

PC를 이용한 선박 조종 시뮬레이터의 개발에 관한 연구

손 경호*, 이 성욱**, 배 정철**

〈 목 차 〉

- | | |
|------------------------|------------------------------|
| 1. 서 언 | 4. 시뮬레이터의 개발 환경 및 부산항에의 적용 예 |
| 2. 시뮬레이션 수학 모델 | 5. 결론 |
| 3. 시계 영상 모의를 위한 그래픽 기법 | 참고문헌 |

1. 서 언

항만내에서 선박을 안전하게 운항하기 위해서는, 선박 고유의 조종 성능 뿐만 아니라 선박을 조종하는 인간의 능력 즉, 해기 기술 또한 중요한 요소로 작용하게 된다. 아울러 항만의 환경 특성도 운항 성능에 큰 영향을 미친다.

따라서 항만내에서의 종합적 운항 안전성 평가의 필요성이 대두되며, 이를 실현하기 위해서는 선박과 인간과 항만 이들 3요소 사이의 적응 관계를 조사해야 한다. 이러한 조사를 공학에서는 인간-기계계(man-machine system) 해석이라고 한다. 인간-기계계 해석의 도구로서 선박 조종 시뮬레이터가 요구되고 있다[1]. 구체적으로, 선박 조종 시뮬레이터는 항만역에서 선원의 반복 훈련이나, 출입항 가능한 선박의 크기의 결정, 또는 특정 규모의 선박이 출입항 가능한 항만 배치의 결정 등을 효과적으로 수행할 수 있도록 한다.

한편, 실제 선박의 브릿지 형태를 취하고 있는 대형 시뮬레이터는 도입 가격이 고가이고 유지 운영비도 고비용이 요구된다. 아울러 초등 해기 교육의 목적에 비해 장비의 규모가 너무 크다. 이러한 점을 고려하여, 대형 시뮬레이터에 비해 기능이나 성능이 떨어지지 않으면서 PC상에서 구현될 수 있는 데스크

탑형 시뮬레이터가 개발되게 되었다[2]. 데스크탑형 시뮬레이터는 해기 교육 기관 또는 실선에 탑재 되어 교육 또는 훈련용으로 이용될 수 있을 뿐만 아니라, 특정 항만 또는 수로의 안전성 평가 등에도 이용될 수 있다.

본 연구에서는 항만내에서 실제의 출입항 조종 현상을 영상 모의할 수 있는, 데스크탑형 시뮬레이터의 개발에 관해서 논의한다. 구체적으로, 항만내에서의 저속시 조종 운동을 구현할 수 있는 수학 모델을 이용하여[4], 특정 항만내에서의 출입항 조종의 실시간 시뮬레이션 구현을 위한 3차원 그래픽 기법을 소개하고 이 기법을 중형 컨테이너 운반선의 부산항 출입항 조종에 적용시킨 예에 관해서 소개한다.

2. 시뮬레이션 수학 모델

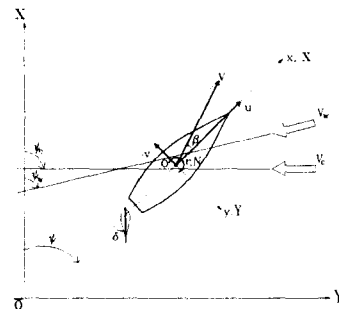


Fig. 1 Coordinate system

* 한국해양대학교, ** 삼양무선공업(주)

일반적으로 선박의 조종 운동 방정식은 전후동요(surge), 좌우동요(sway), 선수동요(yaw)의 복합 운동으로 표현할 수 있다. Fig. 1과 같이 선체중양(midship)에 좌표 원점을 둔 동좌표계를 $O-xyz$ 라 할 때, 조종 운동 방정식은 다음과 같이 표시된다.

$$\begin{aligned} m(\dot{u} - vr - x_G r^2) &= X \\ m(\dot{v} + ur + x_G \dot{r}) &= Y \quad \dots\dots(1) \\ I_{zz} \dot{r} + mx_G(\dot{v} + ur) &= N \end{aligned}$$

여기서 m 은 선체의 질량이고, I_{zz} 는 z 축 주위의 관성모멘트이다. u, v, r 은 각각 선체중양에서의 x 축 방향 속도 성분, y 축 방향 속도 성분, z 축 주위의 각속도 성분이다. $\dot{u}, \dot{v}, \dot{r}$ 는 각각 u, v, r 의 시간에 대한 1차 도함수이고, x_G 는 무게중심의 x 좌표이다. X, Y, N 은 각각 x, y 축 방향 힘과, z 축 주위의 모우멘트이다.

