

PC를 利用한 ARPA RADAR SIMULATOR의 開發에 關한 研究

宋 在 旭* · 金 煥 秀**

A Study on the Development of the Personal Computer
Aided ARPA RADAR Simulator

Chae-Uk Song* · Whan-Soo Kim**

〈 목 차 〉

Abstract	3.1.2 엔진입력
제1장 서론	3.2 레이더의 기능 및 조작
제2장 시뮬레이터의 구성 및 설계	3.2.1 Radar Work
2.1 시뮬레이터의 구성	3.2.2 Presentation
2.1.1 Hardware의 구성	3.2.3 ARPA Function
2.1.2 Software의 구성	제4장 시뮬레이터의 실행 예 및 개선점
2.2 시뮬레이터의 설계	4.1 실행 예
2.2.1 본선의 종류 및 운동방정식의 구 성 방법	4.2 개선점
2.2.2 상대선의 설계	4.2.1 조작장치
2.2.3 본선과 상대선의 관계 해석	4.2.2 상대선의 움직임
2.2.4 프로그래밍 방법	4.2.3 항행환경
제3장 시뮬레이터의 기능과 조작 방법	4.2.4 사용 프로그래밍 언어
3.1 입력 기능	제5장 결론
3.1.1 타각입력	참고문헌

Abstract

In recent years, through the rapid development in personal computer technology, it has become possible to make a radar simulator based on the personal computer. The advantage of the personal

* 한국해양대학교 대학원 해사수송과학과 해사정보전산전공.
** 정회원, 한국해양대학교 해사안전관리학과 부교수.

computer aided radar simulator lies in its cost effectiveness, when comparing with that using the real radar.

Although there have been studies carried out to develop radar simulator using PC and the products of this kind is in the commercial market already, they are all using the mockup of the real radar, and therefore, the price of the simulator is still rather expensive. In this respect, this thesis aims to develop a ARPA radar simulator which is running on the sole PC, so that the students of the maritime educational institution may get ARPA radar training easily and cheaply.

The simulator developed in this thesis using Visual Basic is found to run successfully on the 486PC, and it is expected that this new simulation system designed first time in Korea would be used as an easily accessible ARPA radar training equipment.

1. 서론

해상에서의 인명과 재산의 안전에 관한 관심이 국제적으로 높아져 가고 있음과 동시에, 해운 생산성의 증대와 조선기술의 발달로 선박이 대형화, 고속화, 자동화되고 있는 것이 최근의 해운산업 환경 변화의 추세이다. 이와 함께 충돌, 좌초, 침몰 등과 같은 해상사고, 해기사에게 요구되는 선박조종기술정도, 사고발생시 손해액 등도 동시에 증가하는 추세에 있다. 이와 같이 선박이 대형화, 자동화되고 선박이 수송하는 화물중 원유, 액화천연가스 등과 같은 위험화물의 수송량도 증가되어 선박의 안전운항에 있어서 인간에게 허용될 수 있는 실수의 허용치도 과거에 비해 그만큼 크게 감소하게 되었다. 한편 지난 수십년간 해난사고는 꾸준히 증가되어 왔고 이러한 해난사고의 원인중 약 80%가 기계적 요인이 아닌 인간적 요인임이 여러 연구조사에서 밝혀진 바 있다[1]. 이러한 이유로 인해 해운산업 분야에서는 해기사의 기술 및 기능 향상에 특별한 관심을 가지게 되었고, 이러한 해기사 교육을 위해 full mission bridge 시뮬레이터나 part task 시뮬레이터를 사용한 교육 프로그램이 개발되어 왔다.

레이다 시뮬레이터는 part task 시뮬레이터의 일종으로서, 1950년대 중반부터 선박에 레이다 장치를 적재하는 것이 보편화 되기 시작하고 그 설치가 법으로 의무화되면서 교육용 기구로써 항해 관련 교육기관에서 사용되기 시작한 장비이며, 현

재 STCW협약에서는 레이다 시뮬레이터 교육을 강제화하고 있다.

그러나 초기의 레이다 시뮬레이터는 실제의 레이다를 사용했고 그 신호를 제어하는데 많은 부속장치가 필요했을 뿐만 아니라, 기기 자체의 크기가 커서 제작하는데 많은 비용이 소요되었다. 이와 같은 고가의 시뮬레이터를 학생교육과 같은 교육용으로 사용하기에는 비용면에서 문제가 되었고, 특히 초기의 시뮬레이터는 기동순서나 장치가 복잡하고 크기가 커서 장소적으로도 문제점이 있었다. 이에 따라 아직도 레이다 시뮬레이터를 갖추지 못하고 있는 해기교육기관이 많고, 갖추고 있는 경우에도 1~2기 정도가 보통이어서 많은 학생이 동시에 실습하기 어려운 것이 현실이다.

한편, 계산기 분야의 급진적인 기술발달로 최근에는 대용량 고속처리가 가능한 저가의 개인용 컴퓨터가 등장하게 되었고, 이에 따라 시뮬레이터 제작 업체에서 PC를 이용한 레이다 시뮬레이터의 개발에 관심을 가지기 시작하였으며, 일부 선진국의 경우, 80년대 말에 PC를 내장한 실물형 레이다 시뮬레이터를 개발 완료한 바가 있다[2]. PC를 내장한 실물형 레이다 시뮬레이터는 실물 레이다 시뮬레이터보다는 가격이 낮았으나 여전히 고가의 장비이고, part task 시뮬레이터가 아닌, full mission bridge 시뮬레이터의 레이다 부분을 구성하기 위한 목적으로 개발되었으며, 독립된 PC형 레이다 시뮬레이터는 아직 시판되지 않고 있다. 그리고 우리나라 국내의 경우에는 선박조종 시뮬

레이더 제작 산업이 전무하다시피하여, 레이더 시뮬레이터 개발에 관한 연구는 지금까지 한건도 없는 실정이다.

따라서 본 논문에서는 향후 우리나라 독자 모델의 선박조종 시뮬레이터를 개발하기 위한 기초적인 기술을 개발, 축적함과 동시에, 많은 학생들이 손쉽게 레이더 실습을 할 수 있는 저가의 시뮬레이터를 개발하기 위한 목적으로 PC를 이용한 ARPA Radar 시뮬레이터의 개발에 관해 다루었다. 본 논문에서 개발된 시뮬레이터는 현재 대량 보급중에 있는 486 DX급 개인용 컴퓨터를 이용한, ARPA Radar의 조작 훈련이나 레이더 플로팅 연습뿐만 아니라 피항조선 훈련까지 할 수 있는 desktop 형태의 시뮬레이터로서, 개인용 컴퓨터 이외에는 별도의 장치가 필요없기 때문에 비용면이나 관리면에서 매우 유리한 장비라 할 수 있다.

본 논문의 제2장에서는 시뮬레이터의 구성 및 설계 방법에 대해 다루었으며, 제3장에서 개발된 시뮬레이터의 기능과 조작방법에 대해 다룬 뒤, 제4장에서는 시뮬레이터의 실행 예를 보이고 향후 개선점에 대해 기술하였다. 그리고 제5장에서 결론을 기술하였다.

제 2 장 시뮬레이터의 개요

2.1 시뮬레이터의 구성

2.1.1 Hardware의 구성

본 시뮬레이터는 요즘 보편화되어 흔히 사용되고 있는 개인용 컴퓨터 기종인 486DX를 사용했다. 본체는 Intel 80486DX CPU와 메인메모리 8 MB, 5.25인치 플로피디스크 드라이브 1개, 3.5인치 플로피 디스크 드라이브 1개, 400MB의 하드디스크 드라이브를 갖추고 있으며, 그래픽카드는 VGA카드를 장착하고 있다. 여기에 키보드, 마우스, 14인치 칼라 모니터가 주변장치로 구성되어 있다.

2.1.2 Software의 구성

MS-DOS 버전 5.0, 한글 윈도우즈 버전 3.0,

Visual Basic 버전 3.0를 갖추고 있다.

화면구성이나 상대선 및 그의 레이더의 여러 기능들을 디스플레이하는 데는 그래픽기능이 탁월한 Visual Basic을 사용하였으며, 이의 사용을 위해 윈도우즈를 구비하였다.

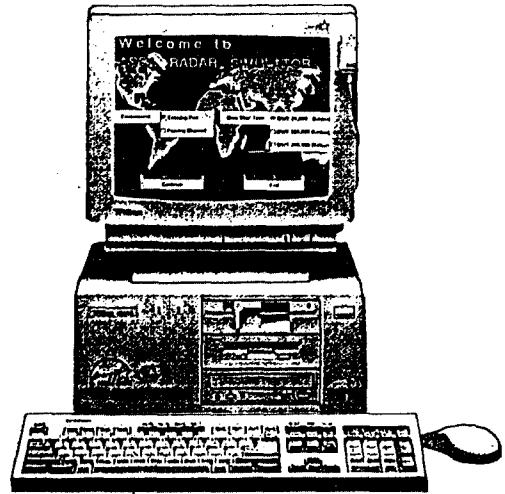


Fig. 1 Desktop ARPA Radar Simulator

2.2 시뮬레이터의 설계

본 시뮬레이터는 한국해양대학교 실습선 한나라호에 적재되어 있는 ARPA Radar(ATLAS 9600)를 모델로 하여 만들었다. 재래 레이더의 기능에 Auto Plotting 기능을 갖춘 것이 ARPA Radar이며 현재 여러나라의 제품이 실선박에서 사용되고 있으나 본 대학 실습선에 적재되어 있는 제품을 기준으로 하여 제작하였으며, 실제 레이더의 여러가지 부가기능 중 중요하지 않은 부분은 생략하여 간략화하였다. 레이더의 전 기능을 구현하는 것이 원칙이었으나 모니터의 크기, 인터페이스의 부족 등으로 레이더 본래의 기능에 크게 중요하지 않은 것은 제외하였다. 이러한 간단한 기능만으로도 충분히 교육효과를 얻을 수 있다고 판단되며, 차후에 기능들을 추가할 수 있도록 여유를 두고 설계하였다.

2.2.1 본선의 종류 및 운동방정식의 구성방법

2.2.1.1 본선의 종류

Table 1. Specifications of the Own Ships

명세	선종	
	DWT 21,000톤급 화물선	DWT 200,000톤급 유조선
수선간장(LBP)	161.0 M	305.0 M
선폭 (B)	23.2 M	47.2 M
흘수 (d)	7.5 M	18.4 M
최대속력	20.0 Kts	16.5 Kts
Block Coefficient	0.6	0.825

본 시뮬레이터에 사용된 본선은 Laden 상태의 DWT 21,000톤급 화물선과 Ballast 및 Laden상태의 DWT 200,000톤급 VLCC 유조선으로 구성되어 있으며, 두 선박의 제원은 다음의 Table 1과 같다.

