



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

DEVS 형식론 기반 핵심 선박장비 및
에이전트 모델링과 항해 시뮬레이션

A Modeling of Core Systems and Agents in Vessel
based on DEVS Formalism and Navigation Simulation

지도교수 이 장 세

2019년 8월

한국해양대학교 대학원

컴퓨터공학과

우 상 민

본 논문을 이상민의 공학석사 학위논문으로 인준함

위원장 : 박 휴 찬



위 원 : 김 재 훈



위 원 : 이 장 세



2019년 6월 27일

한국해양대학교 대학원

목 차

List of Tables	iii
List of Figures	iv
Abstract	vi
제 1 장 서 론	1
제 2 장 관련 연구	4
2.1 모델링 및 시뮬레이션 기술	4
2.1.1 SES/MB 프레임워크 개요	4
2.1.2 DEVS 형식론 개요	8
2.1.3 에이전트 기술 개요	11
2.2 핵심 선박장비 및 선박 충돌회피 기술	12
2.2.1 핵심 선박장비 특성 및 기술	12
2.2.2 Hasegawa의 충돌위험도 평가 모델	16
2.2.3 COLREGs 개요	19
제 3 장 항해 시뮬레이션을 위한 모델링	22
3.1 SES/MB를 이용한 항해 시뮬레이션 시스템 모델링	23
3.2 DEVS 형식론 기반 핵심 선박장비 모델 설계	28
3.3 COLREGs 기반 에이전트 모델링	32
제 4 장 COLREGs 기반 항해 시뮬레이션	35
4.1 시뮬레이션 시나리오 및 초기조건 설정	35
4.2 시뮬레이션 결과 검증	44

제 5 장 결론 및 향후 연구	50
참고문헌	52
부록 A 항해 시뮬레이션 시스템 구성 모델 의사 코드	55

List of Tables

Table 2.1	List of major navigation and communication systems	13
Table 2.2	Requirements for TCS according to Category	15
Table 2.3	Fuzzy rules for ship Collision-Risk (by Hasegawa)	18
Table 3.1	Navigation cases of COLREGs for other ships	33
Table 3.2	Knowledge base of the officer model	33
Table 3.3	Knowledge base of the helmsman model	34
Table 4.1	General characteristics of the simulation ship	36
Table 4.2	ROT calculation by the turning test	38
Table A.1	The pseudo-codes of DEVS-based experiment frame model	56
Table A.2	The pseudo-codes of DEVS-based Navigation SPACE model	57
Table A.3	The pseudo-codes of DEVS-based Sensors model	60
Table A.4	The pseudo-codes of DEVS-based AISs model	62
Table A.5	The pseudo-codes of DEVS-based TCS model	63
Table A.6	The pseudo-codes of DEVS-based Propulsion and Steering Systems model	65
Table A.7	The pseudo-codes of DEVS-based Agent model	66

List of Figures

Fig 2.1	A concept of the SES/MB	5
Fig 2.2	An example of the SES	6
Fig 2.3	An example of the PES	7
Fig 2.4	An example of the model base and the simulation model structure	8
Fig 2.5	A structure of knowledge base of intelligence units	11
Fig 2.6	A structure of core navigation and communication systems ...	13
Fig 2.7	Concept of HCS and TCS	14
Fig 2.8	Membership function for ship Collision-Risk (by Hasegawa) ...	17
Fig 2.9	The rule of navigation lights and navigation cases of COLREGs	21
Fig 3.1	A SES of navigation simulation systems	26
Fig 3.2	An example of the PES	27
Fig 3.3	A simulation model structure of Fig. 3.2	28
Fig 3.4	Simulation model structure of navigation	29
Fig 4.1	Turning test for the simulation ship	37
Fig 4.2	Newton's optimal difference interpolation polynomial for multivariate function	39
Fig 4.3	The initial situation of COLREGs-based 1:1 navigation simulation	42
Fig 4.4	The initial situation of COLREGs-based multiple ship (1:3) navigation simulation	43

Fig 4.5 Trajectory of collision avoidance simulation for Crossing	45
Fig 4.6 Trajectory of collision avoidance simulation for Head-On (OS, TS avoidance)	46
Fig 4.7 Trajectory of collision avoidance simulation for Overtaking (OS avoidance)	48
Fig 4.8 Trajectory of multiple ship (1:3) collision avoidance simulation	49

DEVS 형식론 기반 핵심 선박장비 및 에이전트 모델링과 항해 시뮬레이션

우 상 민

한국해양대학교 대학원
컴퓨터공학과

초록

과거부터 꾸준한 해상 교통량 증가로 선박 사고 발생은 지속해서 증가하고 있으며, 이에 따라 항해사 의사결정 지원 등을 위한 안전 지원 IT 서비스 연구가 활발히 진행되고 있다. 이러한 IT 서비스들을 선박에 탑재하여 운영하기 위해서는 그 유용성 검증이 필수적으로 요구된다. 선박 조종시뮬레이터(FMSS, full mission ship simulator)와 같은 시험 장비 대역 및 실제 선박 탑재를 통한 시험 등은 막대한 비용과 위험 등의 한계를 가진다. 이에 따라 저비용, 쉬운 접근성을 가진 컴퓨터 기반 항해 시뮬레이션 모델링 방법을 제안한다.

본 논문에서는 항해 시뮬레이션을 통해 IT 서비스의 신뢰성 있는 유용

성 검증을 위해 다음의 핵심 요소를 모델링한다. 첫째, 항로 계획에 따른 목적지까지의 자율적인 항해 시뮬레이션을 위해 핵심 선박장비와 항적제어 기술을 분석하고 모델링한다. 둘째, 항해 중 발생하는 충돌위험 상황에 대해 사용자 미개입의 능동적인 회피를 수행하기 위해 국제해상충돌예방규칙(COLREGS, international regulations for preventing collisions at sea)에 따른 의무를 분석하여 에이전트를 모델링한다.

핵심 선박장비와 에이전트는 많은 모델로 구성되어 상당히 복잡한 구조를 가진다. 이를 체계적으로 표현하고 연속적인 시간상에서 이산적인 사건에 따른 모델들의 동적인 변화 표현을 위해 Zeigler가 제안한 DEVS(Discrete Event System Specification) 기반 SES/MB(System Entity Structure/Model Base) 프레임워크를 활용하였다.

제안하는 모델링 방법은 항해 시뮬레이션 시스템 구현에 적용되어 다양한 항해 안전 지원 서비스에 대한 시뮬레이션 및 신뢰성 높은 분석에 활용이 가능할 것으로 기대된다.

키워드 : SES/MB Framework, DEVS Formalism, Agent, Modeling, Navigation Simulation

A Modeling of Core Systems and Agents in Vessel based on DEVS Formalism and Navigation Simulation

Woo, Sangmin

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

Abstract

From the past, maritime traffic accidents have been consistently increased due to a steady increase in marine traffic volume. Thus, the research on safety-assisted IT services has been actively conducted for decision-making of navigators and others.

In order to operate the IT services on board, its utility must be verified. Equipment rental such as FMSS (full mission ship simulator) or ship test, however, takes significant cost and risk limitations. Accordingly, this thesis proposes a computer-based

navigation simulation modeling method with low cost and easy accessibility. For this, this thesis examines the following key elements for reliable validation of IT service through simulation of navigation:

- Analyzing and modeling core vessel equipments and track control technology for autonomous navigation simulation based on destination route planning.
- Modeling agents and its obligations according to COLREGs, international regulations for preventing collisions at sea, in order to actively avoid collision risk during navigation.

The core vessel equipments and agents consist of many models and have a fairly complex structure. For systematically representation of the structure and dynamic changes of discrete event on continuing time, DEVS (discrete event system specification) based SES/MB (system entity structure / model base) is used.

We expect that the proposed modeling method will be applied to the implementation of the navigation simulation system and used for simulation of various navigation safety support services and highly reliable analysis.

Key Words: SES/MB Framework, DEVS Formalism, Agent, Modeling, Navigation Simulation

제 1 장 서 론

최근, 해상 교통량이 증가함에 따라 선박사고 위험이 지속해서 발생하고 있다. 해양수산부의 해양 사고 통계[1]에 따르면 지난 5년간(2013~2017년) 연평균 1,833건의 사고가 발생하였고, 2017년은 총 2,582건(전년 대비 11.9% 증가)으로 사고 발생은 꾸준히 증가하고 있다. 사고 원인 중 단순 기관 고장 등을 제외하면 경계 소홀 등 인적과실(human factor)이 대부분을 차지하고 있는 만큼 선박사고에 대한 종사자들의 안전의식 강화뿐만 아니라 항해사의 업무 지원을 위한 IT 서비스의 필요성 또한 강조되었다. 이에 따라, 인적과실 및 해양 사고를 최소화하기 위해 선박 충돌 위험에 대한 항해사 의사결정 지원, 지능형 항해 지원 등의 여러 서비스의 제안이 점차 증가하고 있다.

항해 안전 지원 서비스는 과거부터 다양한 접근을 통해 제안되었지만, 그 유용성을 검증함에서는 다음의 어려움이 따른다. 첫째, 테스트 선박을 통한 시험은 예상치 못한 사고로 인한 위험과 함께 막대한 비용이 들 수 있으며 제안 시스템의 탑재 및 테스트를 허용하는 선박 섭외 또한 어려운 실정이다. 둘째, 항해사 교육 목적으로 사용되는 선박 조종 시뮬레이터(FMSS, full mission ship simulator)에서 발생하는 항해 데이터 활용이 가능하나 사용 비용이 상당히 많이 들어 연구 검증을 위한 도구로서의 접근성은 떨어진다. 셋째, 선박 운항 관련 지식이 없는 서비스 개발 연구자들의 조종에 의한 FMSS의 데이터는 신뢰성 있는 검증에 적절하지 않다. 이러한 어려움의 해소 및 다양한 서비스의 검증을 위해 모델링 및 시뮬레이션(M&S, Modeling and Simulation) 기법 등을 도입한 항해 시뮬레이션 연구[2-6]가 활발히 이루어지고 있다.

항해 관련 M&S 시스템 연구들로는 선박 충돌 위험도 및 회피 경로를 시뮬레이션하는 연구들[2-5]이 있으며, 항해 시뮬레이션과 서비스 유용성 평가를 지원하는 연구[6]가 진행된 바 있다. 기존 시스템에서는 COLREGs를 기반으로 회피 경로를 판단하였지만, 실제 선박의 항해 지휘체계에 따라 구별되는 특징이 없고 판단 수준은 SM(state machine) 기반으로써, 상황 변화에 따른 묘사가 어려운 점이 있다. 또한, 충돌 위험도 및 회피 경로 추정 서비스에 대한 검증에 국한된 연구가 대부분으로 목적 경유지까지의 항로 계획, 항적 제어, 침로 유지 등의 고려를 하지 않아 전반적인 항해의 시뮬레이션이 어렵다는 한계가 있다.

본 논문에서는 이러한 한계점을 극복하기 위해 컴퓨터 기반 항해 시뮬레이션을 위한 모델링 방법을 제시한다. 제안하는 모델링 방법은 Zeigler가 제안한 DEVS(discrete event system specification) 기반 SES/MB(system entity structure/model base) 프레임워크[7-8]로 시스템 구조를 모델링하고 이산시간 및 이산사건 기반 시뮬레이션으로 시간과 상태 변화에 따른 상황을 묘사한다. 또한, 목적 경유지까지 항로 계획에 따른 항적 제어를 위해 실제 선박에 탑재되는 핵심 선박장비의 기능과 특성을 분석하고, 가상 항해 요소 등을 COLREGs[9]에서 정의하는 항법 관계를 지식으로 가진 에이전트를 모델링한다.

이를 통해, 목적지까지의 자율적인 항해와 선박 충돌위험 상황에 대한 에이전트의 자율적인 판단으로 능동적인 회피를 할 수 있다. 제안하는 시뮬레이션 모델링 방법의 검증을 위해 해양전자장비 간의 통신 표준화를 위해 정의된 데이터 프로토콜인 NMEA 0183 Sentence[10]를 준용하여 항해 데이터를 출력한다.

제안하는 모델링 방법은 다음의 다섯 가지 장점을 제공한다. 첫째, 기존 연구의 기능적인 측면만 중점을 둔 모델링이 아닌 핵심 선박장비와 에이전트를 SES/MB(system entity structure/model base) 프레임워크로 모델링함에 따라 시스템의 구조를 체계적으로 표현한다. 둘째, 기존 관련 연구

의 스크립트 또는 SM 기반 시뮬레이션이 아닌 DEVS 기반 시뮬레이션으로 시간 및 상태의 변화에 따른 복잡한 상황 묘사를 한다. 셋째, 현행 연구들의 항해 경로 제어보다 항로 계획에 따라 전체 경유지를 경유하기 위해 실제 선박에 탑재되는 항적제어시스템(TCS, track control system)을 구현한다. TCS는 항해사의 개입 없이 항로 계획에 따라 자동으로 항적 제어를 하는 시스템으로써, 구현을 위하여 TCS 국제표준인 IEC 62065 Edition 2.0[10] 기반으로 그 기능 요구사항을 분석하였다. 넷째, COLREGs 기반 에이전트는 선박 충돌위험 상황을 지식 기반으로 판단하여 자율적인 충돌 회피를 한다. 다섯째, 항해 시뮬레이션을 위한 모델 구조는 항해 시뮬레이션 시스템 구현에 활용되어 다양한 항해 안전 지원 서비스에 대한 시뮬레이션 및 분석에 활용될 수 있다. 또한, 시뮬레이션 중 각 모델에서 출력되는 데이터는 NMEA 0183을 준용하여 각종 항해 디스플레이 장치의 기능적인 시험 및 실제 선박 탑재를 위한 유용성 검증을 위해서도 충분히 활용할 수 있을 것으로 기대된다.

본 논문의 구성은 다음과 같다. 제2장에서는 본 논문과 관련된 기술과 관련 연구를 소개한다. 제3장에서는 항해 시뮬레이션 시스템의 모델링에 관해 설명하고, 제4장에서는 제안하는 모델링 방법의 유용성 검증을 위해 구현하는 항해 시뮬레이션 시스템의 COLREGs 기반 시뮬레이션을 수행하고 그 결과를 분석 및 검증 이후 제5장 결론 및 향후 연구로 마무리하게 된다.

제 2 장 관련 연구

2.1 모델링 및 시뮬레이션 기술

2.1.1 SES/MB 프레임워크 개요

Zeigler에 의해 제안된 SES/MB(system entity structure/model base) 프레임워크[7-8]는 각종 지식이 포함된 구조적 지식 표현 기법으로, 복잡한 시스템 구조의 설계 및 분석을 계층적으로 용이하게 하며 계획-생성-평가(plan-generate-evaluate) 패러다임을 제공하는 M&S 환경이며, 기본적으로 SES과 MB로 구성되어 시뮬레이션 모델을 이룬다. SES는 복잡한 시스템의 구조를 체계적으로 표현할 수 있는 구조체를 말하며, 나타내는 모델의 구성 관계, 종류, 결합구조, 그리고 제약조건 등의 특정 형식으로 표현한 것으로서 선언적 성격을 가진다. MB는 시스템의 행위적 및 동역학적 특성을 나타내며 기호적인 표현 수단을 제공하는 절차적 성격을 가진 모델들로 구성된다.

Fig 2.1은 SES/MB 프레임워크의 개념을 나타낸다. SES는 구성된 모델의 결합 관계를 저장하고 있으며 단말 노드(leaf node)인 사각형, 삼각형 등은 MB에 저장된다. 이들은 변환(transformation operation)에 의해서 시뮬레이션 모델이 자동 생성되며, 이 모델과 실험 장치(EF, experimental frame)가 결합하여 최종적인 시뮬레이션 모델이 구성된다. 여기에서 말하는 EF는 시뮬레이션 모델의 행위 발생을 위해 다양한 입력을 하고, 효과적인 결과 분석을 수행하기 위해 출력된 결과를 받아 정리하는 장치를 말한다. 이와 같은 모델의 행위와 구조의 분리를 통해 복잡한 시스템의 구조를 체계적으로 표현할 수 있다. 따라서, SES/MB 프레임워크는 본 논문

에서 제안하는 모델링 방법을 위한 핵심 선박장비 및 가상 항해 요소 등의 체계적인 구성과 시스템의 동작을 분석하는데 효과적으로 적용될 수 있다. 본 절에서는 SES/MB의 기본 구성인 SES과 MB에 대해 살펴본다.

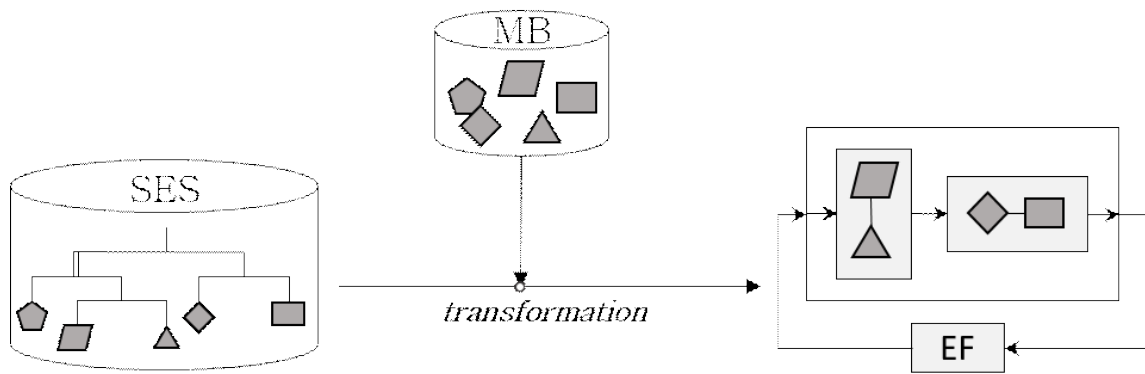


Fig. 2.1 A concept of the SES/MB

(1) SES(system entity structure)

SES는 시스템의 구조를 나타내는 지식을 분할(decomposition), 분류(component taxonomies), 결합 관계(coupling specification), 제약 조건(constraints) 등의 특정 형식으로 표현할 수 있는 구조체를 말한다. SES는 Fig 2.2와 같이 트리 형태로 계층적 관계를 가진 모델들을 표현하며, 이를 위해 Entity, Aspect, Specialization의 세 가지 형태의 노드를 가진다.

