

지능형 자율운항제어를 위한 선박운동제어시스템

(A Ship Motion Control System for Autonomous Navigation)

이 원 호 [†] 김 창 민 ^{**} 최 중 락 ^{***} 김 용 기 ^{****}
 (Won-Ho Lee) (Chang-Min Kim) (Joong-Lak Choi) (Yong-Gi Kim)

요 약 선박 자율운항시스템(autonomous navigation system)이란 선박운항에 있어 항해계획을 수립하고 현재의 선박운항 상태를 파악하여 주변 상황변화를 예측하고 대처하는 항해 전문가의 능력을 전산화한 것이다. 선박 자율운항시스템은 항해, 충돌회피, 선체유지, 자료융합, 운동제어 그리고 이를 통합하는 아키텍처로 구성되어 있다. 운동제어시스템은 선박의 유체학적 특성을 고려하여 해도 상에 원하는 위치로 선박을 이동하기 위해서 추진 및 조타장치를 제어하는 시스템으로 자율운항제어시스템에 필수적인 구성 시스템 중에 하나이다. 본 논문은 운영플랫폼인 가상세계시스템을 기반으로 운영되는 선박자율운항시스템과 운동제어시스템의 연동과 구현에 관한 연구이다. 운동제어시스템은 충돌회피시스템으로부터 상위 레벨의 고수준제어 요구치를 전달받아 조타 및 추진치로 변환하고 조타장치와 추진장치를 제어하는 시스템이다. 본 논문에서 선박 운동 특성을 수학적으로 모방하는 Oldenburger의 제어 이론에 기반하여 선박운동제어기를 개발하고 성능검정을 위해 선박시뮬레이터에서 다양한 시나리오를 바탕으로 시뮬레이션 한다.

키워드 : 자율운항시스템, 선박운동제어시스템, Oldenburger의 제어이론, 선박시뮬레이터

Abstract Ship autonomous navigation is designated as what computerizes mental faculties possessed of navigation experts, which are building navigation plans, grasping the situation, forecasting the fluctuation, and coping with the situation. An autonomous navigation system, which consists of several subsystems such as navigation system, a collision avoidance system, several data fusion systems, and a motion control system, is based on an intelligent control architecture for the sake of integrating the systems. The motion control system, which is one of the most essential system in autonomous navigation system, controls its propulsion and steering gears to move the ship satisfying its hydrodynamic characteristics. This paper is the study on the ship movement control system and its implementation which are totally developed and run on virtual-world system. Receiving the high-level control values such as a waypoint presented from the collision avoidance system, the motion control system generates them to low-level control values for propulsion and steering devices. In the paper, we develop a ship motion controller using Oldenburger's theory based on mathematical fundamentals, and simulate it with various scenarios in order to verify its performance.

Key words : autonomous navigation system, ship movement control system, Oldenburger's control theory, ship simulator

1. 서 론

최근 들어 두드러지고 있는 선박업계의 인력부족 현

상을 해결하기 위하여 숙련된 승조원의 지식을 수집 정리하여 승조원의 역할을 지능적으로 수행할 수 있는 고수준 선박 무인화전문가 시스템이 개발되고 있다. 그 중의 하나가 선박의 지능화 및 자동화에 관한 연구인 지능형 자율운항제어시스템(Autonomous Navigation System using Intelligence Techniques)이다. 지능형 자율운항시스템은 선박의 플랫폼통합관리체계(IPMS, Integrated Platform Management System)를 마치 숙련된 승조원이 작동하는 것처럼 제어하고 상황에 따라 적절한 대처안을 제시하는 자율 대처 능력을 갖춘 지능시스템이다. 이는 선박 운항에 소용되는 승조원의 수를 감축

· 본 연구는 국방과학연구소 기초연구사업(과제번호: ADD-00-6-2)에 의해 수행되었습니다.

[†] 학생회원 : (주)윙몰 연구원

windmill@ailab.gsnu.ac.kr

^{**} 학생회원 : (주)삼진기술 연구원

nuno@ailab.gsnu.ac.kr

^{***} 비 회원 : 국방과학연구소 연구원

jlchoi@add.re.kr

^{****} 종신회원 : 경상대학교 컴퓨터과학과 교수

ygkim@nongae.gsnu.ac.kr

논문접수 : 2002년 11월 13일

심사완료 : 2003년 7월 28일

하는데 그 목적이 있다. 선박의 플랫폼통합관리체제(IPMS)는 선박에 장착된 장비를 디지털화하고, 네트워크로 각 장비들을 연결, 감시, 제어할 수 있는 통합적 관리제어체제를 말한다.

지능형 자율운항제어시스템은 실제적인 조타와 같은 추진제어 명령으로 선박을 이동시키기 위해서는 조타수의 영역전문가 수준의 선박조종제어시스템이 필요하다. 선박조종제어시스템은 지능형 자율운항제어시스템에서 고수준의 제어 요구치를 저수준 제어 요구치로 변환하여 해도상의 원하는 위치로 선박을 이끌어 가는 시스템이다. 여기에서 고수준의 제어는 충돌회피시스템으로부터 출력인 선박이 가야할 다음 위치의 좌표이고, 저수준 제어는 조타 및 추진치를 이용하여 조타장치와 추진장치를 제어하는 것을 의미한다.

본 연구에서는 블랙보드 아키텍처에 기반한 지능시스템 운영플랫폼인 가상세계시스템을 소개하고 이에 기반한 선박자율운항시스템을 설계, 구현한다. 선박자율운항시스템은 센서(sensors)와 액추에이터(actuators)와 같은 하드웨어와 항해시스템, 충돌회피시스템, 자료융합시스템, 선박운동제어시스템과 같은 소프트웨어로 구성된다. 특히, 선박운동제어시스템은 추진장치와 조타장치의 직접제어에 의한 선박의 유체역학적 특성을 반영하여 실제에 근접한 운동제어를 가능하게 하는 소프트웨어로서 선박자율운항시스템에서 없어서는 안될 중요한 부분 중 하나이다. 본 연구에서는 가상세계시스템을 통하여 제공되는 고수준 선박 운동제어를 유체역학을 고려한 저수준 제어로 변환하여 선박을 제어하는 선박운동제어시스템을 구현한다.