본 시뮬레이터를 이용하여 레이다의 조작훈련 뿐만 아니라 피항조선 훈련도 할 수 있도록 하기 위해 선박모델을 톤수별로 두 가지로 나누고 톤수가 큰 유조선은 다시 Ballast상태와 Laden상태로 나누어 모두 3가지의 선박모델을 구축하였다. 이와 같이 여러 형태의 선박모델을 구성하여 각각의 선체운동특성이 다른 선박들을 이용하여 시뮬레이션 해봄으로써 선박의 크기와 적재상태에 따른 선박의 운동특성을 조종자가 익힐 수 있도록 하였다.

2.2.1.2 선박운동방정식의 기본원리

시뮬레이션을 위한 선체운동 수학모델 중 McCallum에 의해 개발된 모델[3]을 기초로 하여 선박운동방정식을 구성하였으며, McCallum Model의 기본 원리는 다음과 같다.

흐르는 유체에 선체를 고정시킬 경우 선체에 미치는 힘은 Fig. 2와 같이 크게 유체의 흐름방향에 직각으로 작용하는 lift force와 유체의 흐름방향에 평행하게 작용하는 drag force 두 가지로 나눌 수 있으며, 이 힘들은 다음과 같이 계수로 표시될 수 있다[4].

$$\text{Lift Coefficient} = C_L = \frac{L}{1/2 \rho AV^2} \quad (2.1)$$

$$\text{Drag Coefficient} = C_D = \frac{D}{1/2 \rho AV^2}$$

여기서, ρ 는 유체밀도,
A는 유체를 받는 면적,
V는 유체속력을 말한다.

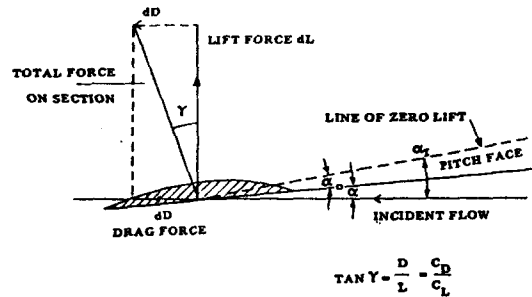


Fig. 2 Forces on a Hydro-Foil

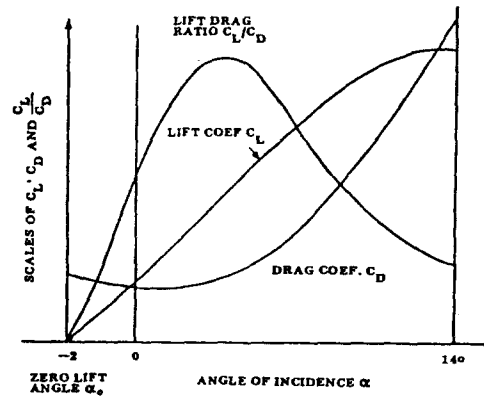


Fig. 3 Characteristics of Lift and Drag Forces

선체와 유체가 이루는 각(drift angle)을 α 라 하면 Fig. 4와 같이 선체에 미치는 유체의 속도는 surge 속도(u)와 sway 속도(v)의 벡터 합이 된다.

$$\vec{u}^2 = u^2 + v^2 \quad (2.2)$$

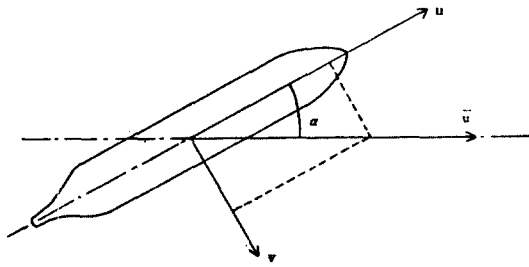


Fig. 4 Surge and Sway Velocities for a Hull at Drift Angle α

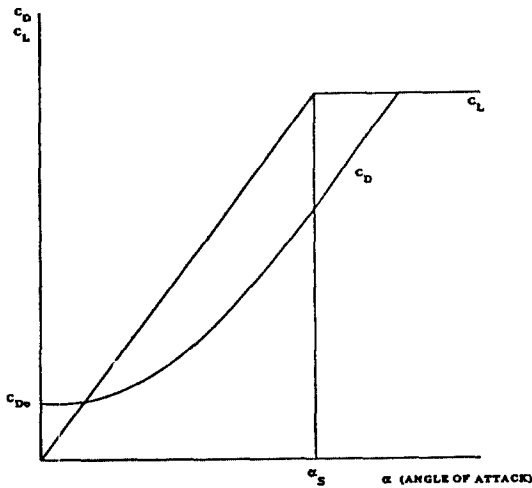


Fig. 5 Idealized Lift and Drag Characteristics for Inclined Hull Form

그리고 drift angle α 는 surge와 sway 속도와 관계가 있으므로 다음과 같이 표현할 수 있다.

$$\alpha = \tan^{-1} \frac{v}{u} \quad (2.3)$$

특정한 형태의 선체모양에 미치는 lift와 drag force를 정확하게 구하기는 불가능하므로 선체에 미치는 lift와 drag force는 Fig. 5와 같이 이상적 형태로 나타낼 수 있다고 가정한다.

Fig. 5에 나타나 있는 것과 같이 lift와 drag force는 stall angle α_s 까지는 거의 수직적으로 상승한다. 작은 aspect ratio(중형비)를 가질수록 stall angle은 커지며, 보통 선박의 경우 아주 작은

aspect ratio를 가지므로 45° 이상의 stall angle을 가진다. 실제로 선박의 drift angle이 45° 를 넘는 경우는 드물기 때문에 Fig. 3을 고려할 때 Fig. 5와 같이 이상화시킨 lift와 drag 특성을 이용하더라도 수학 모델 구성에 큰영향은 받지 않는다.

그리고 Fig. 6과 같이 이상적 lift force의 특성을 나타내는 limited drift angle의 개념을 도입하면,

$$\begin{aligned} | \alpha | \leq K1 \text{ 일 경우 } & a' = \alpha \\ \alpha > K1 \text{ 일 경우 } & a' = K1 \\ \alpha < -K1 \text{ 일 경우 } & a' = -K1 \end{aligned} \quad (2.4)$$

으로 표현할 수 있다[5].

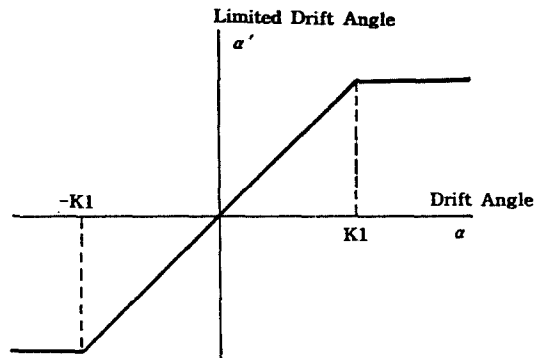


Fig. 6 Limited Drift Angle

2.2.1.3 선체에 미치는 유체역학적 lift와 drag force

식 (2.1)에 나타나 있는 바와 같이 흐르는 유체에 놓인 선체에 미치는 lift와 drag force는 유체의 속도의 제곱에 비례한다.

따라서 lift force는 다음과 같이 나타낼 수 있다.

$$L_H = K7 \times u^2 \times a' \quad (2.5)$$

선체에 미치는 drag force는 α 가 0일 때의 작은 잔류 drag와 α 의 제곱에 비례하는 drag force로 구성된다. 이것을 수식으로 표현하면,

$$D_H = K8 \times \bar{u}^2 + K9 \times \bar{u}^2 \times \alpha^2 \quad (2.6)$$

이 된다. 여기에서 $K8 \times \bar{u}^2$ 은 α 가 0일 때의 잔류 drag량을 나타낸다.

2.2.1.4 프로펠라 유효 슬립과 후류 속도

선박 프로펠라의 실 슬립비는

$$Sr = 1 - \frac{V_a}{p \times n} \quad (2.7)$$

이다. 여기서 V_a 는 프로펠라 대수 전진 속도,
 p 는 pitch,
 n 은 회전속도를 의미한다.

그리고 프로펠라 전진속도 대신에 선박의 속도 V 를 사용하여 또다른 슬립비를 구할 수 있는데 이를 명목슬립비라 하고 다음과 같이 표시할 수 있다.

$$Sa = 1 - \frac{V}{p \times n} = 1 - \frac{u}{p \times n} \quad (2.8)$$

보통 선박조종을 하여 보면 속력이 작을수록 슬립비는 커진다. 그러나 실제 조종중에 정확한 실 슬립이나 명목슬립을 구하기가 어려우므로 이를 간편하게 하기위해 유효슬립비라는 것을 생각해 내고 이를 수식으로 표시하면

$$Se = 1 - \frac{u}{U_0} \quad (2.9)$$

이 된다. 여기서 U_0 는 throttle setting speed를 말한다.

선박의 후류속도(slipstream velocity) U_s 는 평정상태(equilibrium condition)에서는 평정속도 U_0 에 가까워진다. 이때 $u = U_0$, $Se = 0$ 가 된다.

선박조종중 선속 u 는 drift angle α 로 인해 선체의 저항이 증가하므로 감소한다. 그리고 프로펠라 회전이 감소할수록 유효 슬립비는 증가한다. 따라서 스크류 뒤쪽의 후류속도는 선박조종중에 감소를 하게 된다. 다만 선속과 같은 양으로 감소하지는 않는다. 따라서 후류속도 U_s 는 다음과 같이 표현할 수 있다.

$$U_s = U_0 (1 - f(Se)) \quad (2.10)$$

유효슬립비와 후류속도 사이에는 선형적인 관계가 있다는 것이 확인되어 있으므로 [6], 유효슬립비와 후류속도와의 관계는 다음과 같이 표시할 수 있다.

$$U_s = U_0 (1 - K4 \times Se) \quad (2.11)$$

그리고 선미부분의 유체흐름이 선수미선과 α_e 라는 각을 이루고 흐르고 있으므로 후류벡터속도는 다음과 같이 표시될 수 있다.

$$\bar{U}_s = \frac{U_s}{\cos(\alpha_e)} \quad (2.12)$$

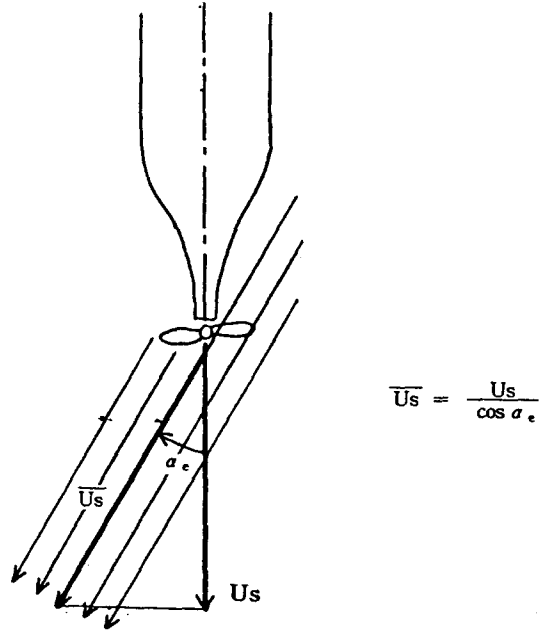


Fig. 7 Vector Slipstream Velocity

2.2.1.5 타에 미치는 lift 와 drag force

Fig. 8에서 알 수 있듯이 타에 미치는 힘은 실제 타의 각에 좌우되는 것이 아니고 타와 그 주위의 유체흐름이 이루는 각에 의해 좌우된다. 만약 타각은 주어져 있으나 그 때 타가 그 주위의 유체흐름 방향과 평행하다면 타에 미치는 lift force는 없을 것이다.

