

CAN 통신을 이용한 다중 직류 모터 제어기 구현

김현성*, 권만오, 이건영
 광운대학교 전기공학과

An Implementation of the Controller for Multiple DC Motors Using CAN

Hyun Sung Kim*, Man Oh Kwon, Keon Young Yi
 Dept. of Electrical Eng. Kwangwoon University

Abstract - This paper presents the controller of multiple DC motors using the network. This controller has been built with 16-bits one chip microprocessor (87C196CA) which includes the integrated CAN serial communication and position control for two motors. Since only one microprocessor is needed, the proposed controller is not only cost effective but also powerful. The system is composed of one main controller, trajectory planner, and the other sub controller, position controller.

The main controller which has been built using Visual Basic programming on the Pentium PC, generates the trajectory and then transmits it to the sub controller. The trajectory transmitted from the PC will be processed by the sub controller. Two motors are controlled using the conventional position control, PID, to reach them the same target position but with different velocities at the same time. The communications between the main controller and sub controller is performed through the RS-232 or the CAN communication. The CAN would be safer and faster than serial communication network since it has non-destructive bitwise arbitration specification. In this paper, we consider the CAN communications generally and then show the usefulness of the proposed controller by demonstrating position control of two DC motors.

1. 서 론

오늘날의 제어시스템은 지능화와 대형화 그리고 보다 복잡한 추세로 발전하고 있으며, 이에 따라 지능화, 대형화, 복잡한 시스템을 통합하는 네트워크의 필요성이 더욱 더 중요하게 부각되고 있다. 복잡하고 대형화된 프로세스를 제어하기 위해 여러 가지 표준이 제정하기 위한 노력이 진행되었으며, 1980년대 초에 최초로 MAP(Manufacturing Automation Protocol)이 제정되었다. 이러한 시스템 통합화에 많이 사용되는 네트워크에는 Map을 포함해서 Minimap, Fieldbus 등이 있으며, Fieldbus에는 Profibus, CAN, Fip 등이 있다. 자동화된 생산 시스템이나, 통합환경에 쓰이는 통신 네트워크는 복잡하고 다량의 신호를 처리해야 하기 때문에 신호를 시스템의 환경에 맞도록 신속한 동작을 필요로 하며, 또한 여러 신호가 동시에 접근하더라도 충돌을 일으키지 않는 네트워크를 필요로 한다.[7] 그러나 PROFIBUS는 CAN의 전송속도 5K(bps)에서 1M(bps)보다 다소 느린 9.6(bps)에서 500(bps)까지 [6]이며, 각각의 컴퓨터에 제어기를 내장해야 하기 때문

에 CAN에 비해 효율성이 떨어진다. 7층의 MAP은 일반적으로 실시간 통신을 하기에 너무 느리고 비용이 비싸기 때문에 MAP의 추가적인 형태인 MINIMAP이 소개되었다. MINIMAP은 MAP의 하위 두개 층을 채택하고 나머지 층은 하나의 층으로 혼합해서 층의 수를 감소하지만, 잘 알려진 방법은 아니다.[2] CAN은 단일 칩(one chip) 마이크로 프로세서에 내장되어 보다 효율적이며, 높은 수준의 신뢰를 바탕으로 실시간 분산제어에 효과적으로 사용되고 있다. CAN의 응용영역은 자동차, 철도, 항공 등 고속의 네트워크에서 저가의 다중 집적 영역 등에 걸쳐 사용되고 있다. 기존의 CAN의 사용은 통신선과 통신보드를 사용하여 CAN통신망을 구축한 후 제어보드를 연결하였으며, 이 통신보드와 제어보드 사이에는 규약을 일치시키기 위한 추가적인 채널이 요구되었다. 그러나 16비트 마이크로 프로세서에 CAN기능을 내장하면 하나의 보드로 제어기능과 통신기능을 동시에 만족시킬 수 있어 효율성과 경제성을 높일 수 있다.

본 논문에서는 각 모터의 기존제적 산출 역할을 하는 PC와 CAN기능을 내장한 16비트 제어기인 인텔의 87C196CA 마이크로 프로세서와 LMD-18200으로 구성된 두 개의 PWM 드라이버를 갖는 위치 제어기를 구성하여 동일시간에 서로 다른 목표 위치로 두 모터를 제어한다. 이 실험을 통하여 제시한 다중 모터제어의 효율성을 확인하고, 현재 개발중인 보행 로봇의 자세제어(두 개 이상의 위치 제어기를 PC에 연결하여 여러 개의 모터를 제어함)에 활용할 예정이다.