식(1)의 좌변에 부가질량력을 고려하고, 우변에는 힘과 모우멘트를 원인 구성별로 나타내면 다음과 같다.

$$\begin{aligned} (m + m_x)\dot{u} - (m + m_y)vr - (mx_G + m_y a)r^2 &= X_H + X_P + X_R + X_T + X_W \\ (m + m_y)\dot{v} + (m + m_x)ur + (mx_G + m_y a)\dot{r} &= Y_H + Y_P + Y_R + Y_T + Y_W + Y_D \\ (I_{zz} + J_{zz})\dot{r} + (mx_G + m_y a)\dot{v} + mx_G ur &= N_H + N_P + N_R + N_T + N_W \quad \dots\dots(2) \end{aligned}$$

여기서 첨자 H, P, R, T, W, D 는 각각 주선체, 프로펠러, 타, 예선, 바람 및 파도에 기인하는 유체력 또는 외력 성분을 나타낸다. 파도에 기인하는 외력 성분은 파도

의 표류력을 의미하며 이것은 y 방향 성분만을 고려한다. 그리고 m_x, m_y 는 x, y 축 방향의 부가질량, J_{zz} 는 z 축 주위의 부가관성모멘트, a 는 m_y 중심의 x 좌표를 나타낸다.

한편, 조류에 기인하는 유체력은 별도의 항으로써 고려할 필요없이, 식(2)의 유체력 계산에서 운동 변수로서 상대유속 즉, 대수속도를 사용함으로써 조류의 영향이 포함될 수 있다. 조류의 유속 V_c 와 유향각 ψ_c 를 Fig. 1과 같이 정의할 때 대수속도와 대지속도의 관계는 다음과 같다.

$$\begin{aligned} u &= u^* + V_c \cos(\psi_c - \psi) \\ v &= v^* + V_c \sin(\psi_c - \psi) \quad \dots\dots(3) \\ \dot{u} &= \dot{u}^* + V_c r \sin(\psi_c - \psi) \\ \dot{v} &= \dot{v}^* - V_c r \cos(\psi_c - \psi) \end{aligned}$$

여기서 $u^*, v^*, \dot{u}^*, \dot{v}^*$ 는 대지속도와 가속도를 u, v, \dot{u}, \dot{v} 는 대수속도와 가속도를 나타낸다(이하 마찬가지로). 식(3)의 관계를 식(2)에 대입하면 다음과 같다.

$$\begin{aligned} (m + m_x)\dot{u}^* &= (m + m_y)vr \\ &+ (mx_G + m_y a)r^2 \\ &- (m + m_x)V_c r \sin(\psi_c - \psi) \\ &+ X_H + X_P + X_R + X_T + X_W \\ (m + m_y)\dot{v}^* + (mx_G + m_y a)\dot{r} &= -(m + m_x)ur \\ &+ (m + m_y)V_c r \cos(\psi_c - \psi) \\ &+ Y_H + Y_P + Y_R \end{aligned}$$

$$\begin{aligned}
& + Y_T + Y_W + Y_D \\
(I_{zz} + J_{zz})\dot{r} + (mx_G + m_y\alpha)\dot{v}^* \\
& = -mx_G ur + (mx_G \\
& \quad + m_y\alpha)V_c r \cos(\psi_c - \psi) \\
& \quad + N_H + N_P + N_R + N_T + N_W \\
\cdots\cdots\cdots(4)
\end{aligned}$$

식(4)로부터 축차근사적분법을 이용하여 대지속도 u^*, v^* 와 회두각속도 r 을 구한 후에, 다시 적분을 수행함으로써 선위(船位)와 회두각을 계산할 수 있다. 식(4) 우변의 유체력과 외력 성분의 추정법에 관해서는 문헌[3][4]에 자세히 언급되고 있다.

