



Received: 2026/02/09
Revised: 2026/02/22
Accepted: 2026/03/10
Published: 2026/03/31

***Corresponding Author:**

Seon Han Choi

(03760) 52 Ewhayeodae-gil, Seodaemun-gu, Seoul,
Republic of Korea

Tel: +82-2-3277-3824

Fax: +82-2-3277-3275

E-mail: seonhan.choi@ewha.ac.kr

해상 교전 효과도 분석을 위한 에이전트 기반 시뮬레이션 플랫폼 개발

Agent-Based Simulation Platform for Effectiveness Analysis of Maritime Engagements

김나연¹, 최선한^{2*}, 최창범³, 정의림⁴, 이정근⁵, 양민수⁶

¹이화여자대학교 전자전기공학전공 석사과정

²이화여자대학교 전자전기공학전공 교수

³국립한밭대학교 컴퓨터공학과 교수

⁴국립한밭대학교 인공지능소프트웨어학과 교수

⁵한화시스템(주) 해양연구소 선임연구원

⁶국립한밭대학교 국방우주공학과 교수

Abstract

다양한 위협과 전투 조건의 결합으로 해상 교전은 점차 복잡해지고 있으며, 이를 일관된 구조로 분석할 수 있는 시뮬레이션 모델이 요구된다. 본 논문에서는 해상 교전 효과도 분석을 위한 에이전트 기반 시뮬레이션 플랫폼을 제안한다. 교전 기능을 탐지, 결심, 제어, 기동 모듈로 분해하고 이를 독립적인 에이전트로 구현함으로써, 동일한 구조에서 내부 로직 변경만으로 다양한 교전 결과를 표현할 수 있음을 확인하였다.

As maritime engagements become increasingly complex owing to the combination of diverse threats and combat conditions, simulation models that can analyze such engagements within a consistent structure are required. This study proposes an agent-based simulation platform to analyze the effectiveness of maritime engagements. Engagement functions are decomposed into four core modules—detection, decision, control, and movement—which are implemented as independent agents. Simulation results demonstrate that diverse engagement outcomes can be represented within the same structural framework by modifying only the internal logic of the agents.

Keywords

해상 교전(Maritime Engagement),
에이전트 기반 시뮬레이션(Agent-Based Simulation),
교전 효과도 분석(Engagement Effectiveness
Analysis),
이산 사건 시뮬레이션(Discrete-Event Simulation),
모듈화 모델(Modular Model)

Acknowledgement

이 논문은 2025년도 한화시스템(주)의 재원을 지원받아 수행된 연구임.

Nayeon Kim¹, Seon Han Choi^{2*}, Changbeom Choi³, Eui Rim Jeong⁴,
Jeong Keun Lee⁵, Min Soo Yang⁶

¹M.S. student, Dept. of Electrical and Electronic Engineering, Ewha Womans University

²Professor, Dept. of Electrical and Electronic Engineering, Ewha Womans University

³Professor, Dept. of Computer Engineering, Hanbat National University

⁴Professor, Dept. of Artificial Intelligence Software, Hanbat National University

⁵Engineer, Naval R&D Center, Hanwha Systems Co., Ltd.

⁶Professor, Dept. of Defense and Space Engineering, Hanbat National University

1. 서론

해군 전투 체계는 다양한 구성 요소와 이들 간의 상호작용을 동시에 고려해야 하는 매우 복잡하고 동적인 환경에서 운용된다[1]. 이러한 복합 교전 환경에서 함정은 제한된 자원과 시간 내에 탐지, 결심, 무장 운용, 기동과 같은 일련의 대응을 수행해야 하므로, 전투 체계를 구성하는 각 부체계에 대한 체계적이고 효율적인 분석이 요구된다[2]. 특히 교전 양상이 점차 다양해짐에 따라, 기존 전투 개념을 넘어서 분석 및 평가 방법의 필요성이 증대되고 있다.

모델링 및 시뮬레이션(modeling and simulation, M&S)은 복잡한 전투체계와 전장 환경을 이해하고 분석하며 평가하기 위한 핵심적인 방법론으로서 국방 분야에서 널리 활용되어 왔다[3]. M&S는 전장 환경과 무기체계, 교전 전술을 수학적·논리적으로 표

현함으로써 실제 전력을 투입하지 않고도 교전 상황을 분석할 수 있는 효과적인 도구이다.

