공학석사 학위논문

어선 자동운항용 제어알고리즘 및 시스템소프트웨어 개발

DEVELOPMENT OF CONTROL ALGORITHM AND SYSTEM SOFTWARE FOR AUTOPILOT SYSTEM OF FISHING BOAT

지도교수 오 세 준

2001년 2월

한 국 해 양 대 학 교 대 학 원

운항시스템공학과 선박자동화시스템 전공

천 승 호

이 논문을 천승호의 공학석사 학위논문으로 인준함

주 심 공학박사 이 상 태 인

부 심 공학박사 소 명 옥 인

부 심 공학박사 오 세 준 인

2001년 2월

한국해양대학교 대학원

운항시스템공학과 선박자동화시스템 전공

천 승 호

목 차

Development of control algorithm and system software for the autopilot system of fishing boat

Seung-Ho Cheon

Department of Marine Engineering, Graduate School
Korea Maritime University

Abstract

A development of the system software for a small class ship autopilot is presented. A dynamics modeling for small class ship is firstly described. A control system with several operating conditions for the autopilot system is constructed. This is based on the conventional PID control technique. The system is composed of power amplifier, gear unit, and ship dynamics. The ship dynamics is realized by microprocessor. A system software is built up using 80C196KC microprocessor on an one chip format. The autopilot system proposed is simpler to operate comparing to others due to the simple design of the operating panel and economical due to the design of the one chip microprocessor. This also has varieties of expansion to a large scale system because of using high level

microprocessor. The experimental results by using a model plant for the autopilot system as well as computor simulation results are given to illustrate the useful ability of the proposed system.

제 1 장 서 론

1.1 배경 및 필요성

현재 선박의 자동조타에 관한 기초 연구 및 기술 개발은 주로 네덜란드, 노르웨이, 일본 등 선진국을 중심으로 활발히 연구되어 오고 있다. 최근에는 대표적인 현대 제어방법인 적응제어 등을 응용하여 선박의 비선형성을 보상하는 방법으로 선박의 항로 유지 제어, 롤-타각 안정화 제어, 선박의 위치 제어, 선박 자동 접안 등에 관한 연구를 수행하고 있으며, 실제의 선박을 대상으로 그 응용 연구가 활발히 진행중에 있다.

자동 조타의 목표는 목표 선수각과 실제 선수각 사이의 오차를 감소시키기 위해서 타를 자동적으로 조작되게 하는 것을 말하며 선박의 항로 유지제어, 항로 변경제어 등에 이용되는 것으로 자동 조타기가 있다. 자동조타기는 선박에 가해지는 많은 외란 등에도 불구하고 원하는 방향으로 오차없이 항해하기 위해, 또한 불필요한 조타기 작동으로 인한 에너지 손실을 막기 위해 적절히 제어되어야 한다.

국내에서 사용되고 있는 소형 선박용 자동조타기는 외국기술 또는 제품에 의존하고 있으며, 그 가격이 고가일 뿐만아니라, 고장이 발생할 경우 A/S 문제가 크다. 국내 어선 자동 조타기 분야에 현재 상용되고 있는 FA-45 전자동 조타시스템 등은 MC 6802를 이용한 어셈블러로 구성되어져 있으며, 프로그램은 자체 기술에 의해 개발된 것이 아니고 대부분 수입에 의존하고 있는 실정이다. 그리고 MC 6802는 8bit data 처리 기능을 가지고 있으므로, 복잡한 연산을 할 수가 없다. 그러므로 이것은 복잡한 제어 알고리즘을 사용하지 못하고, 간단한 P게인을 사용하고 있으므로 제어 성능이 떨어진다. 따라서 현재 국내 소형 선박의 보유 상황 및 관련 조선 산업 분야의 현황을 감안하면 이 분야에 대한 국산화 기술 개발이 시급한 실정이다.

1.2 연구 목적 및 연구 내용

본 연구에서는 어선 자동운항용 제어알고리즘과 개발된 알고리즘을 응용할 수 있는 시스템 소프트웨어를 개발하고 이를 인터페이싱하여 실제 어선에 장착 가능한 자동운항 제어시스템(Autopilot System)을 개발한다. 이를 수행하기 위해 MC 6802 보다 상위 프로세서인 Intel사의 80C196KC를 사용한 자체 운용 프로그램 개발과 이에 적합한 제어알고리즘을 개발하여 기존의 제품보다 성능은 우수하고 국제적으로 가격 경쟁력이 있는 자동운항 제어시스템을 구축한다. 자체 시스템 소프트웨어를 개발함으로써 종래에는 사용대상과 목적에 따라 자주 프로그램을 하던 불편을 해소할 수 있고 제어 구조를 크게 변경하지 않고도 소형 어선 뿐만 아니라 특수선박, 대형 선박에 까지 용이하게 그 사용 범위를 확대시키는 것이 가능하다.

본 연구에서는 수치 시뮬레이션 뿐만 아니라 실선에서 주로 사용되고 있는 자동조타기와 동일한 형태의 유압 엑츄에이터 방식의 시뮬레이터를 이용하여 개발한 자동조타시스템을 시험한다. 실선에서 발생할 수 있는 상황과 유사한 제어 실험을 위하여 선체 운동 부분은 CPU로서 80C196KC를 사용하여 구현하였다. 수치 시뮬레이션과 실험을 통해 본 연구에서 개발한 자동조타시스템의 유효성을 확인한다.

제 2 장 제어알고리즘 및 소프트웨어

2.1 전자동 조타기

자동운항 제어시스템은 목표 선수각과 실제 선수각 사이에 오차를 감소시키기 위해서 조타기를 자동적으로 작동시킨다. 일반적으로 전자동 조타기 제어시스템은 간단한 PID 제어에 바탕을 두고 있다. PID형의 전자동 조타 제어시스템의 목표 작동 수행을 하기 위해서는 컨트롤 파라메타들은 선박의 다이나믹스(Dynamics)와 환경적인 외란의 변동에 따라서 알맞게 적용되어져야 한다. 여기서, 선박의 다이나믹스는 작동 상태 때문에 변하고, 주요 환경적 외란에는 바람, 파도, 조류 등이 있는데, 이것들은 날씨와 해상의 상태에 따라서 변화한다. 그러므로, 시스템의 제어파라메타들을 매번 상황에 따라 적당하게 적용시키는 것은 실제로 번거롭고 어려운 작업이다. 그러므로 보통 항해 중에는 대략적이거나 일정한 값으로 고정시켜 적용한다. 이것은 과도하게 타를 움직임으로써 추진 에너지 손실과 같은 좋지 못한 조타 특성을 유발시킬 수도 있다.

전자동 조타장치는 준비 단계에서 현재의 선수각이 표시되고 필요할 때 콤파스의 편차를 수정해야 한다. 또 U-회전 기능 등 여러 가지 설정값들이 표시되고 목표 방위각을 설정해야 한다. 조타 단계에서는 목적 방위에 따라 조정되는 현재의 선수각이 표시되고 목표치에 맞게 선박을 좌우로 조타한다. 현재 가장 일반적으로 상용화되어 있는 전자동 조타장치 중의 하나인 FA-45 전자동 조타기의 개략도를 Fig. 1에 보인다.

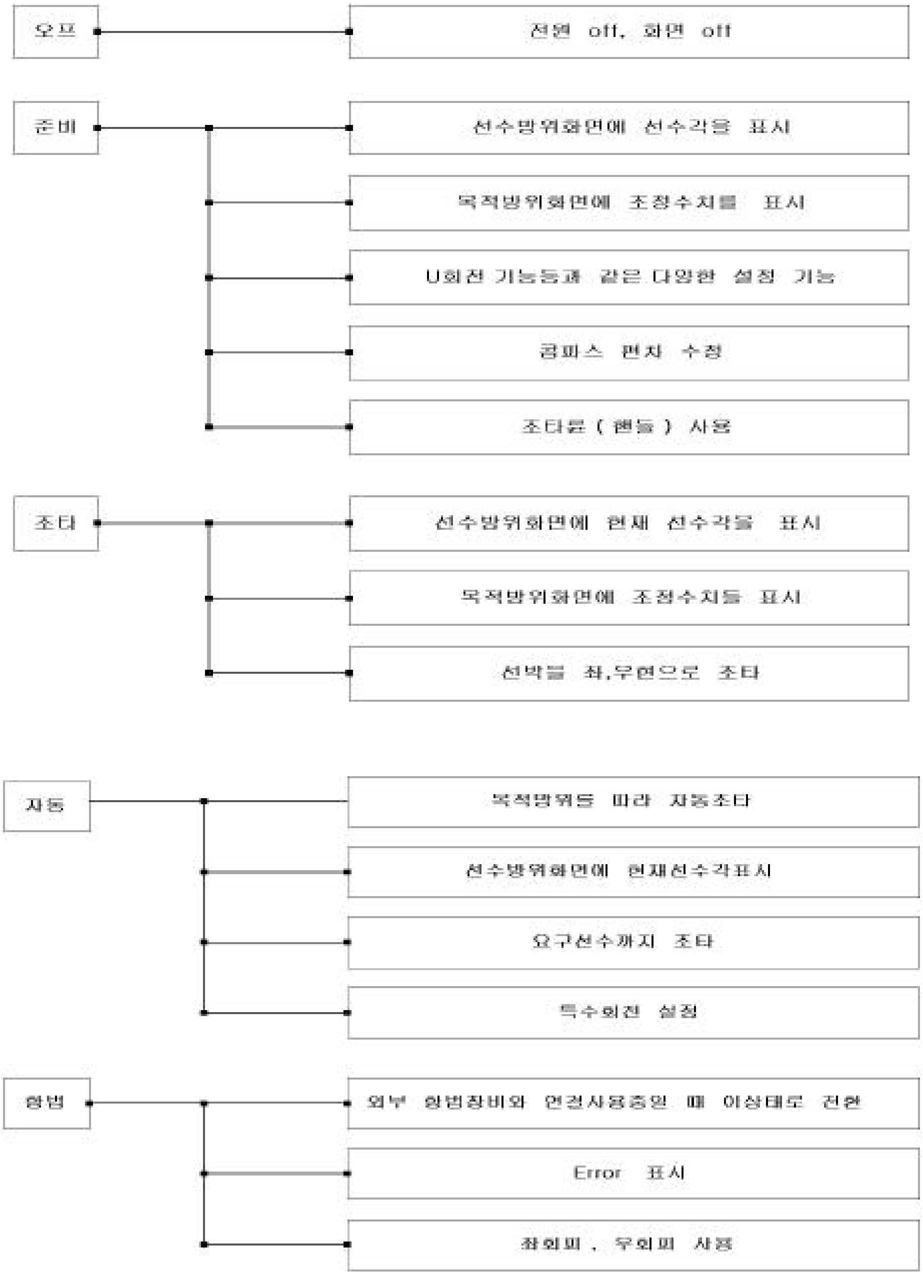


Fig.1 FA-45 Autopilot system

2.2 선체 모델링

Fig. 2와 같이 나타낸 좌표계에 대한 선체의 운동을 나타내는 Newton의 방정식은

$$X_0 = m x''_{0G}$$
$$Y_0 = m y''_{0G} \qquad (2.1)$$
$$N = I_z \psi''$$

단, X_0, Y_0 : 각각 x_0과 y_0 방향의 합력

m : 선박의 질량

N : 선박의 무게중심을 지나고 z_0축에 평행한 축에 관한 전체 모멘트

I_z : 위에서 말한 축에 관한 선박의 관성모멘트

ψ : 수평면 위에서 연직면 $x_0 z_0$ 으로부터 선박의 x 축까지 잰 선수 진행각

지구에 고정된 좌표계에 관한 식 (2.1)을 움직이는 배에 고정된 좌표계에 관한 식으로 변환하기 위하여 x 및 y 방향의 힘 X 와 Y를 x_0 과 y_0 방향의 힘 X_0 과 Y_0 항으로 나타내면,