3. 시계 영상 모의를 위한 그래픽 기법

선박이 운동함에 따라서 시시각각 변화하는 시계(視界)의 영상 모의를 위한 그래픽 기법으로서 3차원 애니메이션을 들 수 있는데, 이에 관해서 언급한다.

3.1 좌표계

Fig. 2에서 $o-xyz$ 는 국부 좌표계로서 개개의 물체를 정의하기 위해 사용되고, $O-XYZ$ 는 실세계 좌표계로서 3차원 장면을 구성하는 데 사용되어진다.

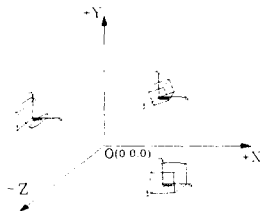


Fig. 2 Local coordinate and world coordinate systems

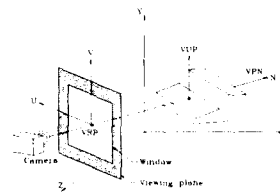


Fig. 3 Viewing reference coordinate system

그리고 Fig. 3에서 $VRP-UVN$ 은 뷰잉 기준 좌표계를 나타내는데, 뷰잉 기준 좌표계를 정의하는 데는 세 가지 매개 변수가 필요하다. 첫번째는 실세계 좌표계에서 카메라가 바라보는 방향의 한 점으로 정의되는 뷰 기준점(VRP : view reference point)이다. 두번째는 뷰 평면에 수직인 방향의 뷰 평면 법선벡터(VPN : view plane normal)이다. 세번째는 카메라의 상(up) 방향(V축)이다. 그리고 뷰잉 기준 좌표계의 U축은 뷰평면 법선벡터와 뷰 상향벡터와의 외적으로서 정의된다. 관찰자가 지정한 대로 물체의 모습을 생성하기 위해서는 실세계 좌표계에서 정의된 물체의 좌표값들이 뷰잉 기준 좌표계로 변환되어야 한다. 실세계 좌표계에서 정의된 물체를 뷰잉 기준 좌표계로 변환하는 과정은 다음과 같다(Fig. 3 참조).

- (i) 뷰 기준점을 실세계 좌표계의 원점으로 이동한다.
- (ii) \vec{N} 이 XZ 평면상에 위치하도록 X 축에 대하여 회전시킨다.
- (iii) \vec{N} 이 Z 축과 일치하도록 Y 축에 대하여 회전시킨다.
- (iv) 모든 축이 일치하도록 Z 축에 대하여 회전시킨다.

3.2 3차원 변환

3차원 공간에서의 변환은 다음과 같이 수행된다.

(1) 크기변환

점 (x, y, z) 는 크기변환에 의해 (x', y', z')

로 다음과 같이 크기가 조절된다.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(5)$$

여기서 s_x, s_y, s_z 는 크기변환율이다.

(2) 이동변환

점 (x, y, z) 를 (dx, dy, dz) 만큼 이동시키면 새로운 점 (x', y', z') 는 다음과 같다.

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(6)$$

(3) 회전변환

점 (x, y, z) 를 임의 축 주위로 θ 만큼 회전시키면 새로운 점 (x', y', z') 는 각각 다음과 같다.

$$[x' \ y' \ z' \ 1]_{z'}^{\theta} = [x \ y \ z \ 1] \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[x' \ y' \ z' \ 1]_{y'}^{\theta} = [x \ y \ z \ 1] \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[x' \ y' \ z' \ 1]_{x'}^{\theta} = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dots\dots\dots(7)$$

여기서 아래 첨자 x, y, z 는 회전축을 나타내고, 위 첨자 θ 는 회전각을 나타낸다.

3.3 투영(projection)

3차원 장면에서의 모든 물체를 구성하는 점들이 뷰잉 기준 좌표계로 변환되면, 3차원 좌표를 2차원 좌표로 변환해야 하는데 이를 투영(projection)이라 한다. 본 연구에서는 원근 투영법을 이용하며(Fig. 4 참조), 원근 투영법에 관한 수식 변환은 다음과 같다.

$$\begin{aligned} x' &= d \times x/z \\ y' &= d \times y/z \end{aligned} \dots\dots\dots(8)$$

여기서 d 는 카메라가 위치한 관측점과 투영면과의 거리이다.