여기서 유효 drift angle α_e 가 실제 drift angle α 과 비례한다고 가정하면 다음과

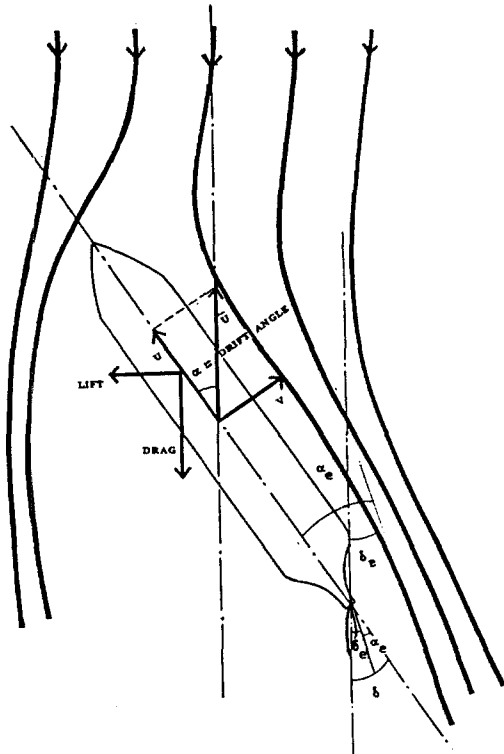


Fig. 8 Lift and Drag Forces and Fluid Stream of a Ship at a Drift Angle α

과 같이 표시할 수 있다.

$$\alpha_e = K2 \times \alpha \quad (2.13)$$

유효타각 δ_e 는 타의 중심선과 그 주위의 유체 흐름 방향이 이루는 각으로 정의되며, 다음과 같이 표시된다.

$$\delta_e = \delta - \alpha_e \quad (2.14)$$

그리고 타도 유효타각이 실효값을 넘으면 선체의 경우와 마찬가지로 stall현상이 일어나므로 제한유효타각(δ_e') 개념을 도입하여 다음과 같이 표현할 수 있다.

$$\begin{aligned} |\delta_e| \leq K3 \text{ 일 경우 } & \delta_e' = \delta_e \\ \delta_e > K3 \text{ 일 경우 } & \delta_e' = K3 \\ \delta_e < -K3 \text{ 일 경우 } & \delta_e' = -K3 \end{aligned} \quad (2.15)$$

따라서 타에 미치는 lift와 drag force를 식으로 표시하면 다음 식 (2.16), (2.17)과 같이 된다.

$$L_R = K10 \times \overline{U_s}^2 \times \delta_e' \quad (2.16)$$

$$D_R = K11 \times \overline{U_s}^2 + K12 \times \overline{U_s}^2 \times \delta_e'^2 \quad (2.17)$$

2.2.1.6 프로펠라에 의한 추력

프로펠라에 의한 추진력은 throttle setting과 프로펠라 슬립의 크기에 의해 좌우된다. Fig. 9에서와 같이 프로펠라에 의한 추력은 유효슬립비에 따라 증가한다. Fig. 10에서 보는 바와 같이 추력계수 K_t 는 직경이 D 인 프로펠라의 경우 명목슬립비 S_a 와 거의 일정한 비율로 변한다. 주어진 throttle setting 값에 대하여 추력 T 가 유효슬립비 S_e 와 일정 비례관계에 있다고 가정하면 추력 T 는 다음 식과 같이 표현된다.

$$T = T_0 (1 + K6 \times S_e) \quad (2.18)$$

여기서, T_0 는 일직선으로 전진운동할 때의 평정 추력이다.

이때 추력과 throttle이 일정 비례관계에 있다고 가정하면 다음 식과 같이 표현할 수 있다.

$$T = K5 \times T_h \times (1 + K6 \times S_e) \quad (2.19)$$

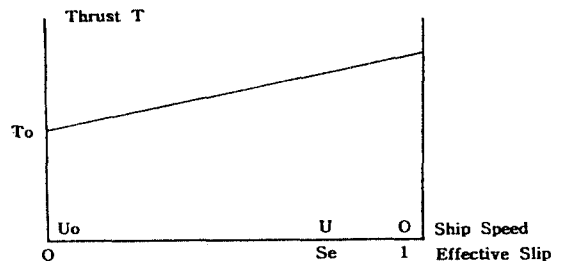


Fig. 9 Assumed Relationship between Thrust and Effective Slip

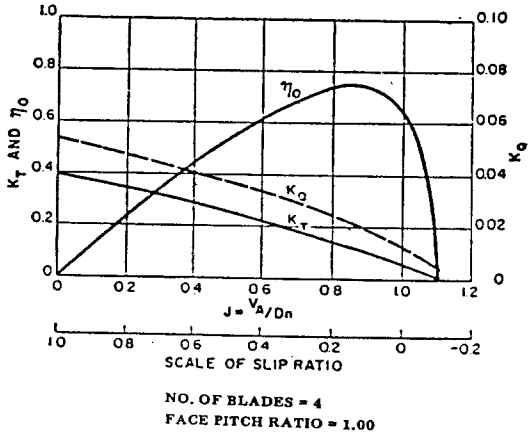


Fig. 10 Typical Curves of Thrust, Torque and Efficiency for Propeller Open Sea[7]

2.2.1.7 선박운동방정식의 구성

본 논문에서 사용한 수학 모델에서는 선체의 움직임을 세 가지의 자유운동 즉 두 가지의 힘과 한 가지의 모멘트로 표시한다. 지구에 고정된 좌표축에 관한 선박의 운동방정식을 뉴턴법칙으로 표현하면 다음 식과 같다.

$$\begin{aligned} X_0 &= m \ddot{X}_0 \\ Y_0 &= m \ddot{Y}_0 \\ N &= I_z \ddot{\psi} \end{aligned} \quad (2.20)$$

여기서 \ddot{x}_0 와 \ddot{y}_0 는 각각의 방향의 가속도를 나타낸다.

X_0 와 Y_0 는 x_0 및 y_0 방향에 작용하는 총힘의 양을 나타낸다.

x_0 와 y_0 는 선체 중심을 기준으로 한 직교좌표 방향을 나타낸다.

m 은 선체 질량

N 은 선체중심에 작용하는, z_0 축과 평행한 총 모멘트를 나타낸다.

I_z 는 z_0 축에 작용하는 선체의 관성모멘트를 나타낸다.

ψ 는 선수미선과 x_0 축이 이루는 yaw 각을 나타낸다.

고정축 x_0 와 y_0 를 가지는 지구에 고정된 좌표계 대신 선수미선을 x 축, 선체정평선을 y 축으로 하는 움직이는 좌표계를 생각해 보면, Fig. 11에서 보는 바와 같이 x 축과 y 축에 작용하는 힘 X 와 Y 를 X_0 와 Y_0 를 이용하여 다음과 같이 표시할 수 있다.

$$\begin{aligned} X &= X_0 \cos \psi + Y_0 \sin \psi \\ Y &= Y_0 \cos \psi - X_0 \sin \psi \end{aligned} \quad (2.21)$$

이때 속도는 다음과 같이 표시된다.

$$\begin{aligned} \dot{x}_0 &= u \cos \psi - v \sin \psi \\ \dot{y}_0 &= u \sin \psi + v \cos \psi \end{aligned} \quad (2.22)$$

여기서 u 와 v 는 각각 x 축과 y 축 방향의 선체 속도를 의미한다.

위의 식을 시간에 대하여 미분하면,

$$\begin{aligned} \ddot{x}_0 &= \dot{u} \cos \psi - \dot{v} \sin \psi - \\ &\quad (u \sin \psi + v \cos \psi) \dot{\psi} \\ \ddot{y}_0 &= \dot{u} \sin \psi + \dot{v} \cos \psi + \\ &\quad (u \cos \psi - v \sin \psi) \dot{\psi} \end{aligned} \quad (2.23)$$

(2.23)식을 (2.20)식에 대입한 뒤 그 결과값 X_0 와 Y_0 를 (2.21)식에 대입하면 다음 식을 얻는다.

$$\begin{aligned} X &= m(\dot{u} - v\dot{\psi}) \\ Y &= m(\dot{v} + u\dot{\psi}) \end{aligned} \quad (2.24)$$

위의 (2.24)식과 (2.20)식의 세번째 식으로부터 (2.25)식과 같은 선체운동방정식을 얻을 수 있다.

$$\begin{aligned} X &= m(\dot{u} - v\dot{\psi}) \quad \text{--- surge equation} \\ Y &= m(\dot{v} + u\dot{\psi}) \quad \text{--- sway equation} \\ N &= I_z \ddot{\psi} \quad \text{---- yaw equation} \end{aligned} \quad (2.25)$$

위의 각 식은 선박의 surge, sway, yaw 운동을 각각 나타내고 있다.

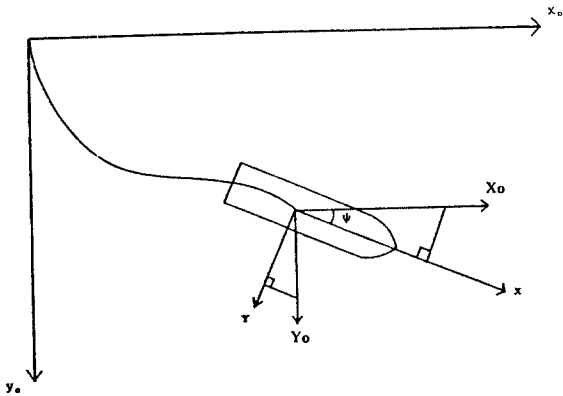


Fig. 11 Relationship Between Axes Fixed in the Earth and Axes in the Moving Ship

위의 (2.25)식에 부가 질량 효과를 적용하면 다음과 같이 표현된다.

$$\begin{aligned} X &= m_1 \dot{u} - m_2 v \dot{\psi} \\ Y &= m_2 \dot{v} + m_1 u \dot{\psi} \\ N &= I_z \ddot{\psi} \end{aligned} \quad (2.26)$$

x축 방향의 부가 질량은 m_1 에, y축 방향의 부가 질량은 m_2 에, 그리고 z축 방향의 부가 관성은 I_z 에 포함되어 있으며, 이들은 크기에 변함이 없이 일정하다고 가정한다[8].

x축과 y축 방향의 힘과 z축 방향의 모멘트는 프로펠러, 타, 유체역학에 의해 선체에 작용하는 외부 힘의 합을 의미한다. Fig. 12는 조종중인 선박에 작용하는 힘과 모멘트를 나타낸다.

