

# 로봇구동용 직류전동기의 동기화 구현

엄한성\* · 안병원\* · 김현수\* · 박영산\* · 배철오\*

\*목포해양대학교

## Synchronization of Robot driving DC motor

Han-Sung Eom\* · Byong-Won Ahn\* · Hyun-Soo Kim\* · Young-San Park\*

Cherl-O Bae\*

\*Mokpo Maritime University

E-mail : mrehs@mmu.ac.kr

### 요 약

본 연구는 인터넷을 통해서 실시간으로 전달되는 속도지령에 대해서 원거리에 위치한 로봇구동용 DC 전동기 동기화 제어에 관한 연구이다. 두 대의 DC 전동기 구동시스템에서 80C196KC  $\mu$ -Processor에 의해서 DC 전동기에 인가되는 PWM전압과 회전수의 관계로 PID제어기를 설계하고 임의의 속도(위치) 지령에 대해서 두 대의 전동기가 빠른 시간에 동기화가 될 수 있게 하였다. 또, 호환성이 뛰어난 Java 플랫폼을 기반의 Java Applet과 Servlet을 연동시켜 보안을 유지하면 인터넷 상에서 전동기를 실시간으로 제어하고, 속도를 모니터링 할 수 있게하였다.

### ABSTRACT

This paper is studied about the DC motor synchronization control for robot driving DC motor where is located in long distance place. The DC motors are received the speed order through internet with realtime. We designed the PID controller that related to PWM voltage and revolution given by 80C196KC  $\mu$ -Processor to move DC motors, so two motors were synchronized quickly about the order of random speed or location by PID controller. Furthermore, we controlled motors with realtime through internet by using Java platform that had a excellent compatibility, and monitored the speed of two motors.

### 키워드

Synchronization, PID controller, PWM, DC motor

### 1. 서 론

바퀴로 구동되는 이동용 로봇은 바퀴들이 상호 동기화가 되지않으면 직진성이 확보되지 않아 주행중 요동주행을 하게된다. 이런 로봇들은 대부분 배터리 전원을 사용하는 DC 전동기에 의해서 구동된다. 직진성이 확보되지 않으면 정확한 목표 위치를 추종할 수 없게되고 요동 주행에의한 에너지 손실도 클 것이다. 본 논문에서는 두 대의 DC 전동기를 빠른 응답속도로 동기화 시킴으로써 외부 조건 변화 에도 바퀴로 구동되는 이동용 로봇의 정확한 직진성을 확보할 수 있도록 80C196KC  $\mu$ -Processor에 의해서 DC 전동기에 인가되는 PWM

전압과 회전수의 관계에서 PID제어기를 설계하여 실험해 보았다. 현재 이동용 로봇들을 RF 모듈로 제어하고 있어 제어거리에 제한을 받고 있다. 이를 해결할수 있는 가장 좋은 방법으로는 인터넷을 이용하는 방법 일것이다. Java 플랫폼 기반으로 인터넷을 통해서 실시간으로 전달되는 속도지령에 대해서 원거리에 위치한 DC 전동기를 제어하고 전동기의 속도를 웹상에서 모니터링 할 수 있게하였다. 본 연구에서는 두 대의 전동기에 동일한 속도 지령을 내렸을때와 하나의 전동기에 속도지령을 내리고 다른 하나의 전동기는 속도지령을 받은 전동기의 현재 속도를 목표값으로 추종하는 두가지 조건에 대해서 실험하였다.

## II. PID제어기에 의한 DC 전동기 제어

### 1. 제어시스템의 구성

PID 속도 제어 시스템의 구성을 그림 1에 나타내었다. DC 전동기의 입력은 전압이므로 PWM, 즉 펄스 폭의 변조에 의해 전동기에 인가되는 전압을 조절한다. 80C196KC에는 PWM 출력이 3개가 있으므로 그 중에서 PWM1, PWM2를 사용하였다. 회전에 따라 출력되는 엔코더의 펄스를 HSI에 인가하여 회전수를 구하였다. 제어주기는 TIMER2 인터럽트를 사용하여 5msec 마다 PID제어를 수행토록 하였다.