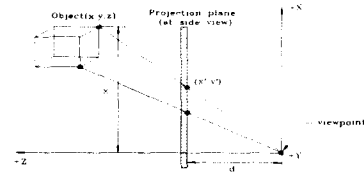


Fig. 4 Perspective projection mechanism

3.4 3차원 절단(clipping)

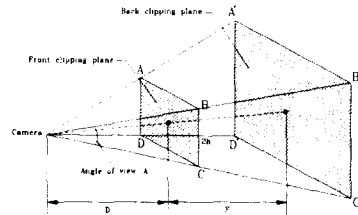


Fig. 5 Viewing frustum

Fig. 5는 뷰잉 프러스텀(frustum)을 나타내고 있는데 여기서 사각형 $ABCD$ 는 전면 절단면(front clipping plane), 사각형 $A'B'C'D'$ 는 후면 절단면(back clipping plane)을 나타내고 있다. 일반적으로 절단면은 원근 투영 전후에 진행되며, 이것을 각각 오브젝트 스페이스 절단과 이미지 스페이스 절단이라 한다 [5].

① 오브젝트 스페이스 절단

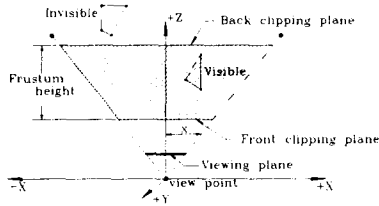


Fig. 6 Clipping layout based on z-axis

오브젝트 스페이스 절단은 기본적으로 인 지적 부피 외부에 놓여있는 다각형과 오브젝트를 제거하는 방법으로서 Fig. 6으로부터 삼각형의 님은꼴의 성질을 이용하여 다음과 같이 쓸 수 있다.

$$\begin{aligned}
 -\frac{W \times z}{2 \times d} &\leq x \leq \frac{W \times z}{2 \times d} \\
 -\frac{W \times z}{2 \times d} &\leq y \leq \frac{W \times z}{2 \times d} \dots\dots\dots(9) \\
 d_b &\leq z \leq d_f
 \end{aligned}$$

여기서 W 는 화면의 폭, d 는 관측점과 투영면 사이의 거리, z 는 물체 위치의 z 좌표, d_b , d_f 는 각각 관측점에서 전후 절단면 사이의 거리를 의미한다.

② 이미지 스페이스 절단

오브젝트 스페이스 절단이 수학적으로는 합리적이지만 3차원 시계 장면의 재현이라는 면에서 비효율적이기 때문에, 본 연구에서는 시계 장면 재현의 초기 단계에서만 오브젝트 스페이스 절단을 수행한 후 최종적인 시계 장면 구현에서 이미지 스페이스 절단을 수행한다. Fig. 7은 이미지 스페이스 절단의 2차원 래스터라이제이션을 설명하고 있다.

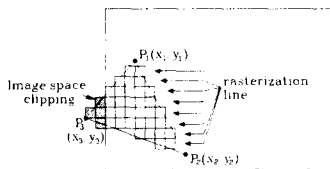


Fig. 7 2-Dimension rasterization of image space clipping

3.5 은면제거 및 렌더링

은면제거란 궁극적으로 올바른 순서대로 3차원 물체를 그리는 것으로서 2가지의 명확한 단계가 있다. 첫번째 단계는 후면발체(back-face cutting)라 부르며, 두번째 단계는 후면발체를 수행한 후 일단 보이는 다각형들의 리스트를 결정하고, 렌더링을 시작하는 것이다. 각각의 단계에 대한 설명은 참고문헌 [6]에 상세히 설명이 되고 있다.

3.6 셰이딩(shading)

셰이딩이란 물체를 현실감 있게 표현하기 위해 물체 표면에 음영을 주는 것을 말하는 데, 본 연구에서는 플랫 셰이딩 기법을 이용한다[6]. 플랫 셰이딩은 어떤 임의의 지점에서의 광원과 셰이딩을 하려고 하는 면의 수직 벡터 사이의 내적을 계산하여 광원의 입사각을 구한 후 64 또는 256가지의 음영 또는 기타 여러 개수의 음영으로 스케일하여 표면에 적용하는 방법이다.