선수미선과, 평행하게 선체에 미치는 힘들을 합하면 다음과 같은 surge 방정식이 얻어진다.

$$m_1 \dot{u} - m_2 v \dot{\psi} = T + L_H \sin \alpha - D_H \cos \alpha - L_R \sin \alpha_e - D_R \cos \alpha_e \quad (2.27)$$

따라서 surge 가속도는 다음과 같이 표현된다.

$$\dot{u} = \frac{1}{m_1} (T + L_H \sin \alpha - D_H \cos \alpha - L_R \sin \alpha_e - D_R \cos \alpha_e + m_2 v \dot{\psi}) \quad (2.28)$$

선수미선과 직각인 방향으로 선체에 미치는 힘들을 합하면 다음과 같은 sway 방정식이 얻어진다.

$$m_2 \dot{v} + m_1 u \dot{\psi} = -L_H \cos \alpha - D_H \sin \alpha + L_R \cos \alpha_e - D_R \sin \alpha_e \quad (2.29)$$

따라서 sway 가속도는 다음과 같이 표현된다.

$$\dot{v} = \frac{1}{m_2} (-L_H \cos \alpha - D_H \sin \alpha + L_R \cos \alpha_e - D_R \sin \alpha_e - m_1 u \dot{\psi}) \quad (2.30)$$

선체에 미치는 저항 토크는 회두방향의 반대방향으로 작용하며, 회두속도의 제곱에 비례한다고 가정하면 저항 토크는 다음과 같이 표시될 수 있다[8].

$$N_v = K_{13} \dot{\psi} |\dot{\psi}| \quad (2.31)$$

yaw 방정식은 선체의 무게중심에 미치는 모멘트로 구성되므로 우선 선체와 타의 압력중심을 알아야 한다. drift angle의 변화에 따라 압력중심이 약간 변하기는 하지만 선체와 타의 압력중심은 선체와 타의 중심선상의 일정한 위치에 존재한다고 가정하여도 무방하다[9].

Fig. 12에서와 같이 선체 압력중심은 선체의 무게중심으로부터 d_1 만큼 앞쪽에 위치하고, 타의 압력중심은 선체의 무게중심으로부터 d_2 만큼 뒤쪽에 위치한다. 선체 무게중심에서의 모멘트를 계산하면 yaw 방정식은 다음과 같이 표시될 수 있다.

$$\ddot{\psi} = \frac{1}{I_z} (-N_v - L_H d_1 \cos \alpha - D_H d_1 \sin \alpha - L_R d_2 \cos \alpha_e + D_R d_2 \sin \alpha_e) \quad (2.32)$$

이들 각 surge, sway, yaw 가속도 \dot{u} , \dot{v} , $\ddot{\psi}$ 를 구한 다음 이들 가속도를 오일러 공식을 이용하여 시간에 대해 적분하면 각각의 속도를 구할 수 있다.

시간 t 에서의 surge, sway, yaw 속도를 각각 u_t , v_t , $\dot{\psi}_t$ 라 하고 시간 t_1 에서의 surge, sway, yaw 속도를 각각 u_{t_1} , v_{t_1} , $\dot{\psi}_{t_1}$ 이라 하면

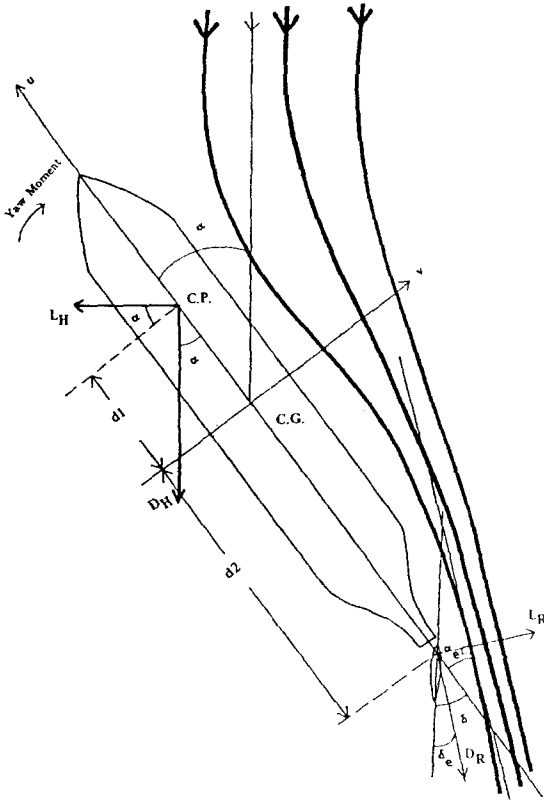


Fig. 12 Forces and Moments Acting on a Manoeuvring Ship

$$\begin{aligned}
 u_{t_1} &= u_t + \int_t^{t_1} \dot{u} dt = u_t + \dot{u} \Delta t \\
 v_{t_1} &= v_t + \int_t^{t_1} \dot{v} dt = v_t + \dot{v} \Delta t \\
 \psi_{t_1} &= \psi_t + \int_t^{t_1} \dot{\psi} dt = \psi_t + \dot{\psi} \Delta t
 \end{aligned}
 \tag{2.33}$$

(단, $\Delta t = t_1 - t$)

이 된다. 그리고 이들 속도를 각각 시간에 대하여 적분하면 일정 시간동안 선박이 이동한 각 방향의 거리 및 회전각이 구해진다.

2.2.2 상대선의 설계

2.2.2.1 입항 환경의 경우

시물레이션 시작시 환경 설정이 입항선으로 설

정되면 현재의 마산항로의 모습이 레이다 화면에 나타나고, 본선은 가덕수도에서 마산항로와 충무항로로 나뉘어지는 지역에 위치하게 된다. 시물레이션 시작과 동시에 상대선들은 부도로 부터 서쪽으로 약 1.4 마일의 거리에 위치한 부도수도 입구에서 출항하도록 설계하였다. 항내의 경우 출항하는 선박은 일정한 항로를 따라 운항해야 하므로 상대선들은 약간의 속력차만 있고 침로는 시물레이션 도중에 변경되지 않도록 하였다. 상대선은 시물레이션 시작시에 가지는 침로와 속력을 시물레이션 종료시까지 계속 유지하도록 하였다. 환경이 입항선일 경우 피항조건보다는 침로유지 및 본선의 위치 설정이 더 중요하므로 상대선은 다섯개만 발생하는 것으로 프로그래밍하였다.

2.2.2.2 협수로 환경의 경우

시물레이션 시작시 환경 설정을 협수로로 하면 한번의 굴곡이 있는 가상의 협수로가 나타나며, 협수로의 폭은 약 3마일이 되도록 설계하였다. 시물레이션 시작시 본선은 협수로로 상행하도록 하였고 상대선은 하행하도록 하였다. 상대선은 10개를 만들어 동시에 두개, 또는 한개씩 본선과 거리차를 두고 하행하도록 하였으며, 각 상대선의 침로와 속력은 조금씩 다르게 하였다. 그리고 상대선의 속력은 처음 설정된대로 중간에 변함이 없도록 하였으나 침로는 협수로의 굴곡부에서 한번의 침로변경이 있도록 설계하였다.

기존의 시물레이터는 일정한 시나리오를 가지고 진행되기 때문에 반복적 효과가 적었으나, 본 시물레이터에서는 10개의 상대선이 시물레이션 시작 시간에 따라 순서를 조합하여 무작위로 발생되도록 하였으므로 한 사람의 조종자가 계속적으로 사용하여 훈련효과를 높일 수 있도록 하였다.

2.2.3 본선과 상대선의 관계 계산

본선과 상대선은 속력 즉 크기와 침로 즉 방향을 가진 벡터로 나타낼 수 있다. 실제 선박에서는 전파를 발사하여 레이다 화면상에 나타난 상대선의 움직임을 계산하여 상대선의 진침로와 진속력, 그리고 CPA, TCPA등을 계산한다. 그러나 본 시

플래이터에서는 상대선의 속력과 침로는 정해주고 본선의 침로와 속력의 합벡터를 구하여 물표(상대선)가 상대적인 움직임을 갖도록 하였다.

2.2.3.1 합벡터의 계산

먼저 상대선과 본선의 침로(방향)와 속력(크기)을 가진 벡터를 본선좌표값(시점)과 일정한 좌표값(중점)을 성분으로 갖는 평면벡터로 바꾸어 준다. 본선좌표값(시점)은 항상 원점이므로 크기와 방향을 이용하여 벡터의 중점 성분을 구한다.

본선이나 상대선의 침로를 θ , 속력을 ω 라 하면, 침로가 $180^\circ \leq \theta < 360^\circ$ 인 경우에는 벡터는 2사분면이나 3사분면에 있게 되고 따라서 x축 좌표값에 - 부호를 붙이고, $270^\circ \leq \theta < 090^\circ$ 인 경우에는 벡터는 1사분면이나 2사분면에 있게 되고 따라서 y축 좌표값에 - 부호를 붙인다. 침로가 $315^\circ \leq \theta \leq 045^\circ$ 이거나 $135^\circ \leq \theta \leq 225^\circ$ 인 경우에는 x축 값은 침로각의 코사인 함수를 이용하여 계산하고, y축 값은 나머지변들의 제곱값 차이에 제곱근을 취하여 다음과 같이 계산한다.

$$\begin{aligned} x &= \omega \times \cos \theta \\ y &= \sqrt{\omega^2 - x^2} \end{aligned} \quad (2.34)$$

침로가 그외의 각일 경우는 동일한 계산과정에서 적용되지만 y축 값을 찾기 위하여 그 각의 사인값을 사용하고 x축에 대하여는 피타고라스의 정리를 적용한다.

$$\begin{aligned} y &= \omega \times \sin \theta \\ x &= \sqrt{\omega^2 - y^2} \end{aligned} \quad (2.35)$$

이는 사인 함수와 코사인 함수가 나타내는 최대 정확도의 범위내에서만 사인 함수와 코사인 함수를 적용한 후 (y축 값이 x축 값보다 클 때 사인 함수를 적용시키고 x축 값이 y축 값보다 클 때는 코사인 함수를 적용시킨다[10]), 피타고라스의 정리를 응용하여 나머지 값을 계산함으로써 결과로 나타날 좌표값이 가능한 한 정확하도록 하였다.