$$X = X_0 \cos\psi + Y_0 \sin\psi$$
$$Y = Y_0 \cos\psi - X_0 \sin\psi \qquad (2.2)$$

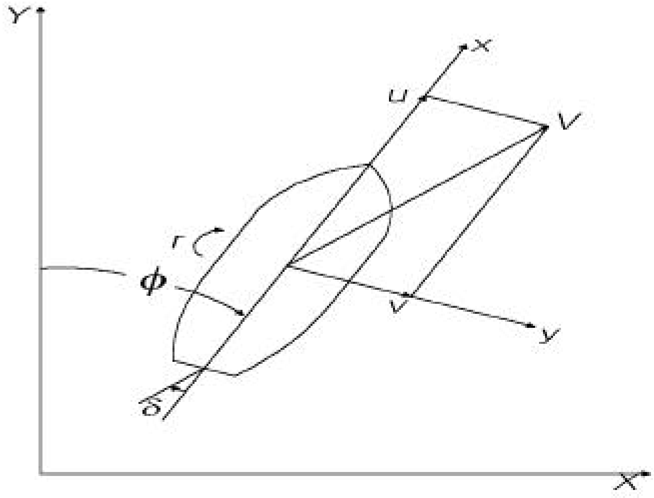


Fig. 2 Coordinate of Ship Dynamics

이것은 다음과 같이 표시되고

$$x'_{0G} = u\cos\psi - v\sin\psi$$

$$y'_{0G} = u\sin\psi + v\cos\psi \qquad (2.3)$$

식 (2.3)를 시간에 관해 한번 더 미분하면

$$x''_{0G} = u'\cos\psi - v'\sin\psi - (u\sin\psi + v\cos\psi)\psi'$$

$$y''_{0G} = u'\sin\psi + v'\cos\psi + (u\cos\psi - v\sin\psi)\psi' \qquad (2.4)$$

식 (2.4)을 식 (2.1)에 대입하여 얻은 X_0 와 Y_0 의 값들을 식 (2.2)에 대입하면,

$$X = m(u' - v\psi')$$

$$Y = m(v' + u\psi')$$

를 얻는다.

식 (2.1) (2.4)로 부터 로울링, 피칭, 및 하이빙을 무시했을 경우의 수평면 안에서의 운동방정식은 다음과 같이 표현된다. 즉,

$$X = m(u' - v\psi')$$

$$Y = m(v' + u\psi') \qquad (2.5)$$

$$N = I_z\psi''$$

X, Y 및 N은 각각 Surge, Sway 및 Yaw방정식을 나타내며 식 (2.5)는 좌표축의 원점 O가 선박의 중심 위치에 있는 경우에 대하여 유도된 식이다. 선박의 무게중심 G로부터 R_G 만큼 떨어진 곳에 좌표의 원점 O를 잡았다고 하면, 위의 식은

$$X = m(u' - v\psi' - y_G\psi'' - x_G\psi'^2)$$

$$Y = m(v' + u\psi' - y_G\psi'^2 + x_G\psi'') \qquad (2.5a)$$

$$N = I_z\psi'' + m[x_G(v' + u\psi') - y_G(u' - v\psi)]$$

이들 식은 수평면 안에서의 운동만을 다루고 있기 때문에 직선거리 z_G는 무시했다.

식 (2.5)와 (2.5a)에 들어 있는 힘 성분 X, Y와 모멘트 성분 N은 선박의 속도와 가속도의 함수를 포함하는 여러 부분으로 구성되었다고 가정하고, 이들이 여러가지 외란으로 인한 선박의 운동에 기인하는 힘과 모멘트만으로 이루어진다고 가정하면, X, Y 및 N은 다음과 같이 표시된다.

$$X = F_x(u, v, u', v', \psi', \psi'')$$

$$Y = F_y(u, v, u', v', \psi', \psi'') \tag{2.6}$$

$$N = F_\psi(u, v, u', v', \psi', \psi'')$$

연직면 안에서의 운동방정식은

$$Z = F_z(z_0, u, \omega, \theta, u', \omega', \theta', \theta'')$$

$$M = F_\theta(z_0, u, \omega, \theta, u', \omega', \theta', \theta'') \tag{2.7}$$

단, Z : z축 방향 힘의 합

M : xz 면 안에 있는 힘의 y축에 관한 모멘트의 합

z_0 : 초기 평형 상태로부터 z_0 축에 연하여 움직인 연직 변위량, 따라서 z_0 는 위치를 나타내는 파라미터이다.

ω : V 의 z 축 방향의 성분

θ : 연직면 $y_0 z_0$ 으로부터 x 축 사이의 세로 경사각

운동 안정성을 나타내는 지수(index)를 표시하기 위해서 식 (2.6)과 (2.7)을 다변수 함수에 대한 Taylor의 전개법을 이용하여 다음과 같이 나타낸다. 여기서 변수 x의 특정한 값에서 그 함수와 모든 도함수들이 연속일 경우에는

$$f(x) = f(x_1) + \frac{1}{1!} \cdot \frac{df(x_1)}{dx} \cdot \Delta x + \frac{1}{2!} \cdot \frac{d^2 f(x_1)}{dx^2} \cdot \Delta x^2 + \cdots$$

$$\cdots + \frac{1}{n!} \cdot \frac{d^n f(x_1)}{dx^n} \cdot \Delta x^n \qquad (2.7a)$$

단, $f(x)$: x_1 **근방의** x **에서의 함수값**

$f(x_1)$: $x = x_1$ **에서의 함수값**

$\Delta x = x - x_1$

$\frac{d^n f(x)}{dx^n}$: $x = x_1$**에서의** n**차 도함수 값**

변수의 변화량 Δx**가 충분히 작다면 식** (2.7a)**에서** Δx**의 고차항들을 무시할 수 있으므로 다음과 같이 변형된다**.

$$f(x) = f(x_1) + \frac{1}{1!} \cdot \frac{df(x_1)}{dx} \cdot \Delta x \qquad (2.7b)$$

두 변수 x**와** y**의 함수를** Taylor**의 급수로 전개하여 선형화하면**

$$f(x,y) = f(x_1, y_1) + \frac{\partial f(x_1, y_1)}{\partial x} \cdot \Delta x + \frac{\partial f(x,y)}{\partial y} \cdot \Delta y \qquad (2.7c)$$

식 (2.6)속의 힘 Y 를 선형화하면 식 (2.7c)로부터

$$Y = F_y(u_1, v_1, u'_1, v'_1, \psi'_1, \psi''_1) + (u - u_1)\frac{\partial Y}{\partial u} + \cdots$$
$$+ (v - v_1)\frac{\partial Y}{\partial v} + (\psi'' - \psi''_1)\frac{\partial Y}{\partial \psi''} \tag{2.8}$$

결국 u_1 만이 0이 아니므로 위의 식을 정리하면

$$Y = \frac{\partial Y}{\partial v} v + \frac{\partial Y}{\partial v'} v' + \frac{\partial Y}{\partial \psi'} \psi' + \frac{\partial Y}{\partial \psi''} \psi'' \tag{2.9a}$$

식 (2.6)속의 힘 X 와 모멘트 N 에 대한 식들을 정리하면

$$X = \frac{\partial X}{\partial u'} u' + \frac{\partial X}{\partial u} u + \frac{\partial X}{\partial v} v + \frac{\partial X}{\partial v'} v' + \frac{\partial X}{\partial \psi'} \psi' + \frac{\partial X}{\partial \psi''} \psi'' \tag{2.9b}$$

$$N = \frac{\partial N}{\partial v} v + \frac{\partial N}{\partial v'} v' + \frac{\partial N}{\partial \psi'} \psi' + \frac{\partial N}{\partial \psi''} \psi'' \tag{2.9c}$$

식 (2.9b)는

$$X = \frac{\partial X}{\partial u'} u' + \frac{\partial X}{\partial u} \Delta u \tag{2.9d}$$

로 표현되고, 배의 무게중심이 서로 대칭면 안에 있는 경우에는 y_G 가 0이므로 식 (2.5a)는 다음과 같이 표현된다.

$$X = m(u' - \psi' v - x_G \psi'^2)$$

$$Y = m(v' + \psi' u + x_G \psi'') \tag{2-10}$$

$$N = I_z \psi'' + m x_G (v' + \psi' u)$$

이들 중에서 Y에 대한 식의 괄호 속을 선형화하면

$$v' + \psi' u + x_G \psi'' = (v'_1 + \Delta v') + (\psi'_1 + \Delta\psi')(u_1 + \Delta u) + \\ + x_G(\psi''_1 + \Delta\psi'')$$

되며 여기서 ψ_1의 값은 무시할 수 있으므로 윗식은 다음과 같이 표시된다.