선박운동제어는 수학적 제어, 퍼지제어, 신경망을 이용한 제어 등이 있다. 수학적 제어이론은 고전적인 선박운동제어에 기반한 제어기로 일정 수준의 선박조종이 가능한 제어기이다. 대표적인 수학적 제어 이론은 Oldenburger의 제어이론[1]이 있다. 퍼지제어이론은 선박제어의 불확실성 해결에 탁월한 기법이다. 대표적인 퍼지선박제어 이론은 Vaneck의 선박퍼지유도제어기가 있다[2]. 이는 이동해야할 선박의 좌표와 이동 후의 선수각을 입력으로 하여 선박을 유도한다. 신경망을 이용한 제어이론은 선수각의 자동제어를 신경망을 이용하여 수행한다. 신경망제어기법은 비선형성이 심한 시스템과 모델링이 정확하지 않은 시스템에 대하여 다른 제어기법들에 비하여 그 적응성이 뛰어나다는 것이 증명되었다[3,4]. 현재 연구가 진행되고 있는 선박운동제어기로는 퍼지와 신경망을 이용한 인공지능기법의 제어기가 있다. 이들은 설계와 구현 시 많은 시간과 개발비용이 소요된다.

본 논문에서는 수학적 제어이론[5]을 이용한 선박운동

제어시스템을 제안한다. 수학적 제어이론은 고전적인 선박운동제어에 기반을 하고 있다. 고전적인 선박운동제어는 선형모델에 근거를 두고 있어 비선형 운동을 하는 선박에는 부적절하다. 그러므로, 우선 비선형성과 불확실성을 고려하여 선박을 선형화하고, 그 후 수학적 제어이론과 종합하면 일정 수준의 선박조종이 가능하다[1,6,7]. 본 논문에서는 선박 선형화를 위해서 nomoto방정식[8]을 사용하였고, 수학적 제어 이론은 Oldenburger의 제어이론을 이용하여 선박운동 제어 시스템을 구성하였다.

2장에서는 지능형 자율운항제어시스템과 운영플랫폼인 가상세계시스템에 관하여 살펴보고, 자율운항시스템을 구성하는 부시스템과 선박운동제어시스템의 연동에 관하여 살펴본다. 3장에서는 Oldenburger의 제어이론에 기반한 선박운동제어시스템에 대해서 살펴본다. 4장에서는 선박운동제어시스템의 시뮬레이션을 살펴보고, 5장에서 결론 및 향후과제에 관하여 살펴본다.

2. 관련 연구

최근 들어, 컴퓨터, 센서, 액추에이터 등 하드웨어 기술의 급격한 발달로 지능시스템 분야의 연구가 활발하게 진행되고 있다. 대표적인 지능시스템 분야로는 소형 로봇, 무인자동차, AUV(Autonomous Underwater Vehicles), UAV(Unmanned Aerial Vehicles), intelligent ship 등을 들 수 있다. 이와 같은 지능시스템을 분석해보면 모든 지능시스템은 문제영역에 의존된 문제해결 기능과 메시지전달, 로그관리, 모듈 연결 등 지능시스템을 운영하는데 요구되는 기능으로 구분됨을 알 수 있다. 이때 후자를 독립된 라이브러리로 개발하고 이를 이용하여 지능시스템을 개발하면 개발비용 및 안정성 면에서 많은 이득을 얻을 수 있다.

이와 관련된 대표적인 연구를 살펴보면, 미국의 NASA(National Aeronautics and Space Administration, 미국항공우주국)에서 개발한 TCA(Task Control Architecture)가 있다. 이는 계층적 아키텍처(hierarchical architecture)에 기초하고 RPC(Remote Process Call) 기반 메시지 패싱(Message Passing) 기술을 이용하여 일반화된 로봇제어 프레임워크(general control framework)이다[9].

GARCS(Generic Autonomous Robot Control Shell)는 계층적 아키텍처에 기반한 지능시스템 운영환경으로써 미국 NIST(National Institute of Standards and Technology)에 의한 지원으로 University of North Texas에서 개발한 소프트웨어이다. 이는 모듈화된 구성, 순수 소프트웨어, 재사용성, 확장성, 그리고 소프트웨어 모듈과 하드웨어 모듈이 같이 참여에 중점을 두며,

상업화가 진행되고 있다[10].

TCA 및 GARCS의 목표는 다양한 이동 로봇에 적용할 수 있는 운영환경을 제공하는 것이다. 이 아키텍처들은 계층적 아키텍처에 기반하므로 태스크간 강한 직접과 빠른 정보교환을 지향한다. 그러나, 이와 같은 특성은 소프트웨어 구성요소 간 제어 흐름 상 접근을 의미하며, 아키텍처 구현 시 필요한 태스크 트리(task tree), RPC, 후킹(hooking) 등 제어 흐름을 일관적으로 통제할 수 있는 관리시스템이 필요하고, 모든 정보교환은 이러한 환경에 기초하여 수행하여야 함을 의미한다. 이는 곧 이 아키텍처를 이용하여 지능시스템 개발 시, 주변장치 즉 노드의 증가에 따른 태스크 트리의 복잡성 크게 증가하므로 많은 센서가 장착되는 대형 자율운송체에는 적합하다고 볼 수 없다. 그리고, 이는 개발자에게 아키텍처의 세부 제어흐름 구조와 방법을 명확히 이해할 것을 요구하며, 이는 개발자에게 또 다른 개발 부담으로 다가온다.