위와 같이, 본선과 상대선의 벡터를 좌표값으로 변환하여 얻은 본선의 벡터변환좌표값을 (x1, y1),

상대선의 벡터변환좌표값을 (x2, y2)이라 한다면 본선과 상대선의 합벡터의 좌표값은 (x1+x2, y1+y2)가 된다.

여기서 $x1+x2=a$, $y1+y2=b$ 라 놓고, 이 합벡터의 좌표를 크기(ω)와 방향(θ)으로 변환하면,

$$\text{합벡터의 크기 } \omega = \sqrt{|a|^2 + |b|^2} \text{ 이 된다.} \quad (2.36)$$

그리고 합벡터의 방향 θ 는

$$\begin{aligned} a > 0 \text{이고 } b > 0 \text{인 경우에는 } \theta &= \text{atan}(a/b) \\ a < 0 \text{이고 } b < 0 \text{인 경우에는 } \theta &= \text{atan}(a/b) + 180^\circ \\ a > 0 \text{이고 } b < 0 \text{인 경우에는 } \theta &= \text{atan}(b/a) + 90^\circ \\ a < 0 \text{이고 } b > 0 \text{인 경우에는 } \theta &= \text{atan}(b/a) + 270^\circ \\ |x| = 0 \text{ 인 경우 } y < 0 \text{ 이면 } \theta &= 0^\circ \text{ 이거나 } 360^\circ \\ y > 0 \text{ 이면 } \theta &= 180^\circ \text{ 이 된다.} \end{aligned} \quad (2.37)$$

2.2.3.2 최근접거리(CPA)와 최근접시간(TCPA)의 계산

본선과 상대선의 합벡터의 좌표를 (a, b), 합벡터의 크기를 ω , 합벡터의 방향을 θ 라 하면 본선의 좌표는 항상(0, 0)이므로 CPA와 TCPA는 다음과 같이 계산될 수 있다.

(1) $\theta = 0^\circ$ 이거나 180° 인 경우

$$\begin{aligned} \text{CPA} &= |a| \\ \text{TCPA} &= \text{CPA} / \omega \end{aligned}$$

(2) $\theta = 90^\circ$ 이거나 270° 인 경우

$$\begin{aligned} \text{CPA} &= |b| \\ \text{TCPA} &= \text{CPA} / \omega \end{aligned}$$

(3) 좌표(a, b)와 방향 θ 로써 직선의 방정식을 유도하면, 그 기울기 m은 다음과 같다.

$$\begin{aligned} 0^\circ < \theta < 90^\circ \text{ 이면 기울기 } m &= \tan(90 - \theta) \\ 90^\circ < \theta < 180^\circ \text{ m} &= -1 \times \tan(\theta - 90) \\ 180^\circ < \theta < 270^\circ \text{ m} &= \tan(270 - \theta) \\ 270^\circ < \theta < 360^\circ \text{ m} &= -1 \times \tan(\theta - 270) \end{aligned} \quad (2.38)$$

따라서 기울기가 m이고 좌표(a, b)를 지나는 직선의 방정식은

$$mx - y - ma + b = 0 \text{ 가 된다. 여기서 } m=k,$$

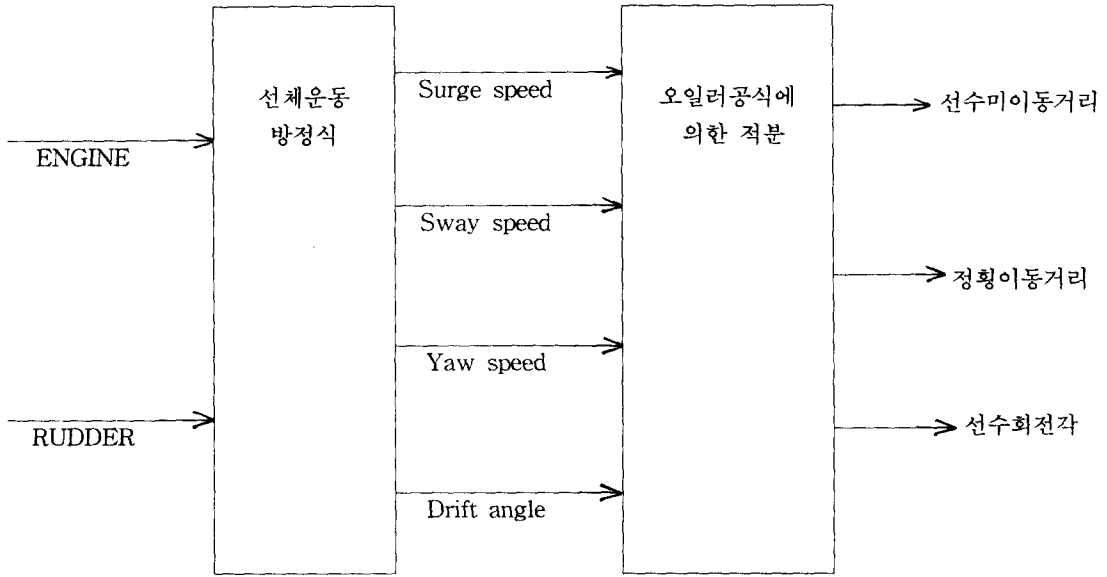


Fig. 13 Concept of Personal Computer Aided ARPA RADAR Simulator

$-1=k_1$, $-ma+b=k_2$ 라 하면 위의 직선과 좌표(0, 0)과의 거리가 CPA가 되므로

$$CPA = \frac{|k_2|}{\sqrt{k_1^2 + k_2^2}} \text{ 이 된다. } \quad (2.39)$$

그리고 좌표(a, b)와 본선좌표(0, 0)간의 거리를 l이라 하면, 좌표(a, b)와 CPA가 되는 지점간의 거리 d는 다음과 같다.

$$d = \sqrt{l^2 - CPA^2} \quad (2.40)$$

따라서 $TCPA = d / \omega$ 가 된다. (2.41)

2.2.4 프로그래밍 방법

2.2.4.1 초기 화면 구성

시뮬레이터 로고나 레이더 화면 등은 Visual Basic을 사용하여 구현하였으며, 시뮬레이션을 시작할 때에 조종자가 시뮬레이션 환경을 설정할 수 있도록 하였다.

환경설정에는 입항 환경이나 협수로 환경중에 하나를 선택하도록 한 항로설정과 3가지 종류의 본선 즉, 만선의 21,000톤급 빠른 화물선, 만선과 공선의 200,000만톤 유조선중에 한가지를 선택할

수 있도록 한 본선설정이 있고 초기에 조종자가 이러한 시뮬레이션 환경을 선택할 수 있도록 하였다.

2.2.4.2 주프로그램의 구성

본선의 운동방정식과 본선과 상대선의 벡터해석은 따로 함수로 만들어 놓고 이를 주프로그램에서 필요시 마다 사용할 수 있도록 하였다. 본선의 운동방정식 함수의 경우 현재 사용되고 있는 타와 엔진의 상태를 입력치로 하여 본선의 속도 및 침로를 출력하고 이 출력치를 주프로그램에서 사용하여 화면에 나타나 있는 물표들의 움직임을 제어하도록 하였다.

벡터해석함수의 경우는 본선운동방정식 함수의 출력치인 본선의 속도와 침로를 상대선의 속력과 침로와 함께 벡터해석 함수의 입력치로 사용하여 상대선의 상대속력, 상대침로, CPA, TCPA를 계산하고 이를 주프로그램에서 화면출력의 정보로 사용할 수 있도록 하였다.

이와같이 주프로그램을 사용하여 매 2초에 한번씩 입력, 계산, 화면출력의 과정을 되풀이 하여 본선의 선체운동방정식 계산, 레이더의 레이다 계산, 각종 계산치의 디스플레이, 상대선의 발생 및 상대

선의 상대속력, 상대침로, CPA, TCPA계산등을 행한다.

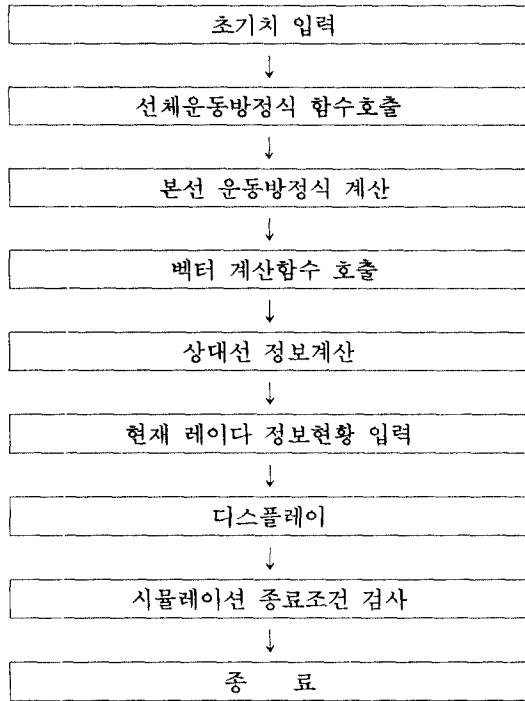


Fig. 14 Main Program Arrangement

2.2.4.3 시뮬레이션 종료

입항 환경일 경우, 본선과 상대선 또는 본선과 육지 물표와의 거리가 50미터 이내로 가까워질 경우 충돌이나 좌초로 간주하여 비상벨소리와 함께 시뮬레이션이 종료되도록 하였고, 본선이 무사히 내항방파제를 통과할 경우는 사용자가 시뮬레이션을 종료하고자 할때까지 계속 실행할 수 있도록 하였다.

협수로 환경일 경우는 본선과 상대선간의 거리가 0.1마일 이내로 접근할 경우 충돌로 간주하고, 본선과 육지물표와의 거리가 0.5마일 이내로 접근할 경우 좌초로 간주하여 비상벨 소리와 함께 시뮬레이션 로고로 되돌아 가면서 시뮬레이션을 종료하도록 하였다. 그리고 상대선 10개중 마지막 상대선이 본선과 거리가 멀어질 경우 시뮬레이션의 훌륭한 수행을 알리는 메시지와 함께 시뮬레

션 로고로 되돌아 가면서 시뮬레이션을 종료하도록 하였다.

제 3 장 시뮬레이터의 기능과 조작 방법

재래의 레이더를 이용할 경우에는 사용자는 레이다 화면상에 발생한 물표에 대해 그 물표의 거리와 방위에 대한 정보만을 얻을 수 있었고 그 물표의 움직이는 방향이나 속도에 대한 정보를 얻고자 할 때에는 일정시간 동안 여러차례 레이다 화면에 색연필로 물표를 표시(plotting)하여 벡터 해석에 의해 그 물표의 실제 움직임을 파악하였으나, ARPA레이더는 이러한 번거로운 작업을 자동적으로 처리할 수 있는 기능을 갖고 있어서 사용자가 물표를 표시만 해주면 자동적으로 물표의 움직임에 대한 정보를 계산하여 그 결과를 사용자에게 알려준다. 이러한 기본적인 auto plotting 기능이 외에도 ARPA레이더에는 협수로를 통과할 경우에 본선의 안전항행구역에 레이다 화면상에 설정하거나 피항조선시에 실제로 피항동작을 취하기 전에 피항동작후의 정보를 예측할 수 있는 기능들이 있다.