Entity는 독립적으로 식별될 수 있는 모델 또는 객체에 해당하며 다수의 Aspect와 Specialization으로 구성할 수 있다. 상위 Entity 노드는 하위 노드와의 구조를 위해 다음의 세 가지 모드 중 하나로 구성하여야 한다. 첫째, Aspect는 하위 노드가 상위 Entity 노드의 필수적인 구성원일 경우 분할 적 표현을 담당하는 모드로서 한 줄의 수직선(Ⅰ)에 의해 표현된다. 둘째, Specialization은 상위 Entity 노드의 분류적 표현을 담당하는 모드로서 하위 노드를 선택함에서 대안을 나타내며, 두 줄의 수직선(Ⅱ)에 의해 표현된다. 셋째, Multiple Entity는 시스템에서 동일 속성을 가진 여러 개의

Entity를 표현할 때 사용되는 것으로, 세 줄의 수직선(III)에 의해 표현되고, 최대 선택 가능한 Entity의 수(~n)를 수직선과 함께 표시한다.

SES는 이와 같은 세 가지의 관계 표현에 의해 시스템의 전체 구조를 트리 형태로 나타내는 것이며, 표현을 함에 있어서 다음의 세 가지 규칙을 필수적으로 따라야 한다. 첫째, SES는 Entity로 시작해서 Entity로 끝나야 한다. 둘째, 연속적인 계층 구조에서는 동일한 Entity가 존재할 수 없으며, 만약 다른 계층에 존재한다면 동일한 구조를 가져야 한다. 셋째, Attribute 변수는 Entity의 특성을 표현하는 것으로 유형 간 유일한 값을 가져야 한다.

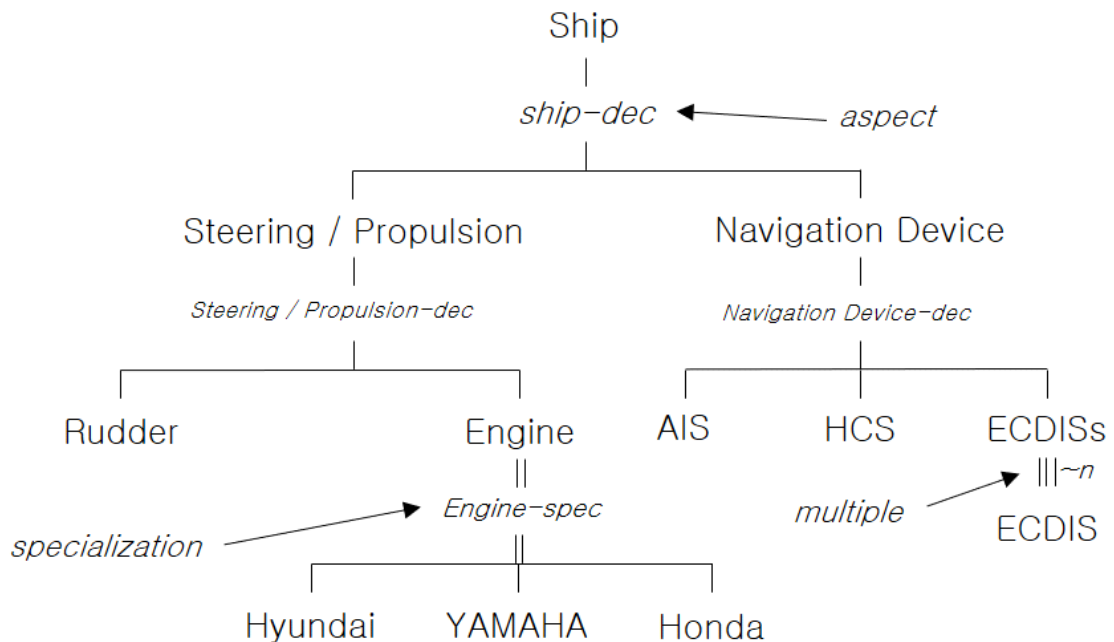


Fig. 2.2 An example of the SES

SES로 나타낸 시스템의 구조를 통해 표현하고자 하는 한 가지의 대상 구조를 선택할 수 있으며, 이는 시뮬레이션을 하고자 하는 대상을 선택하는 과정이라고 할 수 있다. 이 과정은 하나의 구조를 만들기 위한 방법인 가지치기(pruning)를 통해 행하여지며, 선택된 구조를 PES(pruned entity structure)라고 부른다. Fig 2.2의 SES에서 선택된 PES의 예를 Fig 2.3에

나타내었다. SES의 Specialization 및 Multiple Entity 노드에서 단 하나의 Entity를 선택하는 가지치기 과정을 거쳐 오직 Aspect(decomposition)로만 이루는 구조인 PES가 생성된다.

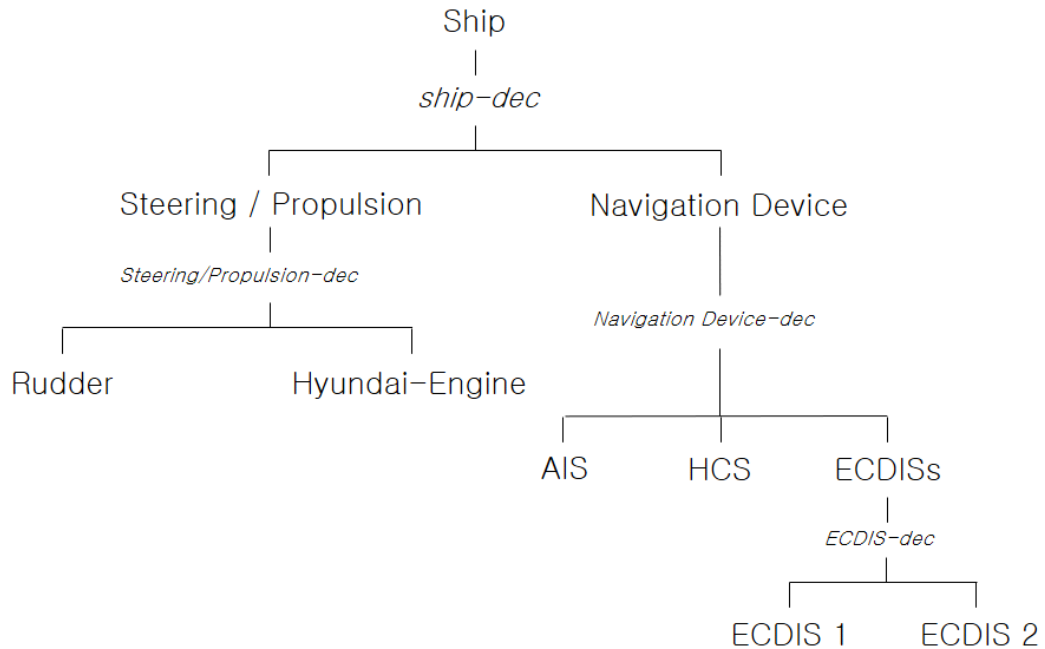


Fig. 2.3 An example of the PES

(2) MB(model base)

MB는 각 모델의 행위적 특성을 나타내는 것으로 절차적 특징을 가지며, 동역학적 및 기호적인 표현 수단을 제공한다. MB에 저장되는 모델들은 SES의 단말 노드들로 구성되며, 이 모델들은 정의된 연결(coupling) 관계에 의해 결합함에 따라 최종적인 시뮬레이션 모델 구조를 구성하게 된다. Fig 2.4는 앞서 예를 들어 설명한 SES의 MB와 그에 대한 PES의 모델 구조를 나타내었다.

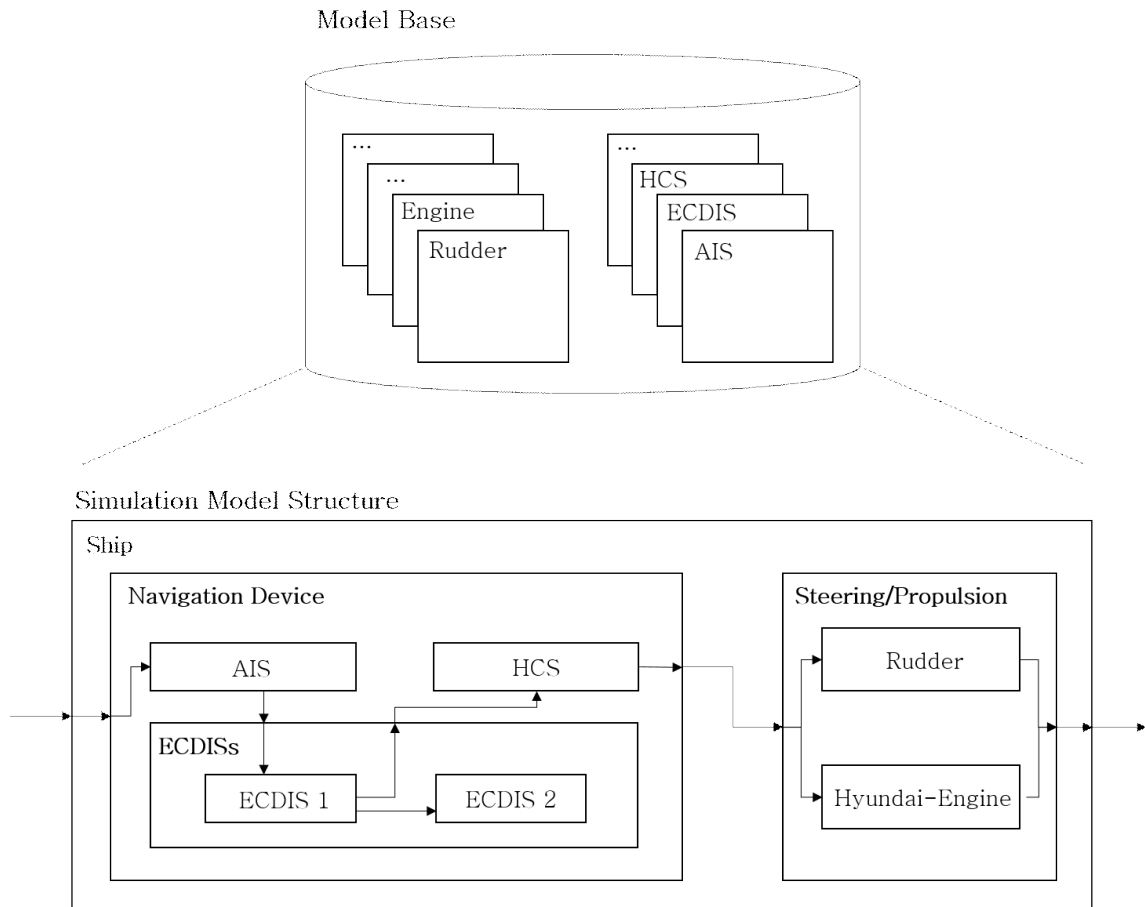


Fig. 2.4 An example of the model base and the simulation model structure

2.1.2 DEVS 형식론 개요

앞서 설명한 MB의 구성을 위하여 본 논문에서는 Zeigler가 제안한 이산 사건 시스템 형식론(DEVS, discrete event system specification)[7-8]을 적용한다. 시간에 따른 동적인 변화와 이산적인 사건에 따른 상태의 전이로 표현되는 항해 시뮬레이션에서의 구성 모델들은 DEVS 형식론으로 효율적인 통합 표현이 가능하다.

기본적으로 DEVS는 연속적인 시간상에서 입력 사건에 의해 상태가 전이 되거나 시간의 흐름에 따른 내부적인 상태 전이에 대한 시스템의 동적

인 변화를 기술한 형식론으로써, 상태의 전이는 입력, 상태, 출력, 시간의 흐름 등의 함수들에 의한 상호작용으로 이루어진다. DEVS 모델은 대상으로 하는 시스템의 동작을 수학적으로 기술한 것으로 원자(Atomic) 모델, 결합(Coupled)모델로 구분되며, 이 두 가지 모델을 사용하여 상태 전이에 대한 체계적인 표현과 시스템 모델의 모듈화를 통한 계층적인 개발이 가능하다. DEVS의 원자 모델은 다음과 같이 표현된다.

$$M = \langle X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta \rangle \quad (2.1)$$

여기에서,

X : 입력 집합

S : 상태 집합

Y : 출력 집합

δ_{ext} : 외부상태 전이함수 ($Q \times X \rightarrow S$)

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

δ_{int} : 내부상태 전이함수 ($S \rightarrow S$)

λ : 출력 함수 ($S \rightarrow Y$)

ta : 시간 진행 함수 ($S \rightarrow R^{+0,\infty}$)

단, $R^{+0,\infty}$ 는 음수를 제외한 실수 집합

원자 모델은 식 (2.1)과 같이 발생하는 사건들의 입력 변수들의 집합을 나타내는 X , 출력 변수들의 집합을 나타내는 Y , 상태 변수들의 집합을 나타내는 S 의 세 가지의 집합과 네 가지의 함수로 표현된다. 외부상태 전이

함수 δ_{ext} 는 시스템 외부에서 X 를 입력받았을 때 모델의 상태를 변화시키며, 내부상태 전이 함수 δ_{int} 는 시간 진행 함수 ta 에 설정된 시간이 지나면 모델 내부의 상태를 전이한다. 내부와 외부상태 전이가 이루어질 때 그에 따른 출력이 존재할 수 있으며, 이 출력 들의 집합을 λ 로 나타낸다.

모델의 행위를 나타내는 원자 모델과 달리, 결합 모델은 원자 모델 또는 다른 결합 모델들로 구성되는 모델로서, 모델을 연결하여 더 큰 시스템을 만들고, 또 그 시스템이 더 큰 시스템의 모델로 사용될 수 있도록 하는 기능을 제공한다. 다음은 결합모델의 정의를 나타낸 것이다.

$$CM = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, SELECT \rangle \quad (2.2)$$

여기에서,

D : 구성 모델의 이름 집합

M_i : 컴포넌트 모델

I_i : i 의 출력에 연결된 컴포넌트 집합

$Z_{i,j}$: i 에서 j 로의 출력전이 함수

$SELECT$: 동 시간 내부상태 전이 모델 우선순위 지정 함수

$$(2^D - \{\} \rightarrow D)$$

원자 모델 또는 다른 결합 모델들을 원소로 가진 결합 모델은 자신의 컴포넌트들을 구성한다. $\{I_i\}$ 와 $\{Z_{i,j}\}$ 는 컴포넌트 간의 결합 관계이며, 시스템의 모델을 계층적으로 구성하기 위해 결합모델과 원자모델을 이용할 수 있다. 이는 수학적 형식론에 기반 하여 여러 구성 모델들이 연결된 큰 시스템을 효율적으로 표현할 수 있는 장점을 가진다.

2.1.3 에이전트 기술 개요

에이전트는 자율적인 의사결정을 위해 지식베이스(KB, knowledge base)를 기반으로 추론, 정보 교환, 학습 등의 수행을 통한 지능을 가진 요소이다[8]. 항해 M&S 분야에서의 에이전트는 항해 환경에서 선박장비가 결정하는 행동 이외의 인간에 의해 행하여져야 할 의무를 특정 지식으로 가진 개체이며, 실제 선박에서는 해당 의무를 수행하는 요소인 항해사와 조타수로 분류할 수 있다.

본 논문에서는 이러한 요소를 에이전트로 모델링하기 위하여 엔진 기반 모델링 기법을 이용하며, 이는 Zeigler와 Chi가 제안한 Endomorphism 개념을 따른다. Endomorphism 개념은 모델링하기 위한 에이전트의 대상이 되는 요소에 대한 추상화된 모델을 가져야 하며, 엔진 기반 모델링 기법은 지식베이스 모델로부터 최적의 해를 찾기 위한 추론엔진(IE, inference engine) 모델이 분리된 구조를 가져야 함을 설명한 모델링 기법이다.

지식베이스는 각 지능 요소에 존재하며 일반적인 규칙 구조인 IF-THEN 형식을 기반으로 하며, 이는 추론엔진을 통해 해당 규칙에 맞는 적절한 결과를 추론한다. 추론엔진은 지식베이스가 가진 규칙의 수행 시점을 결정하여 최적의 해를 찾는다. Fig 2.4에 지식베이스의 구조를 나타내었다.

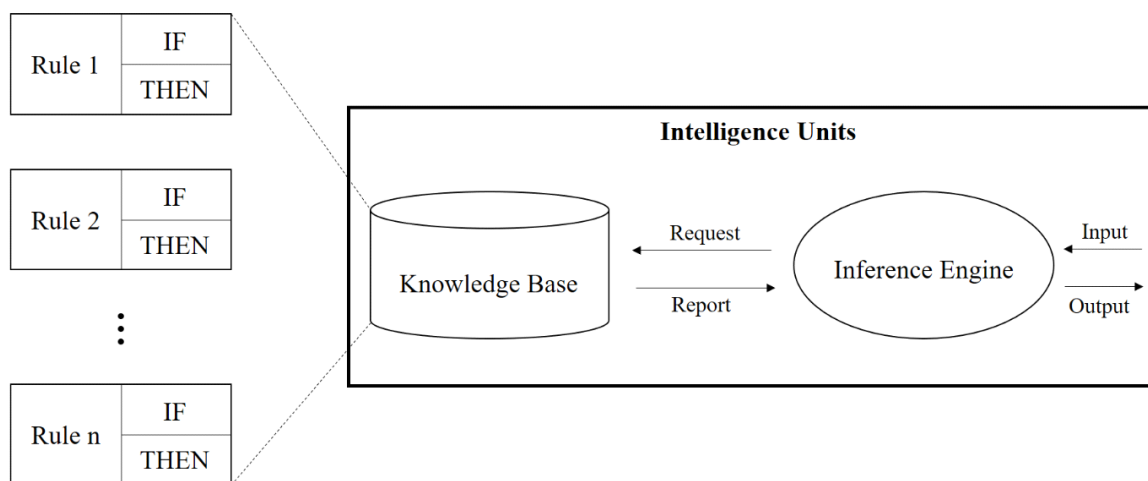


Fig. 2.5 A structure of knowledge base of intelligent units

2.2 핵심 선박장비 및 선박 충돌회피 기술

2.2.1 핵심 선박장비 특성 및 기술

국제해사기구(IMO, international maritime organization)는 해상인명안전협약(SOLAS, safety of life at sea)에서 규정한 선박장비 강제 탑재 규정을 적용하여 국제항해에 종사하는 선박은 종류 및 크기에 따라 의무적으로 규정 장비를 탑재하여야 한다. 본 논문에서는 제안하는 항해 시뮬레이션을 위한 구조 모델링과 각 모델의 구현을 위하여 최소 500톤급 이상의 여객선, 화물선 등에 규정된 탑재 장비 중 주요 항해통신장비를 대상으로 그 특성 및 주요 기술을 분석하였다. 주요 장비의 목록과 특성을 Table 2.1에 정리하였고, 그 구조를 Fig 2.6에 도식화하였다.