이와 같은 특성을 가진 소프트웨어가 등장한 원인을 살펴보면, 지능시스템 소프트웨어의 필요성이 제기된 시기가 로봇의 하드웨어적 처리능력이 충분히 뒷받침되지 못한 20세기 후반이었으므로, 소형 로봇에 적재되는 소프트웨어는 로봇의 메모리 적재 용량과 처리성을 심각하게 고려하여야 하였다. 자율운송체는 소형 로봇과는 달리 넓은 공간을 보유하여 충분한 처리 능력의 전산 자원 장착이 가능하므로 지능시스템 소프트웨어의 소형화 및 처리성능보다는 유연성(flexibility), 안정성(reliability)을 고려하여야 하고, 지능시스템 개발자 중심의 간략화된 개발환경이 제공되어야 한다.

3. 지능형 자율운행제어시스템

자율운행제어시스템은 항해 전문가의 능력을 전산화하는 소프트웨어로 센서로부터 주어지는 입력정보를 처리하여 조타장치 및 추진장치를 조작한다. 자율운행제어시스템은 자동차, 선박, 잠수함과 같이 인간에 의해 직접 조종되는 유인항체에 인간의 역할을 대신할 수 있는 지능시스템을 장착하여 전체적 혹은 부분적으로 무인화된 운송체를 말한다. 무인자율운송체에서 사용되는 소프트웨어는 인식, 사고, 행위와 같은 인간의 지적능력을 내포한 인공지능시스템이어야 한다. 지능형 자율 운항 제어 기술은 운항 중 발생할 수 있는 항해와 충돌회피 문제를 해결하여 운송체를 자율적으로 조종하는 것으로서, 이를 위해서 다양한 인공지능 기법이 활용된다. 본절에서 반응형 및 속고형 기법의 상호 융화 극대화화 시스템 구성요소의 구조적 및 기능적 독립성 보장에 초점을 맞춘 RVC(Reactive layer-Virtual world-Considerative layer) 모델을 소개하고, 이에 기초한 지

능시스템 운영 플랫폼인 가상세계시스템을 소개한다.

3.1 RVC 모델

지능형 자율운행시스템은 다양한 인공지능기법을 사용하는 서브시스템으로 구성된 복잡한 시스템이다. 이와 같이 방대한 시스템을 개발하는데 있어서 효과적인 입·출력 설계와 소프트웨어 공학적 장점을 고려하기 위해서는 통합 아키텍처가 필요하다.

지능형 자율운행시스템에서는 다양한 인공지능기법을 보다 효과적으로 통합하기 위해 RVC(Reactive Layer-Virtual World-Considerative Layer) 모델[11]을 제안한다. RVC 모델은 그림 1과 같이 기초기반 인공지능과 행위기반 인공지능기법을 반응층과 인식층으로 구별하고, 반응층과 인식층 사이에 가상세계라는 공유정보저장영역을 둔 모델이다. 반응층은 저수준의 지적능력이 필요하고 직관적 체계를 바탕으로 하는 자료융합시스템을 배치하고, 인식층은 고수준 지적능력이 필요로 하는 손상감지, 항해, 충돌회피, 선박운동제어시스템을 배치한다. 그리고 이들 모든 시스템들은 가상세계라는 시스템을 통하여 정보를 상호 공유하게 된다[12].

자율운행시스템은 입력장치, 자료융합, 지능제어, 출력장치로 이루어진다. 입력장치는 다시 운항센서류와 장애물감지센서류로 구분할 수 있는데 이는 실세계의 존재하는 수많은 정보 중 자율운행시스템에서 소요되는 정보를 추출하여 가상세계로 주입한다. 자료융합은 입력계층에서 반영한 저수준 정보를 가공하여 고수준 정보를 생산하여 가상세계에 주입한다. 지능제어는 항해시스템, 충돌회피시스템으로 구성되는데, 이는 가상세계에 표현된 실세계 정보를 이용하여 당면 문제점을 해결하여 선박의 고수준 제어치를 출력장치에 전달한다. 출력장치는 시간적 요소를 고려하여 선박의 추진 및 조타장치를 제어하여 충돌회피로부터 주어진 고수준 제어치를 실제화한다. 특히, 선박운동제어시스템은 원하는 위치로 선박을 이동시키기 위해서 추진치와 조타치를 시간에 변화에 따라 제어하는 부시스템인데, 선박의 유체역학적 특성을 모방하여 선박을 제어하기 위한 필수적 도구이다 [12]. 그림 2는 RVC 모델을 적용한 선박자율운행시스템이다. 그림 2에서 센서는 입력장치에, 충돌위험도산출부는 반응형프로세스에 디스플레이와 추진조타제어는 즉

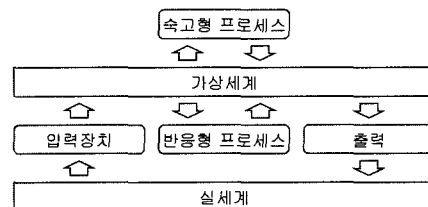


그림 1 RVC 지능시스템 모델

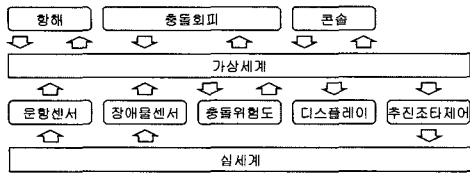


그림 2 RVC 모형 기반 선박자율운항시스템

각적인 처리가 요구되므로 출력장치에 해당하고, 항해시스템, 충돌회피시스템, 콘솔은 시간에 대해서 자유로우므로 속고형 프로세스에 해당한다.

