

論 文

小型 컴퓨터를 이용한 船舶 操縱 시뮬레이터 開發

李 昌 珉* · 姜 昌 求* · 孔 仁 泳* · 金 演 圭*

Development of a PC-based Ship Maneuvering Simulator

C. M. Lee · C. G. Kang · I. Y. Gong · Y. G. Kim

Key Word : 선박 조종 시뮬레이터(Ship Maneuvering Simulator), 소형 컴퓨터(i486 PC), 視界 再現(Computer Generated Imagery), 船舶 操縱 性能 數學 模型(Math. Model of Ship Maneuverability), H/W Interface, 근거리 통신망(LAN), 선원 훈련(Crew's Training), 안전성 검토(Safety Assessment), 항만 설계(Harbor Design)

Abstract

A PC-based ship maneuvering simulator was developed, which was configured in a high performance IBM PC compatible i486 and i286 computer with a TMS 340 graphic signal processor and 10 MBPS Ethernet Cards.

A real-time ship maneuvering simulation program was developed which includes computer generated imagery (CGI) for bird's eye view type and perspective view type. The simulator H/W was designed and manufactured and S/W for interfaces of various navigation equipments was made. Especially, programs for output, analysis, and assessment of simulation results were developed. Communications between PC's are made by using Ethernet bus type LAN system. Simulations could be performed under various environments (current, wind, wave etc.) using data base of harbors and ships. This system can be used for various purposes such as crew's training, harbor and waterway design, and assessment of ship maneuverability in harbor.

1. 서 언

최근에 선박의 대형화 추세 및 효율을 향상시키기 위한 특수한 선형의 개발은 조종 성능이 불량한 선박의 출현을 초래하였으며, 이로 인하여 해운사고의 위험성이 증가하고 있다. 실제 84년에 발표된 한 논문¹⁾에 의하면, 해마다 길이 90m 이상

되는 선박들 중 약 5%가 충돌 사고를 경험하며, 이중 약 80%가 탄 선박과의 충돌 사고인 것으로 나타나고 있다. 이러한 현상, 특히 상대적으로 사고의 위험성이 높은 항만이나 연안 등지에서의 해운사고는 인명에 대한 위협뿐 아니라 심각한 환경 오염의 원인이 될 수 있으며 화물 운송에도 적지 않은 위험 요소로 작용한다.

* 定會員, 韓國海事技術研究所 運航制御室

해상 사고의 이유로는 우선 그 선박의 조종 특성에 관한 조타실의 이해 부족, 그리고 해상 교통량과 Human Factor에 관한 정보 부족 등을 들 수도 있다. 선박의 조종 성능은 선체의 형상뿐만 아니라 선원의 조타 능력 및 자연 환경과도 밀접한 관계가 있다. 즉, 실제 해상에는 바람, 파도 및 조류 등이 존재할 뿐 아니라 부근을 통과하거나 정박하고 있는 선박이 있을 수 있으며, 유한한 수심을 가진 수로나 제한된 항로에서 이러한 여건을 극복하고 목표 지점까지 선박을 조종하여야 하는 어려움이 있다.

이에 따라 이러한 모든 효과를 고려한 상황에서의 선박의 조종 성능을 파악하기 위한 목적으로 선박 조종 시뮬레이터가 널리 사용되고 있으며, 국제 해사기구(IMO)에서도 선원들의 여러가지 상황에 대처하는 훈련용 및 연구용으로 이의 사용을 권장하고 있다. 이러한 여러가지 이유로 각국에서는 船橋의 형상을 그대로 재현한 대형 선박 조종 Simulator의 개발에 힘을 기울이고 있을 뿐 아니라, 이를 이용하여 많은 연구를 수행하고 있는 실정이다.

이에 능동적으로 대처하기 위하여 1984년 Micro Computer를 이용한 소형 선상 조종 Simulator를 개발한 바 있는 데²⁾, 이러한 소형 선상 조종 Simulator는 대형 Simulator에 비하여 조작이 간단하고 선상이나 기타 필요한 장소에 쉽게 설치할 수 있을 뿐 아니라 제작에 소요되는 경비도 비교적 저렴하다. 본 연구소에서는 이를 이용하여 여러가지의 Simulator 연구를 수행한 바 있다.³⁾ 하지만 당시 개발되었던 시뮬레이터는 Basic 언어로 프로그램되어 있어 계산 처리 능력에 한계가 있고 또한 중앙 처리 장치로 사용된 Micro Computer(NEC 9801)가 현재 국내에 많이 보급되어 있는 IBM PC 호환 기종과 자료 교환에 어려움이 있어 1989년 IBM PC/AT 호환 기종을 이용하여 선상 조종 시뮬레이터를 개량한 바 있다. 이것은 과거 본 연구소에서 개발한 바 있는 선박 조종 성능 추정 프로그램을 C 언어로 변환하고 Bird's Eye View type으로 개발한 것이었다.

1990년 부터 1년 동안 수행한 과학기술처의 기업화 과제의 결과로 "DeskPilot"이 개발되었다.²⁾

이 시뮬레이터의 H/W는 IBM PC/AT 386 또는 486 호환 기종을 중앙 처리 장치로 하고 Keyboard, Steering Stand, Engine Telegraph, Tug/Thruster Controller 등의 입력 장치와 VGA Card 또는 TMS340 GSP 등의 Color Graphic Board, Color monitor, Printer 등의 출력장치, 그리고 A/D 변환장치와 동력 공급장치 등으로 이루어져 있다. PC상에서 원하는 화면을 구현하고 여러 H/W와의 Interface를 원활하게 하기 위하여, Graphic 부분에 많은 기능을 추가하여 Bird's-Eye View Type 뿐만 아니라 Perspective View Type의 기본적인 CGI(Computer Generated Imagery)를 구현하였다.

본 프로그램에 입력되는 자료는 선박의 주요 제원을 비롯하여 그 선박의 조종 특성을 나타낼 수 있는 여러 係數 그리고 水深 및 潮流 분포를 포함한 항구자료 등이며, 타각 및 엔진에 대한 제어는 실제 船橋에서 사용되는 것과 유사한 장치들에 의하여 주어지도록 되어 있다. 또한 필요에 따라서 Tug나 Thruster 등의 효과를 포함시킬 수 있도록 되어 있다.

중대형 선박 조종 시뮬레이터 개발의 기초를 마련하기 위하여 선박 조종 시뮬레이션 프로그램을 앞으로 계속 개발될 시뮬레이터 시스템에 적합하도록 보완, 정리하였고, 소형 컴퓨터상에서 작동되는 高速 視界 再現 3次元 映像 處理 기법의 자료조사와 기본 연구를 수행하였다. 그리고 근거리 통신망(LAN)을 도입하여 NetBIOS를 사용한 PC 사이의 통신에 대한 연구를 수행하여 2차원 및 3차원 DeskPilot System을 연결시켜 동시에 작동하도록 시스템을 구성해 보았고 앞으로 개발될 중형이상의 시뮬레이터에서 사용될 데이터 통신의 기본 프로그램을 개발하였다.

2. 船舶 操縱 시뮬레이터 시스템 概要

본 시뮬레이터 시스템은 소형 PC를 이용하여 개발되었지만, 대형 시뮬레이터가 가지고 있는 거의 모든 기능을 구현할 수 있으며 조작과 설치가 간편하고 저렴한 비용으로 시스템을 구성할

수 있는 장점이 있다. 이 시스템은 좁은 공간이나 실제 선실등에 설치 가능한 Portable Type부터 Beam Projector와 Screen을 이용한 중형 시뮬레이터까지 다양하게 시스템을 구성할 수 있으므로, 일반 선박에 탑재되어 사용될 수 있음은 물론, 研究用이나 船員 訓練用 그리고 港灣 開發用으로도 사용될 수 있다.

선박의 조종운동 계산 프로그램, 2차원 및 3차원 그래픽 프로그램, 여러가지 항해 장비와 컴퓨터와의 인터페이스 프로그램, 그리고 시제품 등이 모두 자체 기술로 개발되었으며 이와 관련된 소프트웨어 및 하드웨어도 국내에서 손쉽게 구입할 수 있는 제품을 이용하여 개발되었으므로 향후 시뮬레이션 프로그램의 기능 보완 및 하드웨어의 부품 교체와 기능 개선이 손쉬운 뿐만 아니라 관련 분야의 최신 연구 결과가 즉시 시뮬레이터 시스템에 반영될 수 있다.

특히 본 시뮬레이터는 지난 1984년 연구용으로 최초 개발된 이래 각종 연구 및 항만 설계에 여러 차례 투입되었으며 그간 소형 컴퓨터 기능의 눈부신 발전에 힘입어 지속적으로 기능이 보강되어 왔고, 또한 이러한 연구 결과가 관련 국제 학술대회(MARSIM & ICSM '90, TOKYO)²²⁾에도 보고된 바 있다.

본 시뮬레이터는 Digitizer 등의 장비를 이용하여 입력된 여러 항구 자료와 다양한 종류의 선박 제원 등의 데이터 베이스를 이용하여 시뮬레이션을 수행할 수 있고 여러 가지 해상 상태(조류, 바람, 파도 등)를 시뮬레이션할때에 고려할 수 있다.

이 기능을 사용한다면 준설, 부두 확장, 그리고 부두 신설 등의 경우에 기존의 항만 관련 데이터 베이스를 수정하여 예상되는 항구 내에서의 선박의 운항 안정성 평가 및 항만 설계 적정여부를 다양하게 미리 검토해 볼 수 있다. 또한 새로 건조되는 선박의 조종 성능을 건조 전에 미리 시뮬레이션을 수행하여 검토해 볼 수도 있다. 이를 위해서 기존에 사용되는 선박의 기본 제원에 의한 선박 조종 성능 시뮬레이션 기능 외에 구속 모형 시험(PMM Test)결과 및 실선 시운전 결과를 『System Identification』 기법을 이용하여 해석함으로써 더 정확한 선박조종 시뮬레이션을 수행할 수도 있다.

2.1 船舶 操縱 시뮬레이터의 H/W 構成

본 시뮬레이터의 H/W의 구성은 Fig. 2.1과 같으며, 각각의 기능은 다음과 같다.

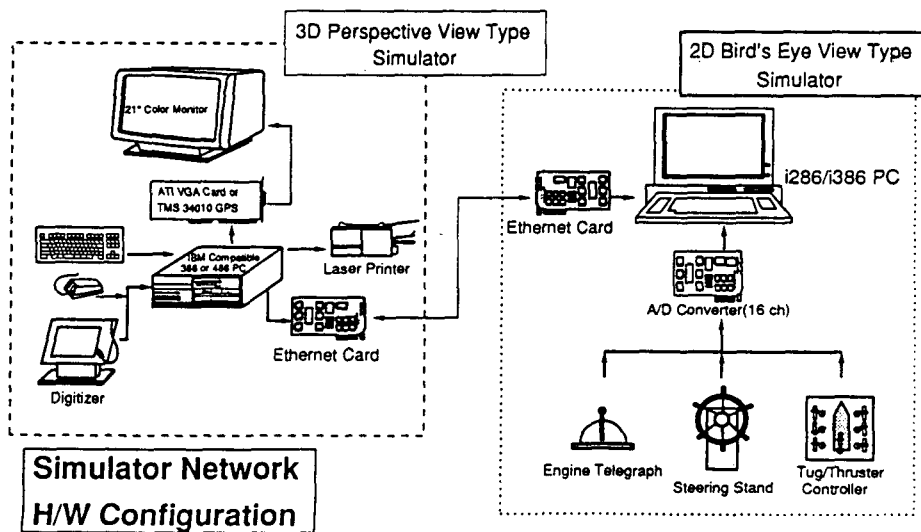


Fig. 2.1 H/W configuration

2.1.1 操船 입력 장치

操船 입력 장치는 네 부분으로 나눌 수 있다. 첫째는 타각을 제어하는 Steering Stand이고, 둘째는 엔진의 출력을 제어하는 Engine Telegraph 이다. 셋째 부분은 Tug나 Thruster 등의 효과를 포함시킬 수 있도록 Tug/Thruster Controller로 이루어져 있는 데, 이를 이용하면 최대 7곳의 위치에서 이들의 효과를 나타낼 수 있도록 되어 있다. 이들 장치에는 Potentiometer가 각각 부착되어 있으며 이들로부터 출력되는 전압이 A/D 변환기를 거쳐 중앙처리장치로 입력되게 된다.

그리고 마지막으로 표준 입력 장치인 Keyboard가 있는데, 이는 Simulation 시작전에 각종 초기치의 입력을 행하거나, Simulation의 종료 및 흐름제어 그리고 그 밖의 여러가지 표준입력 작업에 사용된다.

2.1.2 A/D 변환 장치

A/D 변환장치로는 16 Channel을 가진 A/D Converter가 사용되며 Potentiometer에서 출력되는 Analog신호를 Digital 신호로 바꾸어 중앙처리장치로 보내는 역할을 한다. 본 연구에서 사용된 A/D 변환 장치는 2 Channel의 D/A Converter 기능도 보유하고 있어서 외부의 계기를 통제할 수 있는 기능도 가지고 있다.