4. 시뮬레이터의 개발 환경 및 부산향에의 적용 예

4.1 하드웨어와 소프트웨어

본 연구에서 사용된 하드웨어 사양은 Table. 1과 같으며, OS(operating system)으로는 Windows 95를 사용하였다. 그리고 최종적인 시뮬레이터 개발에 사용되어진 컴파일러는 MicroSoft Visual C++4.0이다.

Table. 1 Specification of computer hardware

CPU	Intel Pentium 60MHz
Memory	24MB
HDD	2.5GB
Video Card	Super VGA

한편, 스캐너로 생성한 데이터를 Auto CAD와 3D Studio max, Adobe Photoshop

등의 그래픽 툴을 사용하여 3차원 물체를 생성하였으며, 그래픽 라이브러리로는 MS의 DirectX를 사용하였다. 이것은 비디오 가속 기능을 하드웨어 독립적으로 일관성있게 사용할 수 있게 한다.

4.2 시뮬레이터의 구성 및 기능

출입항 조종의 실시간 시계(視界) 영상 모의를 위한 데스크탑형 시뮬레이터의 초기 화면 및 정보 표시를 위한 모든 다이얼로그 박스는 Fig. 8과 같다. 본선과 예선의 조종 제어, 초기 조건 및 외란 조건 등은 Fig. 8에서

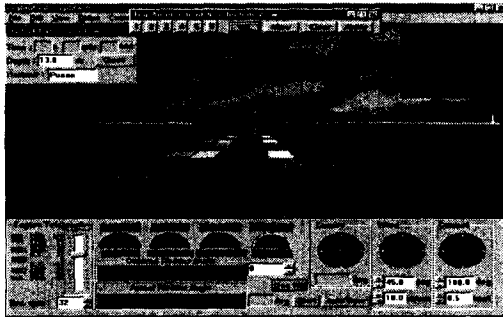


Fig. 8 Initial view of simulator

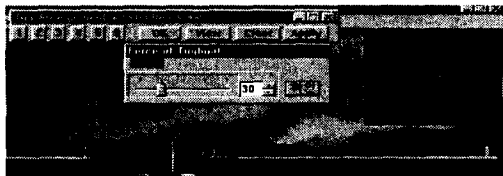


Fig. 9 Bird's eye view and tug controller

보는 바와 같이 하단 부분의 계기판 모양에 마우스와 키보드를 통하여 입력되며, 실시간으로 계산된 선체 운동은 시시각각 화면의 하단 부분에 표시된다. 즉, 선수방위, 회두각 속도, 전진속도, 횡이동속도, 프로펠러 회전수, 타각 등이 화면 하단의 계기판 모양에 정보가 표시되며, 실제의 시계는 Perspective view방식으로 1초, 2초, 5초, 10초 20초 중 선택된 시간 간격으로 표시된다. Fig. 8에서 우측 상단의 사각형은 선박의 충돌이나 접

안시 부두에의 접근, 예선의 사용 등을 보여주기 위해 Bird's eye view방식으로 표현한 것으로 Fig. 9에 확대되어 나타나 있다. 예선의 경우는 Fig. 8, 9에서 보는 바와 같이 화면 우측 상단에 총 6 척중 원하는 것만을 선택하여 볼 수 있다. 예선의 사용은 Fig. 9에서 보는 바와 같이 예선을 나타내는 삼각형을 마우스로 클릭함으로써 선택하고, 사용하고 싶은 예선의 출력은 10, 20, 30, 40 ton의 4 단계 중 하나를 선택하여 사용할 수 있다.