해군 교전 및 무기체계 분석을 위한 M&S 연구는 다양한 관점에서 수행되어 왔다[4]. 교전 시나리오를 기반으로 개별 무기체계의 성능과 효과를 정량적으로 평가하려는 연구에서는 대함유도탄을 대상으로 상태공간 기반 시뮬레이션을 통해 기만기의 효과를 분석하는 등, 특정 무기체계와 교전 조건을 중심으로 한 연구들이 수행되었다[5]. 또한 MBSE와 SysML을 활용하여 교전 과정을 요구사항, 기능, 구조 관점에서 모델링하고, 교전 절차의 타당성과 설계 일관성을 검토하려는 연구도 제안되었다[6].

한편 실제 교전 환경에서 한정된 자원을 효율적으로 활용하기 위한 연구도 진행되어 왔다. 기존 탐지체계가 고정된 성능 범위 내에서 탐지 및 추적을 수행하였다면, 최근에는 소형화·고속화된 위협에 대응하기 위해 탐지 및 무장 체계의 성능 자원을 상황에 따라 할당하는 기법이 연구되고 있다[7]. 이와 함께 단일 함정의 무장 할당, 다수 표적 교전 상황, 자기방어 능력 평가를 위한 시뮬레이션 연구도 수행되었다[8].

그러나 이러한 선행연구들은 대부분 특정 무기체계나 교전 시나리오를 전제로 모델이 구성되어, 탐지 방식이나 결심 로직, 기동 알고리즘과 같은 요소가 변경될 경우 모델의 수정이나 재구성이 요구된다는 한계를 가진다. 교전 상황을 구성하는 기능적 요소가 유사함에도 불구하고, 기능 구조와 알고리즘이 분리되지 않은 채 설계되는 경우가 많아 다양한 교전 시나리오에 범용적으로 적용하기에는 제약이 존재한다. 특히 교전급 시뮬레이션에서 다양한 함정, 무기, 의사결정 알고리즘을 유연하게 교체·비교하면서 시뮬레이션을 통해 효과성을 분석할 수 있는 통합적 구조에 대한 연구는 상대적으로 제한적이다.

이러한 한계를 극복하기 위해, 교전 상황을 이산사건 기반으로 표현하여 DEVS 형식론을 적용한 시뮬레이션 프레임워크를 제안한 연구도 있었다. 해당 연구에서는 전투 개체를 전투 객체, 논리 객체, 환경 객체 등으로 구분하고, 전투 객체를 core-shell 구조로 분리하여 교전 조건 변화에 따른 모델의 재사용성과 확장성을 높이고자 하였다[8]. 그러나 이러한 접근 역시 특정 교전 시나리오나 객체 구성을 전제로 모델이 설계되는 경우가 많아, 탐지 방식이나 결심 로직, 기동 알고리즘과 같은 교전규칙이 변경될 경우에는

여전히 모델의 수정이나 재구성이 요구되는 한계를 가진다.

이러한 한계를 해결하기 위해, 본 연구에서는 교전 시나리오가 주어졌을 때 특정 상황이나 무기체계에 종속되지 않고 다양한 해상 교전 시뮬레이션을 수행할 수 있는 agent 기반 교전 시뮬레이션 플랫폼을 제안한다. 제안하는 플랫폼은 교전 객체를 탐지, 결심, 제어, 기동의 네 가지 핵심 기능으로 분리하여 설계하였으며, 각 기능은 독립적인 모델로 구성되어 알고리즘의 선택 및 교체가 가능하다. 또한 전체 모델은 DEVS 형식론을 기반으로 설계되어 모듈화, 재사용성, 확장성을 확보하였다.

이를 통해 본 연구는 다양한 교전 시나리오에서 동일한 구조의 시뮬레이션 모델을 적용할 수 있는 기반을 제공하고, 향후 교전 분석 및 운용 개념 평가를 위한 범용적인 시뮬레이션 플랫폼으로 활용될 수 있음을 보이고자 한다.

2. 시스템 모델 설계구현

2.1 제안 시뮬레이션 모델 구조

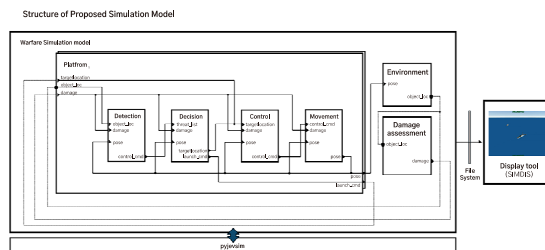


Fig. 1. Structure of proposed simulation model

Fig. 1은 본 연구에서 제안하는 해상 교전 시뮬레이션 모델의 전체 구조를 나타낸다. 제안 모델은 교전에 참여하는 다수의 플랫폼(platform)과 이들 간의 상호작용을 관리하는 environment 모델, 교전 결과를 평가하기 위한 damage assessment 모델, 그리고 시뮬레이션 결과를 가시화하기 위한 display 도구로 구성된다. 전체 시뮬레이션은 DEVS 형식론 기반 이산사건 시뮬레이션 엔진인 pyjevsim 상에서 실행된다.