2.1.3 중앙처리장치

중앙처리장치는 각종 입력장치로부터 필요한 입력을 받아, Simulation에 필요한 모든 조종 운동 수학 모형의 계산을 수행하여 선박의 위치 및 자세 그리고 각종 변수를 Real-Time 또는 Real-Time보다 빠르게 계산하고 화면에 CGI(Computer Generated Imagery)를 구현하기 위한 2차원 및 3차원 그래픽 Algorithm을 계산하여 화면 출력 장치로 보낸다. 본 장치는 현재 국내에 광범위하게 보급되어 있는 IBM PC/AT 호환기종을 사용하였으며, 3차원 Perspective View Type Simulator를 구성할 경우는 Intel 80486 CPU(25 MHz, 11 MIPS)를 사용한 PC를 사용하여 주로 Engineering Workstation에서만 구현되던 3차원 컴퓨터 그래픽 알고리즘을 소형 PC에서 구현시켰다. 그리고 레이다 기능을 하는 2차원 시뮬레이터는 In-

tel 80286 CPU(16MHz)를 사용하는 PC를 사용하였다.

2.1.4 화면 출력 장치(Graphic Board)

화면 출력 장치는 크게 두 가지 종류로 나눌 수 있는데, 하나는 Hercules Mono Graphic Card, EGA, VGA와 같은 Graphic Controller Type Adapter이다. 일반적으로 많이 사용되는 이러한 그래픽 보드들은 가격이 비교적 저렴하고 사용법이 간단하여 프로그램 개발이 용이하다. 그러나 Raster 주사방식으로 컴퓨터 그래픽스 명령어를 처리하는 경우 선, 다각형, 원 등을 그리거나 칠할 때 Bitmap/Pixmap을 만들기 위한 계산을 Main CPU가 담당하므로 그래픽 업무가 많은 소프트웨어의 경우 전체적으로 처리속도가 떨어지게 되는 단점이 있다.

또 하나의 방법은 그래픽 보드에 그래픽 명령어 처리 전용 CPU와 전용 DRAM, VRAM을 설치하고 그래픽 명령어 처리를 이 CPU를 사용하여 병렬 처리 개념(Paralell Processing)으로 처리함으로써 위의 단점을 제거한 새로운 방식으로, Graphic System Processor(GSP)라고도 한다. 이러한 장비는 값은 비싸지만 일반 VGA Board의 8-10배의 속도로 그래픽 기능을 수행하므로 빠른 속도로 그래픽을 구현해야 하는 시뮬레이터에 적합한 장비이다.

화면 출력 장비로 중요한 또 하나는 Monitor이다. 본 연구에서는 14'와 21' 두가지 종류의 고해상도 Multi-Syne 모니터를 사용하였으며 더욱 현실감을 부여할 수 있도록 본 연구팀이 보유하고 있는 Beam Projector와 Screen도 사용할 수 있도록 하였다.

2.1.5 시뮬레이터용 근거리 통신망(LAN)

시뮬레이터 구성시 현실감 재현 및 실제 운항 중인 선박의 항해 장비의 동작 상태를 재현하기 위해서는 여러대의 PC를 사용하여 각각의 기능을 수행시켜야 한다. 특히 현실감 있는 3차원 Perspective View를 재현하기 위해서는 여러대의 영상출력용 PC와 모니터나 Beam-Projector를 통한 CGI 시스템 구성이 필요하게 된다. 또한 선원 훈련을 위한 경우도 교관이 임의의 돌발 사태를 부

여하기 위해서는 선원의 훈련 시뮬레이션 도중에 외부에서 시뮬레이션 상황을 변화시킬 수 있어야 한다. 이러한 기능을 구현하기 위하여 본 연구에서는 근거리 통신망(LAN)을 사용한 PC간의 통신 방법에 대한 연구를 수행하여 10MBPS Ethernet을 사용한 Ethernet Bus type 통신망을 구성하고 NetBIOS통신방식을 사용한 PC사이의 통신 프로그램을 개발하였다. 이 프로그램을 사용하여 3차원 영상 출력용 PC시스템과 레이더 기능을 하는 2차원 시뮬레이터 시스템을 연결하여 동시에 동작하도록 개발하였다.

2.1.6 시뮬레이터용 Data Base 제작/수정을 위한 User Interface용 장비

정확한 선박 조종 시뮬레이션을 위해서는 대상 선박과 대상 항구의 정확한 자료가 준비되어야 한다. 시뮬레이션을 수행할 항구의 지형을 海圖로부터 읽어서 항구 Data Base에 입력하는 작업은 복잡하고 시간이 많이 걸리므로 본 연구에서는 이 작업을 Digitizer를 사용하여 짧은 시간에 더 정확한 항구관련 Data Base를 제작할 수 있도록 하였다. 또한 시뮬레이션 중에도 Mouse나 Digitizer 등을 사용하여 항구의 지형, 수심 등을 변경할 수 있도록 관련 S/W와 H/W를 설계하였다.

2.1.7. Graphic Hardcopy 장비

Simulation 결과의 해석과 검토를 위하여 선박의 제적 및 각종 상태 변수(Heading Angle, Command Rudder Angle, Current Rudder Angle, Yaw Rate, Command Engine Throttle, Current Engine Throttle, Speed, Propeller RPM, Roll Angle, Drift Angle)들의 시간이력(Time History)을 흑백 또는 컬러로 Hardcopy할 수 있는 장비들로 Laser Printer(HP LaserJetIII)와 Color Ink Jet Printer(HP PaintJet)를 사용하였고 Plotter(HP-GL 또는 기타 방식)나 다른 프린터들도 지원 가능하도록 프로그램을 개발하였다. 시뮬레이션 결과의 다양한 해석과 항해자의 선박 운항 상태 및 결과 분석에 도움을 줄 수 있도록 하였다.

2.2 시뮬레이터의 Software 구성

본 Simulator의 S/W 구성도는 Fig. 2.2와 같으

며, 과거 본 연구소에서 개발한 바 있는 선박 조종 성능 추정 프로그램¹¹⁾ C언어로(Microsoft C, Ver. 6.0) 변환시키고 그 결과를 Color Monitor에 재현하기 위한 Graphic Display 부분과 여러가지 제어장치로부터 명령을 받아들이기 위한 H/W Interface 부분을 추가하여 개발된 DeskPilot System²¹⁾용 S/W에 Radar기능을 수행할 16MHz i286 PC와 통신을 하기 위한 NetBIOS 통신 프로그램을 추가하였다.

전체 Program은 3차원 Perspective View를 재현하는 컴퓨터와 2차원 Radar기능을 수행하는 컴퓨터에서 동작되는 각각의 Main Program과 여러개의 시뮬레이션 준비, 조종 성능 수치해석, H/W interface, 2차원 및 3차원 그래픽, 그리고 통신 Sub Program으로 이루어져 있는데, 각 Main Program은 Simulation의 전체 흐름을 제어한다. 또한, 각각의 Sub Program은 다시 기본적인 기능들을 가진 여러개의 Sub Module들로 이루어져 있는데 이는 후일 Program을 Module별로 개선시킬 수 있도록 하기 위함이다.

가. Bird's Eye View Type S/W

"N2DMAIN"이라는 프로그램을 수행시키면 "DeskPilot" 시스템의 2차원 시뮬레이터와 유사한 기능에 Network기능이 추가된 프로그램이 시작된다. 중요한 Subroutine들은 다음과 같다.

PREPARECOM1

NetBIOS 상에서 통신을 하기위해서 Server의 이름을 등록하고 Session 통신을 위한 준비를 한다. 또한 Client인 3차원 시뮬레이터의 동작 상태를 점검한다.

SHPCHO & HBRCHO

선박 및 항구의 Data File을 검색하여, Data File을 MENU형태로 보여주고 대상 선박 및 지역을 설정한다. 항구의 지형 및 수로의 형태를 화면에 출력하고, Game Area를 선택할 수 있도록 Zoom In/Out의 기능을 가지고 있다.

FORCES & INTGRATE

선박의 각 방향 속도, 자세, 위치 및 선수각, 수심 변화 및 해상상태 등으로부터 선박에 작용하는 각종 힘과 모멘트를 계산하고 그힘과 현재의 선박의 상태등으로 부터, 선박의 운동방정식을

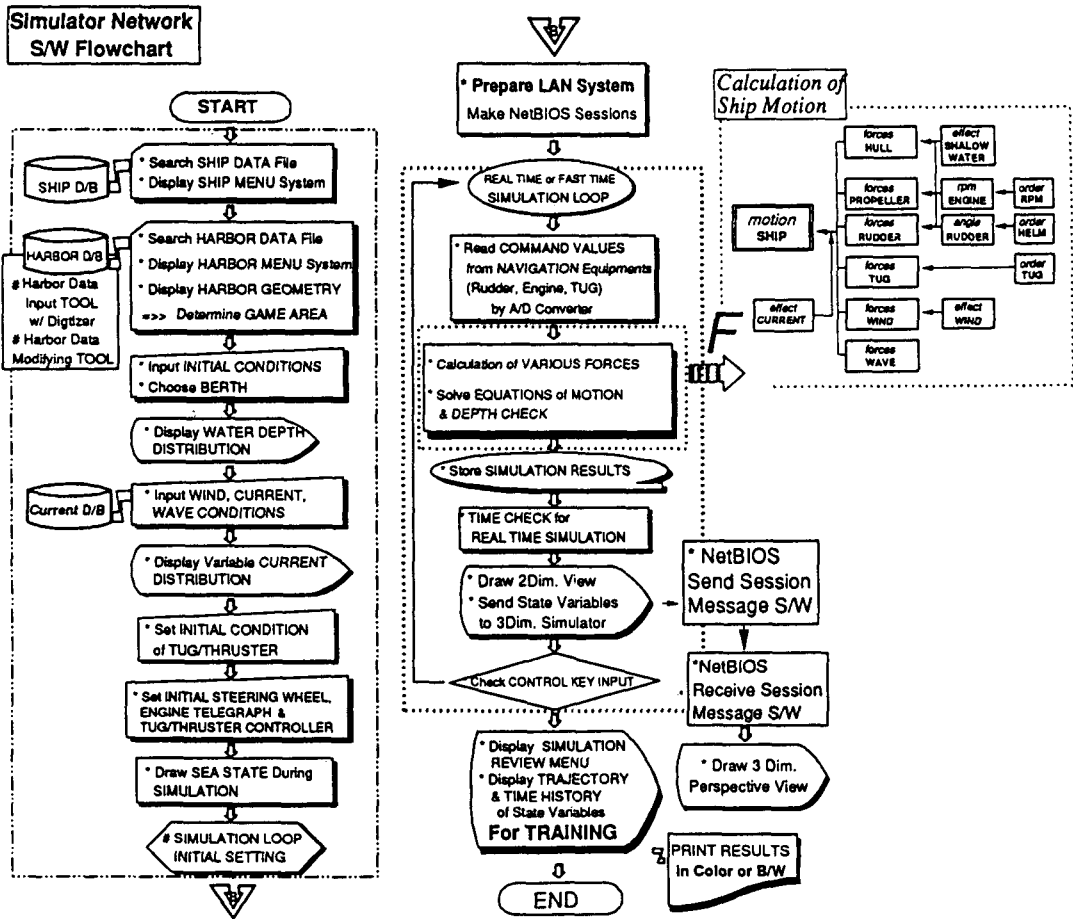


Fig. 2.2 S/W configuration

적분하여 선박의 새로운 위치 및 상태를 계산한다.

COMUNAVIDATA1

계산된 선박의 위치 및 선수각 및 기타 상태변수들을 3차원 시뮬레이터에 보내서 3차원 Perspective View를 재현시킨다.

또한 다른 시뮬레이터로부터 데이터를 받을 수 있도록 설계되어 있으므로 여러 시뮬레이터를 연결하여 서로 데이터를 주고 받을 수 있다. 시뮬레이션 도중에 여러 상황을 부여가능.

DRRADAR

Bird's Eye View 방식(2차원)으로 시뮬레이션 중인 선박의 위치 및 운항 상태를 화면에 표시해주는 Sub Program이다. Tug Boat의 사용 여부에

따라 계기판 배치가 달라진다.

PLTMENU

Simulation이 종료되었을 때, 기록된 결과의 검토를 위한 Sub Program이다. 시뮬레이션 대상 선박의 궤적 및 운항 상태변수들의 Time history를 그래픽으로 나타내준다.

HARBOCOR

시뮬레이션을 위하여 개발된 항구 데이터베이스의 지형 자료를 수정할 수 있도록 개발된 Sub Program이다. 항구 지형 데이터의 첨가나 삭제가 가능하여 항구 부근의 해안선을 간단히 변경할 수 있고 수심도 화면상에서 간단히 수정할 수 있어서 항구 개발을 위한 시뮬레이션에 유용하게 사용될 수 있다.

나. 3차원 Perspective View Type 시뮬레이터 S/W

“TN3DMAIN”이라는 프로그램을 수행시키면 2차원 시뮬레이터에서 계산된 선박의 위치 및 선수각, 그리고 상태 변수들을 Network를 통하여 받아서 3차원 Perspective Viwe를 재현하는 프로그램으로 기능은 “DeskPilot”²¹⁾의 3차원 시뮬레이터에 다음의 Subroutine이 첨가된 것 외에는 비슷하다.

