
|---|
| $Submarine_Controller = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ |
| $X = \{receivingSonar\}$ $Y = \{detected, outofSight, fireDecoy, fireJammer\}$ $S = MODE \times D \times Nd \times Nj$ $MODE = \{READY, RESPONDING, RUNAWAY, ALIVE, CHASED\},,$ $Nd = \{0, 1\}, Nj = \{0, 1\}$ |
| $\delta_{ext} : Q \times X \rightarrow Q, Q = \{(s,e) s \in S \text{ and } 0 \leq e \leq ta(s)\}$ $\delta_{ext}((READY, D, Nd, Nj, t), receivingSonar) = (RESPONDING, Nd, Nj)$ $\delta_{ext}((RUNAWAY, D, Nd, Nj, t), receivingSonar) = (RESPONDING, Nd, Nj)$ |
| $\delta_{int} : Q \rightarrow Q$ $D = \{IN, OUT\}$; 어뢰의 탐지반경 내외 여부 $\delta_{int}(RESPONDING, OUT, Nd, Nj) = (ALIVE, OUT, Nd, Nj)$ $\delta_{int}(RESPONDING, IN, 1, Nj) = (RUNAWAY, IN, 0, Nj)$ $\delta_{int}(RESPONDING, IN, 0, Nj) = (CHASED, IN, 0, Nj)$ $\delta_{int}(CHASED, IN, 0, 1) = (RUNAWAY, IN, 0, 0)$ |
| $\lambda : Q \rightarrow Y$ $D = OUT (d > dc)$ $\lambda(RESPONDING, OUT, Nd, Nj) = outofSight$ $D = IN (d \leq dc)$ $\lambda(RESPONDING, IN, 1, Nj) = fireDecoy$ $\lambda(RESPONDING, IN, 0, Nj) = detected$ $\lambda(CHASED, IN, 0, 1) = fireJammer$ |
| $ta : S \rightarrow R$ $ta(Model, D, Nd, Nj) = \infty, Mode1 = \{READY, RUNAWAY, ALIVE\}$ $D = OUT ta(RESPONDING, OUT, Nd, Nj) = To$ $D = IN ta(RESPONDING, IN, 0, Nj) = Tdelay$ $ta(RESPONDING, IN, 1, Nj) = Tdf = 1-8s$ $ta(CHASED, IN, 0, 1) = Tjf = 1-8s, ta(CHASED, IN, 0, 0) = \infty$ |

<그림 17> The DEVS Formalism of the Submarine Controller Atomic Model

4.4.2 어뢰 대항 체계인 디코이 Atomic 모델과 재머 Atomic 모델

어뢰 대항 체계에는 디코이 Atomic 모델과 재머 Atomic 모델이 있다. 디코이 Atomic 모델은 READY 상태에서 발사 상태인 FIRED 상태, 어뢰가 디코이를 추격하는 CHASED 상태, 어뢰와 충돌하여 소멸되는 END 상태로 구성되어 있고, 재머 Atomic 모델은 READY 상태에서 발사 상태인 FIRED 상태, 재머가 음향 신호를 교란하기 시작하는 ON 상태, 끝나는 OFF 상태로 구성되어 있다.



<그림 18> Atomic Models of the Decoy and the Jammer

| |
|---|
| Jammer = < X, Y, S, δ_{int} , δ_{ext} , λ , ta > |
| X= {fire }, Y= {on, off } S = { READY, FIRED, ON, OFF } |
| $\delta_{ext} : Q \times X \rightarrow Q, Q = \{ (s,e) s \in S \text{ and } 0 \leq e \leq ta(s) \}$ $\delta_{ext}((READY, t), fire) = (FIRED)$ |
| $\delta_{int} : Q \rightarrow Q$ $\delta_{int}(FIRED) = (ON)$ $\delta_{int}(ON) = (OFF)$ |
| $\lambda : Q \rightarrow Y$ $\lambda(FIRED) = on$ $\lambda(ON) = off$ |
| ta : S \rightarrow R ta(READY)= ta(END) = ∞ ta(FIRED) = Tjs ta(ON) = Tjo |