3.2 RVC 모형 기반 가상세계시스템

RVC 모형은 반응층-가상세계-속고층으로 구성되며, 반응층과 속고층은 가상세계를 통하여 서로 연결된다. 이때, 반응층과 속고층에 속하는 부시스템은 가상세계에 표현되는 정보에 자유롭게 접근 가능해야한다. 이에 반해, 가상세계는 항상 대기하며 부시스템의 정보 접근 요구가 있을 때 즉각 서비스 해야한다. 이것을 클라이언트/서버 구조에 투영해보면, 가상세계는 서버의 조건을 만족하고 반응층과 속고층의 부시스템은 클라이언트의 조건을 만족함을 알 수 있다. 그리고, 속고형 프로세스, 반응형 프로세스, 입력장치, 출력장치는 모두 기호화된 정보의 교환으로써 가상세계에 연결되므로 상호 공통적인 인터페이스를 가지게 되므로 단일화된 형태로 추상화될 수 있다. 따라서, 그림 3과 같이 RVC 지능시스템 모형 기반 자율운항시스템은 서버 역할을 하는 가상세계가 기호화된 정보 교환의 중심이 되는 클라이언트/서버 환경 구축으로 가능하다[12].

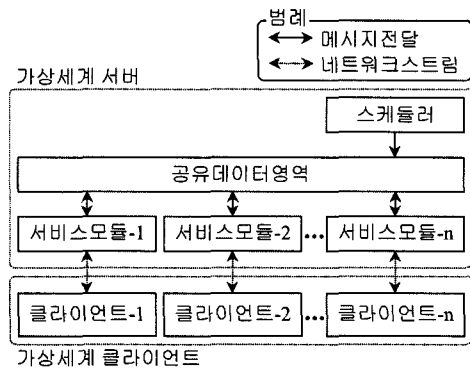


그림 3 가상세계시스템

4. 선박운동제어시스템

선박운동제어기술은 선박 운항의 자동화에 있어 반드시 필요한 기술 중 하나이다. 신뢰성 있는 선박운동제어

기술은 선박의 정교한 조종을 가능하게 하므로, 선박 운항 자동화 기술에는 반드시 선박운동제어기술이 필요하다. 본 절에서는 가상세계시스템과 연동하는 선박운동제어시스템에 관하여 살펴본다.

4.1 선박운동제어시스템 모델링

선박의 운동은 추진장치와 조타장치의 직접제어에 의한 유체역학적 산물이다. 따라서 의도한 대로 선박의 운동 상태를 이끌어내기 위해서는 유체역학을 적용하여 추진장치와 조타장치를 조작하여야 한다. 선박운동제어시스템은 선박의 운동에 관한 고수준제어요구를 반영하기 위한 추진치 및 조타치를 산출하고 이를 추진장치와 조타장치에 전달하는 역할을 한다.

본 연구의 선박운동제어기는 그림 4와 같이 고수준 제어치 R_{high} 와 현 선박의 운동상태 S 가 주어질 때 저수준 제어치 R_{low} 를 산출하는 제어시스템이다. 이때 저수준제어라 함은 선박의 조타장치와 추진장치를 직접 제어하기 위한 조종치입에 반해, 고수준제어라 함은 선박의 조타장치와 추진장치의 제어에 관한 직접적인 정보를 제공하지 않고 선박을 조종하기 위한 추상적인 정보를 말한다. 대표적인 고수준제어는 웨이포인트에 의한 조종, 선박 heading에 의한 조종 등이 있다.

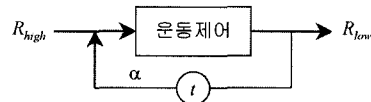


그림 4 선박운동제어모형

이때, 고수준 제어치 R_{high} 는 수식 1과 같이 선박의 다음 경유점(waypoint) w 와 그 순간 선박 운동상태 α 로 표현된다. x 와 y 는 직교좌표계를 기준하여 표현한 웨이포인트의 공간적 위치이다. 선박운동상태 α 는 수식 2, 3, 4와 같이 선박의 주요운동성분인 surge, sway, yaw에 대한 속도와 가속도, 그리고 선박의 현 위치를 표현하는 s 와 선수각 h 로써 표현된다. 저수준 제어치 R_{low} 는 수식 5와 같이 그 순간의 추진치 ρ 와 조타각 γ 로써 표현된다.

$$R_{high} = \{w\} \tag{1}$$

$$w = \{x, y\} \tag{2}$$

$$\alpha = \{s, h\} \tag{3}$$

$$s = \{u, v, r, u', v', r', x_c, y_c, \theta\} \tag{4}$$

$$R_{low} = \{\rho, \gamma\} \tag{5}$$

4.2 선박운동제어시스템과 가상세계시스템의 연동

본 연구의 선박운동제어시스템은 선박자율운항시스템을 구성하는 다른 부시스템과 가상세계시스템을 통하여

연결된다. 특히, 선박운동제어시스템은 충돌회피시스템과 매우 밀접한 관계를 가지는데, 선박운동제어시스템의 입력으로 작용하는 고수준 제어치는 충돌회피시스템에서 생산된 정보이다[12]. 그림 5는 가상세계시스템과 연동하는 선박운동제어기와 주변시스템간의 관계를 표현한 구성도이다.