PREPARECOM2

NetBIOS 상에서 통신을 하기위하여 Client의 이름을 등록하고 Session 통신을 위한 준비를 한다. 또한 Server인 2차원 시뮬레이터의 동작 상태를 점검한다.

COMUNAVIDATA2

계산된 선박의 위치 및 선수각 및 기타 상태변수들을 2차원 시뮬레이터로 부터 받아서 3차원 Perspective View를 재현시킨다. 여러대의 시뮬레이터를 연결하여 서로 데이터를 주고 받을 수 있다.

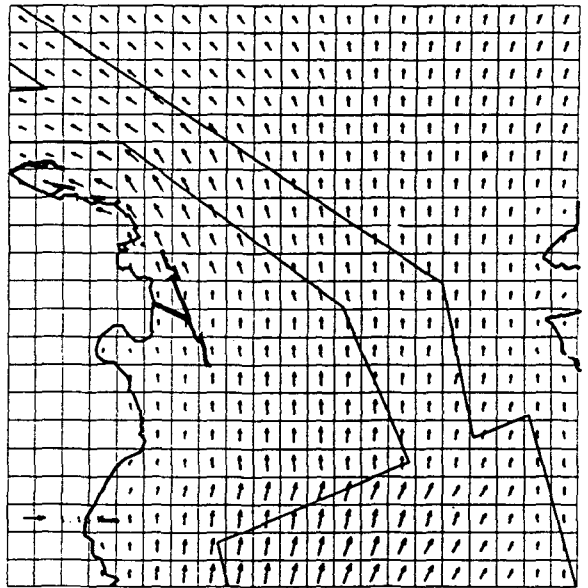


Fig. 2.4 Sample Current Map

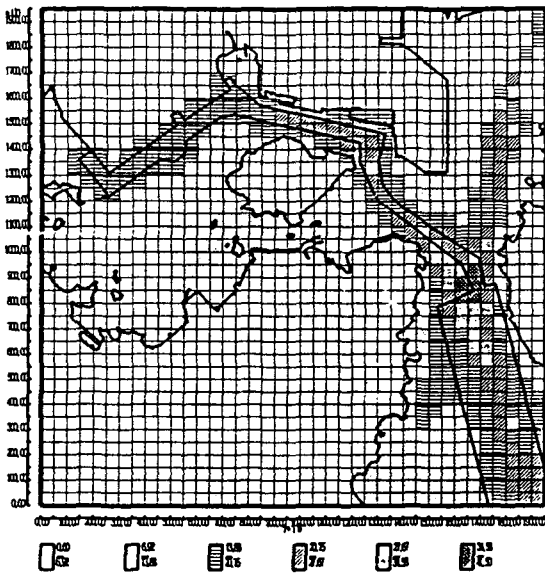


Fig. 2.3 Sample Depth Map

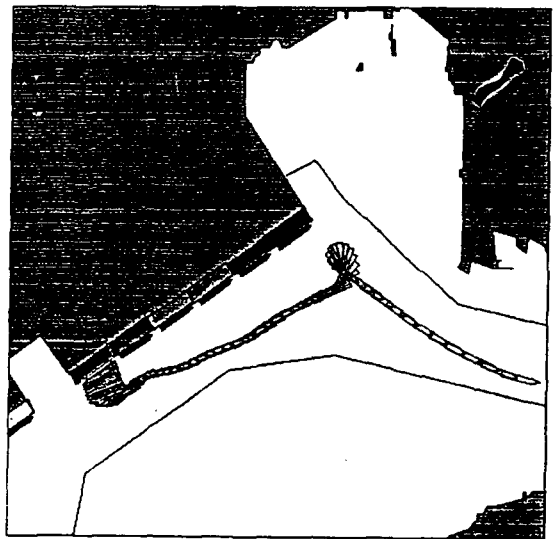


Fig. 2.5 Sample Trajectory of Container ship

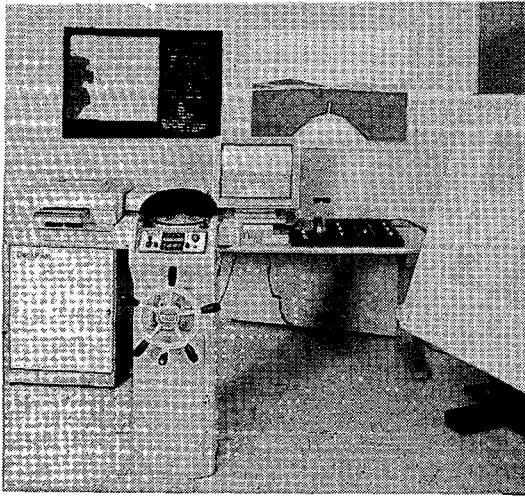


Fig. 2.6 DeskPilot System

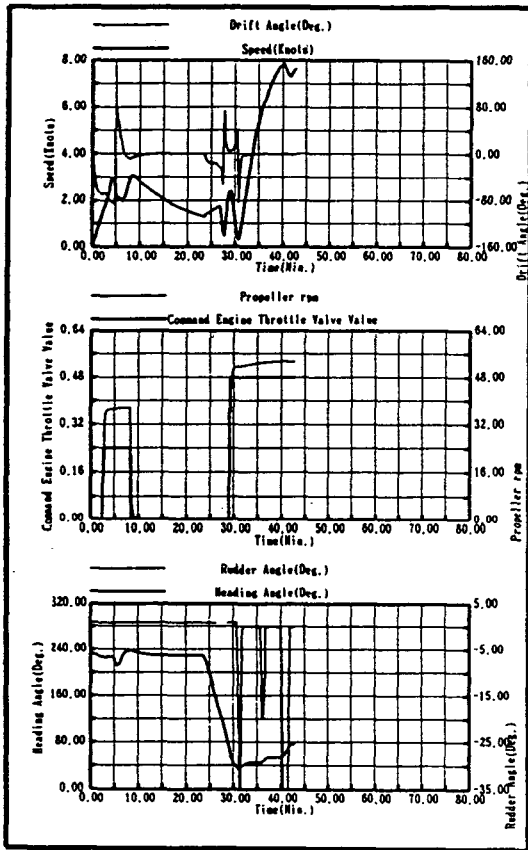


Fig. 2.7 Sample Time History of State Variables

3. Real-Time 선박 조종 성능 해석 프로그램

시뮬레이터에 사용되는 수학 모형은 Computer graphics 기법 및 H/W Interface 기술과 함께 선박 조종 시뮬레이터의 3대 핵심 요소라 할 수 있다. 조종 운동중에 선박이 겪게 되는 여러가지 상황에서 선체에 작용하는 각종 힘들을 정확하게 계산하고, 이를 바탕으로 선체의 복잡한 운동 방정식을 해결함으로써 선박의 위치 및 운동 상태를 신속하게 구할 수 있는 능력은 시뮬레이션 결과의 정확도에 결정적인 영향을 미친다고 할 수 있다. 하지만 여러가지 복잡한 상황에서 선체에 작용하는 힘들을 이론적으로 정확히 추정하는 데에는 아직 한계가 있으며 따라서 선박의 조종성 추정에 사용되는 대부분의 계수를 실험적으로 구하거나 조종 성능 D/B 혹은 경험식에 의존할 수밖에 없는 형편이다.

본 시뮬레이터에서는 실험 자료가 있을 경우에는 여기서 얻어진 자료를 사용하여 필요한 힘들을 계산하도록 되어 있고, 실험 자료가 없는 경우에는 여러가지 경험식을 사용하여 선박의 기본적인 특성치만으로도 필요한 계산을 수행할 수 있도록 되어 있다.

3.1 물리적 현상 가정

선박의 운동은 선박에 작용하는 각종 힘을 포함한 6 자유도의 운동 방정식으로 표현된다. 하지만 수상선의 조종 운동 방정식은 대개 자유표면으로 인한 이력 효과를 무시하고⁴⁾ 정수중에서의 낮은 Froude수를 가정하여 전후동요(Surge), 좌우동요(Sway), 선수동요(Yaw) 그리고 횡동요(Roll)의 4 자유도의 연성 운동 방정식으로 표현하는 것이 보통이다. 또한 선박에 작용하는 힘에는 선박의 관성력, 추진력, 각종 유체력, 조타력 그리고 풍력과 조류력, 파도에 의한 표류력 등이 있으며 이외에도 淺水 效果와 制限 水路의 영향, 인접 선박의 영향 등이 포함될 수 있다.

3.2 좌표계

본 시뮬레이터에서 사용한 좌표계는 Fig. 3.1 과 같다. 여기서 $O_0-X_0Y_0Z_0$ 는 공간 고정 좌표계 이며 $O-XYZ$ 는 원점이 선체의 무게 중심에 있는 선체 고정 좌표계이다. 또한 본 연구에서 사용되는 모든 각도의 정의는 +X축으로부터 시계 방향으로의 각을 양으로 정의하였다.

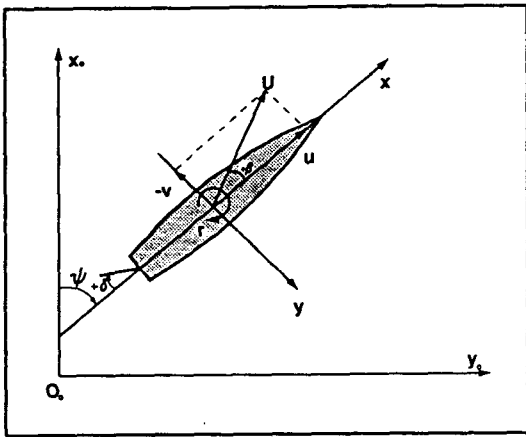


Fig. 3.1 Coordinate

3.3 조종 운동 방정식

원점 O를 무게중심 G에 놓고, x, y 방향 힘 X, Y와 z축을 중심으로 한 모멘트 N을 성분별로 구분하면 선체 고정 좌표계에 관한 선박의 운동방정식과 프로펠러 회전에 대한 방정식은 다음과 같다.

전후 동요 :

$$m(\dot{u} - vr) = X_H + X_P + X_R + X_{WA} + X_{WI} + X_T$$

좌우 동요 :

$$m(\dot{v} - ur) = Y_H + Y_P + Y_R + Y_{WA} + Y_{WI} + Y_T + Y_{HC}$$

선수 동요 :

$$I_{zz}\dot{r} = N_H + N_P + N_R + N_{WA} + N_{WI} + N_T + N_{HC}$$

프로펠러 :

$$2\pi I_{pp}\dot{n} = Q_p + Q_o \tag{3.1}$$

여기서 첨자 H, P, R, T, WA, WI는 각각 선체,

프로펠러, 타, 예선 그리고 파도, 바람에 의한 항을 나타내며, Q_p 는 주기관의 토오크, Q_o 는 프로펠러 토오크를 표시한다. 또한 위에서 모든 유체력은 선박의 유체에 대한 상대적 운동으로부터 얻어지므로 조류의 영향은 별도로 선체 고정 좌표계와 공간 고정 좌표계 사이에서 고려된다.

3.1.1 선체에 작용하는 유체력

선체에 작용하는 유체력 X_H, Y_H, N_H 는 다음과 같이 쓸수 있다.

$$\begin{aligned} X_H &= X_{\dot{u}} \dot{u} + (-Y_{\dot{v}} + X_{vr})vr + X(u) \\ Y_H &= X_{\dot{v}} \dot{v} + X_{\dot{u}} ur + Y_{HO}(v, r) \\ N_H &= N_{\dot{r}} \dot{r} + N_{HO}(v, r) + Y_{HO}(v, r) + x_o \dots \tag{3.2} \end{aligned}$$

여기서 X_H 의 마지막 항 $X(u)$ 는 선속 u 의 함수로 표시되는 선체저항으로서 본 연구에서는 EHP-Curve로부터 그 특성을 고려하기로 하며 후진시 저항은 같은 속도에서의 전진시 저항의 1.3배로 추정한다.

$Y_{HO}(v, r), N_{HO}(v, r)$ 은 횡요 효과를 포함하지 않는 기본적인 힘과 모멘트를 나타내며 다음과 같이 표시할 수 있다고 가정하였다.

$$\begin{aligned} Y_{HO}(v, r) &= \frac{1}{2} \rho L^2 U^2 (Y'_{\dot{v}} \dot{v} + Y'_{\dot{r}} \dot{r} + Y'_{\dot{v}|\dot{v}} |\dot{v}'| |\dot{v}'| \\ &\quad + Y'_{\dot{v}|r} \dot{v}' |r'| + Y'_{\dot{r}|r} \dot{r}' |r'|) \\ N_{HO}(v, r) &= \frac{1}{2} \rho L^3 U^2 (N'_{\dot{v}} \dot{v} + N'_{\dot{r}} \dot{r} + N'_{vrr} v^2 \dot{r} + \\ &\quad N'_{vr} v \dot{r}^2 + N'_{r|r} \dot{r}' |r'|) \dots \tag{3.3} \end{aligned}$$

3.3.2 프로펠러 추진력과 토오크

프로펠러 단독특성과 선미에서의 흐름상태로부터 반류비, 추력감소율, 추력계수, 토오크계수 등이 결정된다. 이 계수와 프로펠러면에서의 유속과 방향으로부터 추력과 토오크를 구할 수 있게 된다.