각 platform은 교전 객체를 구성하는 기본 단위로, 탐지(detection), 결심(decision), 제어(control), 기동(movement)의 네 가지 기능을 수행하는 agent들로 구성된 결합 모델로 표현된다. 이들 agent는 기능

별로 분리된 DEVS 원자모델로 설계되었으며, 포트 기반 메시지 교환을 통해 상호 연동된다. 이를 통해 교전 객체 내부의 기능 구조를 명확히 분리하고, 동일한 구조 내에서 다양한 알고리즘을 적용할 수 있도록 하였다.

Detection agent는 환경으로부터 전달받은 객체 위치 정보(object_loc)와 플랫폼의 상태 정보를 바탕으로 탐지를 수행하고, 탐지 결과를 decision agent로 전달한다. Decision agent는 탐지된 객체 정보를 기반으로 위협 평가 및 교전 평가를 수행하며, 표적 위치(targetlocation) 및 발사 명령(launch_cmd)과 같은 결심 결과를 생성한다. Control agent는 결심 결과를 기동 및 제어 명령(control_cmd)으로 변환하여 movement agent로 전달하며, movement agent는 해당 명령을 바탕으로 플랫폼의 위치 및 자세(pose)를 갱신한다.

Movement agent에서 생성된 플랫폼의 상태 정보는 environment 모델로 전달되어 전장 상태를 구성하는 데 사용된다. Environment 모델은 모든 플랫폼의 상태 정보를 통합 관리하며, 이를 각 플랫폼의 detection agent에 다시 제공함으로써 교전 객체 간 상호작용이 가능하도록 한다. 또한 교전 중 발생하는 피격 및 손상 정보는 damage assessment 모델로 전달되어 교전 결과를 평가하는 데 활용된다.

시뮬레이션 실행 과정에서 기록된 플랫폼의 위치 및 상태 정보는 파일 시스템을 통해 display 도구로 전달되며, 이를 통해 교전 상황을 시간 흐름에 따라 가시적으로 표현할 수 있다. 이러한 구조를 통해 2.1절에서 설명한 시뮬레이션 모델은 교전 객체 내부의 기능 분리, 환경 기반 상호작용, 그리고 실행·분석·가시화의 연계를 하나의 통합된 프레임워크로 구현한다.

2.1.1 DEVS 원자모델 설계: Detection

Detection agent는 교전 객체의 탐지 기능을 수행하는 DEVS 원자모델로, 환경 모델로부터 전달되는 상대 객체 위치 정보(object_loc)와 플랫폼의 위치 정보를 기반으로 탐지 결과를 생성한다. Fig. 2는 detection agent의 동작에 대한 세부 모델링 내용을 자세히 설명한다.

Detection 모델은 시뮬레이션 시작 전 wait 상태로 대기한다. 시뮬레이션이 시작된 후, object_loc 입

력을 받으면 detect 상태로 천이되어 탐지 로직을 수행한다. 이후 탐지주기에 따라 탐지 데이터를 업데이트한다. 만약 감지된 물체가 탐지반경 내에 위치한 경우, 모델은 decision 모델에 threat_list를 전송한다.

또한, detection 모델은 교전 과정에서 외부로부터 damage 입력을 수신할 수 있다. Detection 모델이 wait 또는 detect 상태에 있을 때, 객체가 destroyed 상태로 판단되는 경우, detection 모델은 더 이상 탐지 로직을 수행하지 않으며, 이후 탐지 주기 갱신이나 탐지 결과 생성을 중단한다. 이 경우 detection 모델은 추가적인 object_loc 입력에 대해서도 탐지 상태로 전이하지 않고, 내부 상태를 유지한 채 시뮬레이션 종료 조건을 대기한다. 반대로 damage 입력이 발생하였으나 객체가 완전히 파괴되지 않은 경우에는, 현재 상태를 유지하면서 이후 탐지주기에 따라 기존 탐지 로직을 계속 수행한다.

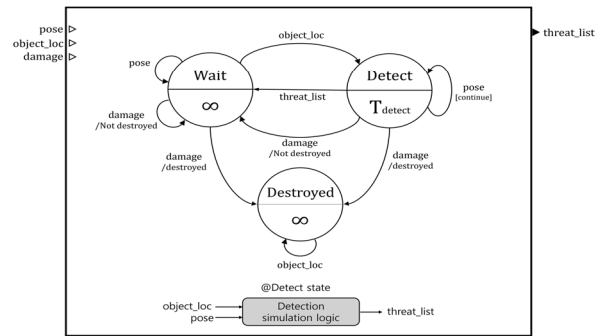


Fig. 2. Design of atomic model: Detection

2.1.2 DEVS 원자모델 설계: Decision

Decision 모델은 교전 결심 기능을 수행하는 원자 모델로, 탐지 결과를 입력으로 받아 교전 여부 및 대응 명령을 결정한다.