$$v' + \psi' u + x_G \psi'' = v' + \psi' u_1 + x_G \Delta\psi''$$

유사한 방법으로 식 (2.10)**의** X**와** N**을 선형화하면**

$$X = m \cdot u'$$

$$Y = m(\Delta v' + u_1 \Delta\psi + x_G \Delta\psi'')$$

$$N = I_z \psi'' + m x_G (v' + \psi' u)$$

의 운동법칙으로부터 얻어진 일반화된 선박의 운동방정식을 평형점 $u = u_0,\ v = r = \delta = 0$**에 대하여 선형 근사화시킨 후, 단위를 무차원화하여 얻은 식이 다음과 같다.**

$$\begin{pmatrix} m' - Y'_{\dot v} & L(m'x'_G - Y'_{\dot r}) \\ m'x'_G - N'_{\dot v} & L(I'_z - N'_{\dot r}) \end{pmatrix} \begin{pmatrix} \dot v \\ \dot r \end{pmatrix}$$

$$= \begin{pmatrix} \frac{V}{L} Y'_v & V(Y'_r - m') \\ \frac{V}{L} N'_v & V(N'_r - m'x'_G) \end{pmatrix} \begin{pmatrix} v \\ r \end{pmatrix} + \begin{pmatrix} \frac{V^2}{L} Y'_\delta \\ \frac{V^2}{L} N'_\delta \end{pmatrix} \delta \qquad (2.11)$$

여기서, 배의 길이의 단위는 L, 시간의 단위는 L/V, 무게의 단위는 $\rho L^3/2$ (ρ = 물의 밀도), r 은 회전율(turning rate), v 는 횡속도(sway velocity), δ 는 타각이다. 식 (2.11)로부터 회전율과 타각 사이의 전달함수를 구하면

$$G_{r\delta}(s) = K\frac{(s+1/T_3)}{(s+1/T_1)(s+1/T_2)} \tag{2.12}$$

으로 된다. 위 식을 $Nomoto$는 다음과 같이 근사화한다.[1]

$$G_{r\delta}(s) = \frac{K}{(s+1/T)} \tag{2.13}$$

단, $T = T_1 + T_2 - T_3$

여기서, T_1, T_2는 추종 및 안정성 계수, T_3는 타각에 대해 미분제어 효과를 발휘함으로써 침로 안정에 도움을 주는 계수이며, K는 선회 능력에 관련된 계수이다. 식 (2.13)에서 선수 각속도를 적분함으로써 다음과 같은 선수각과 타각 사이의 전달함수를 구할 수 있다.

$$G_{\phi\delta}(s) = \frac{K}{s(s+1/T)} \tag{2-14}$$

2.3 제어계 설계

2.3.1 자동운항 제어시스템

자동운항 제어시스템은 목표치 설정부와 피이드백 요소, Power Amplifier, Gear Unit 및 Ship Dynamics로 구성된다.

Power Amplifier와 Gear Unit의 전달함수는 각각 다음과 같다.

Power Amplifier : $\dfrac{0.5}{\tau_A S + 1}$

Gear Unit : $\dfrac{0.5}{\tau_G S + 1}$

여기서 τ_A , τ_G는 시정수이고 일반적으로 대단히 작은 값이므로 무시한다. Fig. 3은 자동운항 제어시스템의 블록 다이어그램을 나타낸다.

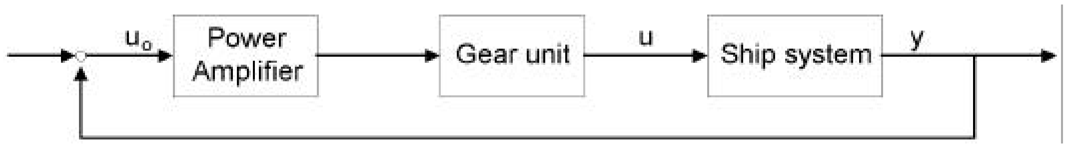


Fig. 3 Block diagram of Auto-steering system

2.3.2 제어계 설계

여기서 사용하는 제어입력은 다음과 같다.[2]

$$v_k = A_1 y_{(k)} - A_2 y_{(k-1)} + B_1 u_{(k)} + B_2 u_{(k-1)}$$

또한 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} y_k \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ y_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_k \qquad (2.15)$$

윗식의 출력이 다음과 같은 목표값을 추종한다면

$$\begin{bmatrix} r_k \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ r_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r_{k+1} \qquad (2.16)$$

식 (2.15),(2.16)**으로 부터 다음을 얻을 수 있다**. (**단** $e_k = r_k - y_k$)

$$\begin{bmatrix} e_k \\ e_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix} \begin{bmatrix} e_{k-1} \\ e_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_k' \qquad (2.17)$$

여기서 $v_k = r_{k+1} - v_k$ 이다. 따라서

$$v_k = r_{k+1} - v_k - k_1 e_k - k_2 e_{k-1} = 0$$

이 되도록 하는 것이 목적이므로 윗 식을 정리하면 식 (2.18)이 되고

$$0 = r_{k+1} - B_1 u_k - B_2 u_{k-1} - A_1 y_k + A_2 y_{k-1} + \\ - k_1 e_k - k_2 e_{k-1} \quad (2.18)$$

다시 정리하면 제어입력 u_k는

$$u_{(k)} = \frac{1}{B_1}(r_{k+1} - B_2 u_{k-1} - A_1 y_k + A_2 y_{k-1} + \\ - k_1 e_k - k_2 e_{k-1}) \quad (2\text{-}19)$$

이다.

2.3.3 제어시스템

● 비례제어

먼저 가장 간단한 비례제어기를 고려하면 비례제어칙은 시스템의 입력과 출력 사이의 오차 e에 선형적으로 비례하는 공정계 입력 u를 가지므로 다음 식으로 표현된다.

$$u(t) = k_p e(t) \tag{2.20}$$

단, $e(t) = r(t) - y(t)$

이와 같이 간단한 궤환 제어는 단일매개변수 k_p만을 조작하여 복잡한 설계 요구조건을 만족할 수 있다. 비교적 느린 응답속도는 적은 k_p의 값에 의하여 발생하므로, 흔히 k_p를 상향 조정한다. 그러나 큰 진동을 갖는 응답은 k_p의 높은 값에서 발생한다. 한편 빠른 응답을 얻고 정상상태 오차를 줄이기 위해 값이 큰 k_p를 선택하는 경향이 있지만, 아직도 큰 진동을 갖는 과도응답을 피하기 위해 값이 작은 k_p를 필요로 한다. Fig. 4는 P제어기의 블록다이아그램을 나타낸다.

● 비례-적분 제어

비례-적분 제어의 경우 제어입력 u는 다음과 같다.

$$u(t) = k_p e(t) + k_I \dot{e}(t) \tag{2.21}$$

Fig. 5는 비례제어에 적분제어 요소를 병렬로 추가한 비례-적분 제어기를 나타낸 블록선도이며, 적분게인은 정상상태 오차를 없애주는 역할을 한다. 만일 비례제어의 경우 정상상태 오차가 존재한다면, 정상상태 오차를 0으로 떨어뜨리는 적분제어는 제어계를 원하는 상태로 유도하기 위해 동작신호를 발생할 것이다. 따라서, 비례이득 k_p는 과도응답을 향상시키기 위해 조절할 수 있다.

따라서 비례-적분제어계의 입력 u는 다음과 같이 결정된다.

$$u = k_P e(t) + k_I \int e(t)\, dt \tag{2.22}$$

단, $e(t) = r(t) - y(t)$ 이다.

● 비례-미분 제어

미분제어의 경우는 입력 u를 e와 $\dot{e}$의 선형결합으로 구성함으로써, 과대한 발진을 발생시키지 않고, 오차를 0의 값으로 떨어뜨릴 수 있다. 미분제어 동작과 비례제어를 채용하는 비례-미분 제어기의 제어입력 u는 다음과 같다.

$$u(t) = k_p e(t) + k_D \dot{e}(t) \tag{2.23}$$

Fig. 6은 비례-미분 제어기의 블록선도를 나타낸다.

● 비례-미분-적분 제어

현재 까지 산업계에서 가장 일반적으로 사용되고 있는 제어기라 할 수 있는 비례-미분-적분제어기는 오차에 대해 게인이 각각 선형 결합으로 이루어져 있으며 과대한 발진을 발생시키지 않고, 오차를 0의 값으로 떨어뜨리며, 보다 빨리 목표치에 추종할 수 있다.

그러나 자동 조타기에 대해 이상적인 PID 게인을 구하는 것은 많은 시행착오를 요하는 문제가 된다. PID 제어기의 제어입력은 다음과 같이 결정되며 Fig. 7은 PID제어기의 블록선도를 보인다.