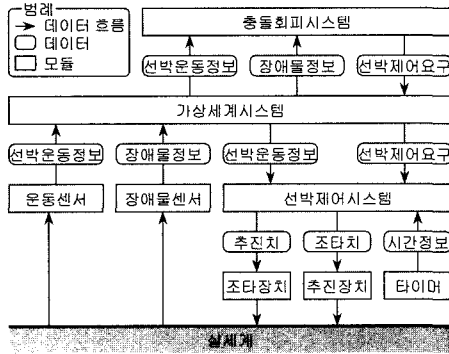


그림 5 선박운동제어시스템과 주변장치

4.3 수학적 이론에 기반을 한 선박운동제어기

선박의 운동모형은 비선형 모형이다. 이를 선형화하여 수학적 제어이론[1]과 종합하면 일정 수준의 선박조종[4-6]이 가능하다. 대표적인 수학적 제어 이론이 Oldenburger의 시간준최적화 제어 이론이다. 우선, 선박의 횡운동에 대한 방정식을 선형화한 후 1차 nomoto 방정식[8]으로 변환하면 수식 6과 같은 기초선형모형이 산출된다[1].

$$T \dot{r}' + r' = K' \delta_R \tag{6}$$

여기에서 T '는 차원화된 시간지연이고, K' 는 이득, r' 은 선미 각 속도, δ_R 는 조타 제어치가 되겠다. 이때 이상적인 수학적 조타모델은 아래 수식 7과 같이 정할 수 있다

$$\delta_R = \epsilon_m \frac{L}{V} U \tag{7}$$

$$\bar{u} = T r' + \psi; \bar{v} = \delta_R \tag{8}$$

$$x = \frac{V}{L \epsilon_m} \cdot \frac{\bar{u} - \psi^*}{K'}; y = \frac{V}{L \epsilon_m} \bar{v} \tag{9}$$

여기에서 ϵ_m 는 타의 최대 각, L 의 배의 길이, V 는 배의 속도, U 는 수식 10에서 산출되는 Oldenburger 제어 계수, \bar{u} 와 \bar{v} 는 상태변수, ψ 는 선미각을 의미하고 있다. 위와 같은 모형이 제공될 때, Oldenburger 제어 이론은 다음 수식 10과 같다.

$$U = \text{sign}(\sigma(x, y)) \tag{10}$$

$$\dot{x} \equiv y; \dot{y} \equiv U \tag{11}$$

이때 수식 12와 같은 간단한 교환함수를 이용할 수 있다. 여기서 σ 가 선박의 방향을 변화시키는 조타치이다. sign 은 σ 의 결과 값이 최대타각 이내의 값인지 비교하고 초과하면 최대타각으로 변환하는 함수이다.

$$\sigma = -x - \frac{1}{2} |y| y \tag{12}$$

4.4 선박운동제어 함수

그림 6은 Oldenburger의 제어이론을 C++로 구현한 것이다. 이 모듈을 살펴보면, 입력으로는 현 시점에서 이동해야 할 선박의 방향(psiOr), 현재의 선미각(psi), 차원화되어진 선박의 속도(rs), 현재의 조타각(d), 현재의 선박의 속도(spd)를 입력받고, Oldenburger의 제어 이론에 의한 수식 12를 적용으로 새로운 조타각 U 를 출력한다.

선박운동제어시스템은 State_Control, Initial, Start_state, Order_Rudder 모듈들로 구성되는데 이 모듈들은 시간에 맞추어 순차적으로 처리된다. 이 순차적 처리를 제어하는 모듈은 State_Control모듈이다. State_Control 모듈은 가상세계의 스케줄러의 상태값을 보고 Initial, Start_state, Order_Rudder모듈 중에서 실행할 모듈을 선택하는 역할을 한다.

```

1: const double PHI 3.141592
2: double CShipMotionSimulator::Con_Old(
3:   double psiOr, // required rudder order
4:   double psi,  // current heading
5:   double rs,   // current dimensionless angular
6:               // velocity
7:   double d,    // current rudder angle
8:   double spd  // current ship speed
9:   double ts    // first order time lag
10: )
11: {
12:   double fac; // a factor
13:   double ub; // aggregated state variables
14:   double vb; // aggregated state variables
15:   double x;  // normalized aggregated states
16:   double y;  // normalized aggregated states
17:   double sw; // switch fuction
18:   double he0; // formal heading error
19:   double he;  // effective heading error
20:
21:   he0 = psi - psiOr;
22:   if (abs(he0) > PHI)
23:     if (he0 > 0.0)
24:       he = he0 - 2 * PHI;
25:     else
26:       he = he0 + 2 * PHI;
27:   else
28:     he = he0;
29:   if( he0 == 0)
30:     return 0;
31:   ub = ts * rs + he;
32:   vb = d;
33:   fac = a1 * spd;
34:   x = fac * ub / ks;
35:   y = fac * vb;
36:   sw = -x - 0.5 * y * abs(y);
37:   sw = sign(dfIMax, sw);
38:   return sw;
}
    
```

그림 6 C++로 작성된 Oldenburger의 제어이론

Initial 모듈은 선박 제어에 앞서 선박의 제원을 입력 받는 부분이다. 입력으로는 선박의 길이, 넓이, 흘수, 디자인 스피드, 타각의 최대 기울어짐, 제어 타임의 간격을 입력받게 된다. 이렇게 입력받은 선박의 제원은 선박의 초기 파라미터들을 설정하게 되는데 이 파라미터들은 nomoto 방정식의 입력으로 주어진다. Initial모듈은 선박의 제원을 입력받아 유체역학적 힘이 존재하는 비선형 운동체를 선형화시키는 역할을 한다.

Start_state모듈은 선박운동제어시스템이 작동 할 시점에서의 선박의 운동상태를 입력하는 모듈이다. 입력으로는 선박의 surge 속도, sway 속도, yaw 속도, heading, advance, transfer, 타의 기울어짐을 입력받게 된다.