시뮬레이션 수행중 타 선박과의 충돌 회피 훈련을 위하여 상대 선박의 움직임을 고려하게 되는데, 이때 상대 선박의 움직임은 Fuzzy 또는 AI(Artificial Intelligence)이론을 적용하여 제어할 수 있으나[7], 본 연구에서는 상대 선박의 움직임에 대한 고려를 단순화 하기 위해 일정한 방향으로 일정하게 움직이는 것으로 한다. Fig. 10은 시뮬레이션 수행 도중 나타나는 선박의 모습을 나타낸다. 한편, 데스크탑형 조종 시뮬레이터의 3차원 시계는 수평 방향으로 좌현 45°에서 우현 45°까지 15°간격으로 변경될 수 있으며, 수직 방향으로로는 하향 20°에서 상향 20°까지 10°간격으로 변경될 수 있다. Fig. 11은 좌현 45°, 하향 20°일 때의 장면을 나타낸다. Fig. 12는 정보 입력 및 표시를 위한 계기판의 모습을 나타내고, Table 2는 정보 입력을 위한 버튼 및 각종 메뉴의 기능을 나타낸다.



Fig. 10 Perspective view of movement of other target ship

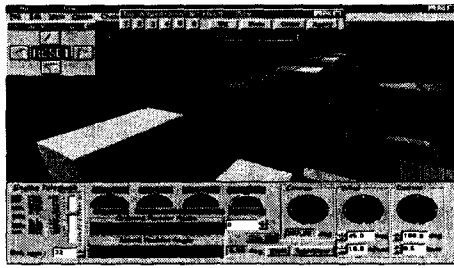


Fig. 11 Perspective view seen 45° left-wards and 20° down-wards

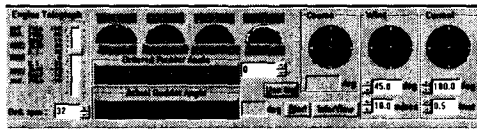


Fig. 12 Simulator console

Table 2. Setup menu and performance of simulator

메뉴	기능
Report	시뮬레이션 결과 그래프 출력
Real-time/Fast-time	시뮬레이션 수행 속도 선택
Change_View_Angle!	뷰잉각 변환(상, 하, 좌, 우현)
Light_Scene/Night_Scene	시뮬레이션 수행의 환경 설정
Use tug	예선 사용
InforView	시뮬레이션 수행 정보 표시
Ord. rpm	명령회전수 변경

4.3 부산항 출입항 조종에의 적용

위에서 언급한 3차원 그래픽 기법을 이용하여 중형 컨테이너 운반선의 부산항 출입항 조종에 적용한 예를 소개한다.

먼저 중형 컨테이너 운반선의 주요목은 Table 3과 같고, 조종운동 시뮬레이션 수학적 모델에서 요구되는, 이 선박의 유체동역학적 계수는 문헌[3]에서 상세히 언급되고 있다.

시뮬레이션 시나리오는 다음과 같다. 부산항 외항 방파제 부근에서 도선사가 승선하여 본선을 부산항 제 5부두(자성대 컨테이너 부두)에 접안시키는 과정을 시뮬레이션 한다.

Table 3 Principal particulars of container ship

HULL		
Length over all	Lo_A (m)	188.0
Length B.P.	L (m)	175.0
Breadth(moulded)	B (m)	25.4
Draft(moulded)	d (m)	8.50
Trim	τ (m)	1.0
Block coefficient	C_B	0.559
RUDDER		
Height	H_R (m)	7.70
Area ratio	A_R/Ld	1/45.8
Aspect ratio	λ	1.827
PROPELLER		
Diameter	D (m)	6.50
Pitch ratio	P/D	1.055
Expanded area ratio		0.730
No. of blades		5

부산 외항 방파제 부근에서 선박의 초기 속도는 6.38 kt(dead slow ahead)이며, 외력으로서 바람 $V_w = 10$ m/sec, $\psi_w = 45^\circ$, 조류 $V_c = 0.5$ kt, $\psi_c = 180^\circ$ 를 상정하였다. 부산항 외항 방파제 부근에서의 초기 화면은 Fig. 8, 항만내의 수로 항진중 타선과의 만남은 Fig. 10에 보인다. 그리고 접안 예정인 부두 가까이에서는 예선을 이용하여 선박을 접안시킨다. Fig. 13은 최종접안시의 화면을 나타내고, 시뮬레이션 전 과정에서의 선체 운동 변수들의 시계열 값은 시뮬레이션이 완료된 후에 Fig. 14와 같이 출력시킬 수 있다. 그리고 Fig. 15는 시뮬레이션 전 과정을 $X-Y$ 플로터를 이용하여 Bird's-eye-view 방식으로 나타낸 것이다.