$$u = k_P e(t) + k_D \dot{e}(t) + k_I \int e(t)\,dt \tag{2.24}$$

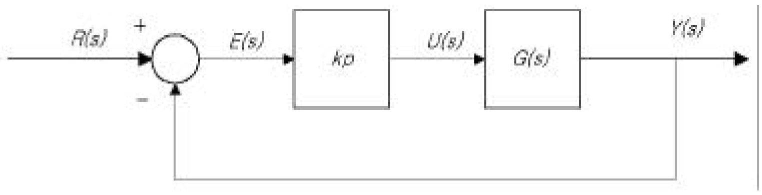


Fig.4 Proportional controller

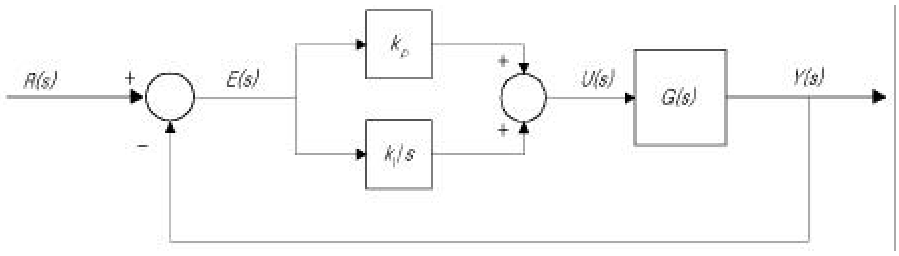


Fig.5 Proportional-Integral controller

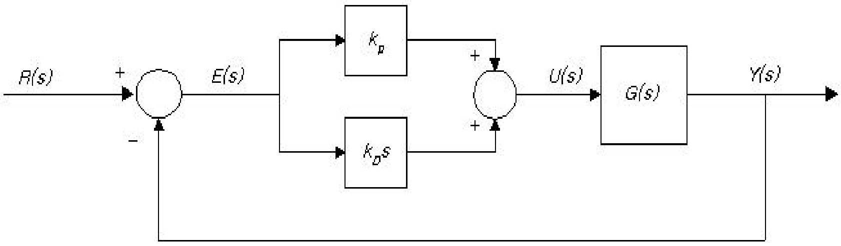


Fig.6 Proportional-Derivative controller

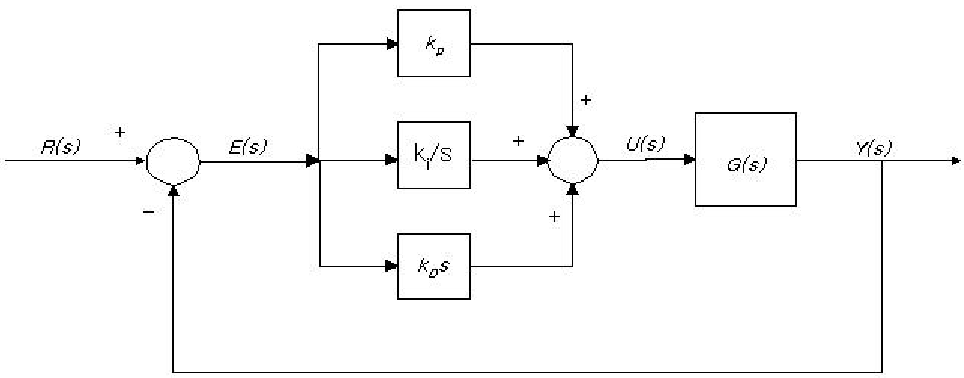


Fig.7 Proportional-Integral-Derivative controller

● 자기동조 제어시스템

앞절에서 언급한 시스템의 상태방정식을 차분방정식의 형태로 표현하면

$$y_{k+1} - A_1 y_k + A_2 y_{k-1} = B_1 u_k + B_2 u_{k-1} \tag{2.25}$$

과 같이 표시할 수 있다. 여기서 파라미터

$$\theta^T = [\ A_1\ A_2\ B_1\ B_2]$$

$$\phi^T = [\ y_k\ y_{k-1}\ u_k\ u_{k-1}\ ] \tag{2.26}$$

라고 두고,

$$J_i = \sum_{i=0}^{k} \| y_{i+1} - \phi_i^T \hat{\theta} \|^2 \lambda^{k-i} \tag{2.27}$$

을 최소로 하는 파라메타를 구하므로서 제어 동작을 수행한다. Fig. 8은 자기동조 제어기의 블록선도를 보인다.

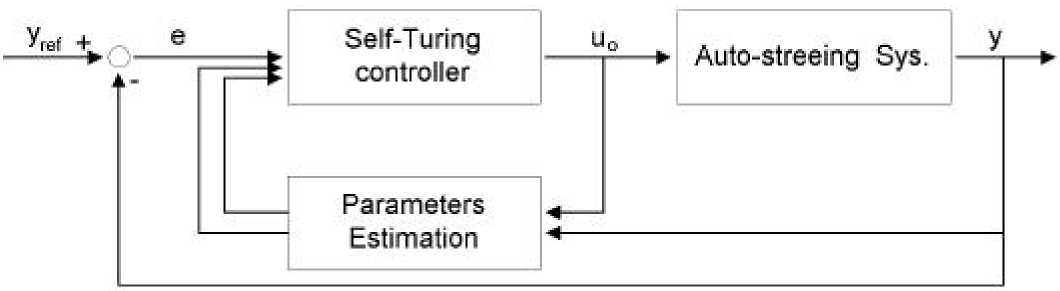


Fig.8 Self-tuning control system

2.4 제어 소프트웨어

제어계의 전체 개략도와 제어소프트웨어는 각 루틴의 섹션별로 설명한다.

2.4.1 전체 제어계 개략도

전체 제어계의 개략도를 Fig. 9에 나타낸다.

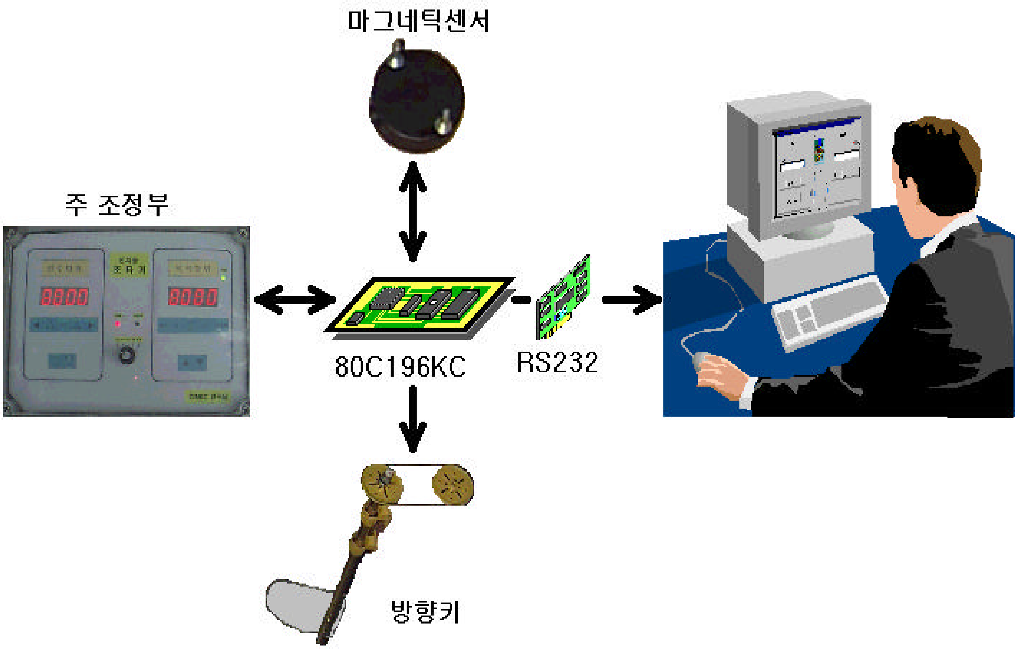


Fig.9 Schematic Diagram of Autopilot System

2.4.2 프로그램

전체 제어 프로그램은 Fig. 10과 같이 구성되어 있으며, 어셈블러를 이용하여 프로그래밍 하였다. 전체적인 프로그램 구성은 준비(READY), 조타(STEERING), 자동(AUTO), 정지(QUIT)이다. 먼저 전체 프로그램의 흐름은 SFR(Special Function Register), 사용자 정의 레지스터와 인터럽트 벡터 테이블을 정의한다. main 프로그램에서는 키패드로부터 지령을 기다린다. 만약 READY라는 키를 누르면 READY 루틴으로 넘어가서 그 루틴을 수행하도록 되어 있다.

- Main program

Fig. 11은 main program의 흐름을 나타낸다.

- Ready program

Fig. 12는 준비(READY)부분의 플로우차트이다.

키패드로부터 준비버턴이 눌러지면 위와 같은 프로그램을 수행하게 된다. 이 부분은 정타(Direct steering), 역타(Inverse steering), 감도(Sensitive), 회전률(Rotating rate) 값을 설정하도록 되어 있다.