본 시뮬레이터는 PC만 사용하여 제작하였고 또한 사용한 프로그램 언어에 한계가 있어서 ARPA 레이더의 전 기능들을 구현하지는 못했지만 레이더의 기본 기능과 auto plotting기능이 포함되어 있다.

이 장에서는 본 시뮬레이터에 구현되어 있는 기능을 타와 엔진의 입력기능과 레이더 본래의 기능으로 나누고 이들 기능과 조작 방법에 대하여 기술한다.

3.1 입력기능

본 시뮬레이터에는 레이더의 조작훈련을 위한 ARPA 레이더의 기능이외에 피항조선 훈련을 위해 사용자가 타와 엔진을 사용할 수 있도록 이들의 입력장치도 설계되어 있다. 엔진은 선박의 전후진 추력을 얻기 위하여, 그리고 타는 선박의 좌

우 움직임을 제어하기 위하여 사용되는데 각각의 입력장치의 사용방법과 그 기능들은 다음과 같다.

3.1.1 타각입력

타각(rudder angle)은 사용자가 본선의 움직임을 제어하기 위하여 사용하는데 화면 좌측 상단에 있는 오른쪽 삼각표(▶)를 마우스로 누르면 타각이 1도씩 증가하고 반대쪽 삼각표(◀)를 누르면 1도씩 감소하도록 설계하였다. 실제 선박에서는 선교(bridge)에서 타의 사용각도를 지시하면 타각 즉시 그 지시각도 만큼 움직이는 것이 아니고 지시후 일정 시간이 지난 다음 지시각도에 도달하기 때문에 시간지연이 발생하며, 이러한 시간지연을 본 시뮬레이터에 적용하기 위해 타각의 증감을 1도로 하여 실제 선박의 타의 움직임과 유사하게 설계되었다. 최대 타각은 좌우 35도로 되어 있고 그 이상의 값을 입력하더라도 타각은 최대 타각의 범위를 넘지 않도록 하였다. 그리고 현재의 타각은 두 삼각형의 가운데 상단에 좌현(port)쪽 타각일 경우 - 부호와 함께 타각이 표시되고 밀바탕이 붉게 변하며, 우현(starboard)쪽 타각일 경우는 부호가 없는 양의 값으로 나타나고 밀바탕이 푸르게 된다. 초기치 타각은 0도이다.

3.1.2 엔진 입력

엔진(engine)은 사용자가 본선의 전후진 추력을 얻기 위하여 사용하는데 본 시뮬레이터에서 사용할 수 있는 엔진상태는 실제 선박에서 사용하고 있는 것과 같이 전진추력을 위한 full ahead, half ahead, slow ahead, dead-slow ahead, 후진추력을 위한 dead-slow astern, slow astern, half astern, full astern, 그리고 엔진의 사용을 멈추기 위한 stop등 모두 9가지로 나뉘어져 있다.

시뮬레이션 시작시에 사용자가 본선의 종류를 결정할 수 있도록 하였는데 각각의 본선 종류마다 최대 전진(full ahead) 때의 추력의 값을 1로 하고 정지(stop)일 때의 추력값을 0으로 하여 각각의 엔진상태의 추력값을 1에서 -1까지 정해 놓았다. 실제 선박에서는 각각의 엔진 상태에 따른 추력으로써 선박의 속력을 계산해 놓고 있는데 본 시뮬레

이터는 역으로 각각의 엔진상태에 해당 선속을 정의해 놓고 이것으로 본선의 추력을 계산하도록 하였다. 사용자가 원하는 엔진상태를 레이다 화면 우측 상단에 설치된 엔진 입력 부분에서 선택하여 마우스의 좌측 버튼으로 누르면 선택된 엔진상태가 회색에서 흰색으로 바뀐다. 초기의 엔진상태는 stop engine 이다.

3.2 레이다 기능 및 조작 방법

본 시뮬레이터가 모델로 사용한 ARPA 레이다는 재래의 레이다 또는 타사의 레이다에 비해 각각의 기능을 조절하는 knob을 가능한 한 없애고 그 대신 화면의 좌측 하단부에 pull-down형식의 메뉴박스를 만들어 사용함으로써 레이다 조절판(control panel)을 간소화한 제품이다. 본 시뮬레이터는 이 모델을 가능한 한 그대로 구현하고자 하였으나 여러가지 software와 hardware적인 제약 조건으로 인해 전 기능을 구현하지는 못하였으며, 실제의 레이다와 조작방법이 조금 다른 기능들도 있다.

화면 좌측 하단부에 있는 메뉴박스의 구성은 다음과 같다.

Radar Work	- Range	- 0.75, 1.5, 3, 6, 12, 24
	- Vector	- True, Relative
Presentation	- Bearing	- Head Up, Course Up, North Up
	- Range Rings	- ON, OFF
	- SHM	
	- EBL	- EBL1, EBL2 - CENT, INDP
	- VRM	- ON, OFF
	- HOME	
	- Motion	- True, Relative
	- Off Center	
ARPA Function	- Guard Ring	- ON, OFF - Fm, To, Range, Draw
	- Nav. Line	- Draw, Clear
	- ARPA	- Manual, Acquire, Clear

이와 같이 사용할 수 있는 기능들을 크게 3가지 그룹으로 나누고 각각의 그룹에 관련 기능들을 포함시켜 놓았으며 각각의 기능을 그룹별로 나누어 설명하면 다음과 같다.

3.2.1 Radar Work

이 선택바를 마우스로 누르면 레이더 화면의 출력범위를 정하는 Range기능과 물표의 방위표시를 위한 Vector기능이 나오며 각각의 기능은 다음과 같다.

(1) Range

레이더 화면의 출력범위(range)를 정하는 것으로 0.75, 1.5, 3, 6, 12, 24마일까지 선택할 수 있으며 초기의 range 범위는 0.75마일로 설정되어 있다. 레이더 화면상에 존재하는 상대선이나 육지물표는 사용자가 선택한 range에 따라 크기와 거리가 변하도록 설계되어 있다. range가 커지면 상대선이나 육지물표의 크기는 작아지고 레이더 화면상의 거리는 본선과 가까워지며, range가 작아지면 상대선이나 육지물표의 크기는 커지고 화면상 본선과의 거리는 멀어진다. 원하는 range에 커서를 놓고 마우스를 누르면 레이더 화면 하단에 설정된 range가 표시되면서 레이더 화면이 설정된 range로 출력된다.

(2) Vector

사용자가 ARPA Function의 ARPA 메뉴에서 상대선의 진속도와 진방향을 표시하도록 지정하였을 때, 해당 상대선의 벡터를 상대벡터로 표시할 것인지 아니면 진벡터로 표시할 것인지, 그리고 그 벡터의 크기는 얼마로 하여 표시할 것인지 등을 정한다. 이 Vector에 속해있는 기능으로 True와 Relative, 그리고 Size가 있으며, True를 선택하면 상대선의 벡터는 상대선만의 벡터인 진벡터로 표시되고 레이더 화면 좌측부분에 Vector T라고 출력된다. 그리고 Relative를 선택하면 상대선과 본선의 합벡터인 상대벡터로 벡터가 표시되고 화면 좌측부분에 Vector R이라 출력된다. 그 벡터의 크기는 Size에서 3, 6, 12분을 단위로 하여 정할 수 있도록 하였다.

3.2.2 Presentation

이 선택바를 마우스로 누르면 ARPA기능을 제외한 레이더 본래의 기능이 8가지 나오며, 사용자

는 원하는 기능을 선택하여 사용할 수 있다. 각각의 기능들에 대한 설계 방법과 사용 방법에 대하여 설명하면 다음과 같으며, 각각의 기능은 단독으로 또는 다른 기능과 동시에 사용될 수 있도록 설계되었다.

(1) Bearing

레이더 화면의 표시방법을 정하는 기능으로서 본선의 heading(선수)을 레이더 화면의 상단 중앙에 고정시켜 놓고 레이더 화면을 보고자 할 때는 Head Up을 선택하고, 정북(true north)을 화면의 상단 중앙으로 하여 보고자 할 때에는 North Up, 본선의 침로(course)를 레이더 화면 상단 중앙으로 하여 보고자 하는 경우는 Course Up을 선택한다. 초기치는 North Up으로 설정되어 있다.

North Up 상태에서는 화면의 중앙 상단이 정북을 나타내고 본선의 선수를 나타내는 직선은 본선의 침로가 바뀔 때 따라 변한다. North Up 상태에서 Head Up상태로 바꾸면 본선의 선수를 나타내는 직선은 화면 중앙 상단으로 옮겨지고 상대선이나 육지물표는 본선의 선수를 중심으로 해서 그려지며, 본선의 침로가 변하더라도 선수선은 일정하게 중앙 상단에 위치하나 상대선이나 육지물표는 본선 침로의 변화각만큼 반대로 이동한 것처럼 그려진다. 현재 설정된 상태는 레이더 화면 상단에 표시된다.

(2) Range Rings

Range Rings는 레이더 화면에 여러개의 물표가 동시에 발생하여 VRM으로 각각의 물표와 본선과의 거리를 일일이 파악하기가 어려울 때 이들 물표의 거리를 쉽게 파악할 수 있도록 레이더 화면의 반지름을 6등분하여 본선을 중심으로 원을 6개 그리는 기능이다. 이렇게 함으로써 동시에 여러개의 물표의 거리를 대략적으로 동시에 파악할 수 있다. 만약 현재 range가 12마일로 설정되어 있는 경우에는 원과 원사이의 거리는 2마일, 6마일로 설정되어 있는 경우에는 원과 원사이의 거리가 1마일이 된다. range rings를 사용할 필요가 없을 때에는 off 기능을 사용하여 그려졌던 range

rings를 없앤다.

(3) Motion

Motion 기능에는 true motion 기능과 relative motion 기능이 있으며, 초기치는 relative motion 으로 설정되어 있다. relative motion은 특수한 경우를 제외하고 실제로 선박에서 사용하는 방법으로서 본선은 화면 중앙에 위치하고 움직이지 않으며, 육지물표나 상대선은 본선과의 상대벡터로 화면상에서 움직인다. 그러나 육지물표를 레이다 화면에 고정시켜 놓고 본선의 움직임을 파악하고자 할 경우에는 true motion 기능을 선택하여 육지물표는 움직이지 아니하고 상대선들은 각각의 진벡터로 움직이며 본선도 화면 중앙으로부터 본선의 벡터로 움직이도록 할 수 있다. 이 경우에 본선이 레이다 화면의 반지름의 2/3정도 레이다 화면 중앙으로부터 벗어나면, 본선은 현재의 본선 침로 반대방향 2/3지점으로 옮겨진다.