Fig 2.6의 항적제어시스템(TCS, track control system)[11]은 기존 분리 운영되었던 전자해도표시시스템(ECDIS, electronic chart display and information system)과 선수방위제어시스템(HCS, heading control system)을 통합함으로써, 선수방위 제어뿐만 아니라 항로 계획에 따라 자동적인 항적 제어를 한다. TCS는 항해사 개입 없이 목적 경유지를 제어하는 핵심 시스템으로써, 본 논문에서 제안하는 모델링 방법의 구현을 위해 적용하며, 그 개념을 설명하기 위해 한 가지 예를 Fig 2.7에 나타내었다.

Table 2.1 List of major navigation and communication systems

Equipment		Description
TCS	ECDIS	Display navigation chart informations on electronic chart including voyage planning and monitoring function
	HCS	Control ship heading
	Speed Log	Sensing current ship speed
	GPS	Calculate current ship position
	Gyro Compass	Sensing current ship heading
	BNWAS	Early detection of neglect of duty
AIS		Identification own and other ships' static and dynamic information
VDR		Record generated voyage data, images and voice
Radar		Identification target and detect target's speed etc.

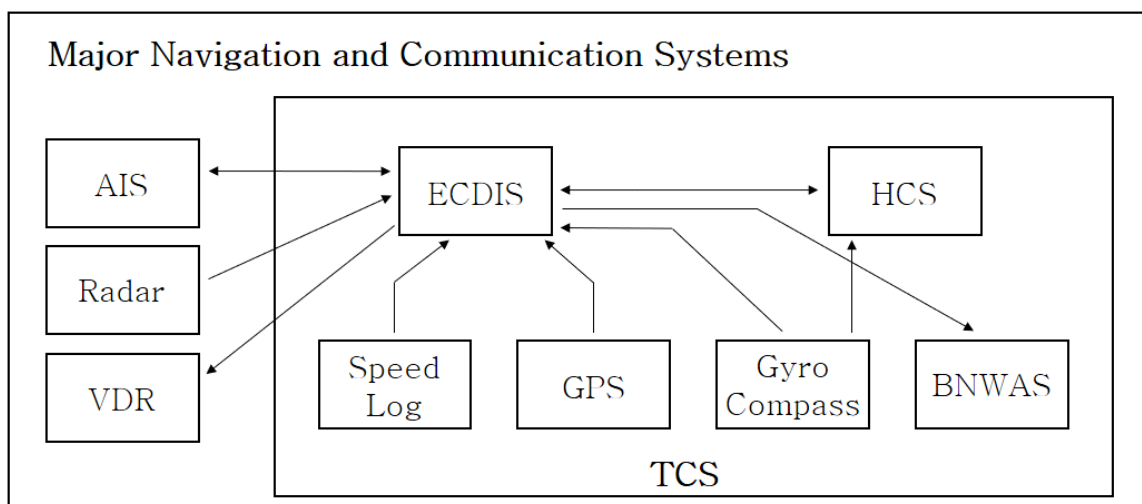


Fig. 2.6 A structure of core navigation and communication systems

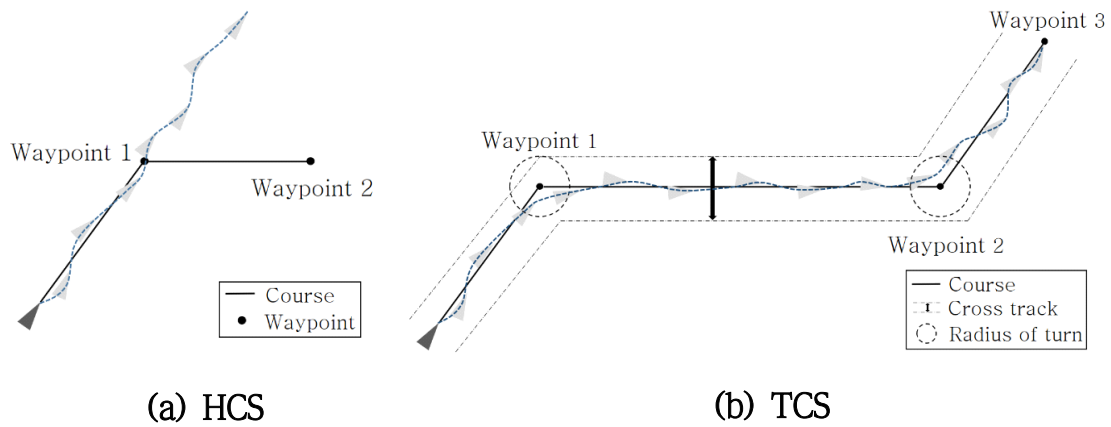


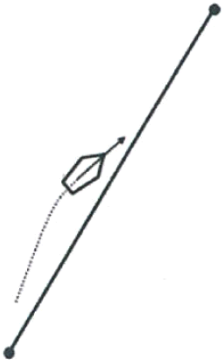
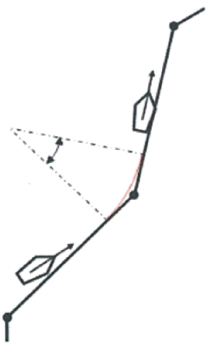
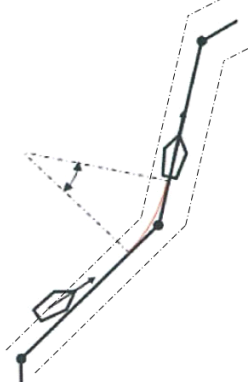
Fig. 2.7 Concept of HCS and TCS

Fig 2.7의 (a)는 출발지로부터 경유지(waypoint) 1을 거쳐 경유지 2로 향하는 항로 계획에 HCS가 적용한 예를 나타낸다. HCS는 설정된 선수방위에 따라 타(rudder)를 조종하여 침로(course)를 유지하는 시스템이다. 항해사의 개입이 없다고 가정하였을 때, (a)의 선박은 약 30°의 선수방위가 설정되었기 때문에 경유지 1을 지나도 기존의 침로를 유지하여 항로 계획을 벗어날 것이다. (b)는 앞서 설명한 TCS가 적용된 항해의 예를 나타내었다. (a)와 유사하게 HCS가 다음 경유지(TO-waypoint, Waypoint 1)까지의 약 30°의 선수방위를 유지하다가, ECDIS에 의해서 선회 반경(radius of turn) 근접이 감지되면 NEXT-경유지로(NEXT-Waypoint, Waypoint 2) 항해하기 위해 반경 내에서 선회하도록 HCS에게 침로 변경을 명령하여 항적을 제어한다. 또한, 교차항적(cross track) 한계를 벗어날 경우 한계 내로 재진입을 위해 HCS는 ECDIS로부터 침로 변경 명령을 받아 선수방위를 제어한다.

TCS의 작동 및 성능요건을 정의한 국제 표준인 IEC 62065 Edition 2.0[11]에 따르면 기능 요구사항에 따라 Table 2.2과 같이 Category A, B, C로 나뉜다. 본 논문에서는 완전한 항해사 미개입을 위해 Category C를 목적으로 하며, 이는 다음의 조건을 따른다. 첫째, 전체 항로에 항적 제어

기능이 요구되어야 한다. 둘째, 항로 계획에 따른 항로를 따라가게 항적을 제어하여야 한다. 셋째, 모든 동작이 자동으로 이루어져야 한다.

Table 2.2 Requirements for TCS according to Category

	Category A	Category B	Category C
Example			
Description	<ul style="list-style-type: none"> ▪ Required leg control between two waypoints ▪ User manual or auto to change heading ▪ Applied straight section 	<ul style="list-style-type: none"> ▪ Required leg control between two waypoints ▪ Changing heading section is controlled only by heading 	<ul style="list-style-type: none"> ▪ Required track control of the overall route plan ▪ Track control to follow route plan ▪ Performed automatically

제안하는 모델링 방법이 Category C를 만족하기 위해 다음의 기술이 요구된다. 첫째, TO-waypoint까지 침로 유지를 위한 선수 방위 제어 구현, 둘째, 항적 제어를 위해 선회 반경 내의 곡선향적 구현, 셋째, 교차항적 한계를 넘어서 항적으로부터 벗어날 경우 재진입을 위한 기술이다.

이러한 기능 요구사항을 충족하기 위해 실시간으로 수신되는 선박의 위치, 속도, 선수방위 등의 정보를 이용하여 적절한 선수방위를 제어한다.

2.2.2 Hasegawa의 충돌위험도 평가 모델

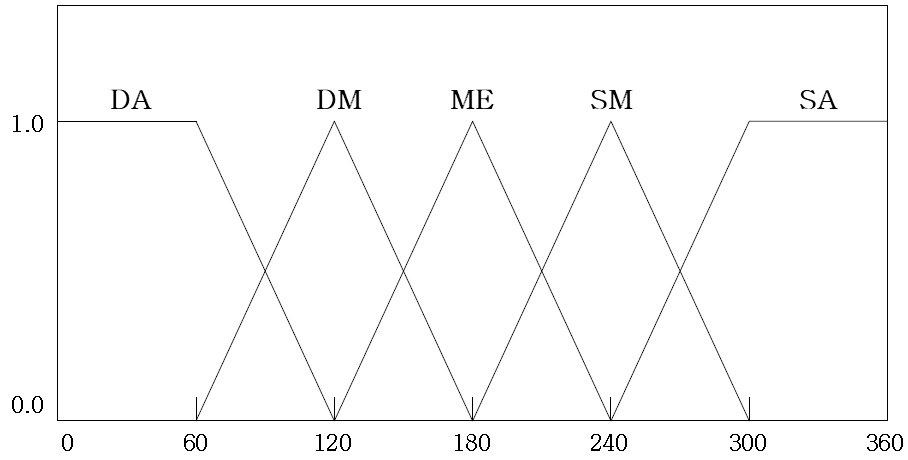
해상에서는 선박 간 충돌을 예방하기 위해 조우 상황을 구체적인 상태로 나누고 있으며, 각 상태에서의 피항선 및 유지선의 동작을 정의한다. 이러한 동작을 결정하는 시점 판단은 선박 충돌을 예방하기 위해 거쳐야 할 중요한 절차이며, 충돌위험에 대한 항해사의 의사결정을 지원하기 위한 연구[12-14]가 과거부터 다양한 방법으로 진행되고 있다.

최근 활발히 진행되고 있는 선박 충돌에 대한 위험 평가 모델 연구로는 fuzzy 이론을 활용한 알고리즘에 관한 것이다. 연구[15]는 본 선박을 기준으로 대상 선박의 충돌위험도(CR, collision risk) 추정을 위해 최근접 거리(DCPA, distance to closest point of approach)와 시간(TCPA, time to closest point of approach)을 활용하고 퍼지 추론을 적용하였다. 또한, 최근에는 선박 항해통신장비 기술의 발달로 인해 기존 퍼지 이론과 선박 자동식별장치(AIS)의 데이터를 이용한 연구[16]와 뉴로-퍼지 기반 충돌위험도 추정 알고리즘 연구[3]로 확장되고 있다.

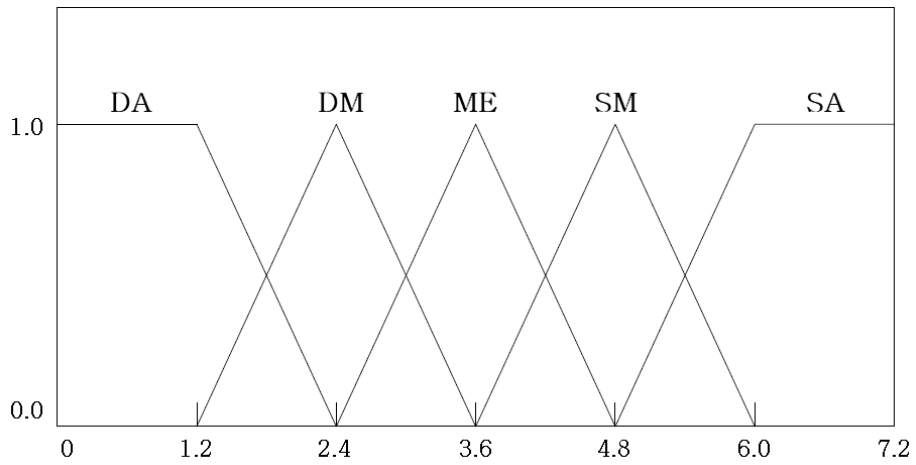
Hasegawa의 충돌위험도 평가 모델은 퍼지 이론을 접목하여 충돌위험 판단의 모호한 부분과 사람이 받아들이기 쉬운 언어적 표현으로 인해 선박 간 충돌 예방에 많은 활용이 되고 있다. 최초 Hasegawa는 SAFES(ship auto-navigation fuzzy system) 소개[17]를 시작으로 계속된 연구를 통해 평가 모델을 개선하고 있다[18].

본 논문에서는 항해 시뮬레이션 중 발생하는 선박 간 충돌 예방과 회피 시점 판단을 위해 Hasegawa 방식을 사용하며, 이는 Fig 2.8과 같은 퍼지 소속함수와 Table 2.3과 같은 충돌위험도 추정을 위한 퍼지 규칙을 기반으로 한다. Fig 2.8의 각 퍼지 소속함수에 대한 설명은 다음과 같다.

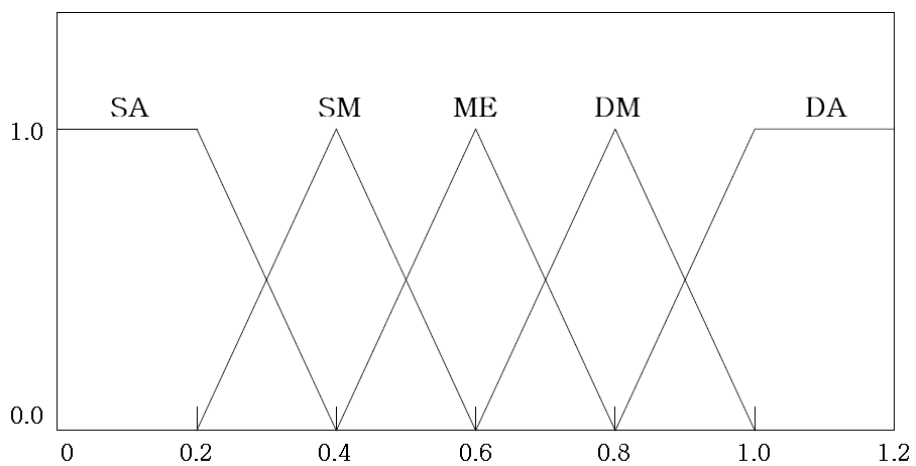
- (a) 초(sec) 단위의 TCPA 값의 위험정도에 대한 퍼지 소속함수
- (b) 해상마일(nm) 단위의 DCPA 값의 위험정도에 대한 퍼지 소속함수
- (c) 충돌위험도에 대한 퍼지 소속함수



(a) A membership function for TCPA(s)



(b) A membership function for DCPA(nm)



(c) A membership function for Collision-Risk

Fig. 2.8 Membership functions for ship Collision-Risk (by Hasegawa)

Table 2.3 Fuzzy rules for ship Collision-Risk (by Hasegawa)

Collision-Risk		TCPA				
		DA	DM	ME	SM	SA
DCPA	DA	DA	DM	ME	SM	SA
	DM	DM	ME	SM	SA	SA
	ME	ME	SM	SA	SA	SA
	SM	SM	SA	SA	SA	SA
	SA	SA	SA	SA	SA	SA

충돌위험정도는 안전부터 위험까지 총 다섯 단계인 SA, SM, ME, DM, DA로 표현하며, 선박 간 상대방위 및 속도에 따라 TCPA와 DCPA가 길수록 그 정도는 안전에 가까워지며, 짧을수록 위험에 가까워진다.

예를 들어, $t=0$ 시점에서 조우 중인 두 선박의 침로와 속도를 고려하여 TCPA가 80초이고, DCPA가 1.8 nm로 계산되었다고 가정한다. 이 두 가지 변수를 이용하여 Hasegawa 퍼지 소속함수에 대입하면, 최근접시간의 위험정도는 (DA:0.7, DM:0.3, ME:0.0, SM:0.0, SA:0.0)이고, 최근접거리의 위험정도는 (DA:0.5, DM:0.5, ME:0.0, SM:0.0, SA:0.0)이다. 계산된 위험정도는 DA와 DM에 다소 높게 추정되며, 어느 정도의 위험이 있다고 판단할 수 있지만 수치적인 계산은 필수적이다.

이를 위해, 맘다니형 추론(Mamdani-style fuzzy interface) 방식으로 추론한다. 퍼지규칙에 따른 충돌위험정도는 (DA:0.5, DM:0.5, ME:0.3, SM:0.0, SA:0.0)이며, 이를 역퍼지화(defuzzification) 하면 충돌위험도는 0.83이다. 충돌위험도의 임계값이 0.6으로 설정되어 있다고 가정하면, 수치화된 결과는 임계 값을 초과하였기 때문에 COLREGs 항법 관계에 따라 $t=0$ 시점에서 초동대응을 수행하여 사고를 예방할 수 있다.

2.2.3 COLREGs 개요

국제해상충돌예방규칙(COLREGs, international regulations for preventing collisions at sea)[9]은 선박 간 충돌을 예방하기 위한 규칙으로 해양 및 모든 수역에 있는 선박 및 수상항공기는 본 규칙의 규정을 이행하여야 한다. COLREGs 제2장 항법 규정 중 충돌위험성을 파악하기 위해 안전속력 유지 및 적합한 경계를 행할 것을 규정하고 있으며, 위험성 판단에 있어서 적절한 항해 장비를 활용하여 상대 선박의 방위 변화에 대한 검토를 규정하고 있다.

선박 항해 중 충돌위험 상황 발생 시 항해사는 본 선박의 피항 또는 유지에 대한 여부를 판단하고, 이는 COLREGs에 따라 타선박이 본 선박을 바라보는 상대방위를 기준으로 결정한다. 이를 위해 COLREGs에서는 선박 간 조우상태에 대해서는 구체적으로 정면으로 마주치는 상태(Head-on), 횡단상태(Crossing), 추월상태(Overtaking)의 세 가지 항법 관계로 정의하며, 자세한 설명은 다음과 같다.