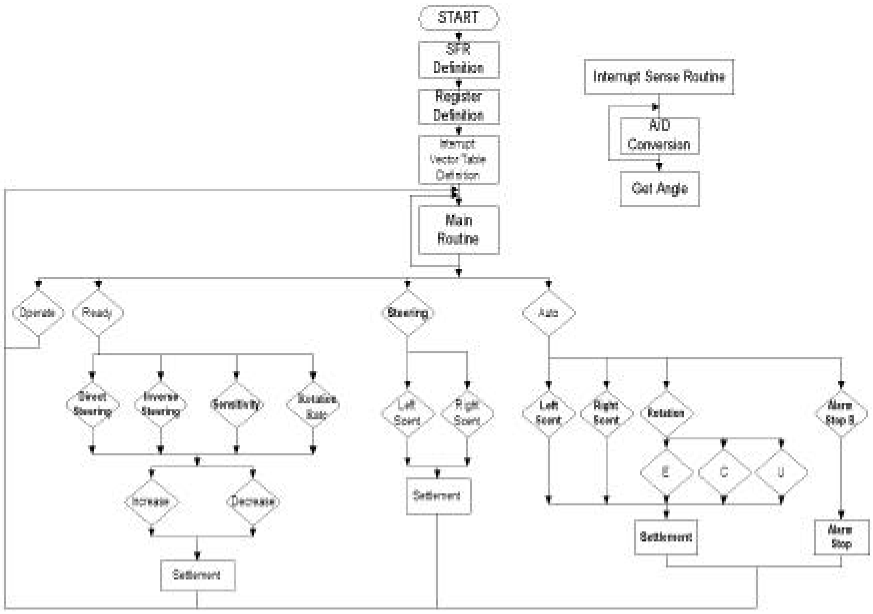


Fig.10 Flowchart of Program

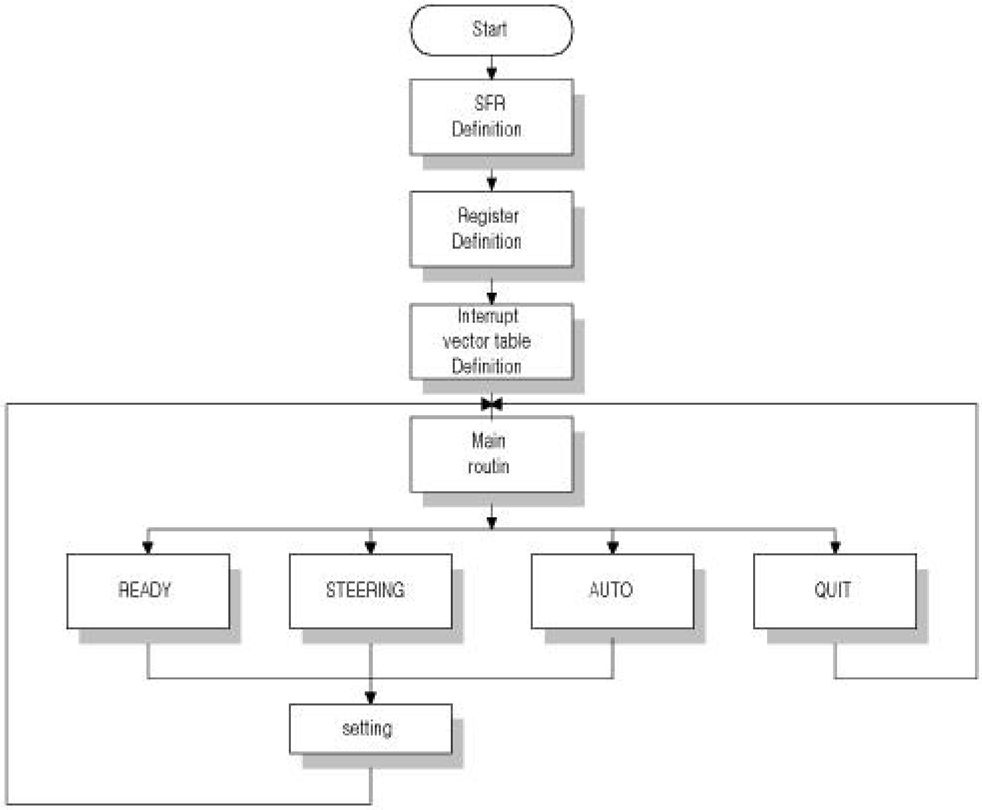


Fig.11 Flowchart of main program

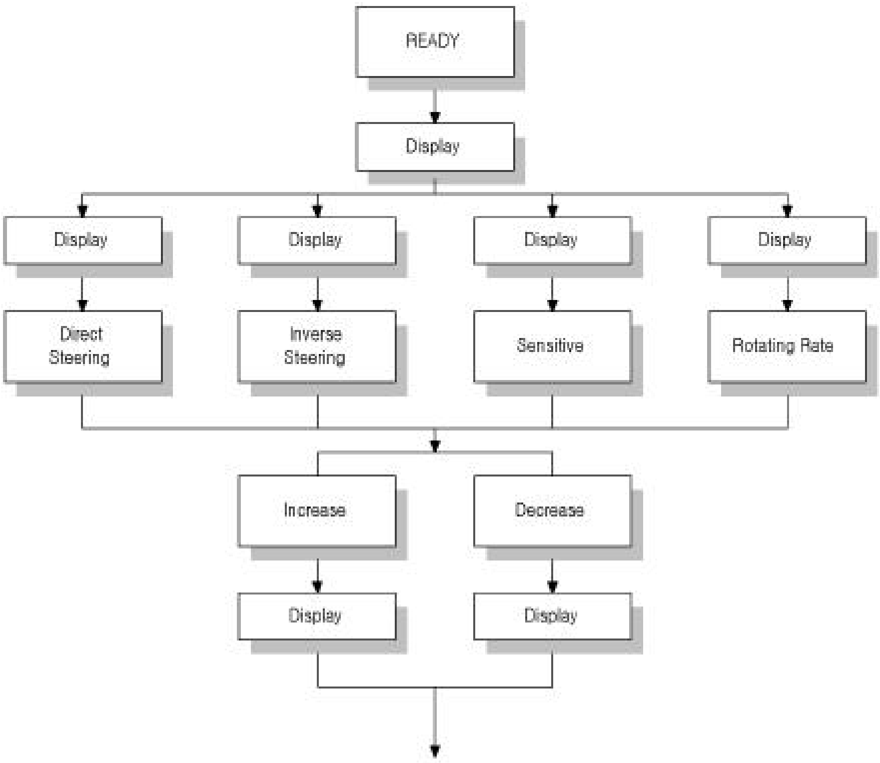


Fig.12 Flowchart of READY part

● Steering program

Fig. 13은 조타(STEERING)부분의 플로우차트이다. 이것은 항해 중 물체가 갑자기 나타나서 회피해야 할 경우에 사용되며, 좌회피(Left Avoiding)와 우회피(Right Avoiding)으로 구성되어 있다.

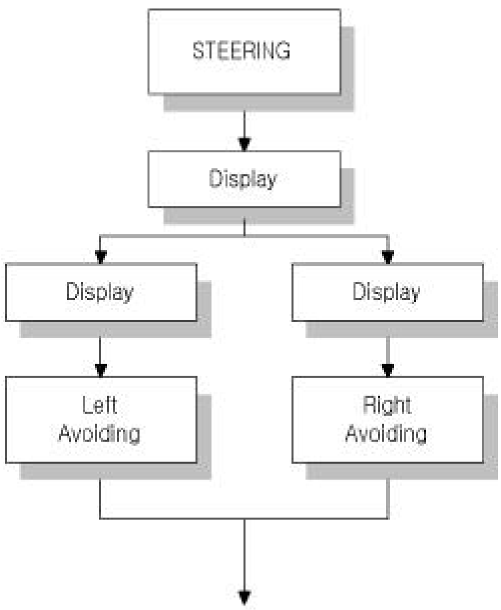


Fig.13 Flowchart of STEERING part

● Auto program

Fig. 14는 자동(AUTO)부분의 플로우차트를 나타내고 있다.

먼저 키패드로부터 자동 버턴이 입력되면 목표값을 설정하게 된다. 다음으로 80C196KC로 부터 선수각을 센싱하고, 이것과 목표각 사이의 오차를 이용하여 제어입력인 타각을 계산한다.

다음으로 실제 타각을 센싱하여 계산된 타각과 비교하며 실제 타각이 계산된 타각에 추종할 수 있도록 port를 작동시킨다.

여기서는 Timer 1 Overflow에 의해서 0.5초 당 한번씩 인터럽터가 수행되도록 되어 있다.

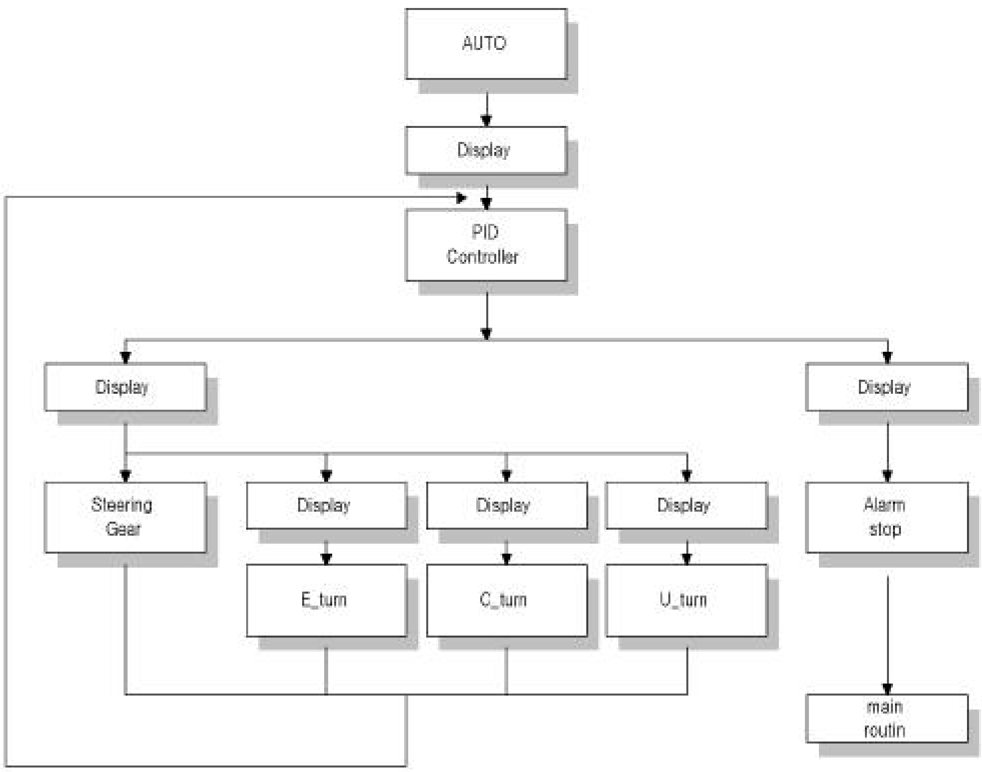


Fig.14 Flowchart of AUTO part

제 3 장 시 뮬 레 이 션 및 실 험

3.1 실험방법

본 연구에서는 Matlab을 이용하여 시뮬레이션을 행하였고, 각각의 게인 값은 시뮬레이터에 대한 입·출력 특성 실험을 통해 P = 2.2, I = 0.001, D = 5.5로 정하였다. 또한 개발한 제어 알고리즘 및 소프트웨어를 80C196KC를 이용하여 구현하고 앞에서 언급한 시뮬레이터를 이용하여 실험 결과를 얻었다. Fig. 15는 80C196KC를 채용한 제어계의 개략도를 나타내고, Fig.16 은 80C196KC 제어기 및 시뮬레이터를 연결한 전체 자동운항 제어시스템의 시뮬레이션 구성에 대한 개략도를 나타낸다.

3.2 시뮬레이션 및 실험결과

Fig. 17는 목표값를 45[deg]로 하였을 경우에 있어 (a)는 PID제어기를 사용하였을 때의 시뮬레이션 결과이고, (b)는 PID제어기를 80C196KC에 적용시켜 직접 실험을 통해 얻은 결과를 보여주고 있다. 또 Fig. 18 21의 (a), (b), (c)는 각각 목표값를 60[deg], 90[deg], 120[deg], 270[deg]인 경우의 P, PID의 시뮬레이션 결과 및 실험 결과를 각각 보인다.

먼저 Fig. 17를 보면 (a)는 약 25[sec]에서 안정하게 되고, (b)는 약 30[sec]에서 안정점에 도달한다. 그리고 Fig. 18 21에서는 (a), (b)는 약 32[sec]에서 안정하게 되고, (c)는 약 40[sec]가 되어서 안정하게 된다.

이러한 결과로 미루어 보면 본 연구에서 개발한 제어시스템은 소형선의 자동조타장치로써 실선 장착에 필요한 충분한 제어시스템의 성능을 가지고 있음을 확인할 수 있고 제안된 제어 알고리즘과 소프트웨어는 실험 결과를 통하여 그 유효성을 충분히 검증할 수 있다.