(4) SHM

만약 본선의 정선수에 작은 물표가 나타난다면, 선수선에 가려서 물표의 확인이 곤란할 경우가 발생할 수 있는데, 이러한 경우에 일시적으로 본선의 선수를 나타내는 선을 지우고자 할 경우에 사용하는 기능이 SHM 기능이다. 이 기능을 ON하면 본선의 선수에 그려진 선이 없어지고 OFF하면 다시 선수선이 나타난다. 실제 레이다에서는 이 스위치를 누르고 있을 때만 선수선이 없어지지만 본 시뮬레이터에서는 선의 유무를 on/off하도록 설계하였다.

(5) EBL

레이다 화면에 발생한 물표의 방위를 알고자 할 때나 그 물표의 방위변화를 점검하고자 할 경우에 사용하는 기능으로서 이 기능을 선택하면 EBL1, EBL2, CLEAR의 세가지 선택 기능이 나오고, 다시 EBL1이나 EBL2 기능을 선택하면 각각 CENT와 INDP의 선택바가 나타난다. CENT를 선택할 경우에는 한쪽 끝이 화면의 중심에 위치하고 다른 한쪽 끝은 화면의 끝부분에 위치한 직선이 나타나

며 메뉴박스에 있는 scroll bar를 사용하여 0도에서 360도 사이의 값을 지정하면 지정한 각도로 직선이 회전한다. 이때 이 직선의 방위각이 화면 좌측 하단에 표시된다. 이렇게 본선을 중심으로 직선을 회전시켜서 파악하고자 하는 물표에 일치시킴으로써 물표의 방위를 파악할 수 있다. INDP를 선택하면 직선의 한쪽끝이 사용자가 커서로써 지정한 곳에 위치하고 다른 한쪽 끝은 화면의 끝부분에 위치한 직선이 나타나며, 이 직선은 사용자가 지정한 위치를 중심으로 회전한다. 이러한 직선을 EBL1과 EBL2에서 각각 한개씩 그릴 수 있으며 또한 CLEAR바를 사용하면 그런 직선을 지울 수 있다.

(6) VRM

레이다 화면에 나타난 물표와 본선과의 거리를 알고자 할 경우에 사용하는 기능으로서 이 기능을 선택하면 레이다 화면 중심을 중심으로 한 원이 나타나며 원의 반지름은 커서와 본선 즉 화면의 중심간의 거리가 된다. 따라서 커서의 위치가 변할때마다 원의 반지름이 변하게 되고 그 반지름이 레이다 화면 좌측 하단에 표시된다. 이렇게 원을 만들어서 원하는 물표에 원을 일치시키면 그 물표와 본선과의 거리를 알 수 있다. 일정한 거리를 반지름으로 하는 원을 화면에 고정시키고자 하는 경우에는 VRM이 만들어진 상태에서 마우스의 왼쪽 버튼을 한번 누르면 원은 그 위치에 고정된다.

(7) Home

레이다 화면상에 있는 커서를 본선의 위치 즉 화면중심으로 옮기고자 할 경우에 이 기능을 사용한다.

(8) Off-Center

항상 레이다는 중앙에 본선이 위치하게 되는데 만약 사용자가 본선의 위치를 화면중심에서 다른 곳으로 옮겨서 레이다 화면을 보고자 할 경우에 이 기능을 사용한다. 이 기능을 선택하고 옮기고자 하는 위치에 마우스로 커서를 옮겨놓은 다음 마우스의 왼쪽 버튼을 누르면 레이다 화면상의 본

선위치가 커서의 위치로 바뀌면서 상대선이나 육지물표도 동시에 옮겨진다. 이 기능은 사용자가 range를 변화시키지 아니하고도 일정방향의 화면을 설정된 range보다 멀리 보고자 할 경우나 피항조선시 주의를 요하는 부분을 한눈에 보고자 할 경우에 사용한다. 다시 한번 이 선택바를 누르면 본선의 위치가 원래대로 화면의 중심으로 옮겨진다.

3.2.3 ARPA Function

(1) Guard Ring

레이다 화면상의 물표가 일정한 거리와 방위를 가지고 본선에 접근하였을 때 경고음이 울리도록 사용자가 거리와 방위를 지정할 수 있는 기능이다. 이때 사용자가 지정하는 거리는 본선으로부터의 거리이며, 방위는 bearing이 North Up상태일 때의 방위로서 시작점 방위와 끝점 방위가 있다. 사용자가 거리와 시작점, 끝점 방위를 지정하면 본선(화면 중심)을 중심으로 하고 지정한 거리를 반지름으로 하며, 시작점 방위로부터 반시계 방향으로 끝점 방위까지 연결하는 붉은 색의 원호가 그려진다. 레이다 화면상의 상대선이나 육지물표가 이 원호에 접근하여 겹치면 경고음이 울리도록 설계되어 있다.

(2) Nav. Line

Nav. Line은 선박이 안전하게 항행하고자 하는, 또는 항행할 수 있는 범위를 레이다 화면상에 여러개의 직선으로 조합하여 만든 항행안전구역선을 말한다. 협수로등을 항행하는 선박이 항행가능한 안전 구역을 미리 설정해 놓고 항행한다면 항행도중에 선박의 안전항행 여부를 항해사나 레이다 사용자가 레이다 화면상에서 쉽게 파악할 수 있다. Nav. Line 기능을 선택하면 Draw, Clear의 두가지 옵션이 나오며, 사용자는 Draw 기능을 사용하여 항행안전구역선을 그릴 수 있고, Clear 기능을 사용하여 그렸던 선을 지울 수 있다. Draw 기능을 선택한 다음, 레이다 화면상의 원하는 곳에 커서를 위치시킨 후 마우스의 왼쪽 버튼을 누르고, 다시 커서를 옮겨 마우스의 왼쪽 버튼을 누르면

두 커서의 위치를 연결하는 직선이 그려지며, 하나의 선을 완성한 후에는 마우스의 왼쪽 버튼을 두 번 연속하여 누른다. Clear 기능을 선택하면 현재 레이다 화면에 그려져 있는 항행안전구역선이 없어진다.

(3) ARPA

ARPA 기능이란 레이다 스스로가 화면상에 나타난 물표의 움직임을 추적하여 그 물표의 진침로와 진속력을 계산하고 이를 벡터 그림이나 숫자로 표시해 주는 기능을 말한다. 재래의 레이다를 탑재한 선박에서는 항해사가 직접 레이다 화면위에 색연필로 물표의 움직임을 표시하여 그 물표의 진침로와 진속력을 파악하였다. 이 기능을 선택하면 MANUAL, ACQUIRE, CLEAR의 세가지 부기능이 나온다. MANUAL 기능을 선택하고 커서를 정보를 얻고자 하는 물표에 놓은 다음 마우스의 왼쪽 버튼을 누르면 물표의 그림 중심으로 부터 물표의 진침로와 진속력을 성분으로 하는 벡터선이 그려진다. ACQUIRE 기능을 선택한 다음 MANUAL 기능에 의해 벡터가 표시되어 있는 물표에 다시 커서를 놓고 마우스의 왼쪽 버튼을 누르면 화면 좌측에 그 물표의 진속력, 진침로, CPA, TCPA에 대한 정보가 출력된다. ACQUIRE 기능에 의해 정보가 출력될 수 있는 물표는 2개이며, MANUAL 기능에 의해 벡터로 표시할 수 있는 물표의 수는 제한이 없다. 이미 선택된 물표에 대한 정보를 없애고자 하는 경우에는 CLEAR기능을 선택한 다음 그 물표에 커서를 옮겨놓고 마우스의 왼쪽 버튼을 누르면 물표에 그려졌던 벡터선과 화면 좌측의 정보가 모두 없어진다.

제 4 장 본 시뮬레이터의 실행 예 및 개선점

4.1 실행 예

시뮬레이션 실행 파일을 실행시키면, 제일 먼저 로고와 함께 시뮬레이션 환경 설정을 위한 입력 화면이 나타난다(Fig. 15). 항로설정은 입항 환경

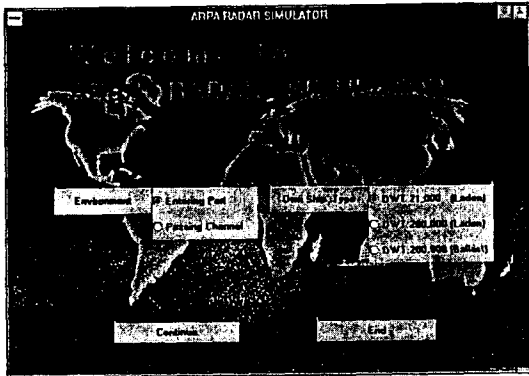


Fig. 15 시뮬레이션 환경 설정을 위한 입력화면

(Entering Port)과 협수로 환경(Passing Channel) 중 사용자가 택일하도록 되어 있고, 본선 설정은 시뮬레이션시 사용할 선박을 DWT 21,000 (Laden), DWT 200,000(Laden), DWT 200,000 (Ballast) 등 세가지 중에서 택일하도록 설계되어 있다. 두가지 설정을 완료하고 Continue 버튼을 누르면 레이더 화면이 나오면서 시뮬레이션이 시작된다. 입항 환경과 협수로 환경의 시뮬레이션 초기 레이더 화면은 각각 다음의 Fig. 16, Fig. 17 과 같다. 그리고 본 시뮬레이터가 가지고 있는 기능들을 사용하여 실행한 시뮬레이션 도중의 레이더 화면은 각각 Fig. 18, Fig. 19와 같다.