선미 반류분포의 불균일성과 프로펠러 회전에 의한 힘 Y 및 N은 실험적으로 혹은 다음 절의 방법에 의한 타력 계산시 타각 0°에서의 타수직력으로부터 구할 수 있다. 즉, 프로펠러에 의한 선체 유기력은 타각 0°에서의 타력으로 볼 수 있다. 1개 이상의 추진기를 장치한 선형에서도 프로펠러의 회전방향에 따라 힘 Y의 방향을 고려하

여 대수적으로 합하면 된다.

Simulation 모형에서는 프로펠러 역회전시의 추력과 토오크 계산을 위해 Patel⁶⁾에 의해 제시된 회귀해석 형태를 전진방향 K_T , K_Q 에 적용한다.

$$\begin{aligned}
 K_T &= K_{T00} + K_{T0} \frac{J}{|J|} + K_{T1} J + \frac{1}{2} K_{T2} J^2 \\
 K_Q &= K_{Q00} + K_{Q0} \frac{J}{|J|} + K_{Q1} J + \frac{1}{2} K_{Q2} J^2 \\
 K_T &= \frac{T}{\rho n^2 D^4} \quad K_Q = \frac{T}{\rho n^2 D^5} \quad J = \frac{u(1-w_p)}{nD}
 \end{aligned}
 \tag{3.4}$$

추력 계수 K_T 와 토오크 계수 K_Q 는 프로펠러 단독 특성 곡선으로부터 전진계수 J 의 함수로 나타낼 수 있다. 추력일치법(Thrust Identity)에 의해 정의되는 w_p 는 여러가지 조종운동시 프로펠러면에서 입사각의 변화에 따라 변하게 되는데, 모형시험으로부터 얻어진 다음의 실험식이 쓰여지고 있다.

$$\begin{aligned}
 w_p &= w_{p0} \exp(K_1 \beta_p^2) \\
 K_1 &= -4.0 \\
 \beta_p &= \beta - \alpha_p' \dots \dots \dots \tag{3.5}
 \end{aligned}$$

프로펠러의 추력, 토오크 계수의 모형 시험 결과가 주어지는 경우에는 J 에 따른 값을 순정(Cure Fitting)하여 사용하고 프로펠러의 주요 치수만 주어지는 경우는 van Lammeren⁸⁾ B-series 프로펠러의 특성으로부터 Interpolation한 값을 쓴다. 본 모형에서는 프로펠러 추력 X_p 와 토오크 Q_p 를 다음과 같이 쓴다.

$$\begin{aligned}
 X_p &= (1-t) \rho n^2 D^4 K_T(J_p) \\
 Q_p &= -2\pi J_p r^3 - \rho n^2 D^5 K_Q(J_p)
 \end{aligned}$$

where

$$J_p = u(1-w_p)/(nD) \dots \dots \dots \tag{3.6}$$

3.3.3 추진기관 모형화

주기관의 토오크 특성은 기관의 종류에 따라 여러가지 형태가 있으며, 동일한 기관에 대해서도 작동 조건에 따라 다음과 같이 분류된다.

- * 정상 주행상태(Normal Running Condition)
- * 연료공급 중단상태(Fuel Cutoff Condition)
- * Braking Condition with Brake-Air

한 예로 저속 디젤 기관과 증기 터빈의 정상 주행상태의 토오크는 다음과 같다.

◇ Diesel engine :

$$\begin{aligned}
 Q_e &= |Q_p| \quad \text{for } |Q_p| \leq Q_{EMAX} \\
 Q_e &= Q_{EMAX} \quad \text{for } |Q_p| > Q_{EMAX}
 \end{aligned}$$

◇ Steam turbine :

$$Q_E = SHP / 2\pi n \dots \dots \dots \tag{3.7}$$

3.3.4. 타력과 타 모멘트

타력과 타모멘트 그리고 타에 의해 선체에 유기되는 유체력과 모멘트는 일본 MMG의 다음과 같은 경험식을 사용할 수 있다.

$$\begin{aligned}
 X_R &= -F_N \sin \delta \\
 Y_R &= -(1+\alpha_H) F_N \cos \delta \\
 N_R &= -(1+\alpha_H x_R F_N \cos \delta \dots \dots \dots \tag{3.8}
 \end{aligned}$$

한편, 타 수직력 F_N 은 아래와 같으며,

$$F_N = \frac{1}{2} \rho \frac{6.13\lambda}{\lambda + 2.25} A_R V_R^2 \sin \alpha_R \dots \dots \dots \tag{3.9}$$

V_R 과 α_R 은 타에 유입되는 유체의 유효 유입속도와 유효 유입각을 나타낸다. 이러한 수학 모형은 선체가 전진방향으로 진행하며 타가 프로펠러 후류속에 있는 경우에 관한 것이므로 본 Simulation 모형에서는 정지 조종, 후진 상태로 부터의 가속 조종 후진 조종도 다음과 같이 포함한다. 선속과 프로펠러 회전 방향에 따라서 다음과 같이 4가지 경우로 나눌 수 있다.

(1) $u > 0, n > 0$:

정상적인 전진 상태로 식 (3.8), (3.9)에 의하여 타력과 모멘트를 계산한다.

(2) $u > 0, n < 0$:

감속 또는 정지 조종시의 경우로서 타 주위의 흐름이 Braking 현상을 일으키며, 이 때는 타력이 존재하지 않는다.

(3) $u < 0, n > 0$:

후진으로부터 전진 방향으로 가속하는 상태에서 프로펠러 후류의 영향이 크게 작용한다.

(4) $u < 0, n < 0$:

정상 후진의 경우로 타주위의 흐름은 개방된

유체흐름과 같고 속도는 선속과 같다고 가정한다.

3.3.5 풍력과 모멘트

풍력과 이로 인한 모멘트는 다음과 같이 표현 된다.

$$\left. \begin{aligned} X_{Wf} &= \frac{1}{2} C_x \rho_a V_{rw}^2 A_T \\ Y_{Wf} &= \frac{1}{2} C_y \rho_a V_{rw}^2 A_L \\ N_{Wf} &= \frac{1}{2} C_N \rho_a V_{rw}^2 A_L L_{OA} \end{aligned} \right\} \dots\dots\dots (3.10)$$

여기서 A_T, A_L 은 각각 수면위의 횡방향, 종방향 투영면적이고, C_x, C_y, C_N 은 상대풍향각 및 선체 형상의 함수인데, Isherwood의 경험식⁶⁾으로부터 구할 수 있다.

3.3.6 파랑력과 모멘트

파랑력은 1차 파랑력(First Order Forces)과 2차 파랑력(Second Order Forces)으로 구분되는데, 선체는 큰 관성력 때문에 Lowpass Filter와 같은 작용을 한다. 따라서 1차 파랑력은 보통 무시되고 있으므로, 본 연구에서는 2차 파랑표류력만을 고려하기로 한다. 파랑 표류력이 파고의 자승에 비례한다고 하면 파랑력, 파랑 모멘트 계수는 다음과 같다.

$$\left. \begin{aligned} X_{WA} &= \frac{1}{2} C_{DX} \rho g L \zeta_a^2 \\ Y_{WA} &= \frac{1}{2} C_{DY} \rho g L \zeta_a^2 \\ N_{WA} &= \frac{1}{2} C_{DN} \rho g L^2 \zeta_a^2 \end{aligned} \right\} \dots\dots\dots (3.11)$$

여기서 C_{DX}, C_{DY}, C_{DN} 은 일반적으로 파장과 조우파각 그리고 조우주파수를 고려한 선속의 함수로 생각된다. 다만, 선속을 고려한 계수는 실험적으로나 이론적으로 충분한 연구가 이루어지지 못하였으므로 본 연구에서는 속도가 0인 경우의 실험 결과를 사용하기로 한다.

Hirano¹⁰⁾는 Y_{WA} 만을 고려했을 때에도 X_{WA}, Y_{WA}, N_{WA} 를 모두 고려한 경우와 큰 차이가 없음을 보인다. 횡방향 파랑력 계수 C_{DY} 는 다음과 같이 쓸 수 있다.

$$C_{DY} = C_{DY}(\alpha_{WA} = 90^\circ) \sin(\alpha_{WA} - \psi) | \sin(\alpha_{WA} - \psi) | \dots\dots\dots (3.12)$$

3.3.7 조류의 영향

본 연구에서는 위치에 따라 조류의 방향과 속도가 변할 수 있다는 가정하에 그 영향을 선체의 운동에 고려하였다. 조류의 속도와 선체의 길이 방향으로의 조류의 변화가 크지 않다고 가정하고, 선체의 유체에 대한 상대 속도와 상대가속도를 $\dot{u}_r, \dot{v}_r, u_r, v_r$ 로 표시하면 선체의 해류에 대한 영향을 다음과 같다.

$$\left. \begin{aligned} \dot{u} &= \dot{u} - u_r \sin(\alpha_c - \psi) \\ \dot{v} &= \dot{v} - v_r \cos(\alpha_c - \psi) \\ u_r &= u - u_c \cos(\alpha_c - \psi) \\ v_r &= v - v_c \sin(\alpha_c - \psi) \end{aligned} \right\} \dots\dots\dots (3.13)$$

3.3.8 예선에 의한 힘과 모멘트

예선에 의한 힘과 모멘트는 다음식에 의하여 추정할 수 있다.

$$\left. \begin{aligned} X_T &= \sum_{i=1}^N (F_i)_{max} R_i \cos \theta_i \\ Y_T &= \sum_{i=1}^N (F_i)_{max} R_i \sin \theta_i \\ N_T &= \sum_{i=1}^N (F_i)_{max} R_i (x_i \cos \theta_i - y_i \sin \theta_i) \end{aligned} \right\} \dots\dots (3.14)$$

여기서 N 은 사용된 예선의 총 수를 나타내고 $(F_i)_{max}$ 는 각 예선의 최대 용량을, 그리고 R_i 는 각 예선의 출력비를 나타낸다. 또한 θ_i 는 예인 방향을 나타내며, (x_i, y_i) 는 예인 점의 (x, y) 좌표를 각각 나타낸다.

3.3.9 Cross Flow Drag에 의한 힘과 모멘트

선박이 저속으로 움직일 경우, 횡방향 및 회전 방향의 속도가 전진 속도에 비하여 상대적으로 커지게 되며 따라서 선박에 작용하는 비선형 항력이 중요하게 된다. 이러한 힘들은 Oltmann¹¹⁾에 의하여 다음처럼 모형화할 수 있다.

$$Y_{HC} = - \frac{\rho}{2} \int_{x_{AP}}^{x_{FP}} T(x) C_{CFD}(x) (v + rx) |v + rx| dx$$

$$N_{HC} = -\frac{\rho}{2} \int_{x_{AP}}^{x_{FP}} T(x) C_{CFD}(x) (v+rx) |v+rx| dx \dots\dots\dots (3. 15)$$

여기서 T(x)는 각 단면에서의 흡수를 나타내고 C_{CFD}(x)는 국부 Cross Flow Drag을 나타내는 계수이다.

4. 視界 再現 컴퓨터 그래픽 기법 개발

이번 절에서는 시뮬레이터의 중요 구성 부분의 하나인 視界 再現 및 선박 운항 상태 재현에 사용된 컴퓨터 그래픽 기법에 대하여 설명한다. 본 연구에서는 25MHz : 486 PC 상에서 Microsoft社의 MS-C ver. 6.0과 TIGA 2.0 SDK, 그리고 Media Cybernetic社의 HALO Professional이라는 범용 Graphic Kernel System을 개발 도구로 사용하여 시뮬레이터용 3차원 컴퓨터 그래픽 소프트웨어를 자체 개발하였다.

4. 1. 3차원 CG의 기초 이론

4. 1. 1. 좌표계

3차원 좌표계는 왼손 좌표계인 視点 좌표계, 오른손 좌표계인 World Coordinate가 있다.

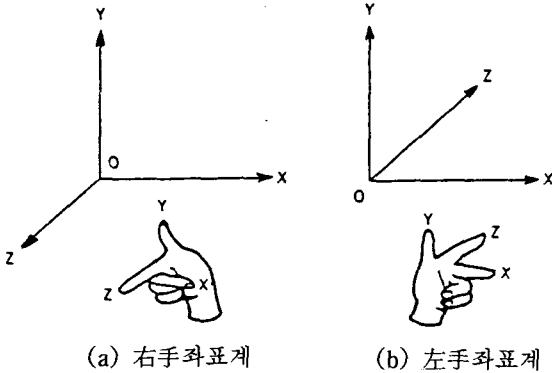


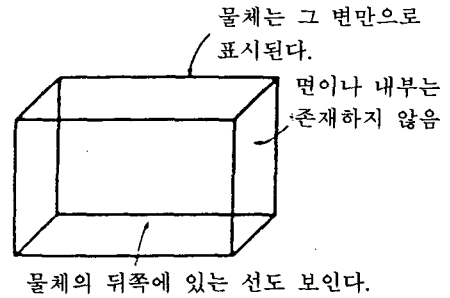
Fig. 4. 1. 3D Coordinate System

4. 1. 2. 형상 Modeling

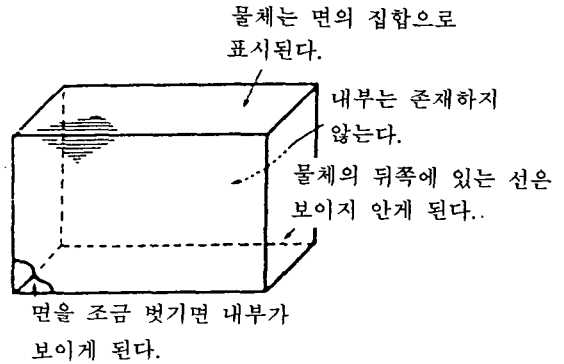
3차원 도형을 정의하는 것을 형상 Modeling이

라 하고 다음과 같이 구분할 수 있다.

