

# Ethernet을 이용한 DC MOTOR 원격 제어시스템 개발

## Development of DC MOTOR Remote Control System using Ethernet

이 재 호\*, 김 이 철\*, 정 준 흥\*, 박 기 현\*  
(Jeaho Lee, Yicheal Kim, Joonhoung Jung and Kiheun Park)

\* 성균관대학교 정보통신공학부(전화:(031)290-7175, 팩스:(031)290-7165, E-mail : ljhhb@hanmail.net)

**Abstract** : Recently, the study on the design of NCS(Network Control System) using Ethernet is being rapidly progressed. NCS can be extensively applied in manufacturing automation, office automation, home automation, remote control and ect. A merit of NCS on based Ethernet is to make good use of advanced Internet environment and to apply a application of abundant TCP/IP upper layer to NCS.

The purpose of this paper is to control a speed of DC-motor using NCS on based Ethernet. The control system is divided into a server part and a client part. A server transfers a value of reference speed of a DC-motor. A client receives a output signal of DC-motor and a reference input obtained from the server. A client computes a error of two signals and then makes a control input. The control input is transferred to a actuator. In this pater, A controller uses a classical control using a general feedback. In this paper, a viewpoint is to compare performance of NCS with performance of a classical control and to analyze the cause of that.

**Keywords** : NCS(Network Control System), Ethernet, TCP/IP, DC-motor

### I. 서 론

최근 네트워크 기술의 비약적인 발전은 제어공학 분야에도 많은 영향을 주어 네트워크 제어시스템(NCS, Network Control System)에 대한 연구가 활발히 진행되고 있다. 중앙집중형 제어방식(Centralized Control System)은 제어기에 집중되는 과도한 트래픽으로 인해 시스템 성능이 저하되며 수 많은 장치와 중앙의 제어기를 일대일로 연결해야 하기 때문에 선로 설치·추가비용이 많이 든다. 이를 보완하기 위해, 플랜트에 산재한 수많은 부 시스템(Subsystem)을 독립적으로 제어할 수 있는 분산형 제어방식(Distributed Control System)이 도입되었으며, 현재 산업 전반에 걸쳐 널리 사용되고 있다. 과거에는 느린 전송속도로 인한 트래픽 증가와 패킷 손실 등 때문에 제어시스템에 네트워크를 적용하는 것을 기피하였으나, 현재 네트워크 성능이 눈부시게 발전함에 따라 많은 분야에 걸쳐 네트워크 제어 시스템이 이용되고 있는 실정이다. 네트워크 제어시스템 구조를 분산형 제어시스템에 도입할 경우 가장 기대되는 효과는 원격제어가 용이하다는 점과 이로 인해 인력 및 시간을 절약할 수 있으므로 시스템 관리비용을 감소할 수 있다는 점이다. 네트워크 제어시스템 구축에 주로 이용되는 프로토콜에는 CAN(Control Area Network), Ethernet(TCP/IP), ControlNet 등이 있다[1],[2].

본 논문에서는 단일침으로 구현된 TCP/IP 프로토콜을 이용하여 Ethernet 상에서 DC-motor를 실시간 제어할 수 있는 네트워크 제어시스템을 구축하였다. 네트워크 제어시스템은 크게 기준입력을 송신하는 서버노드와 이를 수신하여 DC-Motor의 속도제어를 수행하는 클라이언트노드로 구성하였다. 네트워크 제어시스템에 필요한 하드웨어 및 제어기는 직접 설계/제작하였으며 이를 이용하여 네트워크 제어시스템의 실험용 통신시스템을 구축하였다. 그리고 일반적인 피드백 제어시스템의 시뮬레이션 결과와 네트워크를 통한 제어 시스템의 실험결과를 비교, 분석하였다.

### II. 시스템 구성

#### 1. 하드웨어 구성

본 논문에서 제작한 하드웨어는 클라이언트부로서 제어부, 네트워크부, 통신부, 메모리부, 전원부 그리고 액츄에이터부로 구분할 수 있다. 그림 1과 2는 전체 하드웨어의 구성 사진이다. 마이크로 컨트롤러는 80C196KC를 사용하였고[3], 네트워크부는 링크계층의 Ethernet Interface로 RTL8201L(Realteck)을 사용하였으며, 네트워크 계층과 전송계층의 구현은 TCP/IP 프로토콜을 하드웨어적으로 구현한 i2chip W3100(Wiznet)을 사용하였다[4].