- Head-on: 정면으로 마주치는 상태는 타선박의 마스트 정부등(masthead light)과 현등(side light)을 좌우 (5 ~ 6)° 범위 내에서 바라보는 경우 해당하며, 타선박의 좌현을 바라보면서 통과하도록 각기 우현으로 변침하여야 한다(Fig 2.9 참조).
- Crossing: 횡단상태는 서로 다른 선박이 교차하며 항해하는 경우이며, 타선박의 좌현 즉, 홍등(red light)을 바라보는 선박이 우현으로 변침하여 타선박의 전방을 횡단하지 않도록 하여야 한다(Fig 2.9 참조).
- Overtaking: 추월상태는 타선박의 선미등(stern light)만 볼 수 있고 현등(side light)은 볼 수 없는 경우 해당한다. 추월선은 피추월선의 진로를 방해하지 않는 범위 내에서 우현으로 변침하여 완전히 앞질러 갈 때까지 추월선으로써의 의무를 다하여야 한다(Fig 2.9 참조).

Fig 2.9의 (a)는 선박 항해등의 규칙을 나타낸 것으로서, 항해등만으로 선종의 구별, 선박의 항해 또는 정박 여부, 선수방위 등의 다양한 정보를 제공한다. 이처럼 항해등은 선박의 존재와 현재 상태를 타선박에게 알려 줌으로써, 항해사에게 항법 관계 파악, 피항 또는 유지에 대한 판단 등을 돕는 중요한 역할을 담당한다. 항해등은 마스트 정부등(masthead light), 현등(side light), 선미등(stern light)의 세 가지로 구성된다.

- Masthead light: 선박의 마스트에 설치한 백색등(white light)으로 선수 방향을 중심으로 좌, 우 각기 112.5°로 총 225°의 부채꼴 모양의 각을 갖는 형태이며, 원거리에서 항해 중인 선박의 존재를 조기에 알려주고 Head-on 상태에 대한 정보를 제공하는 역할을 가진다.
- Side light: 선박의 좌현을 홍등(red light), 우현을 녹등(green light)으로 표시하는 등화로 선수 방향에서 각 현 쪽으로 112.5°의 각을 갖는 형태로 타선박이 본 선박의 어느 현을 바라보는지를 알려주는 역할을 가진다.
- Stern light: 선박의 후미에 설치되는 백색등으로 선미를 중심으로 135°의 각을 갖는 형태로 추월당하는 선박이 추월하고자 하는 타선박에게 존재를 알려주는 역할을 가진다.

Fig 2.8의 (b)에 앞서 설명한 세 가지(head-on, crossing, overtaking)의 항법 관계를 유형별(Case I, II, III)로 나타낸다. 실제 항해 중에는 타선박의 항해등을 보고 COLREGs에 따른 대응이 가능하지만, 시뮬레이션 환경에서는 이에 대한 대처 방안이 필요하다. 따라서 제안하는 모델링 방법의 시뮬레이션 환경에서는 타선박이 본 선박을 바라보는 각(θ)을 도출하여 어떤 항해등을 바라보고 있는지를 판단함으로써, 본 선박의 유지 또는 피항에 대한 의무를 결정한다.

Case I 은 타선박이 본 선박의 좌현 또는 우현에 위치하는 두 가지의 경우로 나뉘며, 타선박의 어느 현을 바라보는지를 구분하여 네 가지의 경우로 구성된다. 또한 Case II와 III의 두 가지 경우로 총 6가지의 경우로 구분할 수 있으며, 이는 본 논문에서 제안하는 가상 항해 요소인 항해사 에이전트의 지식베이스 규칙으로 적용한다.

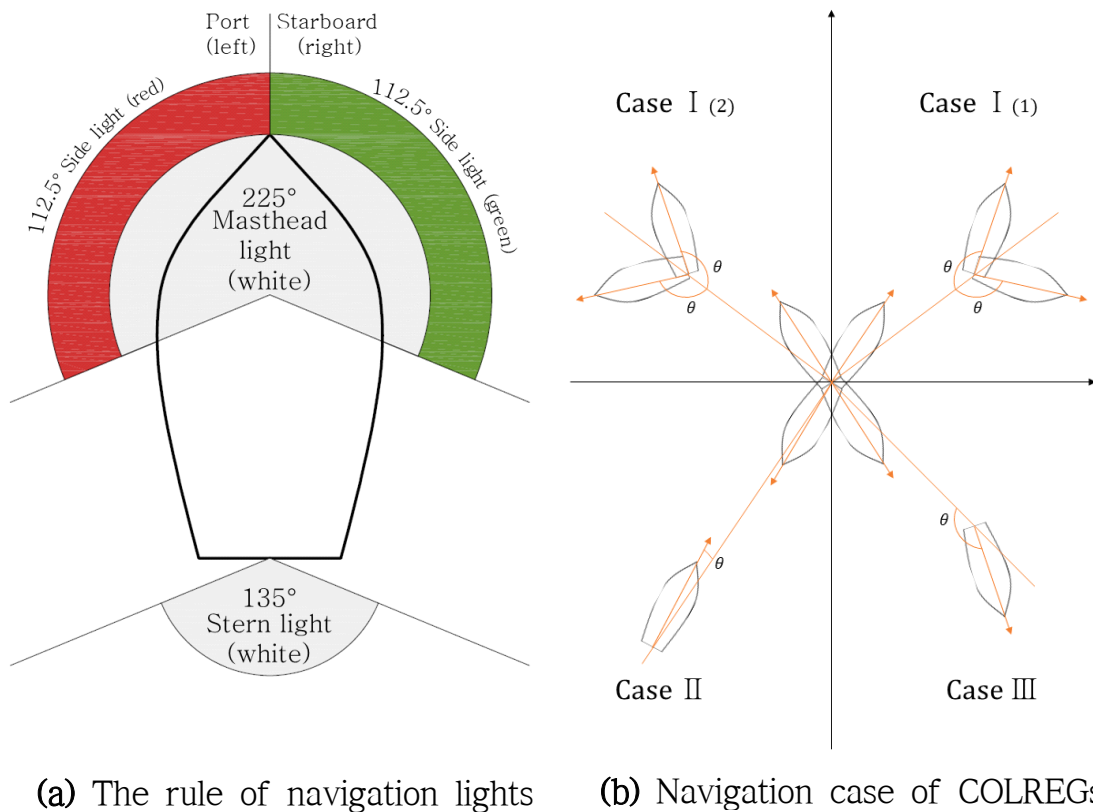


Fig. 2.9 The rule of navigation lights and navigation cases of COLREGs

제 3 장 항해 시뮬레이션을 위한 모델링

본 장에서는 선박 항해를 위한 핵심 선박장비 동작 체계와 선박 충돌 및 위험 상황 조치를 위한 에이전트가 포함된 항해 모델링 및 시뮬레이션 환경을 논한다. 제안된 모델링 및 시뮬레이션 환경은 확장성, 재사용성, 유지 보수성 등의 소프트웨어 공학적 장점을 가진 객체지향 환경인 .NET Framework 기반 C# 언어를 기반으로 한다.

구축된 모델링 및 시뮬레이션 환경을 통해 항해 계획에 따른 다양한 시나리오 등을 시뮬레이션할 수 있으며, 이는 사용자 인터페이스를 통해 선박의 조종, 항로 계획 등이 표시된다. 또한, 선박 충돌위험 상황 회피를 위한 항법 관계를 교육할 수 있을 것으로 기대되며, 시나리오에서 발생하는 시뮬레이션 분석 결과를 활용하여 각종 항해 디스플레이 장치 및 항해 안전 지원 서비스 등의 유용성 분석을 위해서도 충분히 활용할 것으로 판단된다.

따라서, 본 논문에서 제안하는 항해 시뮬레이션을 위한 모델링 방법은 실제 선박에 탑재되는 핵심 선박장비와 항해사와 조타수 의무를 수행하는 에이전트를 DEVS 형식론 기반 SES/MB 프레임워크[7-8]로 모델링함으로써, 높은 성능과 정확도를 지원한다. 또한, 모든 동작이 자동으로 이루어지기 때문에 선박 운항 관련 지식이 없는 서비스 연구자들을 위한 적절한 도구일 것으로 판단되며, 항해와 관련하여 현행 연구되는 다양한 결과의 분석 및 유용성 검증을 위한 쉬운 접근성 가지므로, 그 가치는 클 것으로 기대된다.

3.1 SES/MB를 이용한 항해 시뮬레이션 시스템 모델링

본 절에서는 앞서 설명한 SES/MB를 이용하여 항해 시뮬레이션을 위한 핵심 선박장비와 가상 항해사 및 조타수 모델들의 특성과 구조적 관계를 모델링한다. 본 절에서는 각 모델의 관계와 표현을 위해 설계한 SES/MB를 이용하여 설명한다.

Fig 3.1은 항해 시뮬레이션 시스템의 SES를 도식화하였다. 항해 시뮬레이션 시스템은 크게 항해공간요소(navigation SPACE element)[19]와 선박항해요소(ship navigation elements) 두 가지로 나뉘며, 선박항해요소는 인적요소(human elements)와 선박요소(ship element)로 나뉜다.

항해공간요소는 항해 환경에 관한 부분을 처리하는 것으로서, 선박의 센서 데이터를 수집 및 갱신하는 로거(logger), 새로운 좌표 및 선수방위를 계산하는 공간예측기(SEP, spatial encounter predictor) 그리고 갱신된 선박의 정보를 2D 지도에 업데이트하는 GIS 등으로 이루어져 있다.

선박항해요소의 인적 요소는 기본적으로 인간과 에이전트로 나뉘며, 항해를 위한 인적요소인 항해사(officer)와 조타수(helmsman)를 선택할 수 있는 분류 속성을 가진다.

선박항해요소의 선박요소는 전자적인 통신을 담당하는 항해통신장비(navigation and communication systems)와 선박의 추진과 조향을 담당하는 추진조향장비(propulsion and steering systems)로 나뉜다.

첫째, 항해통신시스템은 항로계획, 항적제어 등을 수행하는 항적제어시스템(TCS, track control system)을 가지며, 이는 HCS와 ECDIS로 구성된다. ECDIS의 충돌위험도 평가 모델은 Hasegawa[17], Imazu[20], Fujii[21] 등의 모델 중 선택할 수 있는 분류 속성을 가진다. 또한, 선박 및 주변 환경을 감지하는 환경감지시스템(environment sensing system), 타선박의 위치, 속도, 선수방위 등을 식별하고 본 선박의 정보를 송신하는 선박식별시스템(ship identification system), 물표 탐지를 위한 물표감지시스템(target

identification system)과 타선박과의 음성 통신을 위한 통신시스템 (communication system)으로 구성된다.

둘째, 추진조향장비는 선박의 조종을 제어하는 타(rudder)와 엔진 (engine)으로 구성된다.

Fig 3.1과 같이 다양한 경우 구성이 가능한 구조체로부터 하나의 항해 구조를 선택하기 위하여 가지치기 과정을 수행한다. 그 최종적인 결과로 나타난 구조의 예는 Fig 3.2과 같은 PES로 표현된다. SES로부터 선택된 요소는 다음과 같다.

- Ship Navigation Elements : 본 선박과 대상 선박을 포함하여 총 네 대의 선박을 선택한다.
- Human Elements : 충돌위험 상황에 대한 자율적 회피를 위해 COLREGs 기반 항해사 에이전트와 그 명령을 따르는 조타수 에이전트를 선택한다.
- Navigation and Comm. Systems : 본 논문에서 제안하는 자율적인 항해 시물레이션을 위한 최소 요소인 Track Control System(TCS), Environment Sensing Systems(ESS), Ship Identification System(SIS)를 선택한다.

TCS의 구성 요소인 ECDIS는 조우하는 선박 간 충돌위험의 정도를 추정하는 평가 모델을 가진다. 본 PES에서는 앞서 설명한 최근 충돌 예방에 많은 활용이 되는 Hasegawa[17] 방식을 선택한다. ESS에서 GNSS는 GPS 그리고 SIS에서는 300톤급 이상 선박 필수탑재 장비인 Class A AIS를 선택한다.

선택된 시나리오의 예는 본 선박을 포함하여 총 네 대의 선박이 선택되었고 인적요소로 모두 에이전트가 선택되어 선박 충돌위험 상황에 따른 회피가 요구될 때 각 상황에 따른 적절한 지식기반 회피를 수행한다. 이는 사용자의 개입이 필요 없이 본 선박뿐만 아니라 대상 선박까지 신뢰성 있는 회피 동작을 보장함으로써 실제 항해사에 의해 발생하는 항해 데이터를 유사하게 모사할 수 있을 것으로 판단된다.

Fig 3.2의 시뮬레이션의 모델 구조는 Fig 3.3과 같으며, 이는 SES에 정의된 단말 모델들의 연결 관계에 의해 결합함에 따라 최종적인 시뮬레이션 모델 구조를 구성하게 되며, 각 모델의 상태 전이에 대한 동적인 변화와 동작에 대한 기술을 위하여 3.2장에서 DEVS 형식론 기반 핵심 선박장비를 설계한다.