<그림 19> The DEVS Formalism of the Decoy Atomic Model

| |
|---|
| Jammer = < X, Y, S, δ_{int} , δ_{ext} , λ , ta > |
| X= {fire }, Y= {on, off } S = { READY, FIRED, ON, OFF } |
| $\delta_{ext} : Q \times X \rightarrow Q, Q = \{ (s,e) s \in S \text{ and } 0 \leq e \leq ta(s) \}$ $\delta_{ext}((READY, t), fire) = (FIRED)$ |
| $\delta_{int} : Q \rightarrow Q$ $\delta_{int}(FIRED) = (ON)$ $\delta_{int}(ON) = (OFF)$ |
| $\lambda : Q \rightarrow Y$ $\lambda(FIRED) = on$ $\lambda(ON) = off$ |
| ta : S \rightarrow R ta(READY)= ta(END) = ∞ ta(FIRED) = Tjs ta(ON) = Tjo |

<그림 20> The DEVS Formalism of the Jammer Atomic Model

4.5 전체 시뮬레이션 Coupled 모델

| |
|--|
| Simulation_Model = < M, IC, SELECT > |
| M = {Submarine, Torpedo} |
| IC= (Torpedo.sendingSonar, Submarine.receivingSonar), (Submarine.detected, Torpedo.subsDetected), (Submarine.outofSight, Torpedo.subsOutofSight), (Submarine.sendingDecSignal, Torpedo.decIsdetected), (Submarine.jammerIsOn, Torpedo.jammerIsOn), (Submarine.jammerIsOff, Torpedo.jammerIsOff), (Torpedo.impactingOnDec, Submarine.decIsImpacted) |
| SEL : Torpedo > Submarine |

<그림 21> The DEVS Formalism of the Whole Simulation Coupled Model

전체 시뮬레이션 모델은 Submarine 모델과 Torpedo 모델의 Coupled 모델로 구성된다. 두 모델 간에 소나 정보를 주고 받게 되며, 이에 따라 잠수함이 탐지되는지 디코이가 탐지되는지의 여부를 판단하게 된다. 그리고 재머에 의해 영향을 받는지의 여부도 메시지로 주고 받게 되어 전체 시뮬레이션이 이루어진다

5. 시뮬레이션 결과 및 분석

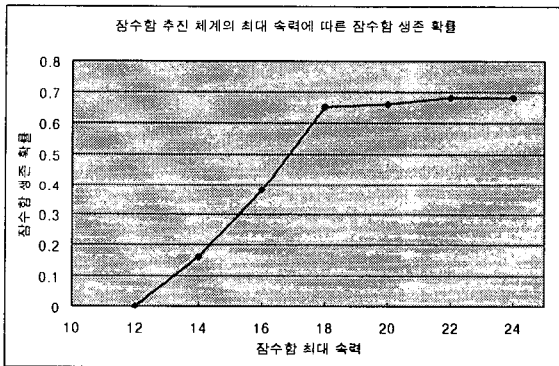
5.1 잠수함의 추진 체계 성능

우선 잠수함의 추진 체계 성능을 시뮬레이션하기 위해 잠수함의 최대 속력과 잠수함의 가

속도를 변경시켰다. 나머지 변수인 어뢰 대항 체계의 디코이 속력은 17kts, 재머의 지속 시간은 30초, 어뢰의 탐지 체계의 탐지 거리는 1,500m로 고정하고 있다

5.1.1 잠수함의 최대 속도 변경

잠수함의 가속도는 0.05%로 고정하고, 잠수함의 최대 속력을 12kts, 14kts, 16kts, 18kts, 20kts, 22kts, 24kts 로 변경시켜 가면서 시뮬레이션을 수행하였고, 다음과 같은 결과를 도출하고 있다.