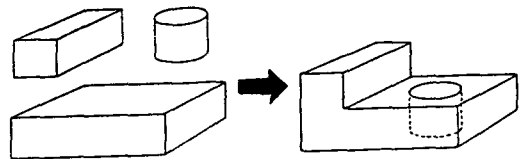
- ① Wireframe Modeling : 물체의 윤곽선으로 물체를 정의하는 방법, 데이터 구조나 처리가 단순하다. 본 연구의 3차원 Perspective View 방식 시뮬레이터에 사용했다.
- ② Surface Modeling : 다각형(Polygon)면의 집합으로써 물체를 정의하는 방법
- ③ CSG Modeling(Constructive Solid Modeling) : 기본적인 몇개의 3차원 물체(원, 원뿔, 직육면체 등)를 이용하여 임의의 3차원 물체를 정의하는 방법



(a) Wire Frame Model



(b) Surface Model



(c) Constructive Solid Model

Fig. 4. 2 3-DIM. Modelings

Wireframe Modeling에서 물체의 각 꼭지점의 좌표와 이것들을 연결한 변의 접속 형태를 Data 로써 정의한다.

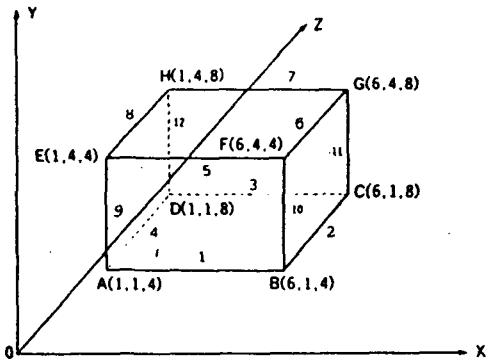


Fig. 4.3 Object in View Coordinate(Right Hand Coord.)

Table 4.1 Wire Frame Modeling
(Data Structure of Fig. 4.3)

꼭지점	좌 표 값			변번호	꼭지점연결
	X	Y	Z		
A	1	1	4	1	A-B
B	6	1	4	2	B-C
C	6	1	8	3	C-D
D	1	1	8	4	D-A
E	1	4	4	5	E-F
F	6	4	4	6	F-G
G	6	4	8	7	G-H
H	1	4	8	8	H-E
				9	A-E
				10	B-F
				11	C-G
				12	D-H

4.1.3. 3차원 도형의 좌표 변환

3차원 공간에서 물체를 이동하는 것은 3차원 좌표 변환을 사용한다. 고려하는 좌표계를 시점 좌표계(左手系)로 한 경우, 어떤 점의 본래 좌표 값을 (x, y, z), 변환 행렬을 T, 변환 후의 좌표치를 (x, Y, Z)라고 하면, 좌표 변환의 일반식은 다음과 같다.

$$[X \ Y \ Z] = [x \ y \ z] T \dots\dots\dots (4.1)$$

여러가지 경우에 있어서 변환행렬 T는 다음과 같이 정의된다.

- ◇ 평행이동 (x축 방향 이동량 l, y축 방향 이동량 m, z축 방향 이동량 n,)

$$T_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \dots\dots\dots (4.2)$$

- ◇ 회전이동 (x축 회전)

$$T_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots (4.3)$$

- ◇ Scaling(확대율 s)

$$T_s = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots (4.4)$$

- ◇ 회전이동 (y축 회전)

$$T_y = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots (4.5)$$

- ◇ 회전이동 (z축 회전)

$$T_z = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots (4.6)$$

이상의 변환 행렬을 연결함으로써 더욱 복잡한 좌표 변환을 할 수 있게 된다.

4.1.4. 3차원 Clipping

3차원 공간에서의 Clipping은 시점에서의 視野의 상하 좌우와 視野의 전후면에서 이루어지는 視野 피라미드의 외부면에 따라 이루어진다. (Fig. 4.4 참조) 視野의 전후면에 따라 Clipping을 한 다음, 視野 피라미드의 Clipping을 하는 순서로 처리한다.

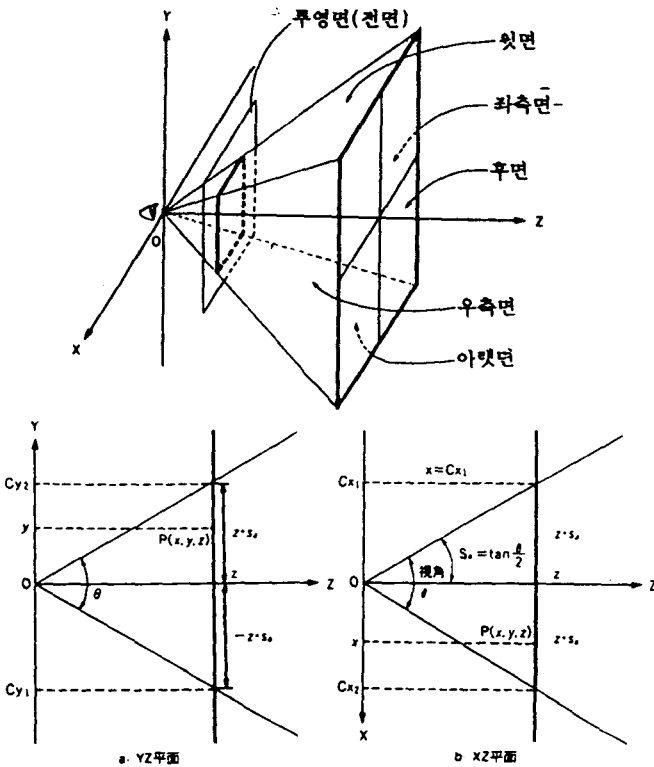


Fig. 4.4 Front/Back face Clipping in View Volume

① 視野의 前後面の Clipping

처리할 변의 꼭지점을 $(x_1, y_1, z_1), (z_2, y_2, z_2)$ 視点과 전면 및 후면까지의 거리를 $z=h, z=f$ 라 하고 Fig. 4.5와 같이 평면을 3구역으로 나누어 거기에 두 점을 주고 이 두점의 2bit code를 C_1, C_2 라고 하면, 다음에 표시되는 순서도(Fig. 4.6)의 방법으로 視野의 전후면의 Clipping 처리가 된다.

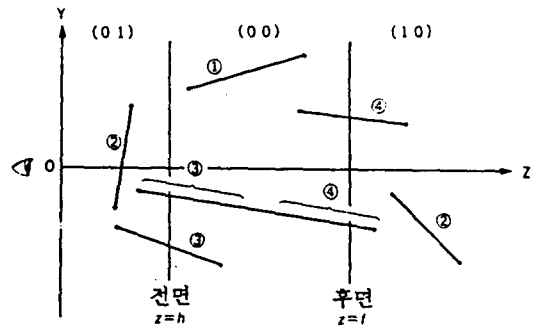


Fig. 4.5 Front/Back face Clipping by 2 bit Code

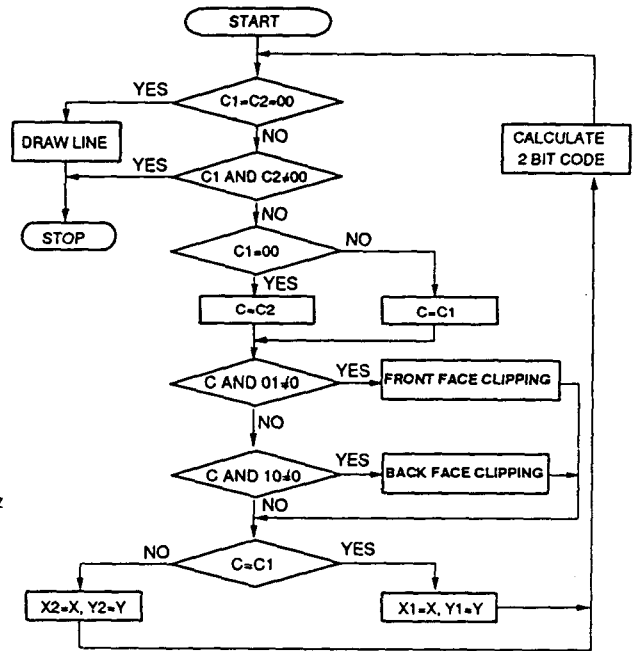


Fig. 4.6 Flowchart of Front/Back face Clipping

② 視野 피라미드의 側面の Clipping

視野 각도를 $2 \cdot S_a$ 처리하는 변의 꼭지점을 $(x_1, y_1, z_1), (x_2, y_2, z_2)$ 로 하면, 2차원 Clipping의 경우와 같이 視野 피라미드의 側面の Clipping 처리를 할 수 있게 된다. 이것은 Fig. 4.7의 4bit Out-code를 써서 순서도의 방법으로 처리하는 것으로 視野 각도의 반을 S_a 라 한다면, 이렇게 3차원 Clipping에 의하여 잘라 내어진 視野 空間을 3차원 View Volume이라고 한다.

(1001)	(1000)	(1010)
(0001)	(0000) 표시영역	(0010)
(0101)	(0100)	(0110)

$x = -S_a \cdot z$ $x = S_a \cdot z$

Fig. 4.7 Clipping by 4bit Outcode

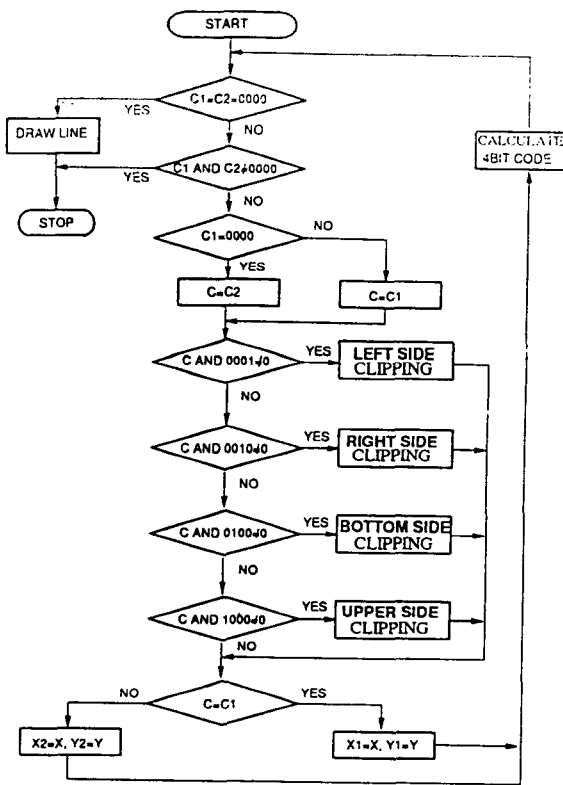


Fig. 4.8 Flowchart of View-Volume Side Clipping

4.1.5. 透視 投影

3차원 도형을 2차원으로 표현하는 것을 投影이라고 하고, 제도등에 쓰이는 平行 投影과 Animation 영화 등의 영사에 쓰이는 遠近法을 도입한 中心 投影이 있다. 視点 좌표계에서 점(x, y, z)를

視点에서 距離 h의 투영면에 중심 투영하면, 그 좌표(X, Y, Z)는 식 (4.7)으로 나타낼 수 있다.

$$\left. \begin{aligned} X &= \frac{hx}{z} \\ Y &= \frac{hy}{z} \\ Z &= z - h \end{aligned} \right\} \dots\dots\dots (4.7)$$

4.2 3차원 그래픽 처리를 위한 좌표변환

이번 절에서는 3차원 도형의 형상을 정의하는 경우의 좌표계의 접근 방식으로, 도형 고유의 좌표계인 Body좌표계와 3차원 공간 내의 도형의 배치를 정의하는 World 좌표계에 대하여 소개하는 한편, 視点 座標도 포함한 相互 좌표 변환에 대해서 설명한다. 움직임이 있는 도형(시뮬레이터의 경우)를 취급하는 경우는 이처럼 형상을 정의하는 부분과 위치를 정의하는 부분, 그리고 관찰자(視点)의 위치를 나누어 생각하는 것이 일반적이다.