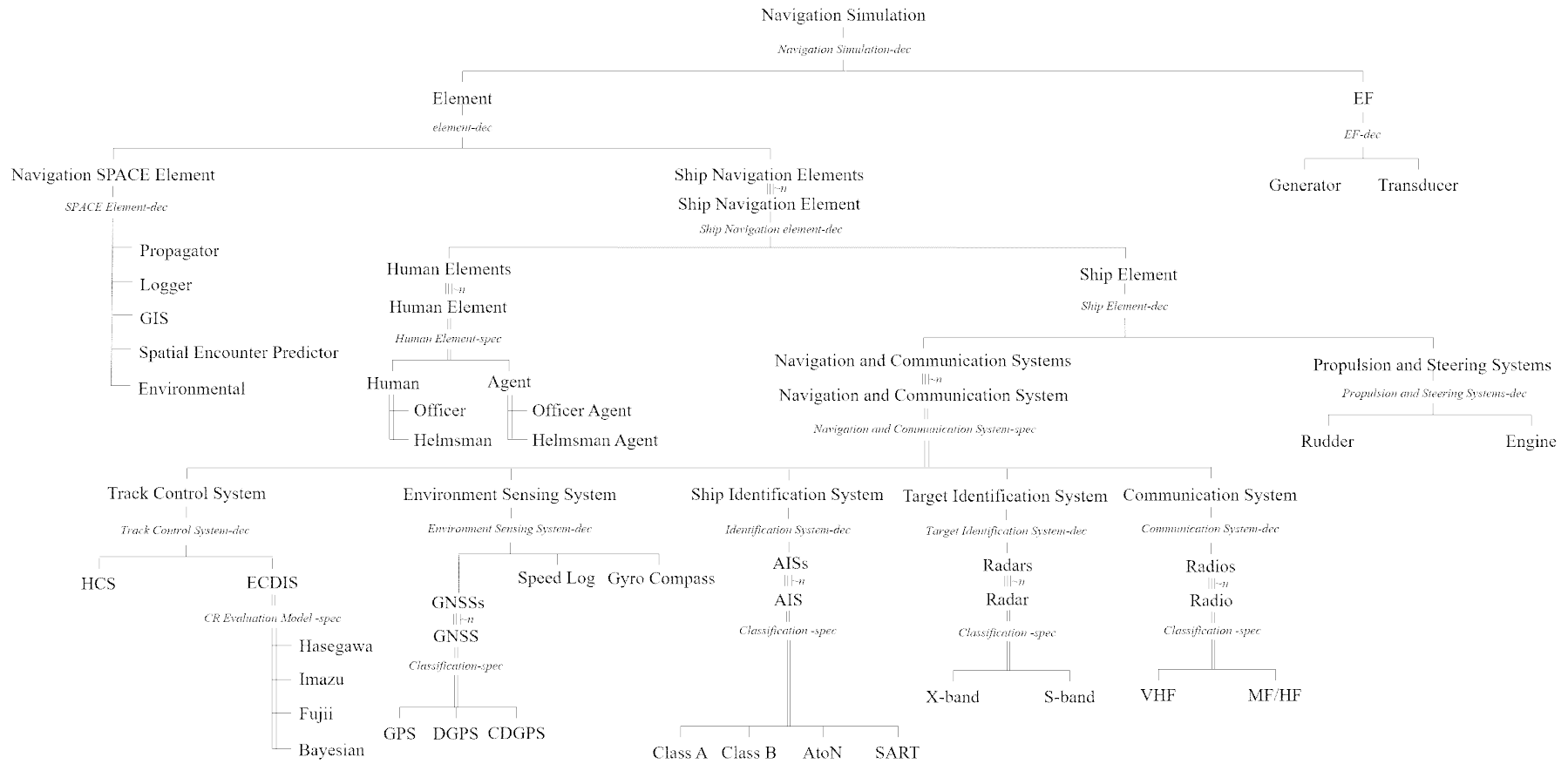


Fig. 3.1 A SES of navigation simulation systems

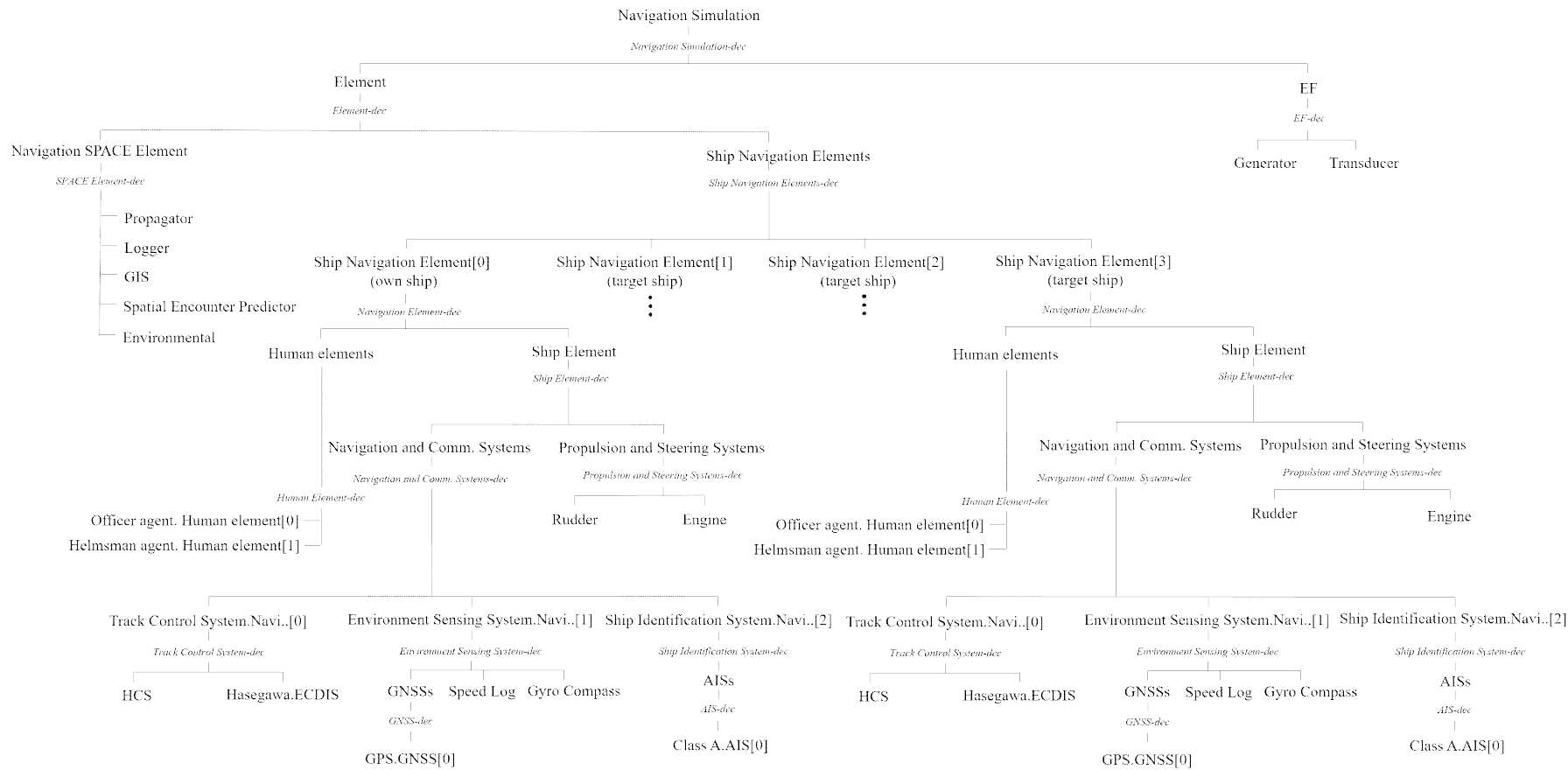


Fig. 3.2 An example of the PES

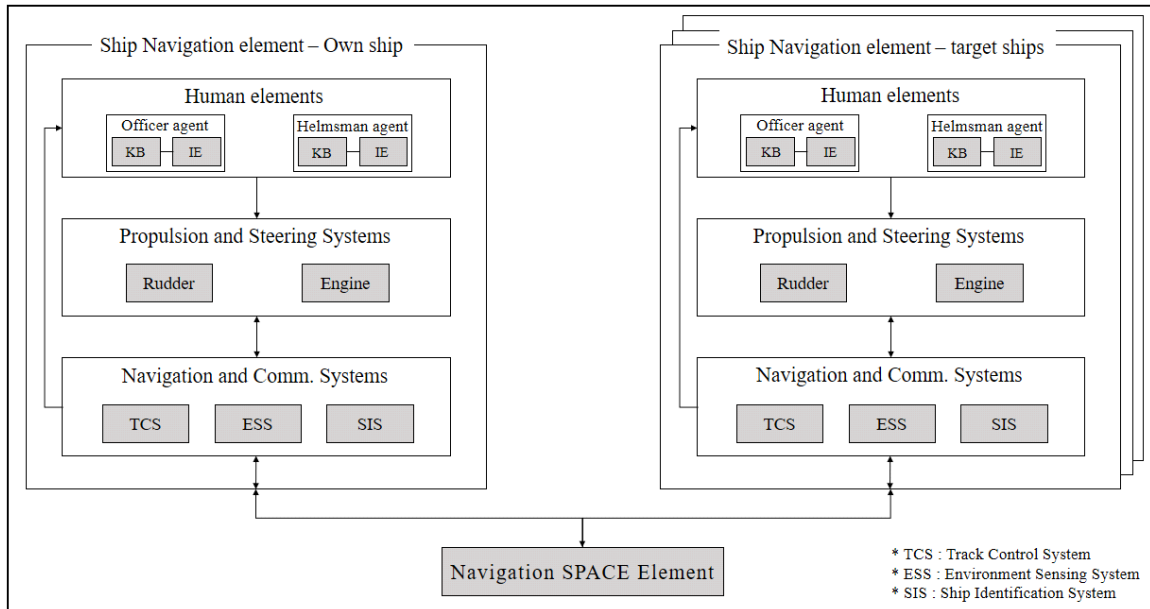


Fig. 3.3 A simulation model structure of Fig. 3.2

3.2 DEVS 형식론 기반 핵심 선박장비 모델 설계

앞선 SES/MB를 이용한 항해 시뮬레이션 시스템 모델링과 충돌위험 상황에 대한 자율적인 회피를 위한 에이전트를 모델링함으로써 항해 시뮬레이션 시스템의 구조를 나타내었다. 핵심 선박장비 모델의 동작을 표현하기 위해 DEVS 형식론을 기반으로 명세한다.

본 논문에서는 Fig 3.1의 SES를 가지치기하여 생성된 PES인 Fig 3.2를 하나의 시뮬레이션 구조로 가정하여 DEVS 기반의 핵심 선박장비 모델을 설계하고, 제4장에서 그 결과를 COLREGs의 항법 관계 기반으로 검증한다.

선택된 시뮬레이션 구조의 시뮬레이션을 위한 모델 구조도를 Fig 3.4에 도식화하였다. 시뮬레이션 모델은 크게 본 선박(own ship)과 대상 선박(target ship)의 항해 요소와 항해 공간모델로 구성된다. 구성되는 모델들의 세부 설명은 다음과 같다.

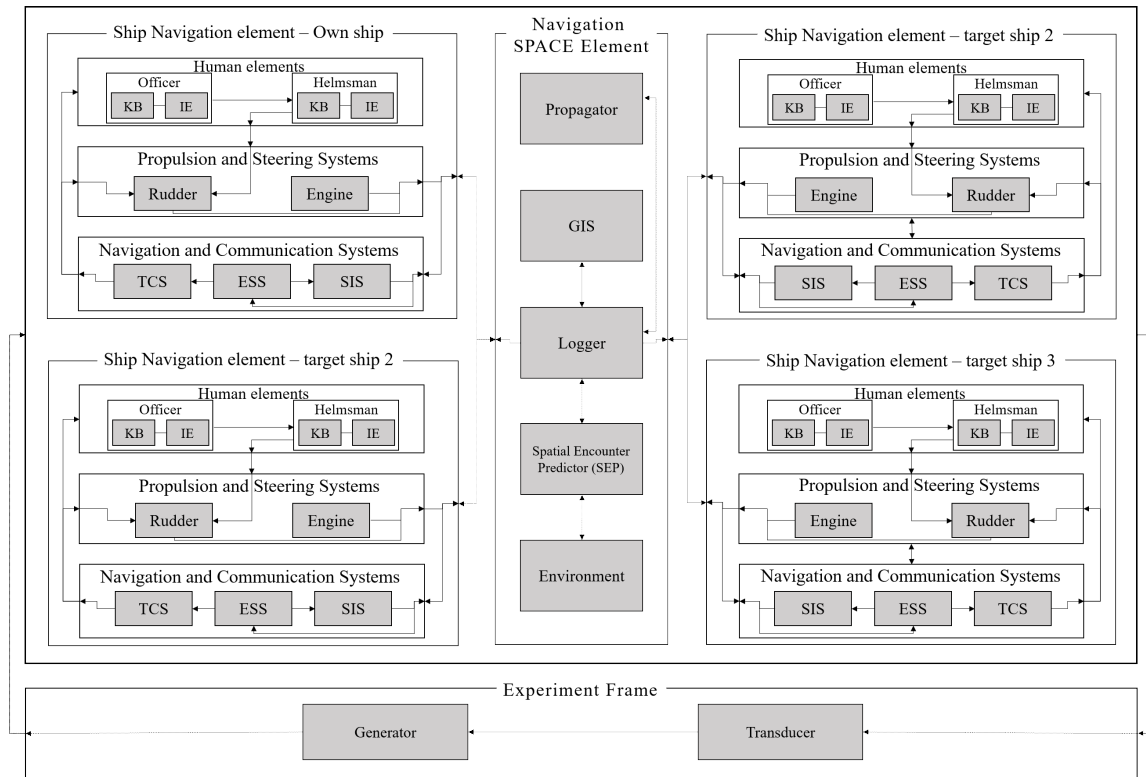


Fig. 3.4 Simulation model structure of navigation

1) Experiment Frame Model

- Generator Model: 시뮬레이션 시스템에서 초기 설정된 선박의 정보 (위도, 경도, 속도, 선수방위, 타각)를 입력으로 가하여 시뮬레이션 시작을 제어하는 모델이다.
- Transducer Model: 시뮬레이션의 분석을 위해 생성되는 정보를 수신 및 데이터베이스에 저장하며, 종료를 제어하는 모델이다.

2) Navigation SPACE Model

- Logger Model: 각 선박의 센서에서 감지되는 정보(위도, 경도, 속도, 선수방위), 추진조향장비의 정보(타각, 엔진 값), 선박식별정보(AIS)를 수신하여 갱신하고 1초 후의 시뮬레이션 선박 위치 계산을 위해 SEP 모델로 정보를 전달하는 모델이다.

- Spatial Encounter Predictor(SEP) Model: 각 선박 모델들의 위치정보 및 타각 변경에 따른 선수방위 변화를 계산하고 Logger 모델로 전달한다.
- GIS Model: 항해 환경 정보를 2D 지도에 갱신하는 모델이다.
- Propagator Model: 각 선박 모델들의 AIS 정보를 수신하여 영향 범위 내에 있는 다른 선박에 전달하는 모델로써, 본 시스템에서 영향 범위는 일반적인 VHF의 전파 도달 범위인 30 km를 고려한다.
- Environment Model: 항해 환경 날씨의 변화 및 시간적 변화를 설정하는 모델로서 파고와 풍속을 SEP 모델로 전달한다. 단, 본 논문에서는 항해상태 0을 고려하여 바람과 파도가 없는 잔잔한 바다의 상태를 적용한다.

3) Sensor Model

- GPS Model: 시뮬레이션 선박 모델의 GPS 정보(위도, 경도)를 감지하여 갱신 및 전달하는 모델이다.
- Gyro compass Model: 시뮬레이션 선박 모델의 선수방위를 감지하여 갱신 및 전달하는 모델이다.
- SpeedLog Model: 시뮬레이션 선박 모델의 속도를 감지하여 갱신하고 Logger 모델로 전달한다.

4) AIS 모델

- Class A AIS Model: 각 선박 모델들의 AIS 정보(위도, 경도, 속도, 선수방위 등)를 갱신하고 Logger 모델로 전달하며, Propagator 모델로부터 수신범위 내에 있는 타선박의 AIS 정보 수신하는 모델이다. 이때, 일반적인 VHF 전파 도달 거리인 30 km를 기준으로 한다.

5) TCS Model

- ECDIS Model: Sensors, AISs 모델로부터 정보(시뮬레이션 선박의 위치, 선수방위, 속도)를 수신하여 타선박과의 충돌위험도 계산, 교차항적 거리 계산, 곡선항적을 제어하는 모델이다. 또한, 충돌위험도가 설정 임계 값을 초과하면 항해사 에이전트에게 위험 대상 선박의 정보(위치, 선수방위, 속도)를 전달한다.
- HCS Model: To-Waypoint까지 선수방위를 제어하며, ECDIS 모델로부터 타각 제어 명령에 따라 동작하는 모델이며, 다음의 네 가지 모드를 가진다. 첫째, HCS 모드는 시뮬레이션 선박의 타(rudder)가 HCS에 의해서만 조종될 때이다. 둘째, Curved 모드는 NEXT-Waypoint로 가기 위한 곡선항적을 수행하며 변경 타각은 ECDIS로부터 수신하여 타를 조종한다. 셋째, Cross 모드는 선박이 침로에서 정해진 범위(교차항적 거리)를 벗어났을 때 재진입하기 위하여 ECDIS로부터 변경 타각을 수신하여 타를 조종한다. 넷째, Agent 모드는 조타수 에이전트에 의해서 타를 조종되며, HCS는 조종이 끝날 때까지 대기한다.

6) Propulsion and Steering Systems Model

- Rudder Model: 시뮬레이션 선박의 타각을 감지하여 갱신하고 Logger 모델로 송신하며, ECDIS 및 Agent 모델로부터의 타각 제어 명령에 따라 동작하는 모델이다.
- Engine Model: 시뮬레이션 선박의 엔진 값을 감지하여 갱신 및 Logger 모델로 송신하는 모델이다.

7) Agent Model

- Officer Model: ECDIS 모델로부터 받은 충돌위험 정보 데이터를 타선박과의 항법 관계 판단을 위해 COLREGs 기반으로 분석하여

Helmsman 모델로 명령하는 모델이다.

- Helmsman Model: Officer 모델로부터 받은 명령에 따라 선박의 타각을 결정하여 Rudder 모델을 제어하는 모델이다.

항해 시뮬레이션 시스템의 각 모델 의사 코드는 부록 A의 Table A.1~A.7과 같다. 본 장에서의 항해 시뮬레이션을 위한 모델 설계를 통해 자율적인 항해 시뮬레이션을 위한 DEVS 기반 시뮬레이션 환경이 완성되었으며, 이를 제4장의 COLREGs 기반 항해 시뮬레이션 사례를 통해 검증할 것이다.

3.3 COLREGs 기반 에이전트 모델링

본 절에서는 앞서 설명한 SES에서 표현된 에이전트의 모델링에 대하여 논한다. 항해 시뮬레이션을 위해 인적요소는 인간 또는 에이전트가 선택될 수 있으며, 제안하는 자율적인 항해 시뮬레이션을 위해서는 충돌위험 상황을 대처하기 위해 선박에서의 명령 체계와 대처 방법에 대한 정의가 필요하다. 실제 선박에서 항해와 관련된 모든 결정을 내리는 항해사는 충돌위험이 있는 선박 간 조우 상황에 대해 COLREGs에서 정의하는 항법 관계에 따라 적절한 판단을 내린다. 결정된 판단은 타의 변경이며, 조타수에 의해 행하여지는 명령 체계를 가진다.

앞선 2.2.3절에서 COLREGs의 항법 관계는 정면으로 마주치는 상황(Head-On), 횡단하는 상태(Crossing), 추월하는 상태(Overtaking)의 세 가지로 나누어 설명하였다. 횡단하는 상태는 본선이 타선의 좌현 그리고 우현에 위치한 두 가지의 상태로 나뉜다. 따라서, 선박 항해 중 발생 가능한 충돌위험 상황은 Table 3.1과 같이 나타낼 수 있으며, 총 6가지의 경우를 가진다. 이를 기반으로 COLREGs의 항법 관계를 판단할 수 있으며, 이는 Table 3.2와 3.3과 같이 항해사와 조타수의 지식베이스로 나타낼 수 있다.

Table 3.1 Navigation cases of COLREGs for other ships

Navigation case of COLREGs					
Head-on	Crossing				Overtaking
Located on the bow	Located on the port		Located on the starboard		Located on the stern
	Heading at the starboard	Heading at the port	Heading at the starboard	Heading at the port	

Table 3.2 Knowledge base of the officer model

	Current state		Output
variable	Own ship		Command to Helmsman
	Position	Heading	
State value	Located on the port	Heading at the starboard	Maintain
		Heading at the port	Crossing (Fine, Broad, Converging)
	Located on the starboard	Heading at the starboard	Maintain
		Heading at the port	Crossing (Fine, Broad, Converging)
	Located on the bow	Heading at the bow	Head-On
	Located on the stern	Heading at the stern	Overtaking

Table 3.3 Knowledge base of the Helmsman model

	Current state	Input	Output
variable	Own ship	Officer command	Control rudder
	Rudder control state		
State value	Normal or Avoid	Maintain	rudder = 0 °
		Crossing - Fine	rudder = 15 °
		Crossing - Broad	rudder = 20 °
		Crossing - Converging	rudder = 30 °
		Head-On	rudder = 10 °
		Overtaking	rudder = 10 °

항해사 에이전트의 추론엔진 모델은 충돌위험에 대한 본 선박과 대상 선박의 위치정보와 선수방위를 수신하고 이 정보를 기반으로 COLREGs 항법 관계를 판단한다. 이 결과는 Table 3.2의 항해사 지식베이스로 요청되어 규칙에 적절한 출력을 조타수 에이전트로 명령한다. 항해사 에이전트의 명령은 침로 유지(Maintain), 횡단 상태(Crossing), 정면으로 마주치는 상태(Head-On), 추월하는 상태(Overtaking)로 크게 네 가지로 나뉜다. Crossing 상태는 두 선박이 교차하는 정도에 따라 변경할 타각은 상이해야 하며 Fine, Broad, Converging의 세 가지로 나뉜다. 이는, [22]의 연구에서 COLREGs 기반 다수의 항해 경험을 가진 항해사를 대상으로 설문한 신뢰성 있는 결과 자료이며, 본 논문에서는 에이전트에 의한 타각 변경의 정도 결정에 적용한다.

조타수 에이전트는 Table 3.3과 같은 지식베이스를 가지며, 항해사 에이전트로부터의 명령에 따라 적절한 타각을 설정하여 피항 또는 유지를 수행한다.

제 4 장 COLREGs 기반 항해 시뮬레이션

본 장에서는 COLREGs 기반 항해 시뮬레이션 사례 연구를 통하여 앞서 제안된 핵심 선박장비 및 에이전트 모델링 및 시뮬레이션 시스템에 대해 검증한다. 이를 위해 본 논문에서는 COLREGs에서 정의하는 항법 관계인 Head-On, Crossing, Overtaking 상태를 1:1 회피 상황으로, 그리고 다중 선박 충돌위험 상태를 1:3 회피 상황으로 테스트 및 검증을 수행하였다. 단, 네 가지로 나뉠 수 있는 Crossing 상태는 본선 피항과 유지의 두 가지로 검증하였다. 이를 통해, 제안하는 모델링 및 시뮬레이션 시스템이 다음을 검증한다. 첫째, 항해 시뮬레이션을 위한 실제 선박에 탑재되는 핵심 선박장비의 묘사가 가능하며, 둘째, COLREGs 기반 에이전트의 지식으로 충돌위험 상황 회피를 위한 사용자 개입이 없는 항해 시뮬레이션이 가능함을 검증한다.