마지막으로 Order_Rudder모듈은 선박운동제어시스템에서 가장 핵심적인 모듈로 Oldenburger의 제어 이론이 실행되는 곳이다. 입력으로는 충돌회피시스템의 출력값인 선박이 다음으로 이동할 waypoint이다. waypoint는 직교좌표계를 이용하여 (x, y)로 표현하고, 북쪽으로 이동할 거리가 x, 동쪽으로 이동할 거리가 y이다. waypoint가 주어지면 현 상태의 운동 정보와 함께 Ctrl_Old모듈을 호출하게 되는데 이때 전달인자는 이동해야 할 선박의 방향과 현재의 선미 각, 차원화되어진 각속도, 그리고 현재 선박의 속도가 전달된다. Ctrl_Old 모듈에서는 각 1/78초 간격으로 타각을 산출하게 된다.

4.5 선박운동제어시스템의 구성

지능형 자율운항시스템은 다양한 인공지능기법을 효율적으로 통합하기 위해 반응층과 인식층으로 시스템으로 분리하고 중간에 가상세계라는 매개체를 둔다. 선박운동제어시스템은 서버와 통신함으로써 모든 시스템의 정보를 얻을 수 있다. 선박운동제어시스템은 크게 두 부분으로 나눌 수 있는데 하나는 가상세계인 서버와 자료를 입·출력하는 모듈과, 다른 하나는 실제적인 고수준 제어치를 저수준 제어치로 변환하는 모듈로 나눌 수 있다. 그림 7은 선박운동제어시스템의 구성도이다.

VWClient모듈부분은 가상세계와 통신하기 위해 TCP/IP 프로토콜을 이용하고, CSocket을 이용하여 통신규약에 맞게 데이터를 전송하게 된다. 이때 먼저 전송을 받는 데이터는 제어를 하게 될 선박의 제원과 제어를 시작할 시점에서의 선박의 상태를 먼저 입력받아 선박 운동 제어시스템을 초기화를 시킨다. 그 다음부터의 데이터는 지능형 자율운항시스템의 충돌회피 시스템의 출력값인 고수준 선박 제어치를 입력받게 된다. 고수준 선박 제어치는 특정시간 후에 선박이 이동해야할 다음 경유점이다. 이러한 데이터를 인터넷을 통해 실시간으로 전송 받는 모듈이 VWClient모듈이다.

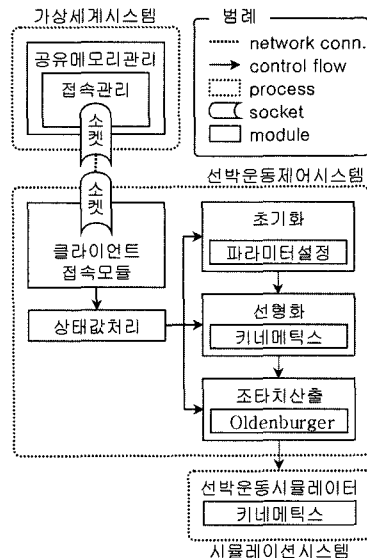


그림 7 선박운동제어시스템 구성도

5. 시뮬레이션

선박운동제어시스템은 자율운항시스템에 포함된 일부 시스템이다. 선박운동제어시스템을 테스트하기 위해서는 자율운항시스템의 모든 시스템이 완전히 구축되고, 실제 선박에 장착되어 실제계에서 운항을 해야 가능하다. 그러나 선박은 고가의 교통기관으로 관련 하부장치의 인터페이스 개발에 많은 노력이 소요되므로 실제 선박에 장착하여 시험하는 데에는 높은 비용이 소요된다 [13-15]. 본 연구에서는 선박 운동제어시스템을 테스트하기 위해서 가상적인 시나리오로 작성하고, 수학적 운동방정식을 기반으로 한 선박시뮬레이션시스템을 통하여 결과를 분석한다.

5.1 선박운동제어시스템 초기화 DB

초기화 DB는 선박의 물리적 특성 및 선박이 초기상태를 부여하며, 지능형 자율운항제어시스템에서 사용할 인터페이스 및 초기화 변수를 등록하고 초기화한다[12]. 이것은 지능형 자율운항제어시스템 내 서버시스템 간의 일관성 유지와 중복을 막기 위함이다. 그림 8은 선박의 제원과 각종 초기 운동상태를 표현한다. 그림 6에서 1번째 줄은 초기화 섹션 정의이고, 2번째 줄부터 시스템간의 섹션 주기(read, write, tail)는 5초, 선박의 길이 70.57m, 넓이 12.3m, 흘수 4.5m, 디자인 스피드 15Knot, 타의 최대 기울어짐 35도인 선박의 제원이 정의된다. 9번째 라인부터 17번째 라인은 선박시뮬레이터에 의해 생산되는 선박운동 상태값을 의미한다.

```

..CLASSF:INITIA:System Initialization
1 VARIAB:NUMBER:INITIA.SESION.PERIOD:1:5:세션 주기
2 VARIAB:NUMBER:INITIA.SHIP.LENGTH:1:70.57:선장
3 VARIAB:NUMBER:INITIA.SHIP.BREATH:1:12.3:선폭
4 VARIAB:NUMBER:INITIA.SHIP.DRAFT:1:4.5:물수
5 VARIAB:NUMBER:INITIA.SHIP.SSPEED:1:15:서비스 선속
6 VARIAB:NUMBER:INITIA.SHIP.MAXRUD:1:35:최대 러드 절환각
7 VARIAB:NUMBER:INIFAC.SMSIM.RELCRD.X:1:0:원지점의 상대 X 위
9 치
10 VARIAB:NUMBER:INIFAC.SMSIM.RELCRD.Y:1:0:원지점의 상대 Y 위
11 치
12 VARIAB:NUMBER:INIFAC.SMSIM.SURGSP:1:15:SURGE 속도
13 VARIAB:NUMBER:INIFAC.SMSIM.SWAYSP:1:0:SWAY 속도
14 VARIAB:NUMBER:INIFAC.SMSIM.YAWDSP:1:0:YAW 각속도
15 VARIAB:NUMBER:INIFAC.SMSIM.HEADDEG:1:0:헤딩각
16 VARIAB:NUMBER:INIFAC.SMSIM.ADV:1:0:ADVANCE
17 VARIAB:NUMBER:INIFAC.SMSIM.TRN:1:0:TRANSFER
: VARIAB:NUMBER:INIFAC.SMSIM.RUDEG:1:0:타각
:
    
```