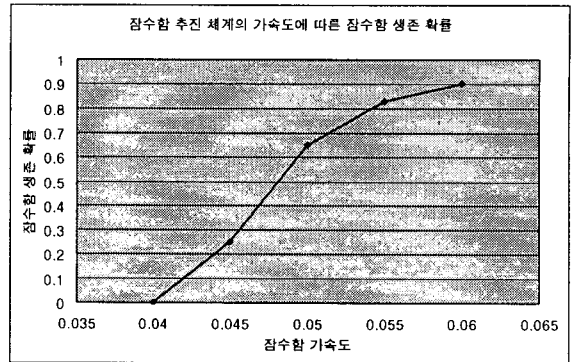


<그림 22> Relationship between Maximum Speed of the Submarine and its Probability of Survival

12kts에서는 거의 생존할 수 없고, 차츰 증가하여 18kts부터는 잠수함 생존 확률이 0.65 정도로 일정하게 유지되는 것을 볼 수 있다. 그 이유는 잠수함의 최대 속력이 커짐에 따라 가속 구간이 길어지는데, 잠수함과 어뢰가 가장 멀리 떨어져 있는 재머 소멸 이후 재탐색 종료 시점까지 가속되는 것은 의미가 없기 때문이다.

5.1.2 잠수함의 가속도 변경

잠수함의 최대 속력은 18kts로 고정하고, 잠수함의 가속도를 0.040%, 0.045%, 0.050%, 0.055%, 0.060% 로 변경시켜 가면서 시뮬레이션을 수행하였고, 다음과 같은 결과를 도출하고 있다.



<그림 23> Relationship between Acceleration of the Submarine and its Probability of Survival

가속도가 0.04% 일 경우에는 생존할 수 없고, 가속도가 커짐에 따라 생존 확률이 증가하는 것을 볼 수 있다. 비교적 적은 차이에도 생존 확률의 차이가 커지는 것으로 보아 민감도가 크다고 할 수 있다.

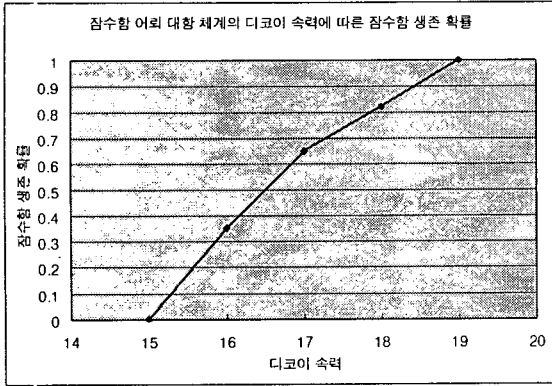
5.2 잠수함의 어뢰 대항 체계 성능

잠수함의 어뢰 대항 체계 성능을 시뮬레이션 하기 위해 디코이의 속력과 재머의 지속 시간을 변경시켰다. 나머지 변수인 잠수함 추진 체계의 최대 속력은 18kts, 가속도는 0.05%, 어뢰의 탐지 체계의 탐지 거리는 1,500m로 고정하고 있다.

5.2.1 디코이 속도 변경

재머의 지속 시간을 30초로 고정하고, 디코이의 속력을 15kts, 16kts, 17kts, 18kts, 19kts 로 변경시켜 가면서 시뮬레이션을 수행하였고, 다음과 같은 결과를 도출하였다. 디코이 속력이 15kts일 때에는 거의 생존할 수 없고, 디코이 속력이 커짐에 따라 생존 확률이 증가하는 것을 볼 수 있다. 디코이 속력이 19kts가 되었을 경우에는 거의 생존하는 것을 볼 수 있다. 잠수함과 유사성을 유지하기 위해 디코이의 속력을 잠수함의 최대 속력과 비슷하게 하면서 증가시킬 수 있다면 생존 확률을 극대화 할 수

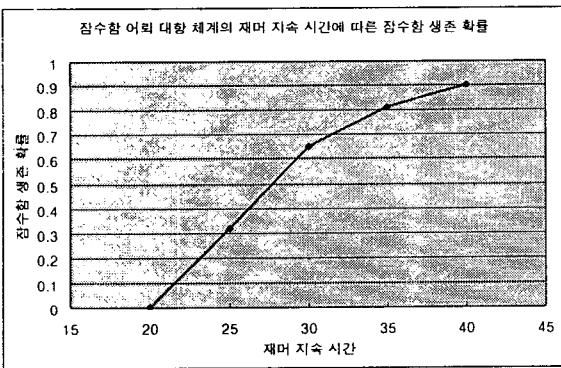
있을 것을 사료된다.