4.1 시뮬레이션 시나리오 및 초기조건 설정


COLREGs 기반 항해 시뮬레이션을 위한 시나리오로써, 에이전트의 지식 기반으로 항법 관계의 피항 또는 유지를 판단하고 상황별 적절한 타의 조종으로 회피를 표현하였다. 시나리오를 위한 시뮬레이션 선박의 신뢰성 있는 동역학 움직임을 표현하기 위하여 속도(knots)와 타각(rudder)에 따른 선수방위 선회율(ROT, rate of turn)을 추정하여 선박의 운동에 적용하는 것은 필수적이다. 하지만, 선박의 조종 운동 방정식을 구하기 위해서는 선박 선형, 계수 등의 제원이 필요하나 연구의 목적으로 이러한 자료를 받는 것은 현실적인 어려움이 따른다.

본 논문에서는 이를 위해 (재)중소조선연구원 “선박ICT 융합장비 시험 인증센터”의 FMSS(full mission ship simulator)를 활용하였다. 이는 선박 제원에 따른 운동, 조종 등 각종 시험을 거친 결과 데이터를 이용해 선박의 동역학 움직임을 실제와 유사하게 모사하므로, 시뮬레이션 선박의 선수방위 선회율 도출을 위한 선회 시험(turning test)을 수행하기에 적절하다.

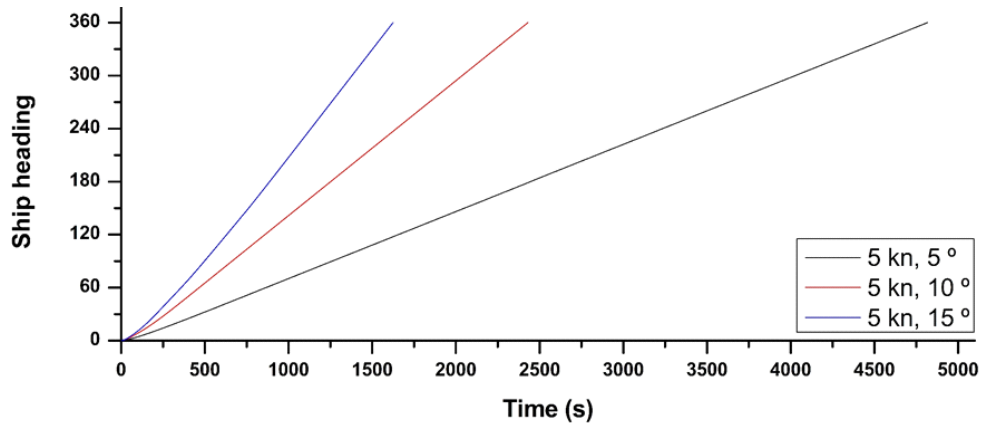
(1) 시뮬레이션 선박 설정 및 선회 시험 방법

시뮬레이션 선박은 “M/S Fantasy”이며 재원은 Table 4.1과 같다. 해상인명안전협약의 선박장비 강제 탑재 규정에 따라, 본 시뮬레이션 선박은 500톤급 이상의 여객선으로 본 논문에서 제안하는 항해 시뮬레이션의 구조 모델링과 핵심 선박장비의 구현을 위해 적절하다.

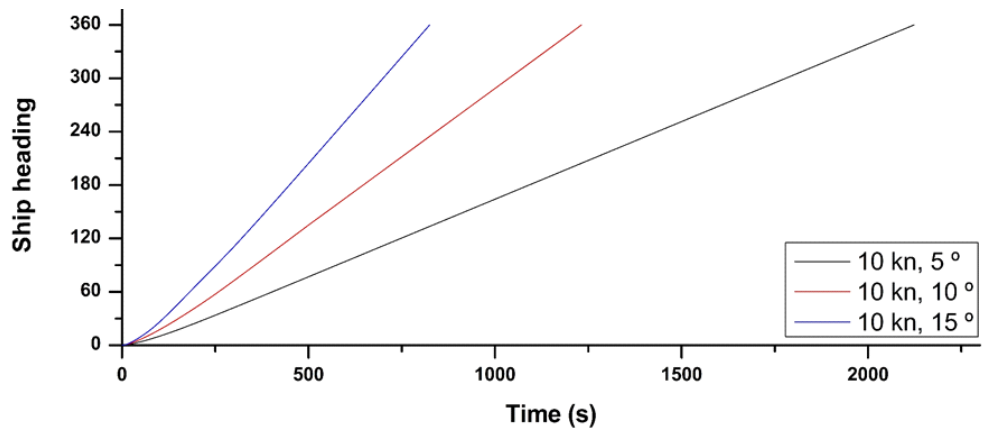
Table 4.1 General characteristics of the simulation ship

M/S Fantasy		
	Summary	
	Class and type	Cruise ferry
	Tonnage	75,027 GT (gross tonnage)
	Length	223.70 m
	Beam	35.00 m
	Draft	6.80 m
	Speed	22.0 knots (40.744 km/h)

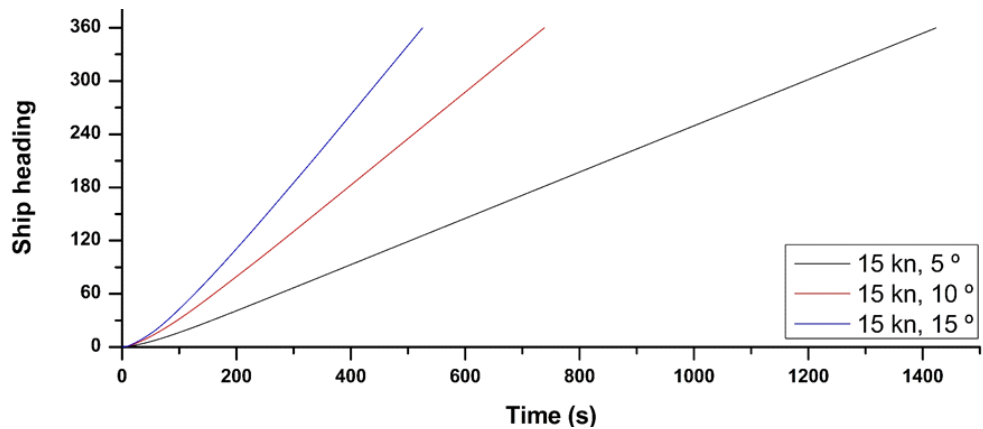
시뮬레이션 선박의 선회 시험은 총 9번을 수행하였고 각 시험은 속도와 타각에 따른 선수방위를 (0 ~ 360)°까지 기록하였으며, 그 결과를 Fig 4.1 과 같이 그래프로 나타내었다.



(a) Velocity: 5 kn, Rudder angle: 5°, 10°, 15°



(b) Velocity: 10 kn, Rudder angle: 5°, 10°, 15°



(c) Velocity: 15 kn, Rudder angle: 5°, 10°, 15°

Fig. 4.1 Turning test for the simulation ship

Fig 4.1의 (a)는 속도가 5 kn이고 타각이 5°, 10°, 15°일 때 선수방위의 변화를 그래프로 나타내었다. (b)와 (c) 또한 속도 10 kn, 15 kn일 때의 그래프를 나타내었으며, 속도가 빠르고 타각이 클수록 선수방위의 변화가 빨리 일어나는 것을 알 수 있다.

시험 결과를 Table 4.2에 정리하였으며, (a)는 각 시험 동안 선수방위 변화의 총 시간(total time)을 나타내었고 (b)는 각 시험의 선수방위 선회율(heading ROT)을 나타내었다. 선수방위 선회율은 초당 선수방위의 변화를 나타내는 것으로서, 360°를 각 시험의 총 시간으로 나누어 계산하였다. 시뮬레이션 중 실시간으로 변하는 속도와 타각에 따른 선수방위 선회율은 보간법을 적용하여 계산하였다.

Table 4.2 ROT calculation by the turning test

(a) Total time for each turning test

Total Time		Rudder angle (°)		
		5	10	15
Velocity (kn)	5	4817.2 sec	2430.1 sec	1624.2 sec
	10	2123.8 sec	1231.7 sec	824.9 sec
	15	1423.4 sec	738.4 sec	525.5 sec

(b) Heading ROT (rate of turn) for each turning test

Heading ROT		Rudder angle (°)		
		5	10	15
Velocity (kn)	5	0.0747 °/s	0.1481 °/s	0.2216 °/s
	10	0.1695 °/s	0.2922 °/s	0.4727 °/s
	15	0.2529 °/s	0.4875 °/s	0.6850 °/s

(2) 선회율 보간 방법

Table 4.2의 (b)는 해당 속도와 타각일 때 선수방위 선회율을 나타낸 것으로서, $f_{x,y} = w_1x + w_2y + b$ 과 같은 다변수 함수(multivariate function)로 나타낼 수 있다. 여기서 x 와 y 는 각각 속도와 타각이며, 시뮬레이션 중 실시간으로 수신되는 시간 t 의 값에 따라 선수방위 선회율, 즉 $f_{x,y}$ 을 도출하여 $t+1$ 의 위치를 계산하고 갱신하여야 한다.

이를 위해, 본 시뮬레이션 시스템에서는 다변수 함수에 대한 연속적인 2차원 보간법인 Newton의 전진차분 보간다항식(Newton's optimal difference interpolation polynomial)을 Fig 4.2와 같이 적용한다. Newton의 전진차분 보간다항식은 Fig 4.2의 (a)와 (b)같이 다음의 두 단계로 나누어 최종적인 값을 도출한다. 첫째, x 축을 고정으로 y 축 방향에 대한 보간을 수행한다. 둘째, 첫 번째 단계에서 보간된 값들을 이용하여 y 축을 고정으로 x 축 방향에 대한 보간을 수행하여 Fig 4.2의 (b)와 같은 최종적인 값을 도출한다.

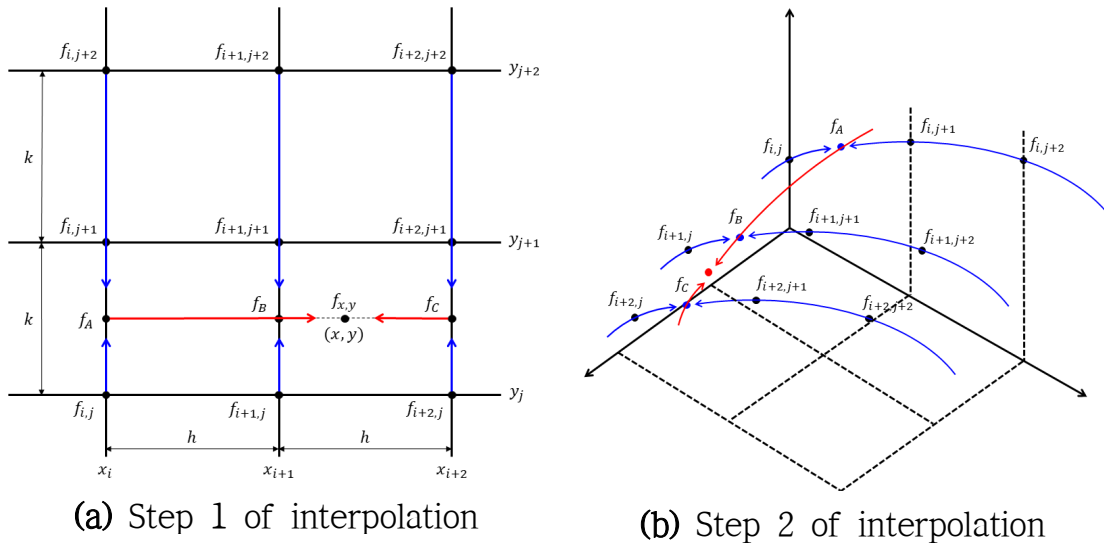


Fig. 4.2 Newton's optimal difference interpolation polynomial for multivariate function

- Step 1 : y 축 방향에 대한(x 고정) Newton의 전진차분 보간다항식

$$f_A = f_{i,j} + \frac{(y-y_j)}{k}(f_{i,j+1} - f_{i,j}) + \frac{(y-y_j)(y-y_{j+1})}{2k^2}(f_{i,j+2} - 2f_{i,j+1} + f_{i,j}) \quad (4.1)$$

$$f_B = f_{i+1,j} + \frac{(y-y_j)}{k}(f_{i+1,j+1} - f_{i+1,j}) + \frac{(y-y_j)(y-y_{j+1})}{2k^2}(f_{i+1,j+2} - 2f_{i+1,j+1} + f_{i+1,j}) \quad (4.2)$$

$$f_C = f_{i+2,j} + \frac{(y-y_j)}{k}(f_{i+2,j+1} - f_{i+2,j}) + \frac{(y-y_j)(y-y_{j+1})}{2k^2}(f_{i+2,j+2} - 2f_{i+2,j+1} + f_{i+2,j}) \quad (4.3)$$

- Step 2 : x 축 방향에 대한(y 고정) Newton의 전진차분 보간다항식

$$f_{x,y} = f_A + \frac{(x-x_i)}{k}(f_B - f_A) + \frac{(x-x_i)(x-x_{i+1})}{2h^2}(f_C - 2f_B + f_A) \quad (4.4)$$

(3) 검증을 위한 시나리오 설정(Case I - 1:1 충돌회피)

본 선박(OS, own ship)과 대상 선박(TS, target ship)의 항로 계획에 따른 침로 유지 및 에이전트에 의한 충돌회피 판단에 대한 수행 결과를 검증한다. 이를 위해, 시뮬레이션의 초기조건으로는 Fig 4.3에서 보는 바와 같이 1:1 상황에 대한 COLREGs의 항법 관계별로 구성하였으며, 세부 설명은 다음과 같다.

(a) Crossing(OS avoidance) : OS과 TS가 횡단하는 상태이며, OS가 TS의 좌현 즉, 홍등을 바라보고 있기 때문에 OS가 우현으로(좌현대 좌현) 피항 의무를 가진다(TS 유지, Fig 4.3 참조).

(b) Crossing(TS avoidance) : OS와 TS가 횡단하는 상태이며, TS가 OS의 좌현 즉, 홍등을 바라보고 있기 때문에 TS가 우현으로(좌현대 좌현) 피항 의무를 가진다(OS 유지, Fig 4.3 참조).

(c) Head-On(OS, TS avoidance) : OS와 TS가 정면으로 마주치는 상태이며, 모두 상대 선박의 선수 즉, 마스트 정부등을 바라보고 있기 때문에 OS와 TS 모두 우현으로(좌현대 좌현) 피항 의무를 가진다. (Fig 4.3 참조)

(d) Overtaking(OS avoidance) : OS는 15 kn로 항해, TS는 5 kn로 항해하여 추월하는 상태이며, OS는 TS의 선미 즉, 선미등을 바라보고 있기 때문에 OS가 우현 또는 좌현으로의 피항 의무를 가진다(TS 유지, Fig 4.3 참조).

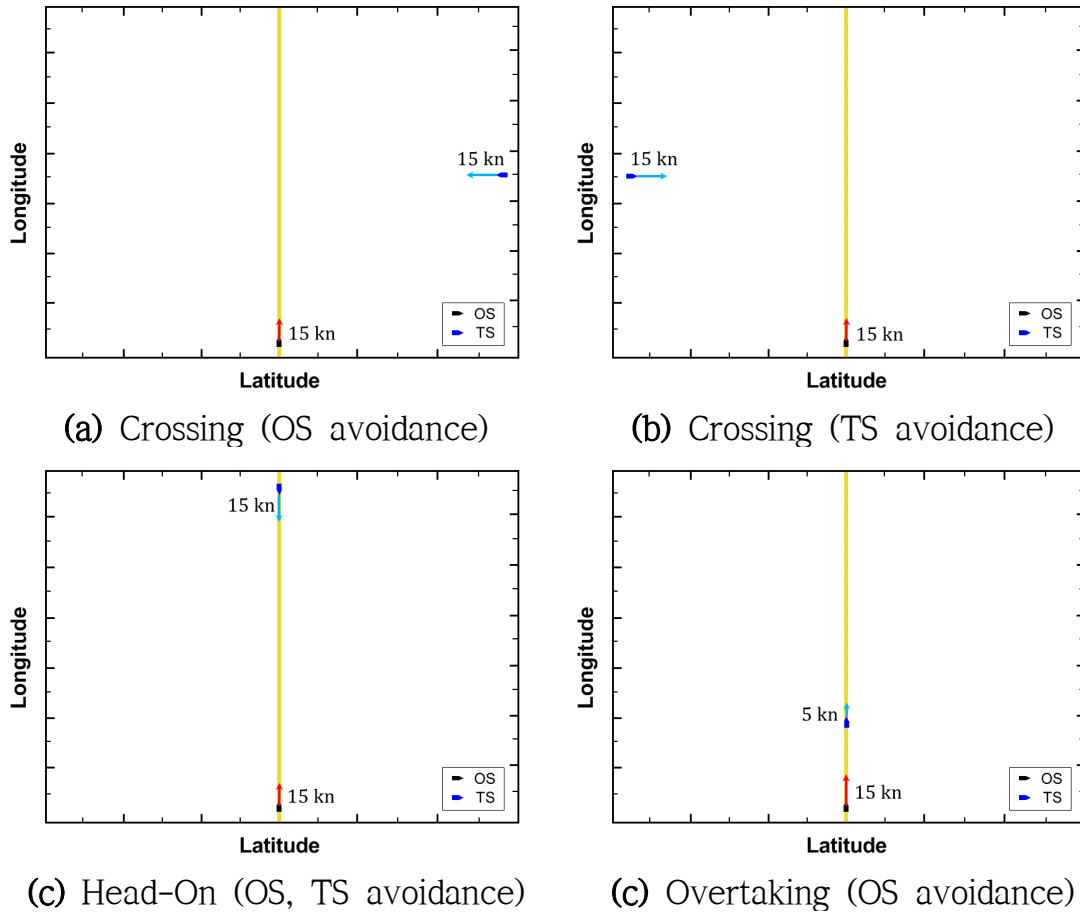


Fig. 4.3 The initial situation of COLREGs-based 1:1 navigation simulation

(4) 검증용 시나리오 설정(Case II - 1:3 충돌회피)

본 선박과 다중 대상 선박의 항로 계획에 따른 침로 유지 및 에이전트에 의한 충돌회피 판단에 대한 수행 결과를 검증하기 위하여 시뮬레이션의 초기조건으로는 Fig 4.4에서 보는 바와 같이 1:3 상황을 구성하여 COLREGs에 따른 다중 선박 충돌회피를 검증한다.