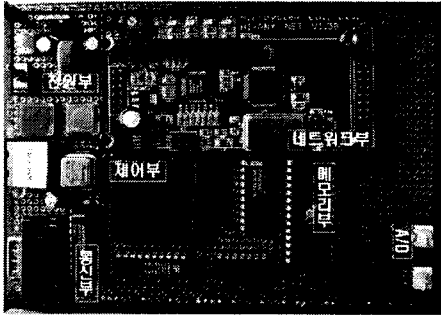


그림 1 클라이언트 노드  
Fig. 1 Client Node

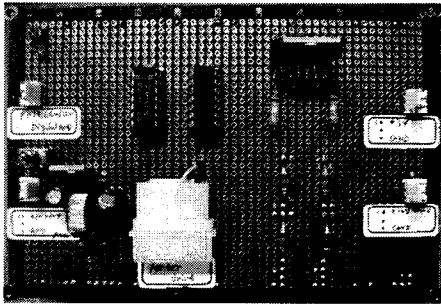


그림 2 액추에이터 노드  
Fig. 2 Actuator Node

여기서, 80C196KC는 크게 두 가지 역할을 담당한다. 첫째, 클라이언트쪽 PC와 80C196KC간에 직렬통신을 이용하여 Network부에서 통신에 필요한 IP주소, Subnetmask, Gateway 그리고 MAC주소를 Serial EEPROM에 저장하는 역할을 하며, 둘째, 서버로부터 기준신호를 입력받아 DC-motor를 제어하는데 필요한 제어입력을 계산하는 역할을 한다.

## 2. 전체 네트워크 구성

그림 3은 본 연구에서 Ethernet을 이용하여 DC모터 원격제어 시스템을 제어하기 위해 구성된 전체 시스템의 블록 선도이다.

그림 3과 같이 구성된 전체 시스템은 크게 서버부와 클라이언트 부로 나눌 수 있다. 서버에서는 기준신호를 클라이언트로 송신하는 역할을 담당하고 있으며, 동일한 Ethernet 상에 접속된 컴퓨터나 기타 장치들도 클라이언트 IP주소를 통해 클라이언트에 접속할 수 있다. 서버부의 소켓 프로그램은 Visual C++ 6.0을 사용하여 구현하였다[5][6].

클라이언트의 네트워크부는 Ethernet을 통해 서버에서 보낸 기준신호를 수신한다. 속도센서(타코미터, Tachometer)로부터 측정된 출력신호는 AD변환을 통해

디지털 값으로 변환된다. 마이크로 컨트롤러에서는 이 두 신호에 대한 오차를 계산하고, 이 오차신호는 단위 변환을 통하여 기계적인 값을 전기적인 값으로 변환한다. 이 변환된 신호는 프로그램으로 구현된 이산 PI 제어기를 거쳐 제어입력을 만들어 내며, 이 입력값에 의해 PWM 신호의 듀티비를 조정함으로써 모터 속도를 제어하게 된다[3].

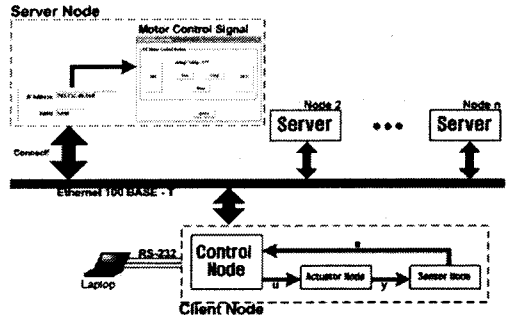


그림 3 전체 시스템의 블록 선도  
Fig. 3 Block diagram of total system

## III. 시스템 모델

직류 전동기에 대한 일반적인 모델은 그림 4와 같다.