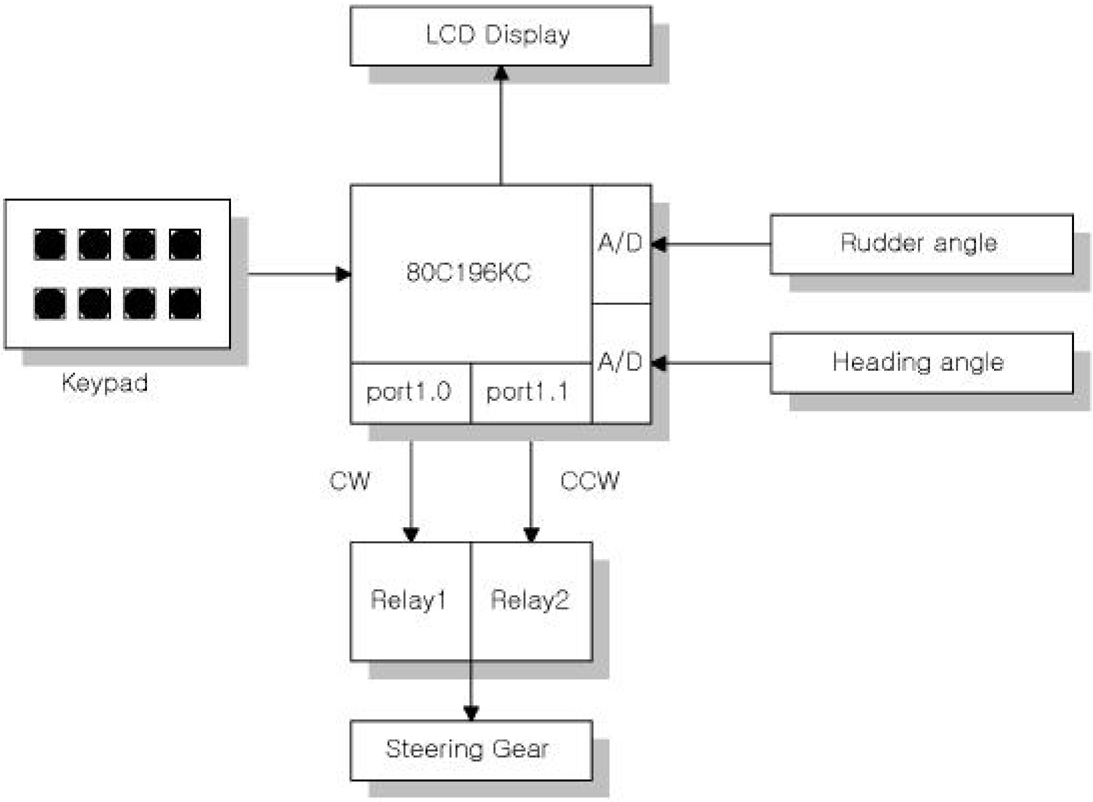


Fig.15 Block diagram of 80C196KC system

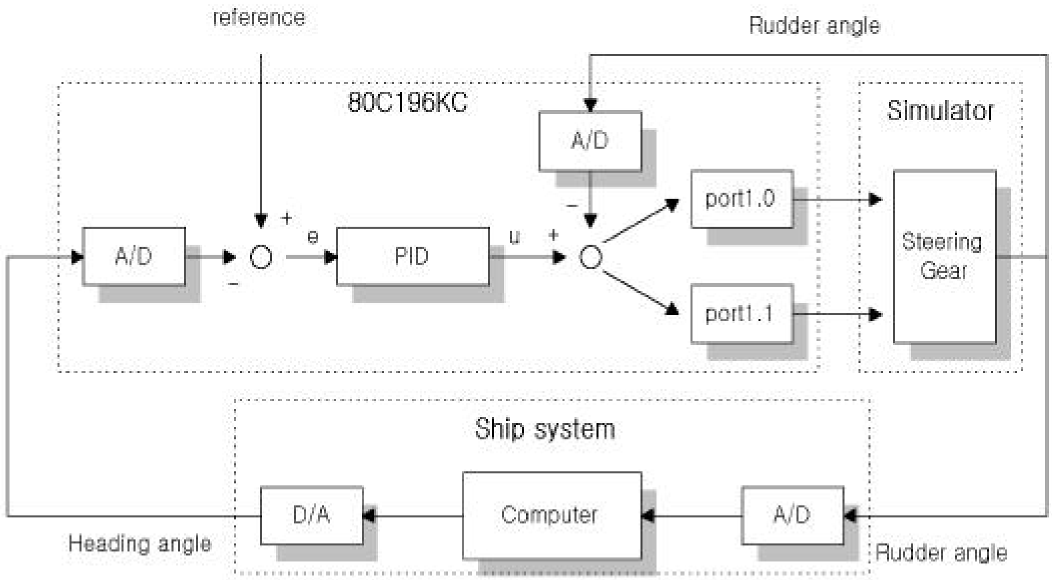
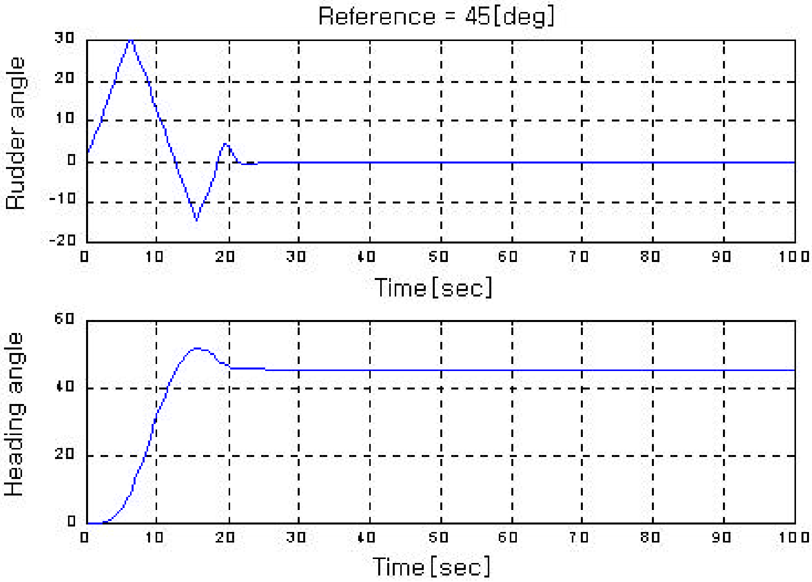
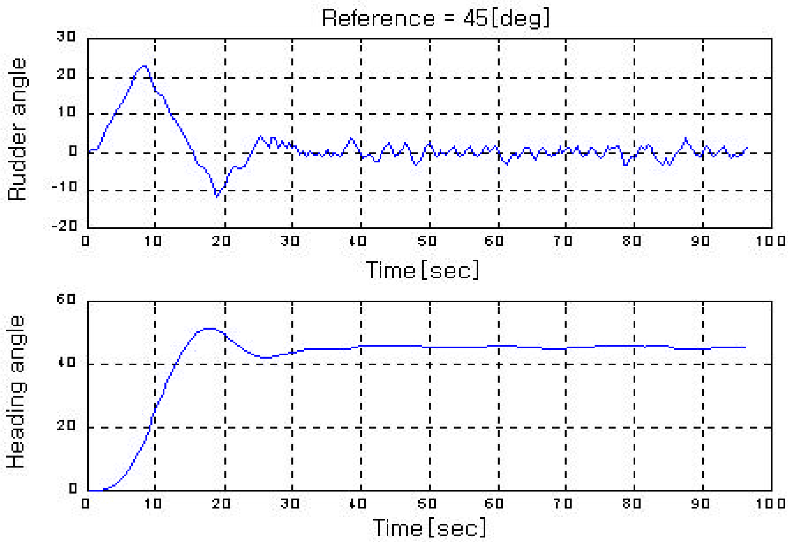