4.2.1 Body 좌표계와 World 좌표계

가. Body 좌표계

Body 좌표계는 정의한 도형의 어느곳에(일반적으로는 꼭지점이나 중심) 기준점을 두어, 그것을 원점으로 하는 것이다. 이렇게 도형 데이터를 정의하여 두면, 도형이 이동하여도 그 도형의 데이터는 변화하지 않는다는 특징이 있다. Body 좌표계의 데이터를 변화시킴으로써 다른 도형과는 독립하여 변화시킬 수도 있다. 앞으로는 Body 좌표계의 각 좌표계를 X_B, Y_B, Z_B 로 표시하고 오른손 좌표계를 사용하여 설명한다.

나. World 좌표계

Body 좌표계는 도형마다 독립하여 설정하므로, 도형이 여러개가 있으면, Body 좌표계도 그만큼 존재한다. 한편, Body 좌표계에서 정의된 도형을 3차원 공간내에 배치하기 위해서는 별도의 좌표계가 필요하게 된다. 이 좌표계를 World 좌표계라고 하고 각 좌표계를 X_W, Y_W, Z_W 로 표시한다.

이렇게 두개의 좌표계를 사용함으로써 각각의

도형의 배치를 정의한다면 각 도형의 Body 좌표계가 World좌표계와 어떠한 위치관계가 있는가를 정의하는 것만으로 단순하게 좌표변환의 처리를 시행할 수 있다(Fig. 4.10) World 좌표계는 일반적으로 右手系가 사용되고 있다.

이것은 예를 들어 방안에 가구나 짐을 배치하는 것처럼 공간내(World 좌표계상)에서 물체(Body 좌표계상의 도형)를 이동(좌표변환)하는 것으로 생각하면 쉽다.

Modeling 변화의 변환 행렬을 T_m 으로 한 경우

$$T_m = T_s, T_r, T_d \dots\dots\dots (4.8)$$

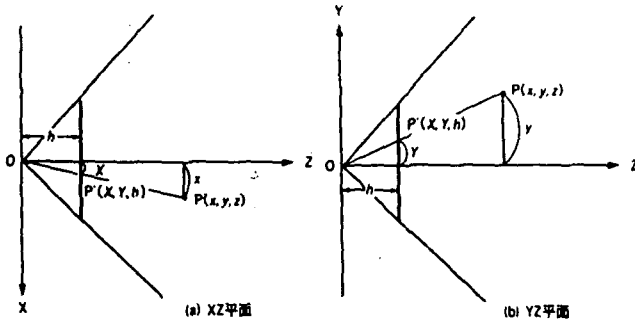
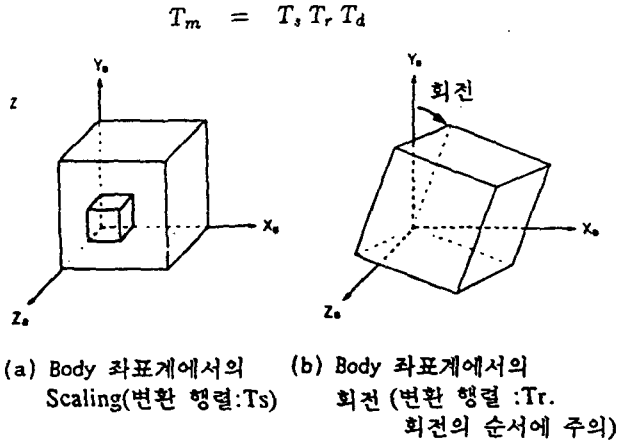


Fig. 4.9 Perspective Projection



(a) Body 좌표계에서의 Scaling(변환 행렬:Ts) (b) Body 좌표계에서의 회전(변환 행렬:Tr. 회전의 순서에 주의)

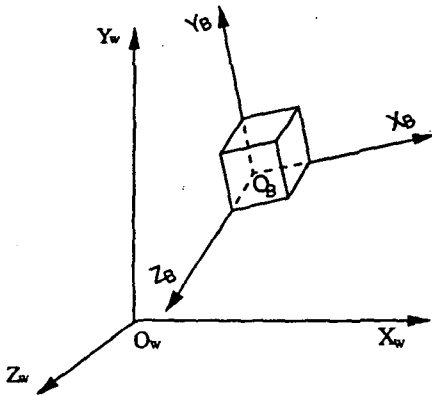


Fig. 4.10 Body Coordinate and World Coordinate

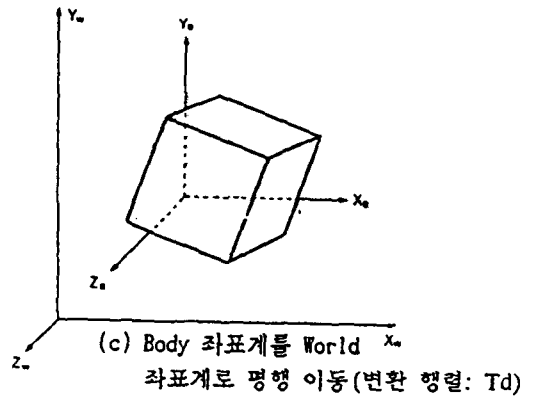


Fig. 4.11 Modeling sequence

4.2.2 좌표 변환의 원리

여기서는 Body 좌표계에서 World좌표계로의 변환 (Modeling 변환)과 World좌표계에서 시점 좌표계로의 변환(視野 變換)의 원리를 설명한다.

가. Modeling 변환(Body 좌표계 ==> World 좌표계)

Body좌표계로 정의한 도형을 World 좌표계상에 배치하는 처리를 Modeling 변환이라 부른다.

나. 視野 變換(World 좌표계 ==> View 좌표계)

3차원 물체는 World 좌표계상에 표시된다. 그러나, 우리가 물체를 볼때에는 우리의 눈(視点)을 중심으로 하는 새로운 좌표계(視点 좌표계)상에서 그것을 본다고 할 수 있다. 그러므로 어떤 물체가 정지해 있고 우리가 그것을 서로 다른 장소

에서 볼 경우, 물체에 대한 World 좌표계 상에서의 좌표는 변함이 없지만 視野좌표계 상에서의 좌표는 관찰자의 움직임에 따라 변할 것이다. 이와 같이 시점에 따라 달라지는 어떤 물체를 2차원적으로 표현하기 위해서는 먼저 World 좌표계 상에 정의된 물체의 좌표를 視点を 원점으로 좌표로 바꾸어야 한다. 이 과정을 視野 變換(Viewing Transformation)이라고 한다.

처리 순서는 다음과 같다. (Fig. 4.12 참조)

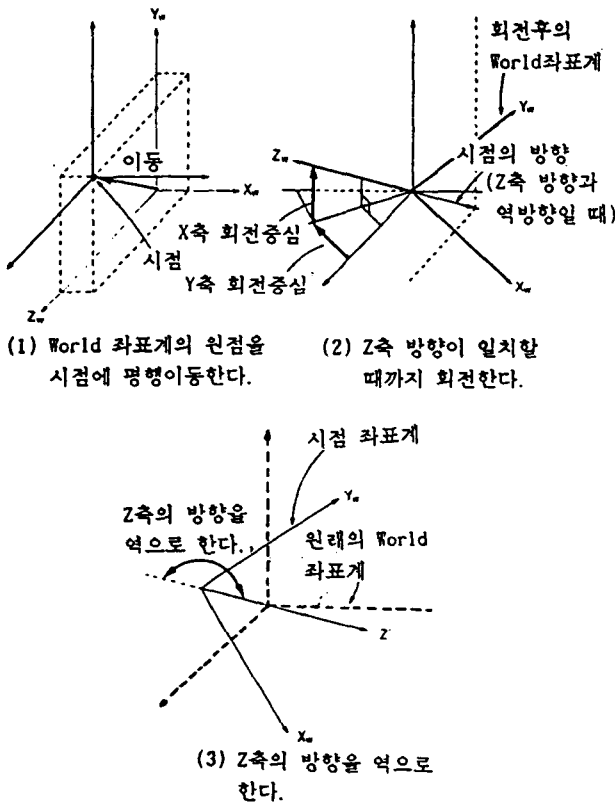


Fig. 4.12 Viewing Transformation Sequence

視野 변환의 계산은 변환 행렬을 위의 순서대로 연결하면 되고, 視野 변환의 변환행렬을 T_v 로 하면 다음 식 (4.9)로 표시된다.

$$T_v = T_d T_a T_\beta T_h$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -V_x & -V_y & -V_z & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4.9)

위에서 설명한 시뮬레이터를 위한 3차원 그래픽 표현 방법을 정리해 보면 다음 Fig. 4.13과 같다.

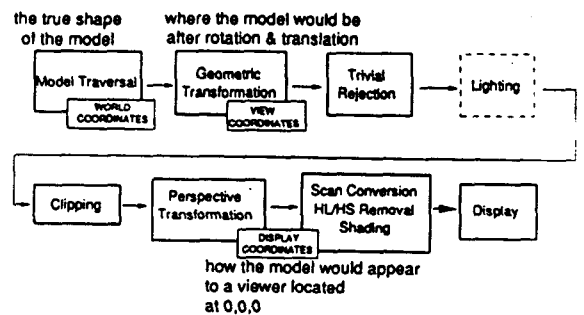


Fig. 4.13 3 Dimensional Pipeline

4.3 숨겨진 면 제거(Hidden Surface Removal)

현실감있는 장면을 만들기 위해 주로 고려할 사항은 선택된 視点에서 물체의 보이지 않는 부분을 찾아내고 이를 제거하는 문제이다. 이 문제를 해결하기 위한 방법은 여러가지가 있고, 서로 다른 응용 형태에 있어 효율적으로 숨은 부분을 제거하기 위해, 무수한 알고리즘들이 고안되어 왔다.

특정 용도를 위한 방법의 선택에는 몇가지 요인이 따르는데, 예를 들면 장면의 복잡도, 물체의 종류, 사용 가능한 장비, 또 정지 화상인가 움직

이는 화상인가 등이다.

4.3.1. 알고리즘의 분류

隱面 및 隱線 제거 알고리즘은 그것이 물체의 정의를 직접 다루는가 투상된 영상을 다루는가에 따라 각각 물체공간법(object-space method)과 영상공간법(image-space method)으로 분류된다. 물체공간법: 물체와 물체의 부분들을 서로 비교하여 전체적으로 어느 면과 선이 보이지 않는 것으로 분류될 것인가를 결정한다.

영상공간법: 이 알고리즘에는 투상면 상의 각 화소 위치에서 점 단위로 가시성(visibility)이 결정된다.

대부분의 숨겨진 면 제거 알고리즘들은 영상공간법을 쓴다. 그러나 어떤 경우에는 물체공간법이 효율적으로 쓰일 수도 있다. 그래픽 워크스테이션에 비해 계산 속도가 상대적으로 느리고 숨겨진 면 처리 전용 H/W(예를 들면 Z-Buffer)가 없는 PC에서 3차원 그래픽을 구현할 때 숨겨진 면을 제거하는 방법은 다양한 여러가지 방법이 있을 수 있지만 다음 3가지 방법이 적당하다.

- ① Radial pre-sort method, ② Radial Sort Method ③ Plane Equation Method(後面 제거, Back-Face Removal)

가. The Radial Pre-Sort Method

프로그래머가 먼저 모델을 관찰할 방향을 결정하면 프로그램이 보이는 면만 그리게 된다. 데이터베이스는 보이는 면의 World 좌표계 값만 가지고 있으면 된다. 이와같은 숨겨진 면 제거 방법은 절보기에는 Solid model과 같지만 실제로는 보이는 부분의 외관만을 표현한다. 이 방법은 관찰될 방향(radial angle)을 미리 정의하여 사전에 정리된 좌표계를 사용하므로 radial pre-sort method 라고 한다. 이 방법의 장점은 빠른 수행사건과 데이터베이스의 크기가 작다는 점이다. 그러나, 이 방법의 중요한 단점은 모델을 회전시킬수 없다는 점이다.

나. The Radial Sort Method

Radial pre-sort 방법에서는 프로그래머가 대부분의 숨겨진 선 제거를 위한 일을 미리 수행해야 했지만, radial sort 방법에서는 컴퓨터가 더

많은 부분의 일을 분담하여 수행한다. 데이터베이스는 모델의 모든 면의 world 좌표계 값을 가지고 있다. 프로그램에서는 관찰하는 각에 따라 보이는 면과 숨겨진 면을 결정한다. 이런 접근은 Radial Pre-Sort 방법보다 다양하며, 하나의 데이터베이스에서 관찰각에 따라 다양한 모델을 만들 수 있다. 다음과 같은 일반적인 명령어를 적용하면,

if (r1 > 3.14159) { } (4.10)

그리는 과정에서 x, y, z 좌표축에서 특정면을 지우거나 포함시킬 수 있다. Radial Sort 방법은 Flight Simulator나 공중전 게임과 같은 상업용 소프트웨어에서 매우 많이 사용되어진다. 이 방법의 장점은 if/then 명령어가 수행이 복잡한 숨겨진 면 제거 방법보다 빠르게 수행되므로 훨씬 빠른 시간내에 숨겨진 면을 제거할 수 있다는 점이다. 그러나 이방법의 큰 단점은 가능한 관찰각도별로 미리 제작된 데이터베이스와 연결되어, 이 방법이 수행되므로 표현할 모델을 첨가하거나 줄일 때 그림 그리는 코드의 중요 부분을 다시 써야 한다.