초기 wait 상태로 대기하던 decision 모델이 detection 모델로부터 threat_list(탐지 결과) 메시지를 받으면, decision 모델은 decide 상태로 천이되어 판단 로직을 수행한다. 이 과정에서 입력된 탐지 정보와 현재 플랫폼 상태를 기반으로 교전 수행 여부를 결심한다.

교전이 필요하다고 결심되는 경우, decision 모델은 표적 위치 정보(targetlocation)를 생성하고, 필요 여부에 따라 launch_cmd를 함께 출력한다. 생성된 결심 결과는 control 모델로 전달되어 이후 제어 명

령 생성에 사용된다.

반대로 교전 조건을 만족하지 않거나 즉각적인 대응이 필요하지 않은 경우에는, decision 모델은 별도의 출력 없이 결심을 종료하고 wait 상태로 복귀하여 다음 탐지 결과 입력을 대기한다.

이와 같은 구조를 통해 decision 모델은 탐지 결과 입력 시에만 활성화되며, 결심이 완료되면 다시 대기 상태로 복귀하는 이산사건 기반 교전 결심 모델로 동작한다.

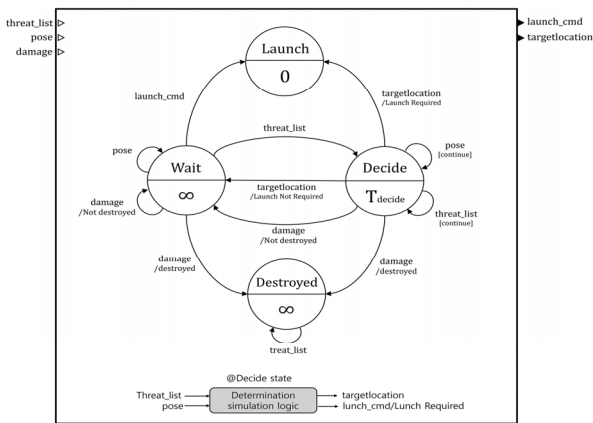


Fig. 3. Design of atomic model: Decision

2.1.3 DEVS 원자모델 설계: Control

Control 모델은 decision 모델로부터 전달된 결심 결과를 입력으로 받아, 기동에 필요한 제어 명령을 생성하는 원자모델이다.

시뮬레이션이 진행되는 동안 decision 모델로부터 targetlocation 입력이 발생하면, control 모델은 제어 상태로 천이되어 제어 로직을 수행한다. 이 과정에서 입력된 표적 위치 정보와 교전 판단 결과를 기반으로 제어 명령(control_cmd)을 계산한다.

제어 명령이 생성되면, control 모델은 해당 명령을 출력하여 movement 모델로 전달한다. 이후 control_cmd에 따라 수행이 완료될 때까지 제어 상태를 유지하다가 수행이 완료되면 wait 상태로 복귀하여 다음 결심 결과 입력을 대기한다.

이와 같은 구조를 통해 control 모델은 decision 모델의 결심 결과를 실제 기동에 적용 가능한 제어 명령으로 변환하는 역할을 수행하며, 결심 결과 입력이 발생할 때에만 활성화되는 이산사건 기반 제어 모델로 동작한다.

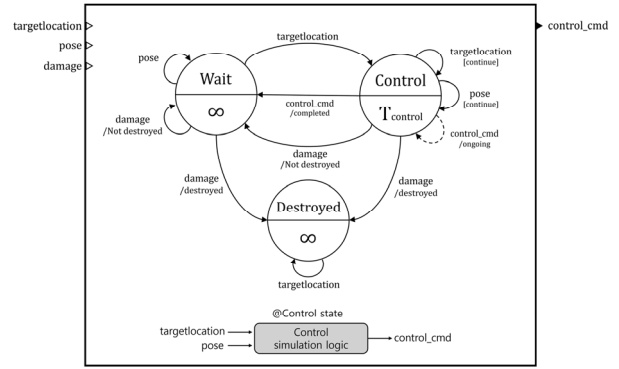


Fig. 4. Design of atomic model: Control

2.1.4 DEVS 원자모델 설계: Movement

Movement 모델은 control 모델로부터 전달된 제어 명령을 입력으로 받아, 플랫폼의 위치 및 heading (pose)을 갱신한다. control_cmd 입력을 받으면, movement 모델은 move 상태로 천이되어 이동 로직을 수행한다. move 상태에서는 입력된 제어 명령을 기반으로 위치, 속도 등의 상태를 갱신하고, 그 결과를 환경 모델에 전달한다. 기동 알고리즘은 단순 운동 모델부터 보다 정교한 운동 방정식까지 확장 가능하도록 설계하였다.