Fig.17 Blockdiagram of test system with the simulator

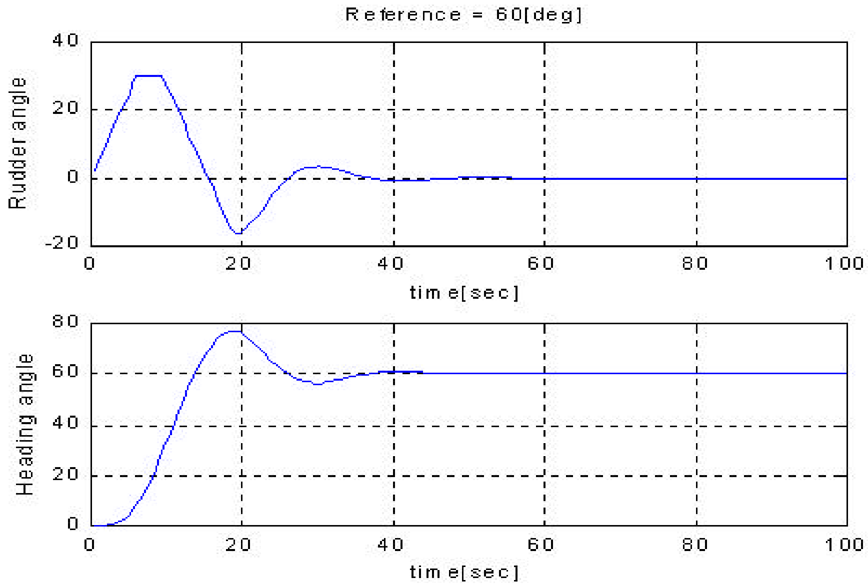


(a)Simulation result for PID control

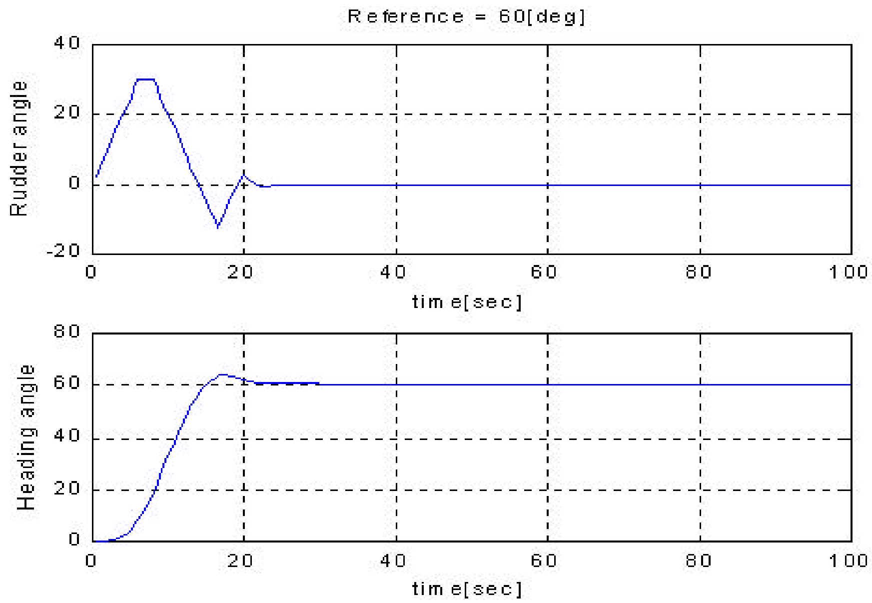


(b) Experimental result for PID control

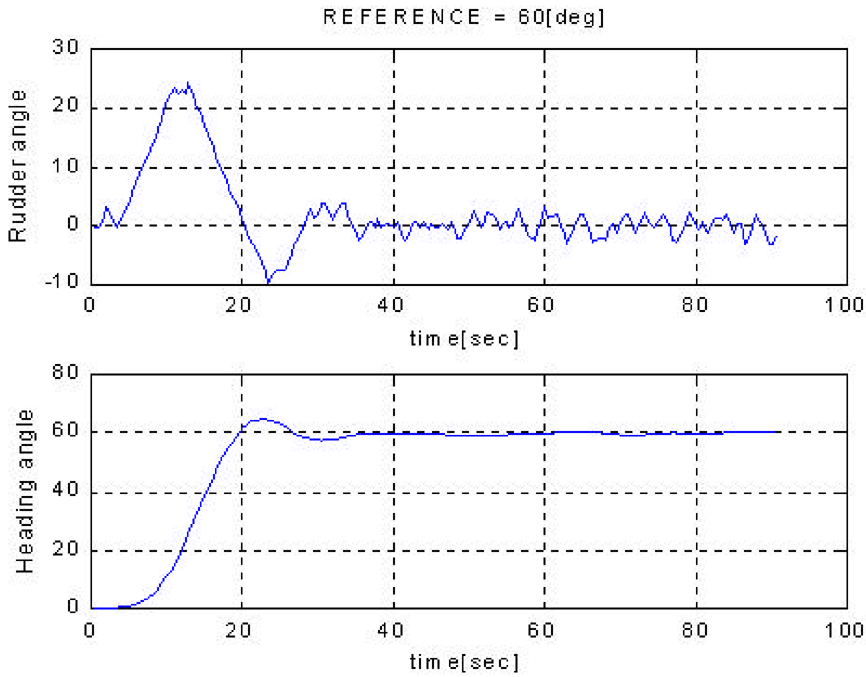
Fig. 17 Results in case of reference 45[deg]



(a) Simulation result for P control

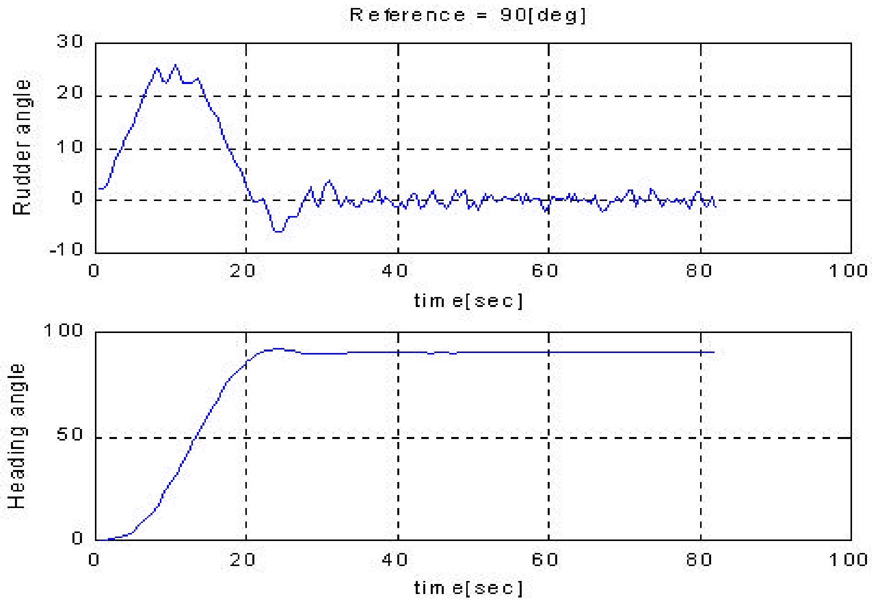


(b) Simulation result for PID control

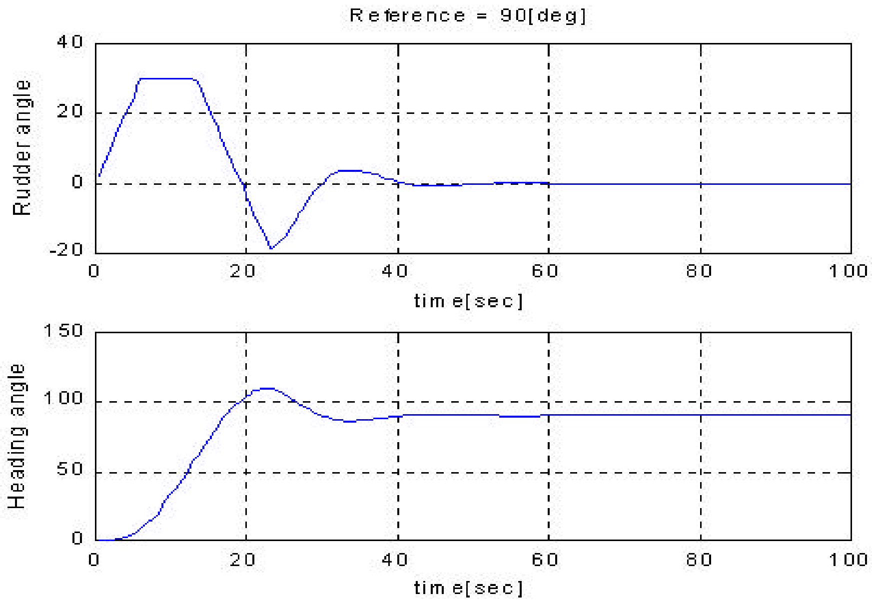


(c) Experimental result for PID control

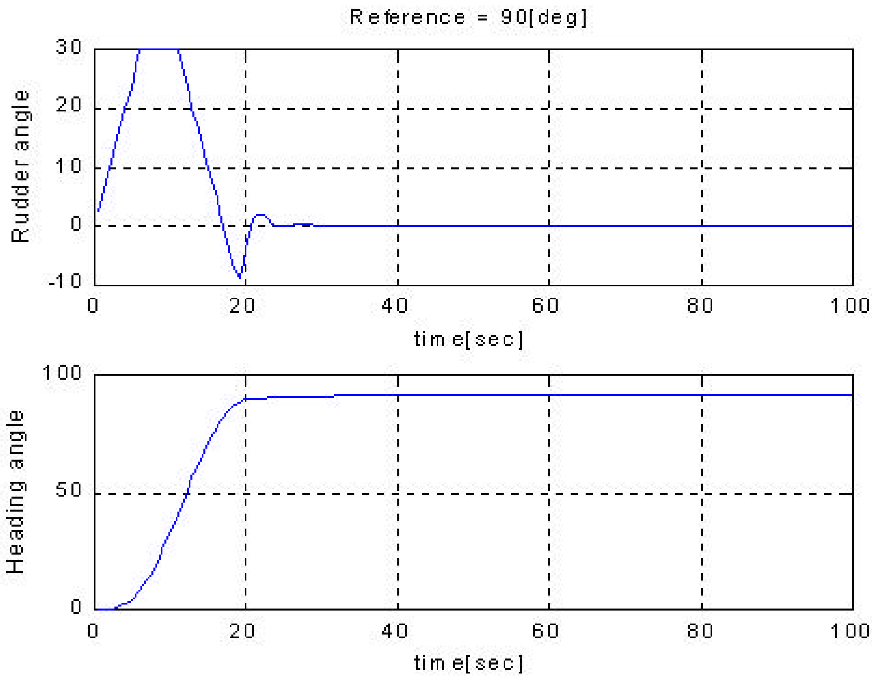
Fig. 18 Results in case of reference 60[deg]