다. Plane Equation Method(後面 제거, Back-Face Removal)

물체의 後面(back face)을 구별해 내는 간단한 물체공간법은 평면의 방정식에 기초를 두고 있다.

Ax+By+Cz+D=0 (4.11)

오른손 좌표계에서 지정된 임의의 점 (x', y', z') 은 다음 부등식을 만족한다면 이 면의 안쪽에 있다.

Ax'+By'+Cz'+D < 0 (4.12)

점(x', y', z')이 視点이라면 부등식(4.12)이 성립하는 모든 면은 後面이 된다. 즉 시점에서는 볼 수 없는 면이다.

식 (4.11)로 표현되는 법선의 벡터를 검사하여 後面 판정을 더 간단히 할 수 있다. 이 법선 벡터는 직교좌표성분(A, B, C)를 갖는다. 관측방향이 음의 z축인 오른쪽 관측계에서 (Fig. 4.14) 법선벡터는 관측 방향에 평행한 성분 C를 가진다.

$C < 0$ 이면 법선벡터는 視点(viewing position)으로부터 멀어져 가는 방향이고, 따라서 이 면은後面이 된다.

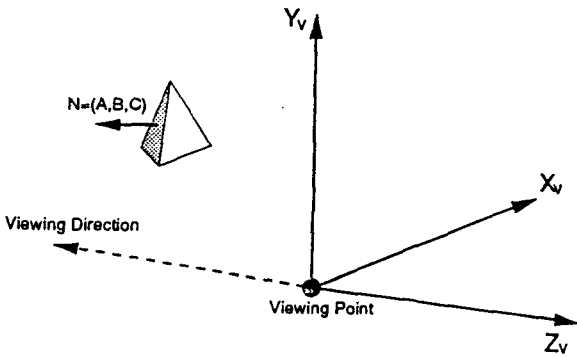


Fig. 4.14 In Case of right-hand viewing coordinate system

4.4 TMS 340 GSP와 TIGA 2.0을 이용한 3차원 Perspective-View 시뮬레이션 프로그램 개발

보통 소형 PC상에서 그래픽을 구현하는 방법은 Graphic Controller Card(VGA, EGA 등)사용 방식과 Graphic Signal Processor를 사용한 방식으로 나눌 수 있다.

이번 연구에서는 기존의 Controller 방식의 그래픽보드(ATI VGA Wonder Card)와 “MS-C ver. 6.0”의 그래픽 내장 함수, 그리고 “HALO Professional”의 그래픽 내장함수를 사용하여 2차원 그래픽 기법을 사용하는 Bird’s-Eye View Type의 시뮬레이터를 구성하였고, 3차원 그래픽 기법을 사용하는 Perspective View 방식의 시뮬레이터는 정밀한 3차원 그래픽을 구현하고 자연스러운 CGI화면을 구성하여 현실감을 더하기 위하여 GSP 방식인 TMS 34010 Board를 사용하였고 TIGA 2.0 Software Development Toolkit을 사용하여 시뮬레이터를 구성하였다.

4.4.1 TMS 340 Graphics System Processor
TMS34010 Graphics System Processor는 컴퓨

터 그래픽 전용의 32bit Microprocessor이며 범용 processor의 기능과 Graphics controller의 기능을 통합하여 강력하고 다방면에 사용 가능한 그래픽 전용 프로세서로써 기능을 구현한다. TMS 34010의 주요 특징으로는 빠른 처리속도와 高難易度의 그래픽 프로그램 기능, 그리고 기존에는 소프트웨어적으로 처리하던 (VGA, EGA graphic card의 방식) 그래픽 기능을 Chip 자체가 효율적으로 처리할 수 있는 기능(Pixels, 2차원 Pixels array등의 처리) 등을 들 수 있다.

TMS34010의 구조는 다양한 Pixel의 크기, frame buffer의 크기, 그리고 Screen 크기를 제공한다. TMS34010의 내장함수(On-chip functions)는 TMS 34010이 기능이 어느 특정 해상도에만 한정되지 않도록 嚴選되었다. 그래서 TMS 34010용 그래픽 소프트웨어의 호환성 및 프로그램의 타 기종 이식성이 증대되었다. 또한 위에 설명한 장점때문에 TMS34010은 그래픽 업계의 표준인 MIT의 X, CGI/CGM, GKS, NAPLPS, PHIGS, 그리고 기타 그래픽 관련 표준시스템을 사용할 수 있는 업계 표준 장비가 되었다. Fig. 4-15은 TMS 34010의 System Block Diagram이다.

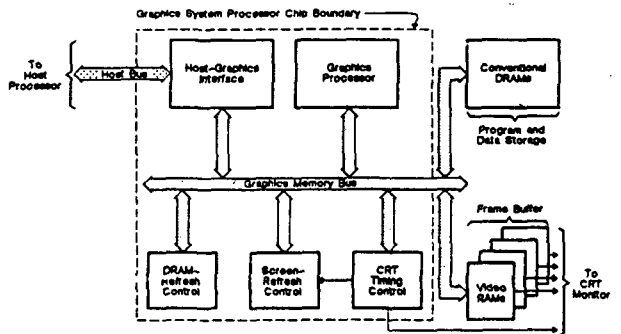


Fig. 4.15 System Block Diagram of TMS340

4.4.2. TIGA(Texas Instruments Graphics Architecture) 2.0

TIGA는 IBM 호환 PC용 모든 TMS 34010 계열의 GSP와 응용 프로그램과의 통신 및 명령어 처리를 표준화시키는 소프트웨어 interface이다. TIGA이 CORE는 응용 프로그램이나 드라이버 환

경하에서 프로토콜을 TMS 34010 프로세서와 링크하여 그래픽 기능을 수행하게끔 한다. 즉, TIGA는 PC 본체 CPU에서 수행중인 Graphics Primitives를 TMS 34010으로 Down-load시켜 프로그램을 수행하도록 하는 것이다. 종래의 컨트롤러 방식의 그래픽 시스템들을 호스트에서 수행되는 응용프로그램의 결과들을 화면에 나타내거나 단순히 그리는 일만을 함으로써 처리속도가 호스트에 상당히 좌우되었다. 그러나 TIGA는 그래픽에 관련된 대부분의 프로그램을 TMS 34010 프로세서가 수행하도록 하였다. 현재 TIGA에서 지원되는 응용프로그램은 Auto CAD를 비롯하여 약 50여종에 이른다.

가. 특 징

① 그래픽 응용 프로그램이 빠르게 작동된다. 프로그램 수행이 호스트의 CPU와 TMS 34010에서 동시에 수행되어 7-8배 빠르게 동작한다. 3차원 Perspective-View 시물레이션에 중요.

② TMS 340 계열 지원

TMS 34010 보다 상위 프로세서인 TMS 34020 프로세서뿐만 아니라 앞으로 개발될 TMS 340 계열을 호환성을 유지하며 지원해준다.

③ 사용이 편리하다

그래픽 표준중에 하나인 CORE와 유사한 명령어 체계를 가지고 있고 기존의 업계 표준의 그래픽 Primitives를 가지고 있어서 사용이 편리하다.

④ 확장성(Extensible)

응용 프로그램에서 TIGA의 기본 그래픽 함수 외에 복잡한 기능의 함수가 필요한 경우 TMS340용 C나 Assembler를 사용하여 사용자가 쉽게 사용자 정의 함수(User-extended functions)를 개발할 수 있다. 시뮬레이터용 그래픽 프로그램을 구조화 시키는데 유리하고, 근래에 S/W 개발에 도입되는 OOP(Object-Oriented Programing) 기법을 활용할 수 있다.

⑤ Hardware호환성

하드웨어의 상태를 확인할 수 있는 Inquiry 함수 기능이 있어서 응용프로그램이 사용하고 있는 TMS 340 그래픽 시스템의 해상도, Pixel크기 등을 스스로 결정할 수 있다.

나. 구 조

TIGA는 다음과 같은 네가지 요소로 구성되어 있다.

- ① Application Interface(AI)
- ② Communication Driver(CD)
- ③ Graphics Manager(GM)
- ④ TIGA Extentions

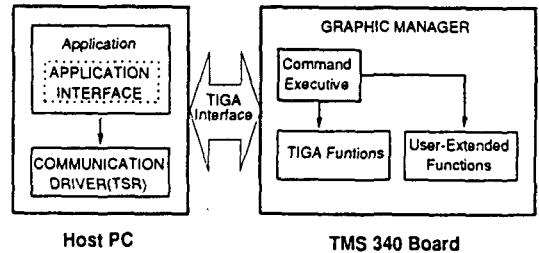


Fig. 4. 16 Block Diagram

5. Simulator용 근거리 통신망(LAN) 구성

현실감 있는 3차원 Perspective View를 재현하기 위해서는 여러 대의 영상 출력용 PC와 모니터나 Beam-Projector를 통한 CGI 시스템 구성이 필요하게 된다.

또한 현실감 재현을 위한 Motion Base의 제어 및 실제 운항중인 선박의 항해 장비의 작동 상태를 재현하고 이에 필요한 A/D, D/A Control 등을 동시에 수행하기 위해서는 여러 대의 PC를 사용하여 각각의 기능을 수행하여야 한다.

그리고 선원 훈련을 위한 경우, 교관이 임의의 돌발 상황을 부여하기 위해서는 선원의 훈련 시물레이션 도중에 외부에서 시물레이션 상황을 변화시킬 수 있어야 한다.

이러한 기능을 구현하기 위하여 본 연구에서는 근거리 통신망(Local Area Network : LAN)을 사용한 여러 PC 사이의 통신 방법에 대한 기초 조사와 연구를 수행하여 다음과 같은 결과를 얻었다.

☑ 10MBPS Ethernet Adapter를 사용한 Ethernet Bus type 통신망의 구성.

☐ NetBIOS 통신 방식을 사용한 PC간의 통신 프로그램을 개발.

☐ 이 프로그램을 사용하여 3차원 영상 출력용 시뮬레이터 시스템과 레이다 기능을 가진 2차원 시뮬레이터 시스템을 연결하여 동시에 작동하도록 시뮬레이터를 개발.

5.1 NetBIOS를 사용한 통신 프로그램 개발

NetBIOS는 유용한 network programming interface이다. 컴퓨터에서 MS-DOS를 사용하는 경우, NetBIOS는 IBM, XNS, TCP, IEEE 그리고 OSI protocols를 사용하는 통신 시스템과 다른 시스템과의 일관성 있는 interface를 제공한다.

Network Basic Input/Output System (NetBIOS)는 컴퓨터간에 데이터를 주고 받는 경우에 사용되는 응용 프로그램 개발 interface이다. 즉, NetBIOS는 컴퓨터 응용 프로그램과 컴퓨터 장비들간에 통신을 수행할 수 있게 해주는 여러가지 지원 기능을 가진 프로그램 개발을 위한 기본 도구이다. 응용 프로그램은 여러가지 정해진 명령어 규칙을 사용하여 다양한 NetBIOS Service들을 수행시켜야 한다. 일반적으로 데이터 교환은 Local Area Network(LAN)으로 연결된 각각의 컴퓨터에서 작동되는 NetBIOS 응용 프로그램에 의해서 수행되어진다.

5.1.1 Programming NetBIOS

NetBIOS program 개발과 disk file 프로그램 개발을 비교해 보면 NetBIOS programing 기법을 쉽게 이해할 수 있다. 실제로 peer-level 통신은 file의 특별한 경우로 취급할 수 있다. 다음 Tabel 5-1은 PC DOS file operation과 그에 대응하는 NetBIOS function의 비교이다.

Table 5.1 File I/O Compared to NetBIOS functions.

File Operation	NetBIOS Equivalent
Open	Add Name, then Call(or Listen)
Read	Receive
Write	Sena
Seek	<None>
Close	Hang Up, then Delete Name

(가) Session Support

일단 LAN adapter가 network상에서 작동되면, workstation상의 응용 프로그램은 동일한 workstation이나 떨어져 있는 다른 workstation에 있는 응용 프로그램과 통신하기 위하여 NetBIOS를 사용할 수 있게 된다. 응용 프로그램은 datagram이나 session을 이용하여 서로 통신할 수 있게 된다.

NetBIOS session 지원 기능은 장기간 진행되는 두 응용 프로그램사이에서 신뢰성있는 two-way data communication을 수행하도록 해 준다. 그러한 연결을 virtual circuit이라고도 한다. 통신 응용 프로그램은 같은 workstation 내에 존재할 수도 있고 (local sessions), 다른 workstation 상에 존재할 수도 있다. (remote session). Datagram communication과 비교했을 때, session communication의 장점은 data를 보낼 때마다 message-receipt status를 data를 보내는 응용 프로그램에서 NetBIOS를 통하여 전달하므로 data를 안전하게 전송할 수 있다는 점이다. 그러나 session communication의 안전성을 위하여 session을 성립시키고 관리하고 LAN adapter간의 packet acknowledgement protocol을 위하여 약간의 overhead가 발생한다.

Session Communication

NetBIOS 명령어 Call, Listen, Hang Up은 session을 관리하기 위한 도구이다. 다음 Table 5.2는 session을 성립시키고 data message를 교환하고 session을 종료하는 일련의 과정을 보여준다.