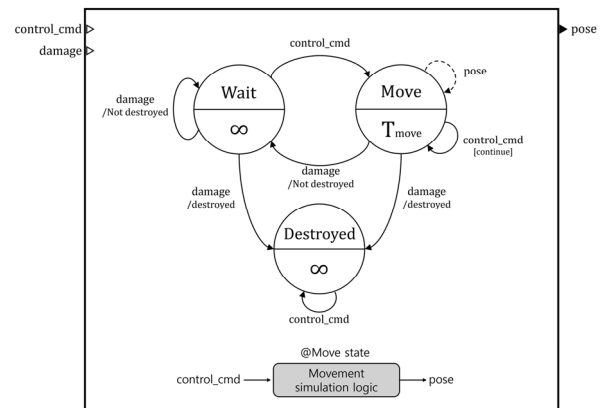


Fig. 5. Design of atomic model: Movement

2.1.5 Agent 간 결합 구조 및 동작 흐름

앞서 설계한 네 가지 원자 모델은 결합 모델(coupled model)로 구성되어 하나의 교전 객체를 형성한다. Detection agent에서 시작된 탐지 이벤트가 decision agent로 전달되어 결심 과정을 거친 후, control agent를 통해 기동 명령으로 변환되고, 최종적

으로 movement agent에서 실제 위치 변화로 반영된다. 이러한 일련의 과정은 메시지 기반 이벤트 흐름으로 표현되며, 각 agent는 독립적인 시간 진행과 상태 전이를 수행한다.

이와 같은 구조를 통해 agent 모델은 교전 기능 간의 명확한 역할 분리를 달성할 수 있으며, 교전 알고리즘의 교체 및 확장을 용이하게 한다. 또한 동일한 구조를 유지한 채 다양한 교전 시나리오와 무기체계 조합을 적용할 수 있는 기반을 제공한다.

2.2 Environment Model

제안하는 시뮬레이션 모델에서 환경(environment) 모델은 교전 객체들이 공유하는 전장 공간을 표현하며, 각 agent 모델 간의 상태 정보를 매개하는 역할을 수행한다. Environment 모델은 교전 객체의 위치 및 상태 정보를 관리하고, 이를 각 agent 모델에 전달함으로써 시뮬레이션 전체의 시간적·공간적 일관성을 유지한다.

각 플랫폼에서 출력된 pose는 environment 모델로 전달되어 전장 상태를 갱신한다. Environment 모델은 수신된 pose 정보를 바탕으로 각 객체의 현재 위치를 업데이트하며, 갱신된 위치 정보(object_loc)를 다시 각 플랫폼에 전달한다. 이를 통해 객체의 detection 모델은 환경 내 최신 위치 정보를 기반으로 탐지 로직을 수행할 수 있다.

환경 모델을 별도의 모듈로 분리함으로써, 교전 객체 내부 로직은 탐지·결심·제어·기동의 기능 수행에 집중할 수 있으며, 모듈의 연결 구조를 유지한 채 알고리즘(탐지 반경, 기동 로직 등)만을 변경하여 시나리오 기반 분석을 수행할 수 있다.

2.3 시뮬레이션 엔진: pyjevsim

본 연구에서 제안하는 교전 시뮬레이션 플랫폼은 탐지-결심-제어-기동의 네 가지 agent와 환경 모델을 DEVS 형식론 기반의 이산사건 모델로 구성한다. 따라서 시뮬레이션 실행 단계에서는 이산 사건 중심의 시간 진행, 모델 간 이벤트 전달 및 스케줄링, 결합 모델(coupled model) 구조 지원이 필수적으로 요구된다. 이에 본 연구는 DEVS 실행 메커니즘을 지원하는 시뮬레이션 엔진으로 pyjevsim을 사용하였다.

pyjevsim은 DEVS 기반 모델(atomic/coupled)의 실행을 위한 이벤트 스케줄링 및 메시지 전달 메커니즘을 제공하며, 모듈화된 모델 구조를 시뮬레이션 실행 단계에서 결합하고 동작시킬 수 있다. 또한 시뮬레이션 과정에서 생성되는 상태 및 이벤트 데이터를 저장하고 불러올 수 있는 snapshot 기능을 통해, 시나리오 반복 실험 및 결과 재현성 확보에 유리하다.