그림 8 선박운동시스템 초기화를 위한 초기화 DB

5.2 실험

시나리오는 충돌회피시스템의 출력값으로 선박의 다 음 위치이다. 그림 9와 10은 실험을 위하여 인위적으로 계획된 시나리오이다. 그림 9의 1번째 라인을 보면, 자 선의 현재위치가 (0,0)이며 다음 이동위치(700, 400)으로 이동하라는 의미를 표현한다.

```

//시나리오 #1
1: Move(Owe_Ship, SPosition(0,0), GPosition(700,400))
2: Move(Owe_Ship, SPosition(700,400), GPosition(0,800))
3: Move(Owe_Ship, SPosition(0,800), GPosition(-400,300))
    
```

그림 9 시나리오 1

```

//시나리오 #2
1: Move(Owe_Ship, SPosition(0,0), GPosition(500,800))
    
```

그림 10 시나리오 2

선박운동제어시스템의 초기화와 그림 9의 시나리오 1 에 의해 약 6분 정도 운행 한 결과는 그림 11과 같다. 선박운동시스템은 초당 78회 주기로 선박의 상태값을 반영한다. 그림 11을 살펴보면, 직선운동을 하고 있는 자선에 시나리오 1이 입력되며, 선박운동제어시스템은 그림 11의 ①에서의 조타치를 제어하여 헤딩을 변화시 킨다. 그 후 차츰 조타치를 감소시키고 헤딩을 바로 잡 기 위해 반대방향으로 조타를 변화시킨다. 그림 12은 시 나리오 2에 대한 타각을 변화를 그래프화 한 것이다.

그림 11을 통하여 선박의 시나리오에 따라 빠르게 이 동했는지 판단하기 힘들다. 그러므로 선박시물레이션시 스템을 이용하여 선박의 이동 경로를 산출한다. 선박 시 물레이션시스템은 수학적 kinematic운동방정식을 이용 하여 선박의 상태를 산출한다[15]. 그림 13과 그림 14는 선박시물레이션시스템이 선박운동제어시스템의 출력을 입력으로 받아 선박의 위치변화를 나타낸 그림이다. 이 는 그림 9와 그림 10에서 계획된 시나리오의 웨이포인 트를 만족한다.

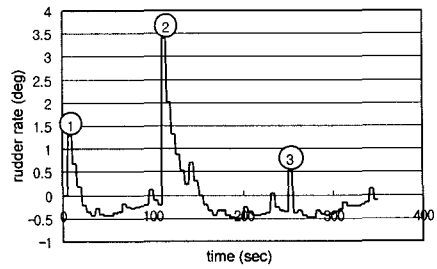


그림 11 시간에 따른 조타의 변환(시나리오 1)

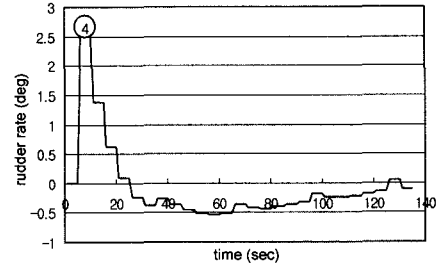


그림 12 시간에 따른 조타의 변환(시나리오 2)

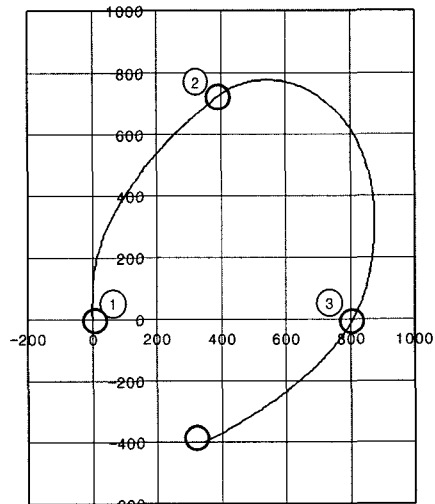


그림 13 조타 제어치에 따른 선박의 위치 (시나리오 1)

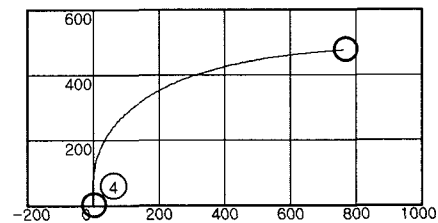


그림 14 조타 제어치에 따른 선박의 위치 (시나리오 2)

6. 결론

본 논문은 가상세계시스템 기반 선박자율운항시스템과 연동하여 운영되는 선박운동제어시스템의 설계 및 구현에 관한 연구이다. 자율운항시스템은 다양한 인공지능기법이 적용된 다양한 서브시스템으로 구성된 복잡한 시스템이다. 이와 같이 방대한 시스템을 개발하는데 있어서 효과적인 입·출력 설계와 소프트웨어 공학적 장점을 고려하기 위해서는 통합 아키텍처가 필요하다. 본 연구에서는 RVC 모형을 소개하고 이에 기초한 가상세계시스템을 선박자율운항시스템에 적용한다. 자율운항시스템을 구성하는 모든 부시스템은 가상세계를 통하여 통신하며 상호 독립적으로 수행된다. 가상세계시스템은 자율운항시스템을 위한 공통적인 운영플랫폼 역할을 구현한다.