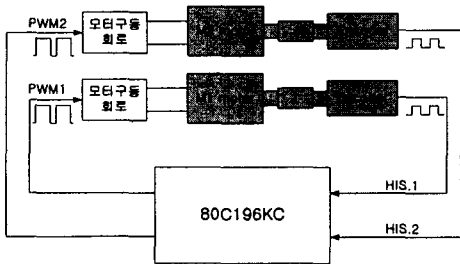


그림 1 제어시스템의 구성

### 2. PID 제어기 설계

본 논문에서 DC 전동기의 모델은 속도제어를 위한 플랜트로 입력되는 전압에 대한 회전수의 관계로 구하였다. 80C196KC로 제어를 수행할 때 플랜트에 인가되는 전압은 PWM값이고 회전수는 입력되는 펄스수이다. 전달함수  $G(s)$ 는 (1)과 같이 나타낼수 있다<sup>[1]</sup>.

$$G(s) = \frac{C(s)}{R(s)} = \frac{1}{K} \frac{1}{\tau s + 1} \quad (1)$$

그림 2와 3은 전동기 M1과 M2의 단위계단 응답을 나타내었다. 전동기의 시정수  $\tau$ 는 M1이 43msec M2는 34msec 이므로 2% 정착시간은  $4\tau$ 로 172msec 와 136msec가 되며 1차 시스템으로 모델링 하였으므로 극점이 실수축 상에 존재한다.

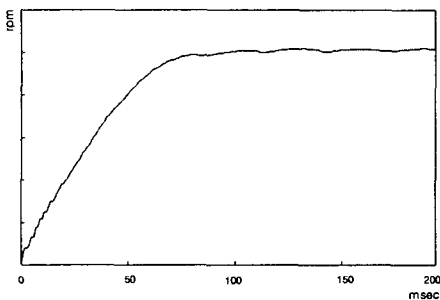


그림 2 M1 전동기 단위계단 응답

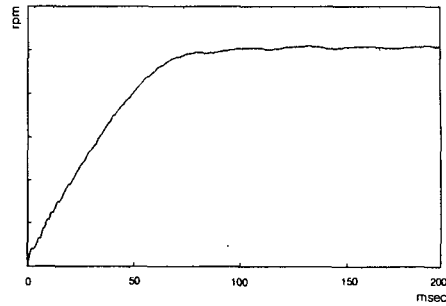


그림 3 M2 전동기 단위계단 응답

PID 제어기를 설계하기 위해서 먼저 전체 시스템의 극점을 설정한다. DC 전동기와 PID 제어기가 합해진 전체 시스템은 2차 시스템이므로 두 극점만 설정하면 된다. 극점 배치를 위해서 감쇠비  $\zeta(\cos \theta)$  0.8, 고유진동수  $w_n$ 은 100으로 설정하였다. 2%기준 정착시간을 사용할 경우 정착시간은 시스템 시정수의 약 4배이므로 (2)와 같다.

$$t_s = \frac{4}{\zeta w_n} = \frac{4}{0.8 \times 100} = 50 \text{ msec} \quad (2)$$

따라서, 전체시스템의 극점은 (3)과 같이 위치된다.

$$p_1, p_2 = -\zeta w_n \pm \sqrt{1 - \zeta^2} = -80 \pm j60 \quad (3)$$

1차 플랜트에 그림 4와 같이 PID제어기를 추가했을 때 전체 시스템의 전달함수는 (4)와 같다.

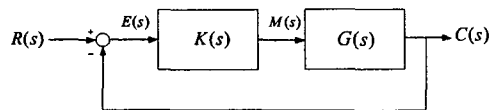


그림 4 PID 제어시스템

$$\frac{C(s)}{R(s)} = \frac{K(s)G(s)}{1 + K(s)G(s)} \quad (4)$$

$$\text{단, } K(s) = \frac{K_p s^2 + K_d s + K_i}{s}$$

$$G(s) = \frac{1}{K} \frac{1}{\tau s + 1}$$

극점은 특성 방정식  $1 + K(s)G(s) = 0$ 을 만족하는 근이다. 전체 시스템의 특성 방정식을 정리하면 (5)와 같다.

$$(K\tau + K_d)S^2 + (K + K_p)S + K_i = 0 \quad (5)$$

여기서  $K$ 와  $\tau$ 는 플랜트에서 구해지는 정수이며 설계시스템 극점  $p_1, p_2$ 를 만족하는 특성방정식을 풀

면  $s^2 + 160s + 10000 = 0$ 을 얻을 수 있고, 따라서  $(K\tau + K_d)S^2 + (K + K_p) + K_i = 0$ 과 일치시키면 (6)의 조건을 얻을 수 있다.

$$\frac{K + K_p}{K\tau + K_d} = 2\zeta\omega_n = 160,$$

$$\frac{K_i}{K\tau + K_d} = \omega_n^2 = 10000 \quad (6)$$

센서잡음에 둔감한 특성을 가지도록 센서잡음이 나타나는 주파수대가 대역폭 외부이기 위한 조건  $|k(j\omega)G(j\omega)|_{\omega=\omega_{noise}} < 1$ 을 만족하는  $K_d$ 의 범위를 구한다. 본 논문에서는  $K_d$ 를 0.01로 설정하고 (6)에서  $K_p, K_i$ 를 구하였다.