Table 5.2 A data packet sequence for NetBIOS sessions.

Workstation A	Workstation B
1. Add Name A	1. Add Name B
2. Liaten(wait for Call)	2. Call Workstation A
3. Send message to B	3. Receive a message
4. Receve a message	4. Send a message
5. Hang Up on B	5. Hang Up on A
6. Delete Name A	6. Delete Name B

5.1.2 NetBIOS 수행

특정한 NetBIOS 함수를 수행시키기 위해서는 NetBIOS Control Block(NCB)를 구성하고, ES :

BX register pair에 있는 NCB에 far pointer를 부여하고 Interrupt 5C(hex)를 수행한다.

```
void NETBIOS (NCB far *ncb_ptr)
{
    struct SREGS srega;
    union REGS regs;

    sregs.es=FP_SEG(ncb_ptr);
    regs.x.bx=FP_OFF(ncb_ptr);
    int86x(OX5c, &regs, &regs, &sregs);
}
```

5.1.3 Network Control Block

Interrupt 5C를 호출할 때 지정해야 하는 것은 Network Control Block의 address이다. NCB는 통신을 위한 특정 명령수행을 위해 필요한 모든 것을 NetBIOS에게 알려주는 媒介體이다. NCB 자체는 network를 통해 전송되지 않고, NetBIOS는 NCB를 수행하고자 하는 통신 명령들의 집합으로 사용한다.

5.2. Simulator용 통신 프로그램 개발

이번 절에서는 앞에서 설명된 LAN 기술을 적용하여 2대의 시뮬레이터를 연결시킨 적용 사례를 설명한다.

시뮬레이터 시스템에서 사용된 LAN 구성 방식은 설치 및 추가 구성이 간단한 Ethernet Bus Type의 구조를 사용하고 NOS는 Peer-to-Peer 방식의 Artisoft의 LANTastic Ver. 4.0을 사용하였다. 시뮬레이터 시스템의 특성상 최대 연결 컴퓨터의 댓수가 20대 미만이고 업무용 LAN과 달리 사용법 및 유지 보수가 간단하고 호환성이 높은 NOS가 유리하여 가격면에서도 유리하고 IBM PC 표준의 NetBIOS를 사용하는 LANTastic NOS를 사용했다. 또한 Peer-to-Peer 방식이므로 특정한 한대의 컴퓨터가 Server역할만 수행하는 Server-LAN 방식보다 경제적으로 시스템을 구성할 수 있다.

시험용 시뮬레이터 시스템의 구성은 Fig. 2.1에 나타낸것과 같이 3차원 영상을 재현시키는 25

MHz i486 PC와 2차원 영상(레이더 기능)을 표현하는 16MHz 286 PC 두 대로 구성되어있다. 두 대의 컴퓨터는 Session Communication을 이용하여 Data를 주고 받을 수 있도록 통신 프로그램이 개발되어 있다. 이 프로그램은 시스템에 컴퓨터가 추가로 설치 되어도 시뮬레이터 작동 중의 선박의 상태 변수들(타각, 기관 출력 현재 선속, Heading Angle, Yaw Rate등)을 1대 1 또는 1대 PC의 정보를 전체 Network상에 전달시킬 수 있도록 개발되어 있다.

6. Maneuvering Controller의 구성 (Interface용 H/W 및 S/W 개발)

실제 船舶의 運航 상태를 再現시켜 주고 실제 操船과 동일한 방법으로 시뮬레이션을 수행할 수 있도록 航海裝備를 재현시키는 일은 선박 조종 시뮬레이터 구성 요소의 중요한 부분 중에 하나이다. 실시간(Real-Time) 시뮬레이션을 위한 模擬裝備로서, 선박의 방향 조종을 위한 舵의 정보를 입력하기 위해, 실선의 Steering Stand와 동일한 기능 및 형상의 Steering Stand를 그리고 주기관의 Throttle값을 조절하기 위해 Engine Telegraph를 설계, 제작하였고, 항구의 淺水영역에서의 入·出港 상황을 시뮬레이션하는 경우, 入·출항 대상 선박을 曳引하는 Tug Boat의 고정 위치 및 추진 방향에 관한 명령을 컴퓨터에 입력하기 위해서 Tug Boat Controller를 개발하였다. 또한 船上 操縱시뮬레이터의 수행에 필요한 입력 데이터를 시뮬레이터 사용자가 船室이나 研究室등 制限된 공간에서 편리하게 사용할 수 있도록 탁상용 조종장치(Portable Controller)를 설계 및 제작하였다.

A/D Converter는 Blue Cip Tech社의 16 Ch. ADC 42를 사용하였고, A/D Converter운용 프로그램은 MS-C 언어를 사용하여 개발하였다. Fig. 6.1는 Maneuvering Controller의 구성도이고, Fig. 6.2는 Portable Type Controller의 구성도이다.

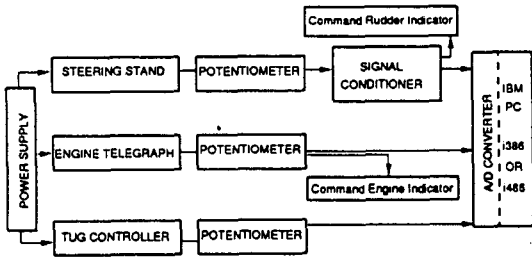


Fig. 6.1 Configuration of Maneuvering controller

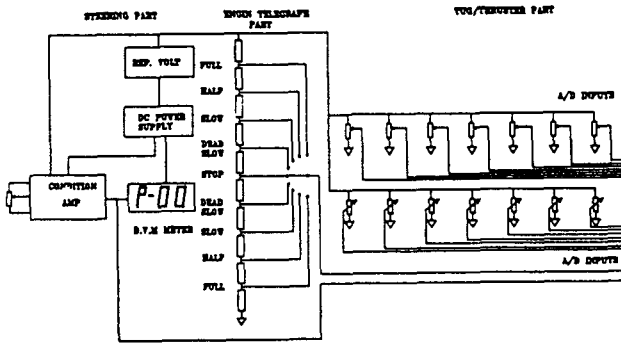


Fig. 6.2 Diagram of Portable Controller

7. 結 言

以上에서 기술한 바와 같이 본 연구에서는 선박 조종 시뮬레이터 개발을 위한 基礎 技術 開發 및 관련 자료 조사를 수행하여 시뮬레이터 개발에 필요한 核心 技術인 물리적 현상을 예측하는 수학 모델, 현실감 재현을 위한 컴퓨터 그래픽 기술, 사용자와 컴퓨터를 연결시키는 H/W Interface 기술 등의 기본 기술을 개발하였고 소형 PC를 이용하여 소형 및 이동형 선박 조종 시뮬레이터를 제작하였으며, 이에 대한 제품기술을 개발하였다.

본 연구결과들을 정리하면 다음과 같다.

- ① 소형 PC를 이용한 Real-Time 선박 조종 운동 시뮬레이션 프로그램 개발

- ② Bird's Eye View 방식 및 Perspective View 방식 視界再現 영상 처리 기법 자료 조사 및 기본 프로그램 개발
- ③ 근거리 통신망(LAN)을 이용한 시뮬레이터용 통신 프로그램 개발
- ④ 시뮬레이터용 각종 항해 장비(조타 장치, Tug/Thruster, Engine 제어) 인터페이스용 H/W 및 S/W 설계 제작
- ⑤ 하드웨어와 소프트웨어를 통합한 시뮬레이터 기술 개발
- ⑥ 사용자 위주의 시뮬레이터 운영 프로그램 개발
- ⑦ 시뮬레이션 결과 출력, 해석 및 평가 프로그램 개발
- ⑧ 항구 지형 자료(지형, 수로, 수심)와 해상 상태(조류분포, 바람, 파도) Data Base 구축 및 수정 프로그램 개발
- ⑨ 소형 및 이동형 시뮬레이터 시제품 제작 그리고 본 연구의 活用 方案은 다음과 같다.

- ① 동일 상황 반복에 의한 선원의 훈련 및 교육
- ② 각종 비상상황 시뮬레이션에 의한 선원의 비상시 대처 능력 향상
- ③ 새로운 항구 입항시 사전 연습에 의한 안전 운항 능력 향상
- ④ 기존 항만 준설 및 수로 변경에 따른 각종 영향 평가
- ⑤ 초기 설계 단계에서의 선박의 조종 성능 평가
- ⑥ 事故 狀況 재현 및 原因 分析에 이용
- ⑦ 해운 및 조선사에 관련 기술 제공
- ⑧ 선원 훈련용 및 연구용 대형 航海 시뮬레이터 개발
- ⑨ Auto-Pilot System 개발 및 제어시스템 연구
- ⑩ 개발된 3차원 및 2차원 視界再現 프로그램과 사용자 인터페이스 프로그램의 타 분야 활용

☞ 초고속선 및 항공기용 시뮬레이터 개발에

이용 가능

8. 후 기

본 논문은 과학 기술처의 1990년 특정 연구 사업으로 한국 해사 기술 연구소와 (주)삼풍엔지니어링이 공동으로 수행한 “소형 컴퓨터를 이용한 선박 조종 시뮬레이터 개발 및 제품 기술 개발” 과제와 과학 기술처의 1990년 기본 연구 사업으로 수행한 “선박 조종 Simulator 개발(I)” 과제의 결과임을 밝히며 과학기술처와 (주)삼풍엔지니어링에 감사를 표한다.

參 考 文 獻

- Samuelides, E. and Frieze, P., “Experimental and Numerical Simulation of Ship Collisions”, Proc. 3rd Int. Congress on Marine Technology, Vol. 1, Athens, Greece, pp. 297-308, 1984.
- 김훈철 외, “최적 운항 조건 도출을 위한 요소기술 개발”, 한국 기계연구소 연구 보고서 UCN 202-492. D, Dec. 1984.
- 서상현 외, “M. 조선소 예인시 좌초한 Continental Friendship호의 조종성능 해석”, 한국 기계연구소 연구 보고서 BIOS 407-607. Oct. 1985.
- Goodwin, A. H. & et. al., “The Practical Application of Computers in Marine Engineering”, Trans. of the Institute of Marine Engineers, Vol. 80, No. 7, 1968.
- 김선영 외, “조종 성능 추정 기법 개발(I)”, 한국 기계연구소 연구 보고서 UCE 337-1082. D, Mar. 1988.
- Patell, J. M., A Digital Ship Maneuvering Simulation for Harbor Traffic”, Rep. No. 72-2, Commodity Transportation and Economic Dev. Lab., MIT, 1972.
- Hirano, M., “On the Calculation Method of Ship Maneuvering Motion at the Initial Design Phase” (In Japanese), Jour. Society of Naval Architects of Japan, Vol. 147, 1980.
- van Lammeren, W. P. A. & et. al., “The Wageningen B-Screw Series”, Trans., SNAME Vol. 77, 1969.
- Tor Einar Berg, Kjell Martinussen, Torbjorn Utnes., “Prediction of Maneuvering Characteristics”, Report R-104. 80, The Ship Research Institute of Norway, 1980.
- Isherwood, R. M., “Wind Resistance of Merchant Ship”, RINA, Vol. 114, 1972.
- 강창구 외 “광양항 1단계(2차) 개발 사업 실시 설계 용역(선박 안전 운항 평가 연구), 한국기계연구소 부설 해사기술연구소, 연구 보고서 BSI-841-1440. D, July, 1991.
- 佐藤義雄, “實習 グラフィックス, アスキー出版局, 1989.
- Lee Adams, “High-Performance CAD GRAPHICS in C”, McGrawHill, 1986.
- Texas Instruments Computer Video Product, “TIGA Inteface User’s Guide”, Texas Instruments 1990.
- Texas Instruments Computer Video Product, “TMS34010 User’s Guide”, Texas Instruments 1990.
- Foley, van Dam, Feiner, Hughes, “Computer Graphics Principles and Practice”, Addison Wesley, 1990.
- Bill Shotts et al. “HALO Professional Programmer’s Guide”, Media Cybernetics, 1991.
- Barry Nance, “Network Programing in C”, QUE, 1990.
- W. David Schwaderer, “C Programmer’s Guide to NetBIOS”, SAMS, 1990.
- 정진욱, 변옥환 공저, “LAN기술과 그 응용 근거리 통신망”, Ohm社, 1986.
- 오영환, 신성용 공저, “컴퓨터 그래픽스”, 흥릉과학 출판사, 1990.
- In-Young Gong, Sun-Young Kim, Chang-Gu Kang, Hee-Chang Park, “Development and Application of an On-Board Manueve-

- ring Simulator” Proceedings, MARSIM & ICSM 90, The Society of Naval Architectures of Japan, Japan Shipbuilding Industry Foundation, June, 1990.
23. 강창구 외 “소형 컴퓨터를 이용한 선박 조종 시뮬레이터 개발 및 제품 기술 개발”, 한국기계연구소 부설 해사기술연구소, 연구 보고서, BSM 215-1533. D, Dec. 1991.
24. 강창구 외 “선박 조종 Simulator 개발(I)”, 한국기계연구소 부설 해사기술연구소, 연구보고서, UCE 511-1577. D, March, 1992.