<그림 24> Relationship between Speed of the Decoy and Probability of Survival of the Submarine

5.2.2 재머 지속 시간 변경

디코이 속력을 17kts 로 고정하고, 재머의 지속 시간을 20초, 25초, 30초, 35초, 40초로 변경시켜 가면서 시뮬레이션을 수행하였고, 다음과 같은 결과를 도출하였다. 예상대로 디코이의 지속 시간이 길수록 잠수함의 생존 확률이 증가하는 것을 볼 수 있다.

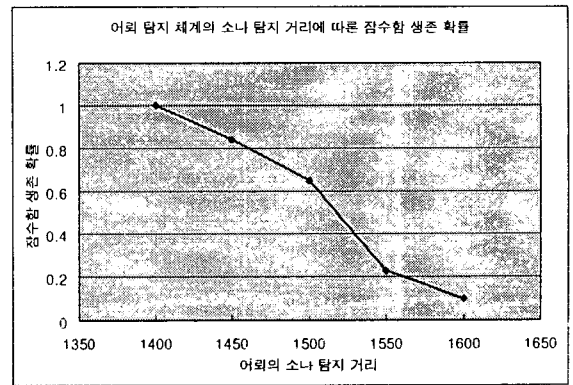


<그림 25> Relationship between Operating Time of the Jammers and Probability of Survival of the Submarine

5.3 어뢰의 탐지 체계 성능

어뢰의 탐지 체계 성능을 시뮬레이션 하기 위해 소나의 탐지 거리를 1,400m, 1,450m,

1,500m, 1,550, 1,600m로 변경시켰다. 나머지 변수인 잠수함 추진 체계의 최대 속력은 18kts, 가속도는 0.05%², 어뢰 대항 체계의 디코이 속력은 17kts, 재머의 지속 시간은 30초로 고정하고 있다.



<그림 26> Relationship between Sonar Range of the Torpedo and Probability of Survival of the Submarine

어뢰의 소나 탐지 거리가 길어질수록 잠수함의 생존 확률은 작아지는 것을 볼 수 있다. 이러한 데이터에 입각하여, 어떠한 어뢰에 대항하여 어느 정도의 생존 확률을 보장하는 잠수함을 설계할 것인가를 판단하여야 할 것이다.

6. 결론 및 향후 연구 계획

본 연구는 재래식 연안 잠수함이 최신 대잠 경어뢰의 공격을 받을 때, 잠수함의 추진 체계 성능 및 어뢰 대항 체계 성능, 그리고 어뢰의 탐지 체계 성능에 따라 어떠한 결과가 나오는지에 대해 DEVS 형식론에 기반한 객체 지향적인 시뮬레이션을 통하여 타당한 근거를 제시할 수 있는 방법을 제안하고 있고, 이는 개략적으로나마 잠수함의 시뮬레이션 기반 획득(SBA)에 있어서 자료로 활용될 수 있을 것이다. 또한 미 해대원 논문(The Relationship Between a Submarine's Maximum Speed and Its Evasive Capability)의 잠수함 최고 속도만을 변경한 연구 결과를 비교해 보면, 본 연구

와 거의 일치하였고, 본 연구가 어느 정도 자체 신뢰도가 보장됨을 알 수 있다. 앞으로 이러한 DEVS 기반 모델링의 시뮬레이션 개발 방법론을 확장하여, 각기 다른 시스템 및 다른 개발자에 의해 개발된 시뮬레이션 모델을 네트워크 상에서 통합 운용할 수 있는 시뮬레이션 모델 구조를 연구하고자 한다.