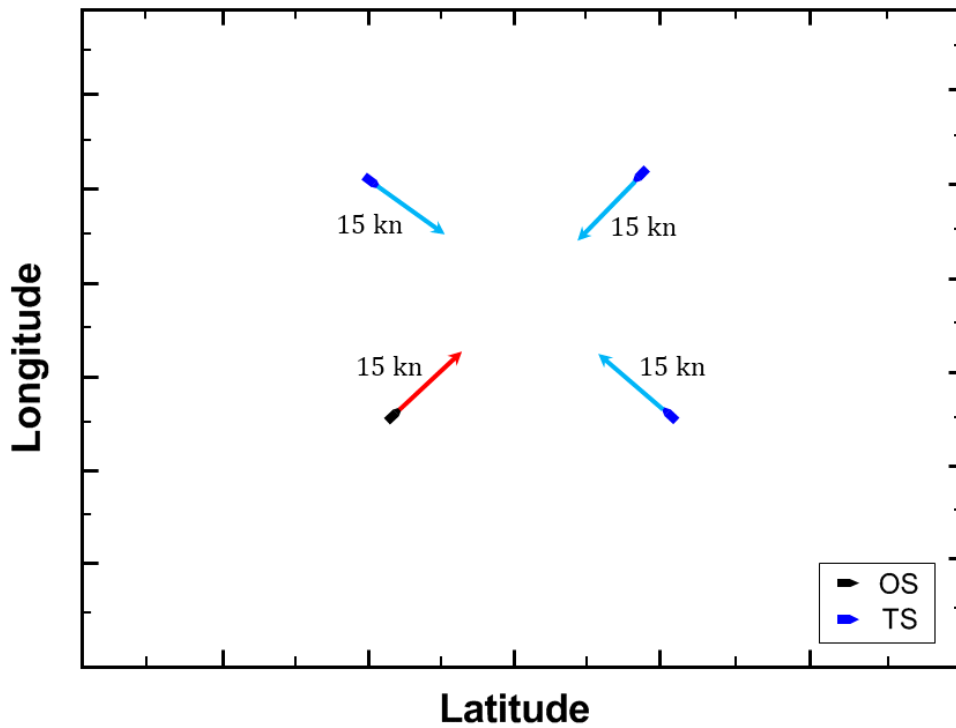


Fig. 4.4 The initial situation of COLREGs-based multiple ship (1:3) navigation simulation

4.2 시뮬레이션 결과 검증

앞서 COLREGs 기반 항해 시뮬레이션 검증을 위해 시뮬레이션 선박 설정과 신뢰성 있는 동역학 움직임을 표현하기 위하여 선회 시험을 수행하였으며, 이 결과로 실시간 선수방위 변화율의 도출이 가능하여졌다. 이를 통해 시뮬레이션 선박은 실제 해상에서의 항해와 유사한 움직임을 가질 것으로 기대되며, 시뮬레이션 중 각 모델에서 출력되는 데이터를 이용한 결과 검증은 신뢰성을 가질 것으로 판단된다.

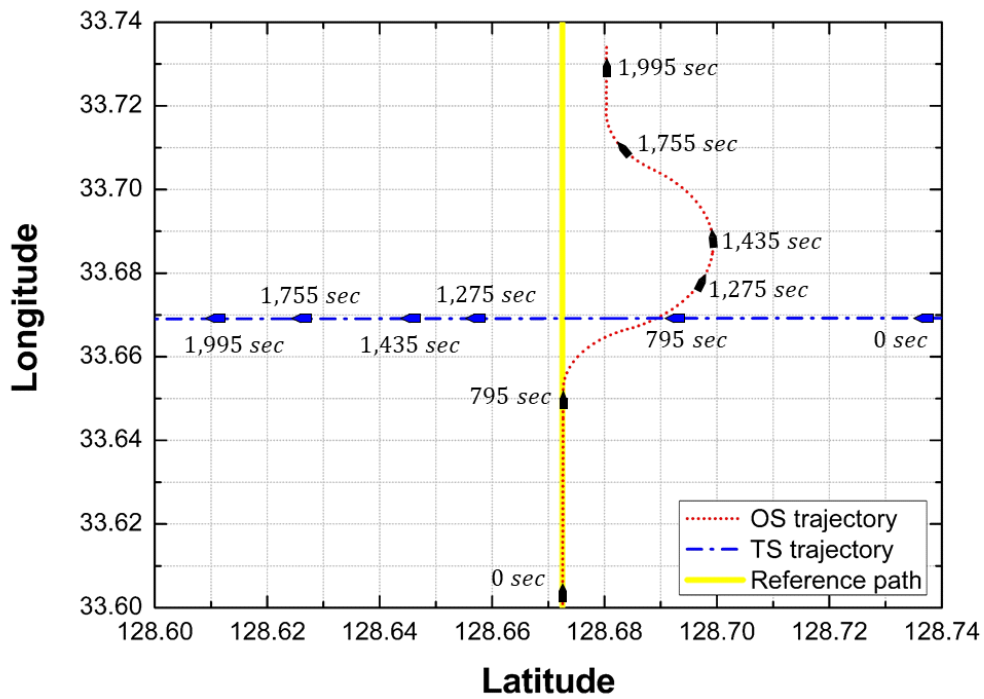
4.1절에서 결과 검증을 위해 설정한 시뮬레이션의 초기조건은 COLREGs에서 규정하는 항법 관계에 따라 1:1 충돌회피를 위한 총 4가지의 상황, 그리고 다중 선박 충돌회피를 위한 1:3 상황을 구성하였다.

(1) Case I - 1:1 충돌회피 검증

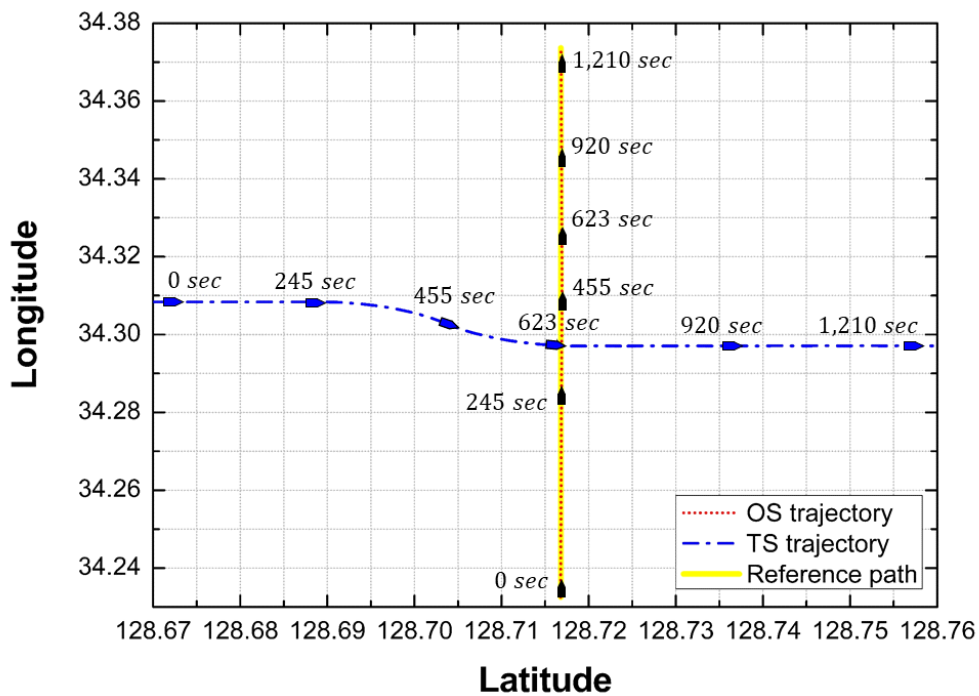
Fig 4.5의 (a)와 (b)는 각각 Crossing 상태에 대한 회피 또는 유지의 결정을 나타내며, Fig 4.3의 (a)와 (b)에서 표현한 시뮬레이션 상황에 대하여 충돌회피를 수행한 결과를 나타낸다.

Fig 4.5의 (a)는 OS과 TS의 상대방위가 약 45°를 유지하고 서로의 양현등을 바라보고 있는 상황으로 Crossing 상태이다. 상대 선박에 대한 회피를 결정하는 시점(795 sec)에서 OS의 항해사 에이전트는 Crossing 조우 상황에 대하여 본 선박이 TS의 좌현, 즉 홍등을 바라보고 있다고 판단하였다. COLREGs에 따라 회피 의무를 지기 때문에 우현으로의 변침을 명령하고, 조타수 에이전트는 이에 따라 타를 변경하여 충돌을 회피한다.

상대선박의 관점에서 TS는 OS의 녹등을 바라보고 있기 때문에 항해사 에이전트에 의해 회피 의무를 지지 않는다고 판단하여 진로를 유지한다. OS는 1,275 sec까지 516 sec 동안 회피를 수행하였으며, 이후, TCS에 의해 원 침로 및 항적 교차 범위 내로의 진입을 수행한다.



(a) Crossing (OS avoidance)



(b) Crossing (TS avoidance)

Fig. 4.5 Trajectory of collision avoidance simulation for Crossing

Fig 4.5의 (b)는 (a)와 반대의 상황으로 TS가 피항하는 경우이다. 이 경우 TS가 OS의 흥등을 바라보고 있기 때문에 피항 의무를 지며, 회피결정 시점(245 sec)부터 455 sec까지 210 sec간 회피를 수행하였다. 원 침로 복귀 후 약 0.75 nm 벗어나 있지만, 이는 초기설정 교차항적 거리(1 nm) 범위 내에 있으므로 침로를 유지하며 항해한다.

Fig 4.6은 Head-On 상태에 대한 피항 또는 유지의 결정을 나타내며, Fig 4.3의 (c)에서 표현한 시뮬레이션 상황에 대하여 충돌회피를 수행한 결과를 나타낸다. OS과 TS는 서로의 선수, 즉 마스트 정부등을 바라보고 정면으로 마주치고 있으므로 Head-On 상태이다. 상대 선박에 대한 충돌 위험도가 219 sec에서 임계 값을 벗어났으며, 이는 회피를 결정하는 시점이 된다.

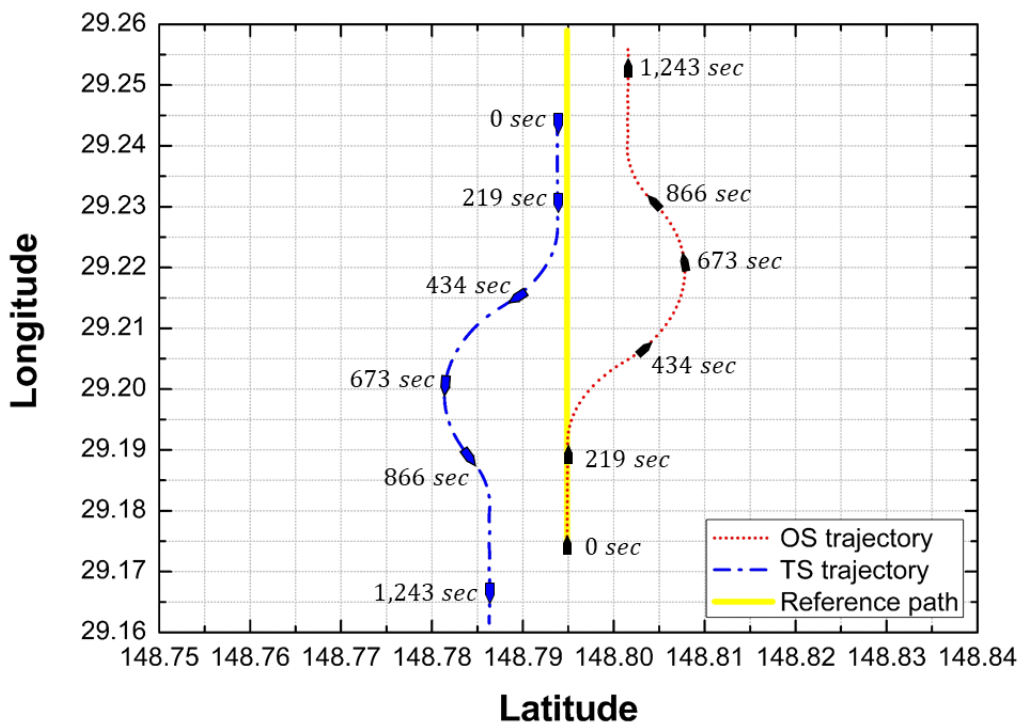


Fig. 4.6 Trajectory of collision avoidance simulation for Head-On (OS, TS avoidance)

OS와 TS의 항해사 에이전트는 COLREGs 기반 지식으로 Head-On 상태임을 판단하고 피항 의무를 지기 때문에 두 선박 모두 우현으로 변침을 명령하고, 조타수 에이전트는 이에 따라 타를 변경하여 충돌을 회피한다. OS과 TS는 회피 결정 시점부터 434 sec까지 215 sec간 회피를 수행하였으며, 이후 원 침로 및 초기설정 교차 항적 거리 범위 내로 진입하기 위하여 TCS에 의해 조종된다.

Fig 4.7의 은 Overtaking 상태에 대한 피항 또는 유지의 결정을 나타내며, Fig 4.3의 (d)에서 표현한 시뮬레이션 상황에 대하여 충돌회피를 수행한 결과를 나타낸다. TS보다 10 kn 빠르게 항해 중인 OS는 TS의 선미, 즉 선미등을 바라보고 있기 때문에 Overtaking 상태이다. 회피를 결정하는 시점(225 sec)에서 OS의 항해사 에이전트는 충돌위험 회피를 위한 판단 결과로 우현 변침을 선택한다. 조타수 에이전트는 이에 따라 타를 변경하여 충돌을 회피하며, 회피 결정 시점부터 710 sec까지 485 sec간 회피를 수행하였다. 이후, 원 침로 및 초기설정 교차 항적 거리 범위 내로 진입하기 위하여 TCS에 의해 조종된다.

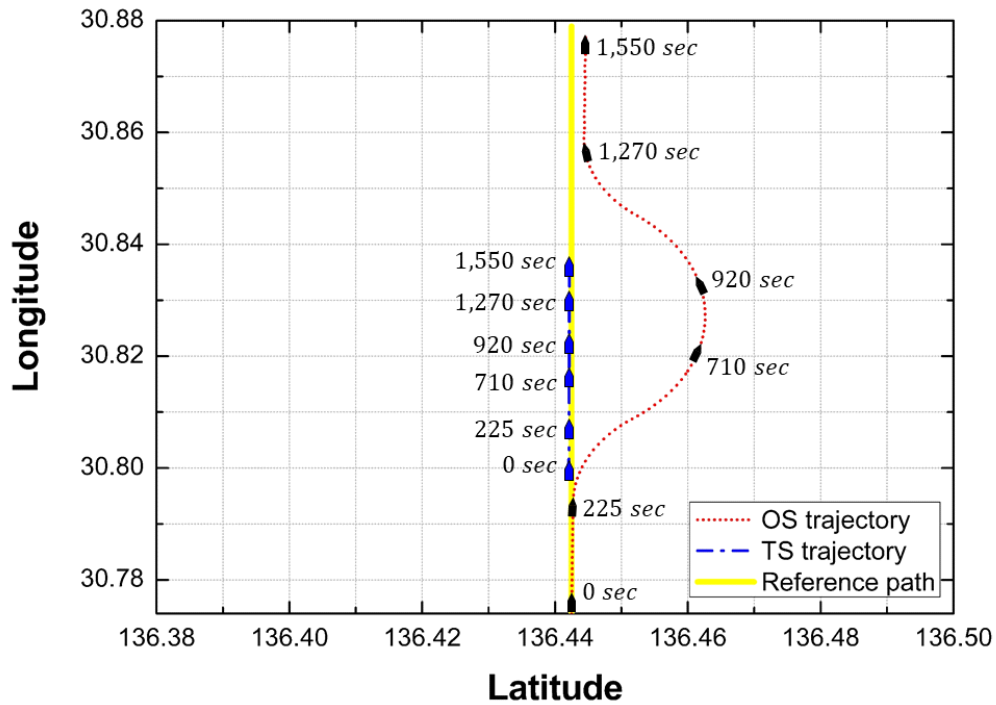


Fig. 4.7 Trajectory of collision avoidance simulation for Overtaking (OS avoidance)

(2) Case II - 1:3 충돌회피 검증

Fig 4.8은 Fig 4.4에서 표현한 시뮬레이션 상황에 대하여 다중 대상 선박의 항로 계획에 따른 침로 유지 및 에이전트에 의한 충돌회피 판단에 대한 수행 결과를 나타낸다. OS뿐만 아니라 TS 또한 설정된 침로를 향해 하는 동안 대상 선박에 대한 충돌위험도를 실시간으로 추정하여 임계 값을 초과하는 충돌위험도 중 가장 높은 대상을 우선하여 회피한다.

OS의 최초 충돌회피를 감지하는 시점(185 sec)에서 타선박 대비 TCPA와 DCPA가 가장 작아서 충돌위험도가 높게 추정된 TS1에 대해 항해사 에이전트는 피항 또는 유지 결정을 한다. TS1 또한 OS에 대한 충돌위험도가 가장 높게 추정된다. 이 상태는 COLREGs의 Crossing 상태로 TS1의 좌현을 바라보고 있는 OS가 피항의 의무를 지기 때문에 우현으로 회피하고 TS1은 침로를 유지한다. OS는 500 sec까지 315 sec간 회피를 수행하는

것을 확인할 수 있다. 이후, 원 침로 및 초기 설정 교차 항적 거리 범위내로 진입하기 위하여 TCS에 의해 조종된다. 대상 선박(TS1, TS2, TS3) 또한 침로 향해 중 발생하는 충돌 위험을 항해사 에이전트에 의해 자율적으로 회피하고 원 침로로 복귀하는 것을 검증한다.