본 플랫폼은 pyjevsim 엔진 상에서 각 agent (detection/decision/control/movement)와 환경 모델을 독립 엔티티(entity)로 등록하고, 포트 기반 coupling을 통해 메시지 흐름을 구성하는 방식으로 실행된다. 교전 객체는 단일한 거대 모델이 아니라, 여러 atomic model의 결합으로 형성되며, pyjevsim은 이들 간의 이벤트 전달 및 시간 동기화를 수행한다.

2.4 동작 가시화: SIMDIS

교전 시뮬레이션은 다수의 플랫폼이 시간에 따라 이동하며 상호작용하는 복합적인 과정을 포함한다. 수치 데이터만으로는 모델의 동작 흐름과 교전 상황을 직관적으로 파악하는 데 한계가 있으므로, 시뮬레이션 결과에 대한 동작 검증과 교전 과정 분석을 위해 SIMDIS로 시각화가 가능하도록 구현했다.

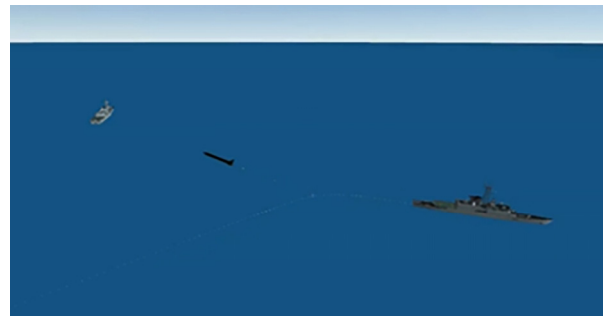


Fig. 6. SIMDIS-based visualization of engagement simulation results

제안한 시뮬레이션 모델은 Python 기반 DEVS 시뮬레이션 엔진 상에서 실행되며, 시뮬레이션 수행 이전에 초기 설정 정보가 로드된다. 시뮬레이션이 진행되는 동안 각 플랫폼의 위치 정보는 시간에 따라 기록되며, 시뮬레이션 종료 후 해당 결과는 SIMDIS에서 시각화가 가능한 asi 형식으로 저장된다. 이를 통해 전투 플랫폼과 무장 객체의 이동 경로, 상대적 위치 변화, 교전 시점 등을 3차원 공간 상에서 확인할 수 있다.

3. 시뮬레이션 실험

본 장에서는 제안한 교전 시뮬레이션 플랫폼이 다양한 교전 시나리오에 대해 동일한 구조로 적용 가능함을 검증하기 위해 설계한 시뮬레이션 실험 시나리오에 대해 설명한다. 이러한 특성을 검증하기 위한 사례로서, 미사일의 동작 구조를 단계적으로 변화시키는 case를 구성하고 이를 시나리오 분석에 적용하였다.

3.1 실험 시나리오

실험은 미사일의 제어 및 탐지 기능 수준에 따라 총 세 가지 case로 구성하였다. 각 시나리오는 미사일 모델의 내부 agent 구성 중 detection, control, movement 기능의 포함 여부 및 역할 분담 방식이 상이하다. 미사일의 에이전트 구성에 따른 세 가지 실험 케이스의 세부 정의는 다음과 같다.

Case 1에서는 발사 시점에 획득한 표적의 최종 좌표를 고정 목표로 비행하는 단순 미사일 모델을 사용한다. 미사일 내부는 control 및 movement agent로만 구성되며, 비행 중 목표 위치 갱신이나 탐지 기능은 수행하지 않는다. 또한, 적군 함정의 기동이나 회피 동작에 대해 반응하지 않는다.

Case 2는 미사일이 비행 중 아군 함정으로부터 주기적으로 갱신된 목표 위치 정보를 전달받아 비행 경로를 수정하는 모델을 사용한다. 해당 시나리오에서도 미사일 내부는 detection을 포함하지 않으며, 탐지 기능은 아군 함정에 의해 수행된다. 미사일은 전달 받은 최신 목표 위치를 기반으로 control agent에서 목표 방향을 재계산하고, 이를 바탕으로 비행 경로를 수정한다.

Case 3에서는 미사일이 자체적인 detection 기능을 포함한 자율 추적 미사일 모델로 구성된다. 미사일은 비행 중 detection agent를 통해 적군 함정을 탐지하며, 탐지 결과를 기반으로 추적 방향을 계산한다. 미사일의 탐지 성능은 아군 함정의 탐지 시스템에 비해 제한적으로 설정되었으며, 이는 실제 교전 상황에서 미사일 탑재 센서의 물리적 제약을 반영한 것이다.