참고문헌

- [1] 손주항, 박상준, 한정안, 김형중, 이종찬, 김병기, 2004, "DEVS 모델링을 적용한 인터넷 위상 생성기", 「한국시뮬레이션 학회 논문지」, Vol. 13, No.3, pp. 43-52.
- [2] Armo, Knut Rief, "The Relationship Between a Submarine's Maximum Speed and Its Evasive Capability", Master's Thesis of Naval Postgraduate School, Monterey, California, June 2000.
- [3] Fujimoto, Richard M., Parallel and Distributed Simulation Systems, A Wiley-Interscience Publication, America, 2000.
- [4] Hu, Xiaolin, "A Simulation-based Software Development Methodology for Distributed Real-Time Systems", ACIMS, The University of Arizona, 2004.
- [5] Park, Seong Bong and Tag Con, Kim "The DEVS Formalism: Hierarchical Modular Systems Specification in C++", Proc. the 1992 European Simulation Multi-conference, 1992.
- [6] Watts, Anthony J, Jane's Underwater Warfare System, Eleventh Edition, 1999-2000, Jane's Information Group Limited, UK, 1999.
- [7] Whitten, Jeffery L, Systems Analysis and Design Methods, Sixth Edition, 1989-2004, McGraw.Hill Higher Education, America, 2004.
- [8] Zeigler, B. P., Multifaceted Modeling and Discrete Event Simulation, Academic Press, 1984.
- [9] Zeigler, B. P., Object-oriented Simulation with Hierarchical, Modular Models : Intelligent Agent and Endomorphic system, Academic Press, 1990.
- [10] Zeigler, B. P., H. Praehofer., and T. G. Kim, Theory of Modeling and Simulation, Second Edition, Academic Press, 2000.

주 작 성 자 : 강 정 호

논문 투고일 : 2005. 03. 30

논문 심사일 : 2005. 04.14(1차), 2005. 04.14(2차),
2005. 04.14(3차)

심사판정일 : 2005. 04. 14

● 저자소개 ●



강정호

2000년 육군사관학교 전자공학과 학사

2004년 - 현재 서울대학교 대학원 협동과정 계산과학전공 석사과정

관심분야 : Modeling and Simulation, Computer Graphics



이성준

2004년 서울대학교 조선해양공학과 학사

2004년 - 현재 서울대학교 대학원 조선해양공학과 석사과정

관심분야 : Modeling and Simulation, 조선 CAD



차주환

2002년 서울대학교 조선해양공학과 학사

2004년 서울대학교 대학원 조선해양공학과 석사

2004년 - 현재 서울대학교 대학원 조선해양공학과 박사과정

관심분야 : 음함수 곡면 모델링 및 가시화, 볼륨모델링,
Modeling and Simulation

유성진

2004년 서울대학교 대학원 조선해양공학과 학사

2004년 - 현재 서울대학교 대학원 조선해양공학과 석사과정

관심분야 : Implicit 모델링, Modeling and Simulation



이효광

2003년 연세대학교 기계공학과 학사

2003년 - 현재 한국과학기술원 대학원 기계공학과 석사과정

관심분야 : 가상현실 멀티채널 시스템, STEP, SEDRIS

● 저자소개 ●



이규열

1971년 서울대학교 공과대학 조선공학과 학사
 1975년 독일 하노버 공과대학 조선공학 석사(Dipl.-Ing.)
 1982년 독일 하노버 공과대학 조선공학 박사(Dr.-Ing.)
 1975년 - 1983년 독일 하노버 공과대학 선박설계 및 이론연구소, 주정부 연구원
 1983년 - 1994년 한국기계연구원 선박해양공학연구센터, 선박설계, 생산자동화 연구사업(CSDP)단장
 1994년 - 2000년 서울대학교 공과대학 조선해양공학과 부교수
 2000년 - 현재 서울대학교 공과대학 조선해양공학과 교수
 관심 분야 : 최적설계, 형상모델링, CALS



김태완

1985년 한양대학교 산업공학과 학사
 1993년 미국 Arizona State Univ., Computer Science 석사
 1996년 미국 Arizona State Univ., Computer Science 박사
 1996년 - 1999년 미국 EDS, Inc.(이전 SDRC) 소프트웨어 엔지니어
 1999년 - 2001년 서울대학교 정밀기계설계공동연구소 특별연구원
 2001년 - 2003년 세종대학교 디지털콘텐츠학과 조교수
 2003년 - 2004년 서울대학교 조선해양공학과 조교수
 2004년 - 현재 서울대학교 조선해양공학과 부교수
 관심분야 : Modeling and Simulation, NURBS 곡선과 곡면, CAGD, Computer Graphics



고용석

1985년 서울대학교 조선공학과 학사
 1985년 - 1988년 해군본부 조함단 해군감독관
 1988년 - 1990년 코리아 타코마(주) 특수선 설계과
 1996년 부산대학교 산업공학과 석사
 1990년 - 현재 국방과학연구소 함정설계팀
 관심분야 : 함정설계, 체계분석, Modeling and Simulation