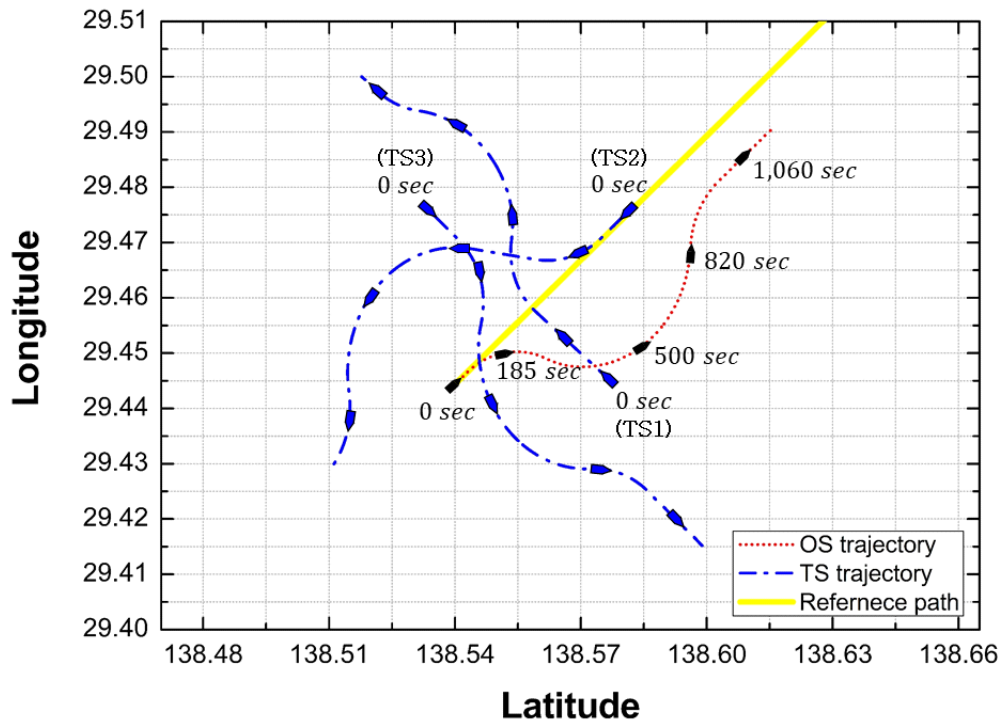


Fig. 4.8 Trajectory of multiple ship (1:3) collision avoidance simulation

제 5 장 결론 및 향후 연구

해상에서 지속해서 발생하고 있는 선박사고의 감소를 위해 항해사의 의사결정을 지원할 수 있는 항해 안전 지원 서비스 등이 과거부터 다양한 접근을 통해 제안되었다. 하지만, 연구 결과물의 검증을 위해 실제 선박 탑재를 통한 시험은 막대한 비용과 위험이 따른다. 이러한 문제를 위해서 항해사 교육 목적의 시스템인 선박 조종 시뮬레이터(FMSS, full mission ship simulator)를 이용하여 다양한 시나리오의 시뮬레이션을 통한 검증이 가능하지만, 이 또한 대여 비용이 상당히 높아서 연구 목적으로의 접근성이 떨어진다.

또한, FMSS는 선박 충돌위험 등과 같은 상황에 대한 직접적인 조종이 요구되며, 운항 관련 지식이 없이 이루어지는 선박 조종으로는 신뢰할 수 있는 검증이 이루어질 수 없을 것으로 판단된다. 그리고 FMSS에서 운항 시나리오 중 본 선박과 대상 선박은 설정된 경유 지점을 따라 운항할 뿐 충돌위험 상황에 대한 능동적인 회피를 하지 않으므로 실제 선박 운항과는 많은 차이를 가진다.

본 논문에서는 이러한 한계점을 극복하기 위해 컴퓨터 기반의 시뮬레이션을 위한 기본 구조를 설계하였다. 선박에 탑재되는 핵심장비의 주요 기술과 특성을 분석하여 모델링하였고, 실제 선박에서 항해에 직접 관여하는 항해사와 조타수의 의무를 분석하여 에이전트로 모델링하였다.

이를 통해, 항해 시뮬레이션 시스템은 여러 모델로 구성되어 복잡한 관계를 가진다. 따라서, 이러한 구조를 체계적으로 모델링하기 위해 Zeigler가 제안한 SES/MB(system entity structure/model base) 프레임워크를 사용

하였으며, 모델들 간의 계층적인 관계와 각 모델의 행위적 및 동역학적 특성을 나타내었다. 모델링 된 구조를 DEVS(discrete event system specification) 형식론 기반으로 각 모델과 여러 모델 간의 동작을 명세하여 시뮬레이션 시스템을 구현하였다.

항해 시뮬레이션을 위해 본 논문에서 설계한 내용 검증을 위해 국제해상충돌예방규칙(COLREGs, international regulations for preventing collisions at sea)에서 정의하는 항법 관계를 시나리오로 그 유용성을 성공적으로 검증하였다.

제안한 항해 시뮬레이션 시스템을 위한 모델링 방법은 컴퓨터 기반 시뮬레이션을 위한 기본적인 구조로 활용될 뿐만 아니라 향후 추가적인 요소의 결합을 통한 확장성을 가진다. 또한, 항해 안전 지원 서비스 및 항해 디스플레이 장치 등의 저비용 및 신뢰성 높은 유용성 검증이 가능할 것으로 기대된다.

향후에는 제안한 모델링 방법을 통한 시뮬레이션으로 수집 가능한 다양한 데이터를 활용 및 분석하여 보다 더 효율적인 회피 경로를 도출하기 위한 인공지능 기반의 연구 등을 진행할 예정이며, 이는 최근 관심이 집중되고 있는 자율운항 선박의 충돌위험 상황 시 회피 경로 도출을 위한 모듈로 사용할 것으로 기대된다.

참고문헌

- [1] 해양수산부, 2018. *2017년 해양사고 통계 발표*, 해양수산부 중앙해양안전심판원.
- [2] 이미라, 2010. 선박의 항행정보시스템을 위한 상황 예측 시뮬레이션 방안 연구. *한국시뮬레이션학회논문지*, 19(3), pp.127-135.
- [3] 이미라, 2017. 뉴로-퍼지 기반의 선박 충돌 회피 조치 분석 시스템 설계. *한국지능시스템학회논문지*, 27(2), pp.113-118.
- [4] 김동균, 정중식, 박계각, 2011. 국제해상충돌예방규칙에 따른 충돌회피 알고리즘에 관한 연구. *한국지능시스템학회논문지*, 21(3), pp.290-295.
- [5] Y. Xue, D. Clelland, B. S. Lee, and D. Han, 2011. Automatic simulation of ship navigation. *Ocean Engineering*, 38(17-18), pp.2290-2305.
- [6] 신일식, 황훈규, 이장세, 유영호, 2015. e-Navigation MSP 평가를 위한 시나리오 기반 시뮬레이션 시스템 개발. *한국마린엔지니어링학회논문지*, 39(1), pp.86-93.
- [7] B. P. Zeigler, 1990. Object-oriented Simulation with Hierarchical, Modular Models : Intelligent Agents and Endomorphic systems. No. 04; QA76. 9. C65, Z4.
- [8] B. P. Zeigler, H. Praehofer, and T.G. Kim, 2000. Theory of Modeling and Simulation, 2nd Edition. *Academic Press*.
- [9] IMO Std. COLREGs, Convention on the International Regulations for

Preventing Collisions at Sea, *IMO*, 1972.

- [10] NMEA Std. 0183, Standard for Interfacing Marine Electronic Devices, Version 4.00, *NMEA*, 2008.
- [11] IEC Std. 62065 ed.2, Maritime navigation and radiocommunication equipment and systems-Track control systems-Operational and performance requirements, methods of testing and required test results, *IEC*, 2014.
- [12] 송병호, 이경효, 정민아, 이성로, 2011. 다중 선박에서 효율적인 충돌 회피를 위한 베이지안 충돌 위험도 추정 알고리즘. 한국통신학회논문지, 36(3), pp.248-253.
- [13] 김동균, 정중식, 박계각, 2011. 국제해상충돌예방규칙에 따른 충돌회피 알고리즘에 관한 연구. 한국지능시스템학회논문지, 21(3), pp.290-295.
- [14] 김성우, 2013. 선박 충돌회피 우선순위지정 및 회피모델 연구. 한국산학기술학회논문지, 14(11), pp.5442-5447.
- [15] 임남균, 2004. 선박충돌회피지원을 위한 자동제어-Ⅲ. 한국항해항만학회지, 28(6), pp.475-480.
- [16] 손남선, 김선영, 2004. 선박충돌사고의 AIS 데이터를 이용한 선박 충돌위험도 추정 알고리즘 검증에 관한 연구. 한국항해항만학회지, 34(10), pp.727-733.
- [17] K. Hasegawa et al., 1989. Ship Auto-navigation Fuzzy Expert System (SAFES). *Journal of the Society of Naval Architecture of Japan*, 166, pp.445-452.
- [18] K. Hasegawa, J. Fukuto, R. Miyake, and M. Yamazaki, 2012. An Intelligent Ship Handling Simulator with Automatic Collision Avoidance

Function of Target Ships. *International Navigation Simulator Lecturers' Conference (INSLC17)*, 23, pp.1-9.

- [19] B. P. Zeigler, 1999. Exploiting HLA and DEVS To Promote Interoperability and Reuse in Lockheed's Corporate Environment, *Simulation*, 73(5), pp.288-295.
- [20] H. Imazu and T. Koyama, 1988. The Determination of Collision Avoidance Action. *Journal of Japan Institute of Navigation*, 70, pp.31-37.
- [21] Y. Fujii and K. Tanaka, 1971. Traffic Capacity. *The Journal of Navigation (Japan)*, 24, pp.543-552.
- [22] 이윤석, 박준모, 안영중, 2014. 충돌위험 회피를 위한 선박 운항자의 초동조치에 관한 연구. *한국항해항만학회지*, 38(4), pp.327-333.

부록 A 향해 시뮬레이션 시스템 구성 모델 의사 코드

Table A.1 The Pseudo-codes of DEVS-based Experiment Frame Model

(a) Generator Model

Experiment frame - Generator

external transition function

when receive value on input port from Transducer model

if phase is "wait"

if (value == "simulation stop") hold-in "passive"

internal transition function

if phase is "busy" hold-in "wait"

output function

if phase is "busy"

output : send the ships initialization message to output port

(b) Transducer Model

Experiment frame - Transducer

external transition function

when receive value on input port from Logger model

if phase is "active"

if (value == "simulation ships")

Save(); // ships data (position, heading etc.) hold-in "active"

else if (value == "stop") hold-in "passive"

internal transition function

if phase is "passive" hold-in "passive"

output function

if phase is "passive"

output : send the stop message to output port

Table A.2 The Pseudo-codes of DEVS-based Navigation SPACE Model

(a) Logger Model

SPACE Model - Logger

external transition function
when receive value on input port from Generator, Sensors and SEP model
if phase is "wait"
if (value == "simulation stop") hold-in "passive"
else if (value == "ship initialization") hold-in "sending sensor"
else if (value == "sensor") hold-in "receiving"
else if (value == "SEP value")
update the receiving SEP value
hold-in "sending GIS"
else if phase is "receiving"
update the receiving sensors, rudder, engine and AIS value
// position, heading, velocity, rudder
if (value == AIS) hold-in "sending Propagator"
else hold-in "sending SEP"

internal transition function
if phase is "sending sensor" hold-in "receiving"
if phase is "receiving"
if phase is "sending SEP" hold-in "wait"
if phase is "sending Propagator" hold-in "wait"
if phase is "sending GIS" hold-in "wait"
if phase is "passive" hold-in "passive"

output function
if phase is "sending sensor"
output : send the sensor message to output port
else if phase is "receiving"
output : send the sensor message to output port
else if phase is "sending SEP"
output : send the sensor message to output port
else if phase is "sending Propagator"
output : send the AIS message to output port
else if phase is "sending GIS"
output : send the GIS message to output port
else if phase is "passive"
output : send the simulation stop message to output port

Table A.2 The Pseudo-codes of DEVS-based Navigation SPACE Model

(b) SEP Model

SPACE Model - Spatial Encounter Predictor

external transition function

when receive value on input port from Logger model
calculate the position and heading information of each ship
// Position after 1 second
hold-in “calculating”

internal transition function

if phase is “calculating”
hold-in “wait”

output function

if phase is “calculating”
output : send the ship location and heading message to output
port

(c) GIS Model

SPACE Model - GIS

external transition function

when receive value on input port from Logger model
update GIS information
hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

Table A.2 The Pseudo-codes of DEVS-based Navigation SPACE Model

(d) Propagator Model

SPACE Model - Propagator

external transition function

when receive value on input port from Logger model
calculate the propagation range of AIS
// Propagation range within 30 km (VHF Communication)
hold-in “propagating”

internal transition function

if phase is “propagating”
hold-in “passive”

output function

if phase is “propagating”
output : send the propagation message (AIS) to output port

(e) Environment Model

SPACE Model - Environment

external transition function

when receive value on input port from SEP model
setup the environment // wind, wave height
hold-in “setup”

internal transition function

if phase is “setup”
hold-in “passive”

output function

if phase is “setup”
output : send the set message (environment) to output port

Table A.3 The Pseudo-codes of DEVS-based Sensor Model

(a) GPS Model

Sensors - GPS

external transition function

when receive value on input port from Logger model
update the GPS information of each ship
hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

if phase is “updating”
output : send the GPS information to output port

(b) Gyro compass Model

Sensors - Gyro compass

external transition function

when receive value on input port from Logger model
update the Gyro compass information of each ship
hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

if phase is “updating”
output : send the Gyro compass information to output port

Table A.3 The Pseudo-codes of DEVS-based Sensors Model

(c) SpeedLog Model

Sensors - SpeedLog

external transition function

when receive value on input port from Logger model
update the SpeedLog information of each ship
hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

if phase is “updating”
output : send the SpeedLog information to output port

Table A.4 The Pseudo-codes of DEVS-based AIS Model

(a) Class A AIS Model

AISs – Class A AIS

external transition function

when receive value on input port from Sensors model

if (value == “sensors”) hold-in “updating”

else if (value == “propagation”) hold-in “sending”

internal transition function

if phase is “updating”

hold-in “passive”

else if phase is “sending”

hold-in “passive”

output function

if phase is “updating”

output : send the AIS information to output port

if phase is “sending”

output : send the propagation information (AIS of other ships) to output port

Table A.5 The Pseudo-codes of DEVS-based TCS Model

(a) ECDIS Model

Track Control System - ECDIS

external transition function

when receive value on input port from Sensors, AISs and
Agent model

if phase is "wait"

if (value == "sensors")

update(); hold-in "updating"

// update simulation ships position, heading, velocity

else if (value == "AIS")

if (CR() > Threshold) hold-in "collision risk"

//calculate CR

else if (value == "Agent") hold-in "manual"

//manual steering by agent

internal transition function

if phase is "updating"

hold-in "wait"

if phase is "collision risk"

hold-in "wait"

else if phase is "manual"

hold-in "wait"

output function

if phase is "collision risk"

output : send the collision information to output port

else if phase is "manual"

output : send the manual information to output port

Table A.5 The Pseudo-codes of DEVS-based TCS Model

(b) HCS Model

Track Control System - HCS

external transition function

when receive value on input port from ECDIS model
if phase is "wait"
if (value.mode == "HCS")
HCS(); hold-in "HCS"
else if (value.mode == "curved track control")
Curved(); hold-in "curved"
else if (value.mode == "cross track control")
Cross(); hold-in "cross"
else if (value.mode == "agent")
agent(); hold-in "agent"
if phase is "curved" || "cross" || "agent"
mode change to the HCS mode
Hold-in "HCS"

internal transition function

if phase is "HCS"
hold-in "wait"
else if phase is "curved" || "cross" || "agent"
hold-in the phase //curved, cross, agent

output function

if phase is "HCS"
output : send the HCS rudder value to output port
else if phase is "curved"
output : send the curved rudder value to output port
else if phase is "cross"
output : send the cross rudder value to output port

Table A.6 The Pseudo-codes of
DEVS-based Propulsion and Steering Systems Model

(a) Rudder Model

Propulsion and Steering Systems - Rudder

external transition function

when receive value on input port from HCS and Agent model
update the rudder value
Hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

if phase is “updating”
output : send the rudder value to output port

(b) Engine Model

Propulsion and Steering Systems - Engine

external transition function

when receive value on input port from Logger model
update the engine value
Hold-in “updating”

internal transition function

if phase is “updating”
hold-in “passive”

output function

if phase is “updating”
output : send the engine value to output port

Table A.7 The Pseudo-codes of DEVS-based agent Model

(a) Officer Agent Knowledge Base Model

Agents - Officer Agent KB

external transition function

when receive value on input port from Officer IE model

rule 1 : **if** located on the port and look at the starboard of target ship
Then report is Maintain hold-in “normal”

rule 2 : **if** located on the port and look at the port of target ship
if (sl.location <= 30) **Then** report is Fine
else if (sl.location <= 60) **Then** report is Broad
else if (sl.location <= 112.5) **Then** report is Converging
hold-in “avoid”

rule 3 : **if** located on the starboard and look at the starboard of target ship
Then report is Maintain
hold-in “normal”

rule 4 : **if** located on the starboard and look at the port of target ship
if (sl.location <= 30) **Then** report is Fine
else if (sl.location <= 60) **Then** report is Broad
else if (sl.location <= 112.5) **Then** report is Converging
hold-in “avoid”

rule 5 : **if** located on the bow and look at the bow of target ship
Then report is Head-On
hold-in “avoid”

rule 6 : **if** located on the stern and look at the stern of target ship
Then report is Overtaking
hold-in “avoid”

internal transition function

if phase is “normal” hold-in “normal”
if phase is “avoid” hold-in “avoid”

output function

if phase is “normal” || “avoid”
output : send the report data message to output port

Table A.7 The Pseudo-codes of DEVS-based agent Model

(b) Helmsman Agent Knowledge Base Model

Agents - Helmsman Agent KB

external transition function

when receive value on input port from Helmsman Inference
Engine model

rule 1 : if order data is maintain
Then order is Maintain

rule 2 : if order data is fine
h.rudder = 15;
Then order is Turning

rule 3 : if order data is broad
h.rudder = 20;
Then order is Turning

rule 4 : if order data is conversing
h.rudder = 30;
Then order is Turning

rule 5 : if order data is head-on
h.rudder = 10;
Then order is Turning

rule 6 : if order data is overtaking
h.rudder = 10;
Then order is Turning

hold-in “decision-making”

internal transition function

if phase is “decision-making”

hold-in “passive”

output function

if phase is “decision-making”

output : send the order data message to output port

Table A.7 The Pseudo-codes of DEVS-based agent Model

(c) Officer Agent Inference Engine Model

Agents - Officer IE

external transition function

when receive value on input port from ECDIS model

check the COLREGs rule

request condition leveling

Hold-in "request"

internal transition function

if phase is "request"

hold-in "interfacing"

output function

if phase is "request"

output : send the ship location message to output port

(d) Helmsman Agent Inference Engine Model

Agents - Helmsman IE

external transition function

when receive value on input port from Officer model

check the order

request condition leveling

Hold-in "request"

internal transition function

if phase is "request"

hold-in "interfacing"

output function

if phase is "request"

output : send the order message to output port