실험의 목적은 적 함선의 탐지 수준을 단계적으로 변화시키며, 미사일의 내부 구성 차이에 따른 교전 효과도의 변화를 비교·분석하는 데 있다. 이를 위해 적

군 함정은 미사일을 발사하지 않고 아군 미사일에 의해 탐지될 경우 생존을 위해 회피 기동을 수행하며, 미사일의 비행 속도는 함정의 기동 속도보다 우세하게 설정하였다. 전장 환경 및 함정의 초기 배치 조건은 모든 실험군에서 동일하게 유지되며, 미사일 모델의 동작 구조만을 변경하여 실험을 수행한다. 시뮬레이션은 아군 미사일이 적군 함정을 요격한 경우 또는 미사일이 목표를 상실하거나 최대 비행 시간이 초과된 경우 종료된다.

3.1.1 시나리오 1: Case 1과 2의 미사일 제어 수준에 따른 교전 효과도

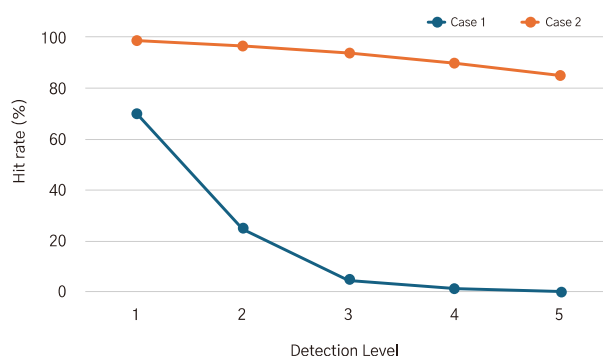


Fig. 7. Analysis of missile hit rate according to detection level: Scenario 1

Fig. 7은 적군 함정의 탐지 수준 변화에 따른 case 1과 case 2의 미사일 명중률 변화를 비교한 결과를 나타낸다. 탐지 수준은 적군이 아군 함정을 인지하고 회피 기동을 개시할 수 있는 조기경보 능력을 단계적으로 증가시킨 변수로 해석할 수 있다. Case 1의 경우, 탐지 수준이 증가함에 따라 명중률이 급격히 감소하는 경향을 보였다. 이는 발사 시점에 설정된 고정 목표 좌표만을 기반으로 비행하는 구조적 특성으로 인해, 적군이 조기에 회피 기동을 수행할 경우 미사일의 목표 위치 오차가 비행 시간 동안 누적되기 때문으로 해석된다. 반면 case 2는 탐지 수준이 증가함에 따라 명중률이 완만하게 감소하는 경향을 보였다. 탐지 수준 1에서는 약 99%의 높은 명중률을 유지하였으며, 탐지 수준 5에서도 높은 수준의 명중률을 유지하였다. 이는 미사일이 비행 중 아군 함정으로부터 주기적으로 갱신되는 목표 위치 정보를 기반으로 비행 경로를 수정함으로써, 적군의 회피 기동에 효과적으로 대응할 수 있음을 의미한다.

3.1.2 시나리오 2: Case 2와 3의 미사일 탐지·추적 기능에 따른 교전 효과도

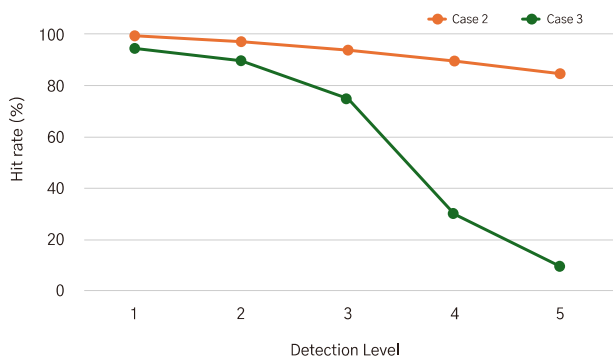


Fig. 8. Analysis of missile hit rate according to detection level: Scenario 2

Fig. 8은 case 2와 case 3의 명중률을 비교하여, 미사일의 자율 탐지·추적 기능이 교전 효과도에 미치는 영향을 분석한 결과를 나타낸다. 두 case 모두 목표 위치 갱신 또는 추적 기능을 포함하지만, case 2는 아군 함정의 탐지 정보를 활용하는 반면, case 3는 미사일 자체 탐지 기능에 의존한다는 차이가 있다. 탐지 수준이 낮은 구간에서는 case 3가 비교적 높은 명중률을 유지하며 case 2와 유사한 성능을 보였으나, 탐지 수준이 증가함에 따라 명중률이 급격히 감소하여 두 case 간 성능 차이가 크게 확대되었다. 이러한 결과는 미사일의 자율 탐지·추적 기능이 구조적으로는 외부 정보 의존도를 낮출 수 있으나, 센서 성능이 충분하지 않을 경우 조기 회피 환경에서 오히려 교전 성능 저하 요인으로 작용할 수 있음을 보여준다.