PID제어기를 80C196KC로 구현하기 위해서는 이산시간 시스템으로 나타내어야 한다. 이산시간 시스템에서의 PID 출력식은 (7)과 같다.  $K_p, K_i, T$ 를 대입하여 구한  $K'_p, K'_i, K'_d$ 는 실제 프로그램에서 사용하는 정수값이다.

$$m[n] = K'_p e[n] + K'_i \sum_{h=0}^n \frac{e[h] + e[h-1]}{2} + K'_d (e[n] - e[n-1])$$

$$\left( K'_p = K_p, K'_i = K_i T, K'_d = \frac{K_d}{T} \right) \quad (7)$$

### 3. 인터넷 원격제어 시스템

제안된 인터넷 원격제어 시스템은 크게 다섯 부분으로 이루어진다. 첫째는 인터넷사용이 가능한 원격지에서 전동기를 제어·모니터링 하는 Client Computer이다. Client Computer의 웹 브라우저를 통해서 직류전동기를 제어·모니터링 한다. 둘째, Client Computer 웹 브라우저의 제어·모니터링에 대한 요청을 처리하기 위한 웹서버이다. 셋째는 DC 전동기를 실제적으로 제어하기 위한 80C196KC, 넷째는 80C196KC로부터 DC 전동기의 회전수를 받아들이고 Client Computer에서 전달되는 속도지령을 전달하기 위한 제어프로그램이며, 마지막으로 웹서버와 제어프로그램을 연결하기 위한 데이터베이스로 구성하였다.

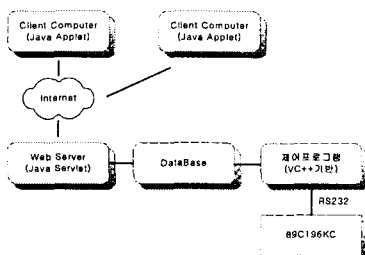


그림 5 인터넷 원격제어 시스템

Java Applet과 Servlet을 연동시켜 보안을 유지하면서 서버측에 연결된 전동기 M1, M2의 회전수를 실시간으로 Client Computer의 웹 브라우저에서 모니터링 할 수 있게 하였다.

### III. 실험 및 고찰

실험은 75[V], 200[W], 3000rpm DC 서보전동기 두 대로 수행하였으며, 전동기 드라이버는 LMD18200을 사용하였다. PID 정수값은 (6)에서 구해진 정수 값을 근간으로 반복 실험을 통하여 M1 전동기는  $K'_p = 3.76, K'_i = 1.25, K'_d = 0.7$ , M2는  $K'_p = 3.25, K'_i = 1.10, K'_d = 0.7$ 로 선정하였다.

그림 6과 그림 7은 400rpm에서의 스텝응답 특성을 나타낸 것이다. 제어주기는 5msec, 매 주기마다 RS-232통신을 이용하여 M1, M2 전동기의 회전수를 PC로 upload 하였다.

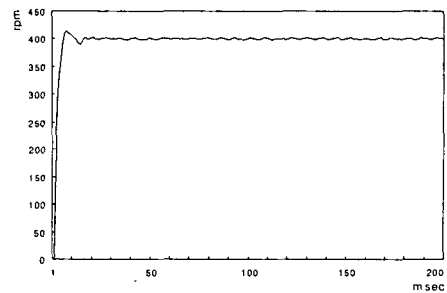


그림 6 M1 전동기의 스텝응답 특성

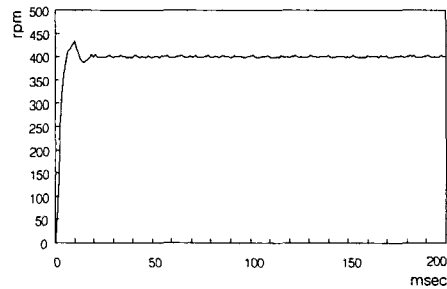


그림 7 M2 전동기의 스텝응답 특성

그림 8과 그림 9는 속도지령을 1초 간격으로 200에서 400으로 변경했을 때의 응답특성을 나타낸 것이다.