선박의 운동은 추진장치와 조타장치의 직접제어에 의한 유체역학적 산물이다. 따라서 의도한 대로 선박의 운동 상태를 이끌어내기 위해서는 선박의 유체역학을 적용하여 추진장치와 조타장치를 조작하여야 한다. 본 연구에서는 선박의 운동에 관한 웨이포인트 형태의 고수준제어요구를 반영하기 위하여 추진치 및 조타치를 산출하고 이를 추진장치와 조타장치에 전달하는 역할을 구현한 선박운동제어시스템을 구현한다. 본 연구의 선박운동제어시스템은 선박 운동 특성을 수학적으로 모방하는 Oldenburger의 이론에 기초한다. 그리고, 비선형 운동체인 선박의 운동을 선형 운동모형으로 만들기 위해 nomoto 방정식[8]을 사용한다.

선박 제어는 크게 수학적 제어, 퍼지이론을 이용한 제어, 신경망을 이용한 제어로 구분할 수 있는데, 본 연구의 선박운동제어시스템은 선박 운동 특성을 수학적으로 모방하는 Oldenburger의 이론에 기초한다. 그리고, 비선형 운동체인 선박의 운동을 선형 운동모형으로 만들기 위해 nomoto 방정식[8]을 사용한다.

선박운동제어시스템 테스트를 위해서 가상시나리오를 구성하고, 충돌회피시스템에서 제공된 고수준제어정보가 그에 적합한 조타 및 추진치로 변환되어 의도한 선박의 조종이 이루어짐을 보였다.

향후에는 약간 거친 운항결과를 보이는 수학적 기반의 운동제어이론 보다는 유연성과 강직성이 높은 퍼지제어나 실세계 조타수의 능력과 아주 유사하게 제어할 수 있는 신경망 기법을 적용한 선박운동제어시스템을 설계하고 이를 가상세계시스템 기반 선박자율운항시스템에 적용하여 통합하여 보다 정교한 선박운동제어 모형을 적용하여야 한다.

참고 문헌

- [1] 연구기관, 조종성능 추정 정도 향상을 위한 요소 기술 개발, 한국기계연구원 선박해양공학연구센터 자체연구보고서, 1997.
- [2] Vaneck, T. W., "Fuzzy Guidance Controller for an Autonomous Boat," IEEE Control Systems, Vol. 17, No. 2, 1997.
- [3] Rozhdestvensky, V. V., Submarine Dynamics.-Leningrad, Sudostroyeniye Publ., 1970.
- [4] Sutulo, S. V., and Polukhin, K. V., Numerical Simulation of Ship Manoeuvring Motion, Schiffbau-forschung, 1988.
- [5] 이영호, 제어기 성능평가를 위한 연구용 비행 시뮬레이터 개발, 서울대학교 대학원 항공우주공학과, 석사학위논문, 1995.
- [6] 성영재, An Investigation on the PMM Test Condition for the Estimation of the Manoeuvring Coefficients, 서울대학교 대학원 선박해양공학과, 석사학위논문, 1998.
- [7] Gordon, G., System simulation(second editon), Prentice-Hall, 1978.
- [8] Nomoto, K., Taguchi, K., Honda, K., and Hirano, S., "On the Steering Quality of Ships," International Shipbuilding Progress, Vol. 4, pp. 354-370, 1957.
- [9] TCA, <http://www-2.cs.cmu.edu/afs/cs/project/TCA/release/tca.html>.
- [10] GARCS, <http://www.unt.edu/robotics/garcs.htm>.
- [11] 김창민, 김용기, "무인자율항체를 위한 지능제어 아키텍처에 관한 연구", 한국군사과학기술학회지, Vol. 4, No. 2, pp.249-255, 2001.
- [12] 김창민, 해양운송체를 위한 자율운항시스템의 지능시스템 아키텍처에 관한 연구, 경상대학교, 박사학위논문, 2003.
- [13] Law, A. M., and Kelton, W. D., Simulation modeling and analysis, McGraw-Hill Book Company, 1982.
- [14] Banks, J., and Carson, J. SII., Discrete-event system simulation, Prentice-Hall, 1984.
- [15] 이원호, 김창민, 김용기, "지능형 자율운항제어시스템을 위한 시뮬레이터 구현", 한국 퍼지 및 지능시스템학회 추계학술대회 학술발표논문집, Vol. 11, No. 2, pp.169-172, 2001.



이 원 호
2001년 경상대학교 컴퓨터과학과(이학사). 2003년 경상대학교 컴퓨터과학과(공학석사). 현재 (주)뮬뎀 연구원. 관심분야는 인공지능, 지식기반시스템, 퍼지정보검색 기법, 선박운동제어, 선박시뮬레이터



김 창 민

1997년 경상대학교 컴퓨터학과(이학사). 1999년 경상대학교 컴퓨터학과(공학석사). 2003년 경상대학교 컴퓨터학과(공학박사). 현재 (주)삼진기술 선임연구원. 관심분야는 퍼지정보검색, 퍼지문서분류기법, 지능제어아키텍처, 자율무인

잠수정아키텍처, 충돌회피



최 중 락

1977년 고려대학교 전자과(공학사). 1987년 서울대학교 제어계측학과(공학석사) 1997년 부산대학교 전기과(공학박사) 1978년~현재 국방과학연구소(책임연구원). 관심분야는 함정조종제어적용, 무인제어, 인공지능



김 용 기

1978년 서울대학교 공과대학(공학사) 1987년 University of Montana(전산학석사). 1992년 Florida State University(전산학박사). 1982년~1984년 KIST시스템공학연구소 연구원. 1992년~현재 경상대학교 컴퓨터학과 교수. 관심분야는

인공지능, 지식기반시스템, 자율무인잠수정, 지능항해시스템, 퍼지정보검색시스템