4. 결론

본 연구에서는 다양한 교전 시나리오에 대해 동일한 구조의 시뮬레이션 모델을 적용할 수 있는 agent 기반 교전 시뮬레이션 모델을 제안한다. 제안한 시뮬레이션 모델은 교전 과정에서 발생하는 탐지, 결심, 제어, 기동 기능을 각각 독립적인 agent로 분리하여 설계함으로써, 교전 로직과 시뮬레이션 모델 구조를 명확히 분리할 수 있도록 한다. 이를 통해 특정 무기 체계나 교전 상황에 종속되지 않고, 다양한 교전 조건을 동일한 모델 구조 내에서 표현할 수 있는 기반을 마련하였다. 시뮬레이션의 각 플랫폼은 환경 모델을 매개로 시공간적 정보를 참조하고 갱신하도록 구성

되었으며, 이산사건 기반의 실행 구조를 통해 교전 흐름을 일관성 있게 표현할 수 있도록 설계되었다. 또한 DEVS 실행 메커니즘을 지원하는 pyjevsim 시뮬레이션 엔진과의 결합을 통해 모델 간 이벤트 전달과 시간 관리를 수행하였고, 가시화 모듈과의 연계를 통해 교전 과정 전반의 동작을 시각적으로 확인하고 검증할 수 있음을 보였다.

제안한 시뮬레이션 모델의 적용 가능성을 확인하기 위해, 동일한 agent 구조를 유지한 상태에서 미사일 모델의 제어 및 탐지 기능을 단계적으로 변경한 교전 시나리오를 구성하였다. 해당 시나리오를 통해 시뮬레이션 모델의 구조를 변경하지 않고도 내부 agent의 동작 로직 차이에 따라 교전 결과의 변화가 관찰되었다. 이는 탐지 방식, 제어 구조, 결심 로직 간의 상호작용에 의해 형성되는 교전 과정을 기능별 agent의 모듈화된 구조를 통해 시뮬레이션 상에서 유연하게 표현할 수 있음을 보여준다.

본 연구에서 제안한 교전 시뮬레이션 플랫폼은 다양한 전장 환경과 교전 조건을 표현할 수 있는 범용적인 시뮬레이션 프레임워크이다. 향후 연구에서는 다수 플랫폼이 참여하는 복합 교전 시나리오로의 확장 및 고도화된 판단 및 협업 로직의 적용을 통해 실제 운용 환경에 근접한 교전 분석을 수행하고자 한다.

참고문헌

- [1] Andreas Tolk, 'Tutorial on the Engineering Principles of Combat Modeling and Distributed Simulation,' in proceedings of 2019 Winter Simulation Conference (WSC), 2019, pp. 18-32.
- [2] Mi Jang, Hee-Mun Park, & Kyung-Min Seo, 'DEVS-Based Simulation Model Development for Composite Warfare Analysis of Naval Warship,' Journal of the Korea Society for Simulation, VOL. 32, NO. 4, 2023, pp. 41-58.
- [3] Andreas Tolk, Erdal Cayirci, Randall Shumaker, Nabil R. Adam, Stefan Pickl, & Joseph A. Sullivan, 'Defense and Security Applications of Modeling and Simulation — Grand Challenges and Current Efforts,' in proceedings of the 2012 Winter Simulation Conference (WSC), Berlin, Germany, 2012.
- [4] Haiwang Cao & Chaogai Xue, 'An Integrated System for Underwater Warfare Simulation,' in proceedings of 2009 Chinese Control and Decision Conference, Guilin, China, 2009.
- [5] Youngjae Kim & Yonghoon Ha, 'Comparative Analysis of Naval Gun Effectiveness Against Unmanned Surface Vehicles (USVs) Using ABM Simulation,' Journal of the KNST, VOL. 8,

NO. 3, 2025, pp. 337–344.

[6] Yuan Feng, Zhong Yao, Pan Pan Xun, & Rui Wang, 'Model-Based Simulation of the Combat Flow of Naval Gun Weapon Systems,' in proceedings of the 2022 International Conference on Big Data, IoT, and Cloud Computing, 2022.

[7] Kookhyun Kim, 'Engagement-Scenario-Based Decoy-Effect Simulation Against an Anti-Ship Missile

Considering Radar Cross Section and Evasive Maneuvers of Naval Ships,' *Journal of Ocean Engineering and Technology*, VOL. 35, NO. 3, 2021, pp. 238–246.

[8] Sang C. Park, Yongjin Kwon, Kilyoung Seong, & JaiJeong Pyun, 'Simulation Framework for Small Scale Engagement,' *Computers & Industrial Engineering*, VOL. 59, NO. 3, 2010, pp. 463–472.