그림 8은 M1 전동기와 M2 전동기에 같은 속도지령을 준 경우이며 두 전동기의 동기화가 잘 이루어짐을 알 수 있다.

그림 9는 속도지령을 M1 전동기에 주고 M2 전동기는 M1의 현재 속도를 목표값으로 추종하게 한 경우이다. M1 전동기의 변속 부분에서 M2 전동기는 큰 오버슈트와 불안정한 응답특성을 보임

을 알 수 있다. 이것은 M1의 오버슈트와 M2의 오버슈트가 더해져서 일어나는 현상으로 생각된다.

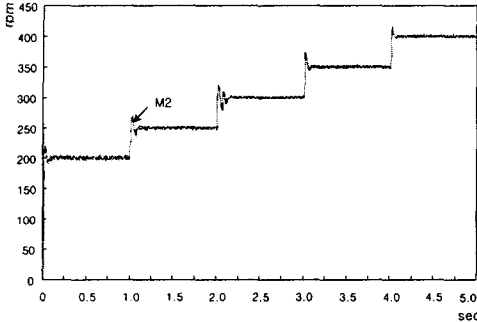


그림 8 M1, M2 동일 속도 지령시 응답특성

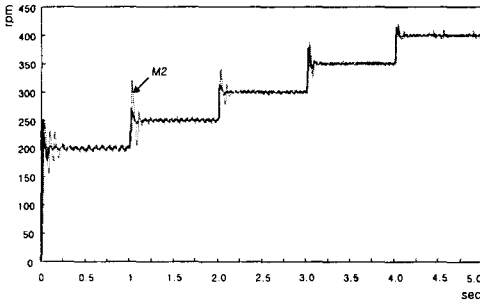


그림 9 M1속도 추종 지령시 응답특성

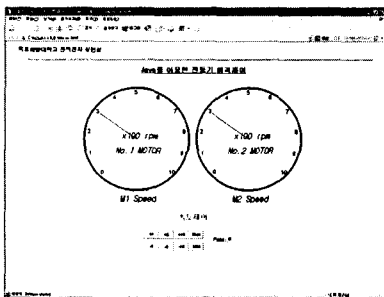


그림 10 제어·모니터링 화면

그림 10은 전동기 제어와 모니터링을 위한 Client Computer의 웹브라우저의 화면을 나타낸 것이다. 전동기의 속도는 버튼을 눌러 변경하게 하였다. 각각의 버튼을 누르면 버튼에 해당하는 속도지령값이 웹서버의 데이터베이스에 저장된다. 제어 프로그램은 데이터베이스 속도지령값의 변경여부를 계속 감시하여 변경된 값을 80C196KC에 전달하여 M1, M2 전동기의 속도를 제어한다.

본 논문에서 제안한 인터넷 원격제어 시스템을 이용하여 인터넷과 연결된 장소의 Client Computer 웹브라우저에서 구현한 결과 오차발생 없이 정확한 제어결과를 얻을 수 있었으며 M1, M2 전동기의 회전수를 실시간으로 모니터링 할 수 있었다.

#### IV. 결론

본 연구는 인터넷을 통해서 실시간으로 전달되는 속도지령에 대해서 원격리에 위치한 DC 전동기 동기화 제어 실험을 수행한 결과 다음과 같은 결론을 얻을 수 있었다.

- 1) 두 대의 전동기에 동일한 속도지령을 준 경우에는 만족할 만한 동기 특성을 나타냈지만, 속도지령을 M1 전동기에만 주고 M2 전동기는 M1 전동기의 현재 속도를 목표값으로 추종하게 한 경우는 큰 오버슈트와 불안정한 응답특성을 나타냈다.
- 2) 제안한 Java 플랫폼 기반의 인터넷을 원격제어 시스템을 통하여 원격리에 위치한 DC 전동기를 제어한 결과 오차발생 없이 정확한 제어결과를 얻을 수 있었고 전동기의 속도를 웹브라우저에서 모니터링 할 수 있었다.

#### 참고 문헌

- [1] 이장명 외3인, "80C196KC를 이용한 제어시스템설계", 북두출판사, 2003
- [2] Oboe. R, Fiorini.P, "Adeign and Control Environment for Internet-Based Tele robotics" The International journal of robotics research, vol 4, pp. 433~449, 1998
- [3] Richaed c.Dorf, "Modern Control Systems", Addison-Wesley Publishig company, 1993