(a) Simulation result for P control

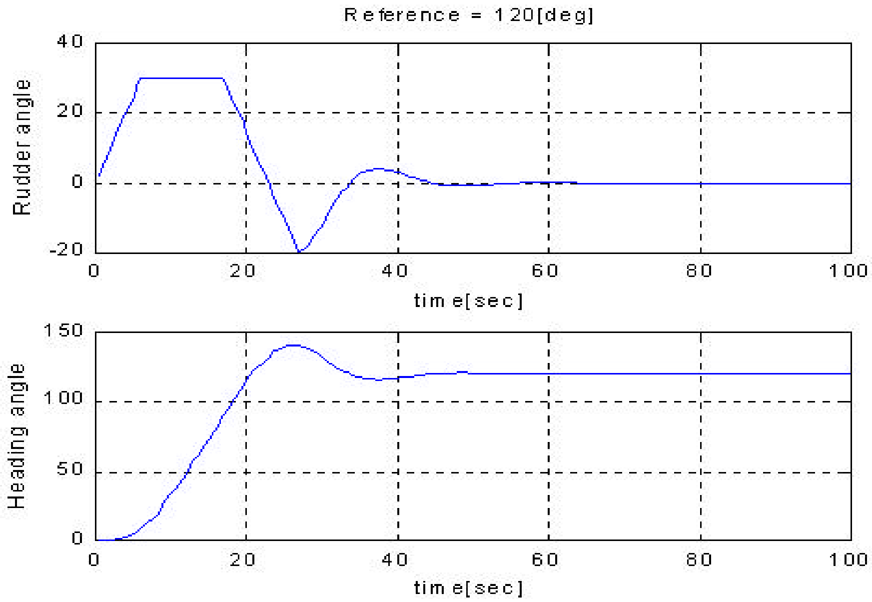


(b) Simulation result for PID control

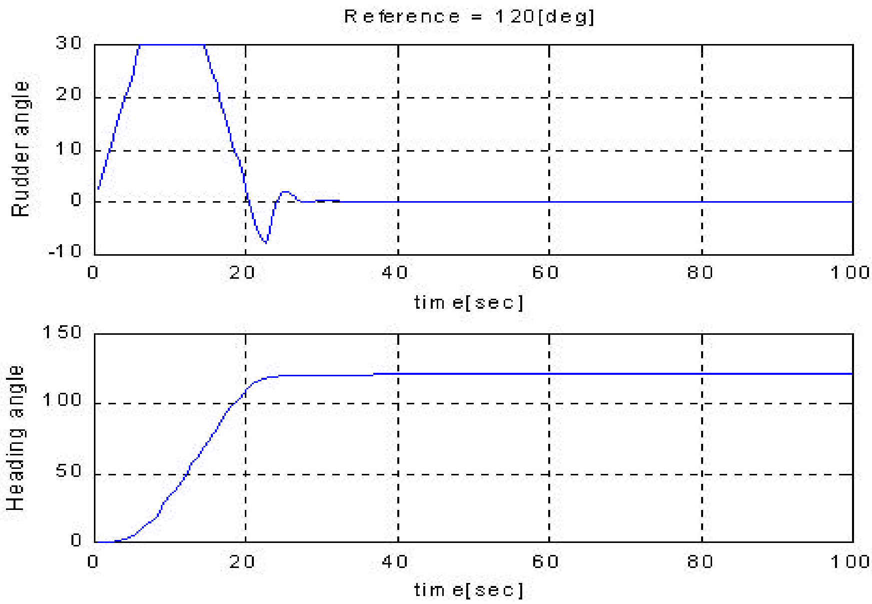


(c) Experimental result for PID control

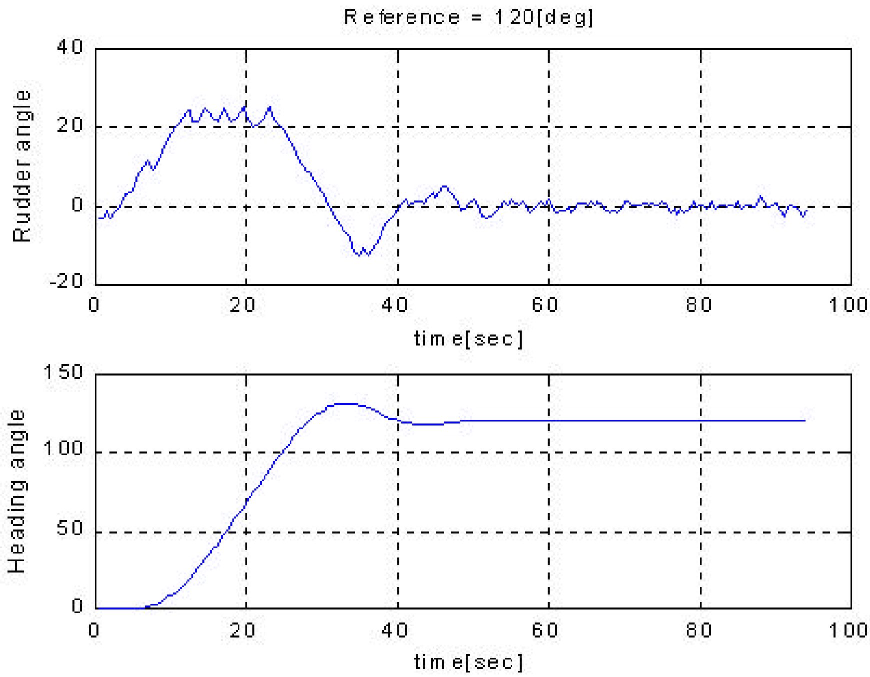
Fig. 19 Results in case of reference 90[deg]



(a) Simulation result for P control

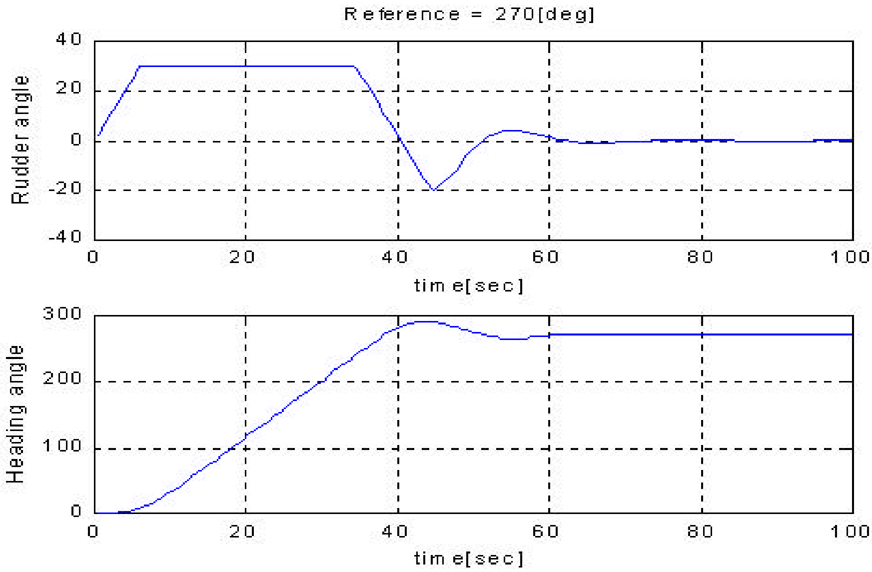


(b) Simulation result for PID control

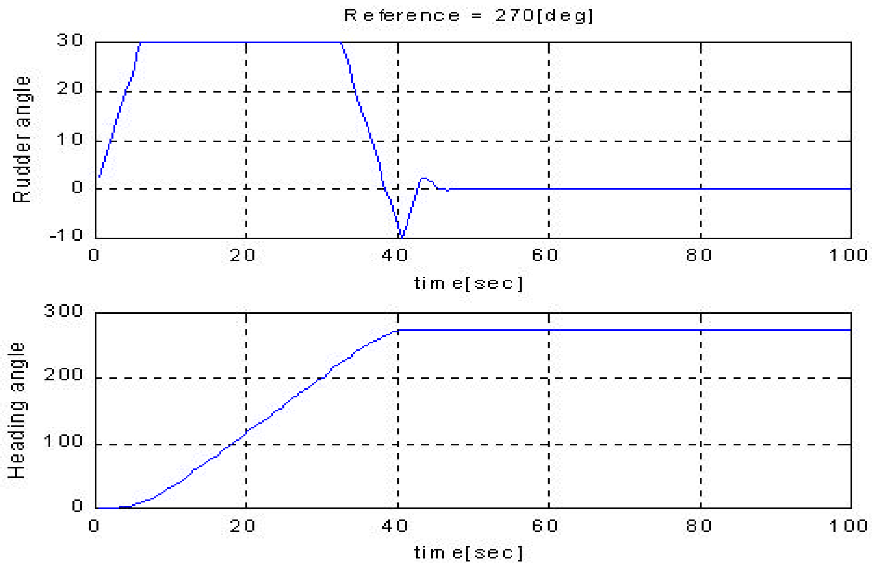


(c) Experimental result for PID control

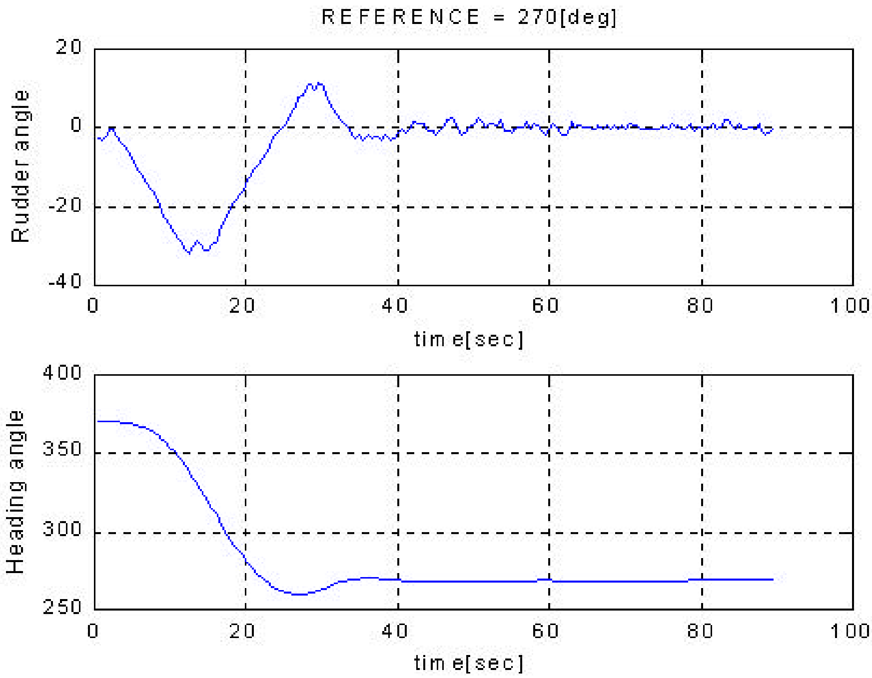
Fig. 20 Results in case of reference 120[deg]



(a) Simulation result for P control



(b) Simulation result for PID control



(c) Experimental result for PID control

Fig. 21 Results in case of reference 270[deg]

제 4 장 결 론

현재 국내에서 사용되고 있는 소형선박용 자동조타기는 외국기술에 의존하고 있어 가격이 비싸고 또한 하위레벨의 마이크로프로페서를 사용하므로 버전변경 내지 장치의 고장시 대처하기가 곤란하다.

이러한 문제점을 해결하기 위해서 본 연구에서는 소형 선박용 자동 조타기 제어 장치를 위하여 80C196KC 마이크로프로세서와 PID 제어기를 적용한 자동 조타 시스템에 대한 제어알고리즘과 소프트웨어를 개발, 제시하였다.

개발된 자동조타장치에 대해 컴퓨터 시뮬레이션 및 시뮬레이터를 이용한 실험결과 다양한 스텝 형태의 목표치에 대해 응답이 잘 추종하고 두 방법의 응답이 잘 일치함을 확인함으로써 본 방법의 유효성을 확인하였다.

향후 과제로는 실제 선박에 적용시켰을 경우 발생하게 될 여러 가지 문제점을 검토·분석하여 이를 효과적으로 이용할 수 있는 방안을 강구하는 것이 필요하다고 사료된다.

참 고 문 헌

1) C. G Kallstrom and K. J. ASTROM, "Experiences of System Identification Applied to Ship Steering", International Federation of Automatic Control p187 - 198, 1981

2) Bystrom, L. and C. G. Kallstrom, "System Identification of Linear and Non-linear Ship Steering Dynamics.", Proc. 5th Ship Control Systems Symposium, Annapolis, maryland, USA, 1978

3) T. Koyama, "On the Optimum Automatic Steering System of Ships at Sea", Journal of the Society of Naval Architects of Japan, Vol. 122, pp. 18-35, Dec. 1976.

4) K. J. Astrom and B. Wittenmark, "Computer Controlled Systems Theory and Design," Prentice-Hall, pp. 221-253, 1984.

5) Tatsuo Arie, Masanori Itoh, Akira Senoh, "An Adaptive Steering System for a Ship", IEEE Control System Magazine, 1986. 10

6) **한봉주**, **배경수**, **김환성**, **김상봉**, "**자동운항시스템의 제어에 관한 실험적 연구**(1)", **대한기계학회논문집**(A), **제**20**권**, **제** 8**호**, pp. 2449 - 2457, 1996

7) **하주식**, **오세준**, **유희한**, **윤지근**,"**선박용 디젤기관의 지능적인 속도제어시스템**", **한국박용기관학회지 제**22**권 제**3**호**, pp. 320-328, 1998