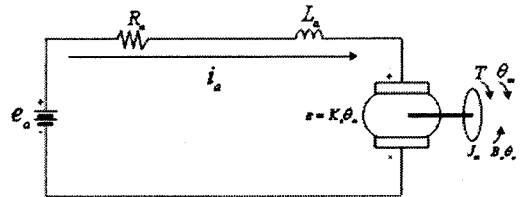


그림 4 직류전동기의 일반적인 모델  
Fig. 4 A general model of DC motor

그림 4로부터 직류전동기의 토크 방정식과 전기 방정식의 유도는 식(1) 과 식(2)와 같다.

$$J_m \ddot{\theta}_m + B_m \dot{\theta}_m = K_t i_a \quad (1)$$

$$L_a \frac{di_a}{dt} + R_a i_a = e_a - K_b \dot{\theta}_m \quad (2)$$

식(1)과 식(2)를 이용하여 상태방정식과 출력방정식을 유도하면 식(3)과 식(4)와 같다.

$$\begin{bmatrix} \frac{di_a}{dt} \\ \frac{dw_m}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_i}{J_m} & -\frac{B_m}{J_m} \end{bmatrix} \begin{bmatrix} i_a \\ w_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} e_a \quad (3)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_a \\ w_m \end{bmatrix} \quad (4)$$

식(3)과 식(4)로부터 입력전압과 모터의 각속도에 대한 전달함수를 식(5)와 같이 유도할 수 있다.

$$\frac{w_m(s)}{E_a(s)} = \frac{K_i}{L_a J_m s^2 + (R_a J_m + B_m L_a) s + (K_b K_i + R_a B_m)} \quad (5)$$

DC 모터의 파라미터 값은 다음과 같다.

Torque constant  $K_i = 17.66 \times 10^{-2} (Nm/A)$

Back-emf constant  $K_b = 17.67 \times 10^{-2} (V \text{ sec/rad})$

Armature resistance  $R_a = 2.0 (\Omega)$

Armature inductance  $L_a = 3.0 \times 10^{-3} (H)$

Rotor inertia  $J_m = 3.6173 \times 10^{-4} (Kg m^2)$

Viscous-friction coefficient

$B_m = 4.91 \times 10^{-2} (Nm \cdot sec/rad)$

식(5)에 위 파라미터를 대입하면 다음 식(6)과 같다.

$$\frac{w_m(s)}{E_a(s)} = \frac{17.66 \times 10^{-2}}{10.852 \times 10^{-7} s^2 + 8.819 \times 10^{-4} s + 12.925 \times 10^{-2}} \quad (6)$$

#### IV. 시뮬레이션 및 실험

##### 1. 시뮬레이션

시뮬레이션은 식(6)의 DC모터 모델을 대상으로 Matlab의 Simulink를 이용하여 실행하였다. Simulink를 이용한 시뮬레이션 블록 선도는 그림 5와 같다.

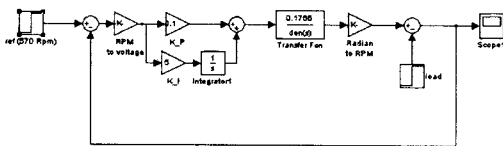


그림 5 시뮬레이션 블록 선도

Fig. 5 Block diagram of simulation

기준신호는 570[RPM]으로 시뮬레이션 상에서 1초 후에 인가하였으며, 2초 후에 부하가 걸리게 된다. PI 제어기 이득은 시행착오적으로  $K_P=0.1$ ,  $K_I=5$ 로 구하였다. 그림 6은 이 때의 시뮬레이션 출력을 나타내고 있으며, 이로부터 설계된 PI 제어기를 이용할 경우 부하가 가해진 후 0.5초 이내에 기준신호에 다시 도달하게 됨을 알 수 있다.

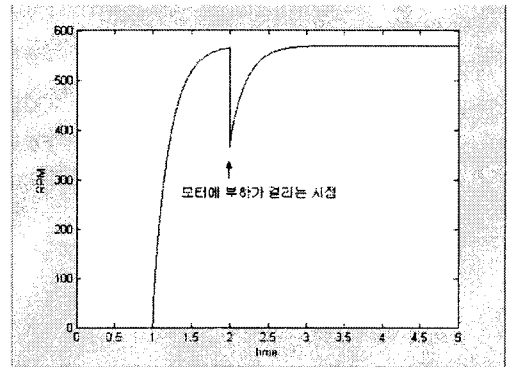


그림 6 부하에 대한 시뮬레이션 그래프

Fig. 6 Simulation Graph for a load

##### 2. 실험

그림 7은 클라이언트에 구현된 이산 PI 제어기의 흐름도이다.