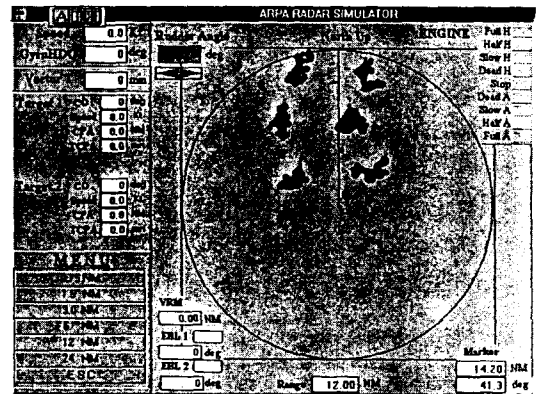


Fig. 17 협수로 환경일 경우의 초기 화면

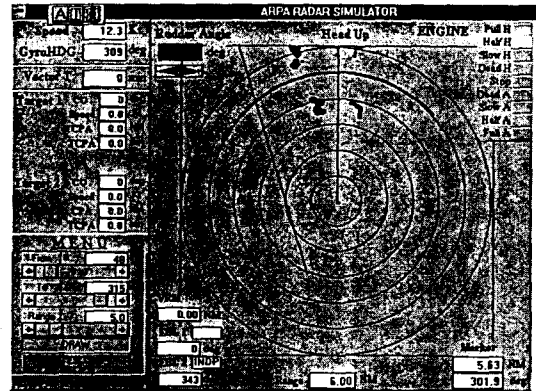


Fig. 18 시뮬레이션 도중의 레이더 화면 (입항 환경의 경우)

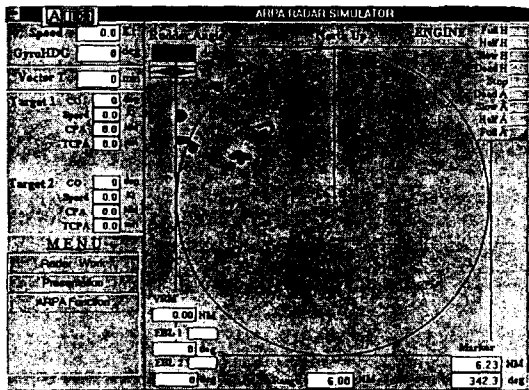


Fig. 16 입항 환경일 경우의 초기 화면

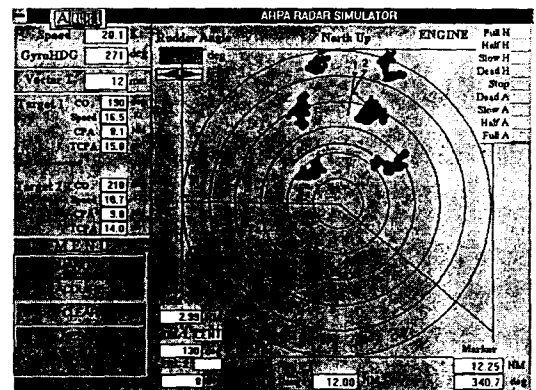


Fig. 19 시뮬레이션 도중의 레이더 화면 (협수로 환경의 경우)

4.2 개선점

본 시뮬레이터의 문제점을 들고 그 개선책에 관하여 기술하면 다음과 같다.

4.2.1 조작장치

본 시뮬레이터는 개인용 컴퓨터에서 쉽게 실행시켜 볼 수 있도록 설계되어 있으나, 실제 레이다가 가지고 있는 모든 기능을 다 갖추고 있지는 않으며, 타각을 주기 위한 wheel이나 엔진을 사용하기 위한 telegraph는 인터페이스를 사용하지 아니하고 레이다 화면에 그림으로 그려져 있기 때문에 실제 레이다 화면보다 화면이 복잡하게 되어 있다.

타와 엔진부분은 인터페이스를 사용하고 좀 더 큰 모니터를 사용하고, 실제 레이다에 있는 전 기능을 본 시뮬레이터에 추가로 부여한다면 좀 더 효과있는 교육용 시뮬레이터가 될 수 있을 것으로 본다.

4.2.2 상대선의 움직임

본 시뮬레이터에서는 상대선을 10개로 한정하였고 상대선은 처음 갖는 속력과 침로를 계속 유지하도록 하였기 때문에 현실감이 조금 결여되어 있다. 만약 상대선의 속력과 침로를 조종자가 아닌 교육자가 옆에서 인터페이스를 사용하여 조종할 수 있고 상대선도 선체운동특성을 가지게 한다면 교육 효과가 더욱 증대될 것으로 본다.

프로그램상으로 상대선의 침로와 속력을 시뮬레이션 도중에 조종하는 것은 가능하나, 본 시뮬레이터가 개인용 컴퓨터를 사용하여 혼자서 실습할 수 있도록 하는데 목표를 두었으므로 상대선의 침로와 속력을 변경할 수 없도록 하였다.

4.2.3. 항행 환경

본 시뮬레이터는 항로 설정에 입항 환경과 협수로 환경을 주어 두가지 환경하에서 시뮬레이션할 수 있도록 하였으나 본선 설정시 조류, 바람등과 같은 기상 조건은 무시하였다. 본선의 운동방식에 조류의 영향, 바람의 영향등을 넣는다면 이러한 조류, 바람의 방향과 세기도 초기에 선택하여 시뮬

레이션할 수 있을 것이다.

4.2.4 사용 프로그래밍 언어

본 시뮬레이터는 MS-Visual Basic, MS-Windows 3.1, MS-Visual C++에 의해 제작되었으나 MS-Windows의 graphic image 처리에 시간상으로 지연이 많아서 한계가 있었다. 먼저 Visual Basic은 컨트롤할 수 있는 object의 수에 한계가 있어서 전체적인 화면구성이나 레이다 화면의 그래픽 출력에 제한이 많고, 스캐너등을 이용해 만든 레이다 화면의 밀그림을 좌우상하로 움직이게 할 수는 있으나 이를 회전시키기 위해서는 다른 부가장치가 필요하였기 때문에 밀그림으로 사용할 수 없었다. 따라서 육지물표 등은 적당한 크기의 타원을 여러개 겹쳐서 만들어야 하는 어려운 점이 있었다. 이러한 문제점들을 해결할 수 있도록 hardware와 software적으로 개선을 한다면 여러가지 시뮬레이션 환경을 만들 수 있어 좀 더 나은 교육용 레이다 시뮬레이터가 될 수 있을 것으로 본다.

제 5 장 결 론

최근, 컴퓨터 산업의 급진적인 발달에 힘입어 대용량 고속처리가 가능한 개인용 컴퓨터가 보편화됨에 따라 이러한 저가의 컴퓨터를 이용한 레이다 시뮬레이터의 개발에 대한 관심이 높아지게 되었고 80년대 말경에는 일부 선진국에서 개발을 완료하기에 이르렀다. 그러나 PC를 내장한 실물형 레이다 시뮬레이터는 기존의 실물 레이다 시뮬레이터에 비하면 가격면에서 저렴한 편이었으나 그래도 여전히 고가의 장비였고 또한 독립형이 아닌 full mission bridge 시뮬레이터의 일부분으로만 개발이 되어왔다. 따라서 본 논문을 통하여 다른 부가 장치는 사용하지 않고 오직 PC만을 이용해서, PC만 가지고 있으면 누구나 손쉽게 레이다 실습을 할 수 있는 저가의 ARPA Radar 시뮬레이터를 개발하였다.

본 시뮬레이터는 레이다의 기본 기능에 auto

plotting, guard ring, navigation line 등의 기능을 추가한 ARPA radar 시뮬레이터로서 현재 운항중인 선박들에 적재되어 있는 제품 중에 가장 최근의 제품을 모델로 하여 제작되었다. 그리고 레이다 조작 훈련뿐만 아니라 피항조선 훈련도 할 수 있도록 타와 엔진의 입력장치를 설치하여, 사용자가 레이다를 조작하면서 선박을 조종하는 것과 같은 피항조선 훈련효과도 얻을 수 있도록 설계하였다. 상대선의 발생 위치를 결정할 때에 랜덤함수를 사용하였기 때문에 시뮬레이션 시작시 마다 상대선의 발생 위치와 속력이 변하고, 따라서 사용자가 연속하여 본 시뮬레이터를 사용하더라도 계속적인 훈련 효과를 얻을 수 있다. 또한 시뮬레이션을 해 보고자 하는 다른 협수로나 항구가 있다면 그 협수로나 항구의 모양을 손쉽게 그래픽으로 구성할 수 있도록 하여 다양한 항로 환경을 만들어 사용할 수 있도록 설계하였다.

그러나 본 시뮬레이터는 외부 입력장치를 별도로 설치하지 않고 타와 엔진의 입력을 위해서 레이다 화면의 일부를 사용했기 때문에 실제보다 화면이 복잡하게 구성되어 있으며, 시뮬레이션시에 사용되는 상대선은 도중에 침로나 속도의 변화가 없고, 또한 바람이나 조류의 영향은 받지 않는 것으로 설계되었다. 그리고 시뮬레이터 제작에 사용된 PC의 그래픽 처리 능력에 한계가 있어서 육지 물표나 상대선의 화면 변화에 시간이 다소 지연되는 문제점도 있다.

따라서 타와 엔진의 입력은 외부입력장치를 별도로 만들어 입력할 수 있도록 하고, 그래픽 처리 기능이 우수한 Software나 Hardware를 사용하거나 그래픽 처리용 장치를 별도로 설치한다면 실제 레이다와 똑같은 화면을 구성할 수 있을 것이며, 교육자가 옆에서 시뮬레이션 상황에 따라 상대선의 침로와 속도를 제어할 수 있는 장치를 개발하고 본선의 운동방정식에 조류와 바람의 영향 등을 포함시킨다면 본 시뮬레이터는 교육용 레이다 시뮬레이터로서의 기능 뿐만 아니라 full mission bridge 시뮬레이터 개발의 기초장비로서 활용될 수 있을 것으로 기대되며 이는 향후 연구 과제라 할 수 있다.

참 고 문 헌

- [1] Kim, W. S., Port Design and Simulation : Validity of Microcomputer Aided Port Design Simulation and its Application, Doctoral Thesis of the University of Walse, Cardiff, June 1990, pp. 12~59.
- [2] Maritime Dynamics, Instructor Station - User Guide, The Publication of Maritime Dynamics Ltd., Llantrisant, U.K., 1990.
- [3] McCALLUM, I. R., A New Approach to Manoeuvring Ship Simulation, Doctoral Thesis of The City University, London, 1976.
- [4] ibid 1, pp. 68~69
- [5] ibid 1, p. 73
- [6] ibid 3, pp. 67~69
- [7] 임상전, 基本造船學, 大韓教科書株式會社, 서울, 1986, p. 603.
- [8] Ibid 3, p. 71.
- [9] Ibid 3, pp. 70~71.
- [10] 최인현, 그래픽 프로그래밍, 大林圖書出版株式會社, 서울, 1991, pp. 321~326.
- [11] 山村三朗, “個人用計算機を用いた簡易レータシミュレータ”, 神戸商船大學紀要 第2類 第29號, 昭和 56年, pp. 37~42.
- [12] 廣田 實 外 4人, “確率を用いたレータシミュレータの試作”, 神戸商船大學紀要 第2類 第29號, 昭和 56年, pp. 21~35.
- [13] MARSIM, MARSIM87, PROCEEDINGS, Fourth International Conference on Marine Simulation, 1987, pp. 5~17.
- [14] Shuko Moriya · Masayoshi Hirano, “Development of a Desktop Simulator and its Application of Simulator Link”, MARSIM & ICSM 90, Tokyo, 1990, pp. 169~174.