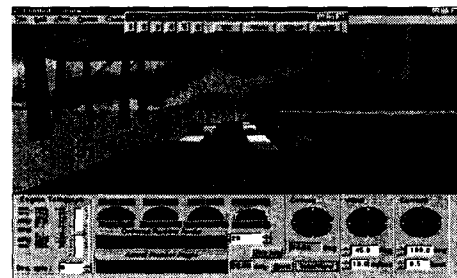


Fig. 13 Perspective view of ship moored at pier

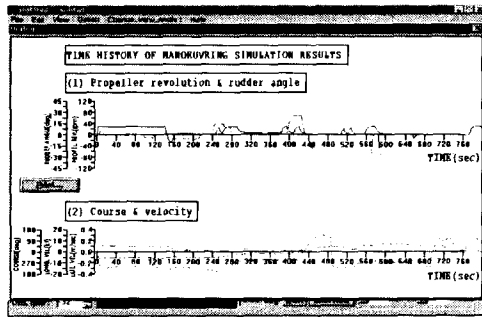


Fig. 14 Time histories of ship motion parameters in harbour manoeuvre

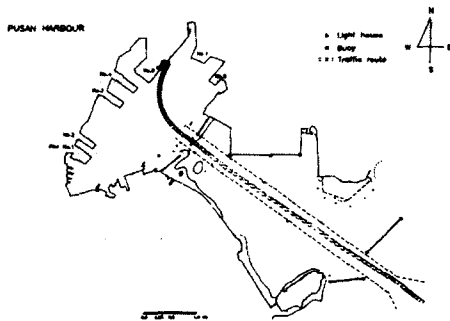


Fig. 15 X-Y Plotter display of berthing manoeuvre

5. 결론

본 연구에서는 항만내에서 실제의 출입항 조종 현상을 시계 영상 모의할 수 있는, 데스크탑형 시뮬레이터의 개발에 관해서 논의 하였다.

본 연구에서 개발된 시뮬레이터는, 선박의 조종운동의 영상 모의를 위하여 3차원 그래픽 기법을 일반 개인용 컴퓨터에 적용한 것으로서, 실용적이고도 범용성있는 시스템이라 생각된다. 또한 본 시뮬레이터는 특정 항만내에서의 선박의 반복 훈련용으로서, 그리고 항만 수로 설계시의 안전성 평가용으로서 유용하게 활용될 수 있을 것으로서 생각된다.

앞으로 본 시뮬레이터의 성능 향상을 위해서는 타선의 움직임을 보다 실제에 가깝게 구축하고, 시뮬레이터 콘솔을 별도의 하드웨어 제어기로 대체할 필요가 있을 것으로 생각한다. 아울러 모의 시계 장치로서 다수개의 모니터를 사용함으로써 수평시야각을 넓힐 수 있을 것이다. 또한 여러 대의 시뮬레이터를 접속 연결함으로써 복수의 선박간의 상호 운항 훈련용으로 발전될 수도 있을 것이다.

참 고 문 헌

- [1] Tsumura, T., et al., "Training Simulator for Ship Handling", JRC Technical Report No. 3, 1967.
- [2] 日本造船學會 新製品・新技術の紹介, "Desktop Ship Handling Simulator 「Harbour Master」", 日本造船學會誌 第716号, 1989.
- [3] 손 경호, et al., "출입항 조종의 실시간 시뮬레이션 기법 및 부산항에의 적용 예에 관한 연구", 해양안전학회지, 제3권 제2호, 1997.
- [4] 손 경호, "Hydrodynamic Force and Manoeuvring Characteristics of Ship at Low Advance Speed", 대한조선학회논문집, 제29권, 제3호, 1992.
- [5] Anand, V. B., "Computer Graphics & Geometric Modeling for Engineers", John Wiley & Sons Inc., 1996.
- [6] LaMothe, A., "Black Art of 3D Game Programming", The Waite Group Inc., 1995.
- [7] Kose, K. et al., "On An Intelligent Harbour Manoeuvring Simulator and Its Applications", Proceeding of MARSIM & ICSM 90, Tokyo, Japan, 1990.