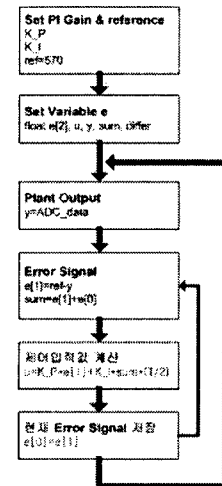


그림 7 제어기 프로그램 흐름도

Fig. 7 Flow chart of control program

DC-motor의 출력은 속도센서를 통해 구해지므로

출력값을 A/D 변환을 한 후 적절한 상수이득을 곱하여 RPM 단위로 환산하였다. 이 후, 기준신호인 570[RPM]과 변환된 측정출력 사이의 오차신호를 계산하고 이산 PI 제어를 통해 제어입력을 계산하였다.

그림 8은 이 때의 출력파형을 관찰한 사진이다.

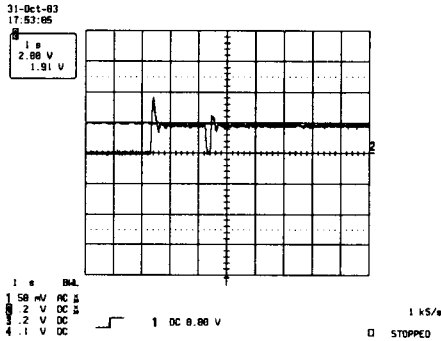


그림 8 실제 출력파형  
Fig. 8 Actual output

회전속도가 1500[RPM]일 경우 속도센서의 출력값은 5[V]이며 회전속도가 0[RPM]일 때는 속도센서의 출력값도 0[V]가 된다. 회전속도와 속도센서의 출력값은 선형적이므로, <그림 8>에서 속도센서의 출력값이 약 1.9[V]이기 때문에 실제 모터의 회전속도는 약 570[RPM]임을 알 수 있다. 즉, 서버에서 송신하는 기준신호인 570[RPM]을 클라이언트의 DC-motor가 정확히 추종하고 있음을 확인할 수 있다. 이로부터, 본 논문에서 구현된 Ethernet을 이용한 DC-motor의 속도제어 성능이 만족스럽다는 것을 입증할 수 있다.

## V. 결 론

본 논문에서는 현재 가장 많이 활용하고 있는 Ethernet 기반의 TCP/IP 프로토콜을 이용하여 네트워크 제어 시스템을 구현하였다. 실험결과로부터, 시뮬레이션 상에서 구해진 DC-motor의 속도제어 성능과 실제 구현한 네트워크 제어시스템의 DC-motor 속도제어 성능이 유사함을 확인할 수 있었다. 본 논문에서 구축한 Ethernet을 이용한 DC-motor 속도제어 시스템은 향후 일반적인 네트워크 제어시스템 설계에 직접 적용할 수 있는 실용적인 기법이 될 수 있으며, 공장자동화, 사무자동화, 홈오토메이션, 원격제어 등 많은 분야에서 직접적으로 적용될 수 있을 것이다.

## 참고문헌

- [1] 정준홍, 최수영, 이종성, 박기현, "Network Control System의 안정도 분석에 관한 연구", 대한전기학회 하계학술대회논문집, pp.2231-2233, 7, 2002
- [2] 강신영, 김은주, 위석오, 김광현, 임영철, "인터넷 기반모터 원격제어 및 모니터링", Trans KIEE.Vol.50D, no.7 pp279-285, 2002
- [3] 차영배, MICROCONTROLLER 80196, 다다미디어, 1997.
- [4] Behrouz A. Forouzan, TCP/IP Protocol Suite, McGRAW.HILL, 2003.
- [5] 이상엽, Visual C++ Programming Bible, (주)영진출판사, 1998.
- [6] Bjarne Stroustrup, The C++ Programming Language, Addison Wesley, 1997.
- [7] 이원무, "CAN을 이용한 아라고 원판 시스템 제어", 성균관대학교 정보통신공학부 석사학위논문, 2002.
- [8] Chi-Tsong Chen, Linear System Theory and Design, OXFORD, 2002.
- [9] Benjamin C. Kuo, Farid Golnaraghi, Automatic Control System, WILEY, 2003.
- [10] Samuel P. Harbison, Guy L. Steele Jr, C A Reference Manual, Prentice Hall, 2002.