2. CAN(Controller Area Network)

CAN은 차량 통신용 네트워크를 위해 독일의 Bosch에 의해 1980년 개발되었으며, ISO의 국제표준(ISO11898)(1)으로 지정되었다. CAN은 데이터를 serial로 전송하기 때문에 보다 적은 wiring을 통해 복잡한 전자장비들로 이루어진 자동차 내부의 통신 네트워크로 사용되고 있다. 각종 네트워크CAN은 ISO/OSI 참조 모델 7계층 중 하위 2계층인, 물리계층과 데이터 링크 계층을 사용한다. 물리계층은 실제적인 신호전송과 Bit Timing, Bit Encoding, 그리고 동기화(Synchronization)등을 정의하고 있다. Data link는 메시지 프레임(Message frame), 중재(Arbitration), 에러검출(Error detection) 및 신호화 역할을 하며 CAN protocol의 핵심을 이루고 있는 MAC(Medium Access Control) sublayer와 LLC(Logical Link Control) sublayer로 나누어진다.

CAN의 메시지 전송은 contention based addressing을 이용하며, 그것은 identifier와 data를 동시에 보내며 MSB부터 차례로 메시지를 비교하여 dominant(0)인 메시지가 선택되고 recessive(1)인 메시지는 탈락되어 우선 순위에 따른 메시지 선택이 이루어진다. 주소를 지정하지 않고 contention based addressing을 이용하기 때문에 새로운 노드를 추가하는데 필요한 별도의 hardware나 software를 구성하지

않아도 되기 때문에 높은 유연성을 발휘한다. CAN은 고유한 메시지가 부여되는 bitwise arbitration을 이용한 Identifier가 메시지의 우선 순위를 결정하기 때문에, 여러 메시지가 동시에 버스로 접근 시 충돌현상과 지연현상을 막아 실시간을 요구하는 분산제어에 효과적으로 적용할 수 있다.

CAN은 버스에 입력과 출력(TXCAN, RXCAN)두 선으로 이루어져 신호 당 하나의 전선(point-to-point)을 사용할 때 드는 wiring을 효과적으로 감소시켜 wiring에 드는 비용을 줄일 수 있다. 또한 CAN의 뛰어난 에러감지 능력과 교정능력 때문에 높은 안전성을 가지고 있기 때문에, 전기적 장애(noise)가 많은 환경에 적합하고 자동차뿐만 아니라, 철도와 항공, 산업 자동화 등에 걸쳐 널리 활용되고 있다. 메시지 프레임은 Data Frame, Remote Frame, Error Frame, Overload Frame의 4개의 프레임으로 이루어져 있으며, 11비트 길이의 Identifier를 갖는 Standard Frames과 29비트 Identifier를 갖는 Extended Frames로 구분된다.[3][4][5]

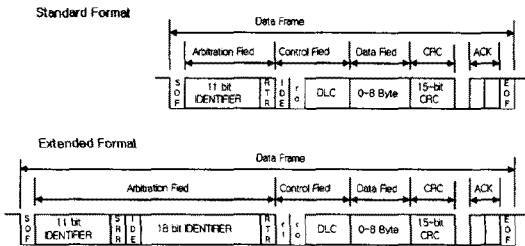


그림 1. CAN 메시지 프레임
Fig. 1. CAN Message Frames

3. 네트워크를 이용한 다중모터 제어

제어기와 PC사이의 네트워크는 직렬통신인 RS-232/CAN을 이용하여 통신을 할 수 있다. 본 논문에서는 한 대의 PC와 한 대의 제어기를 사용하여 두 직류모터가 같은 시간에 다른 속도로 정해진 위치에 도달하는 것을 실험을 통해 보였다.

본 논문에서는 궤적생성기(trajjectory planner)가 생성한 데이터를 RS-232직렬 통신을 통해 제어기에 전송하고, 이 데이터에 의해 두 개의 모터는 궤적생성기가 만들어낸 위치에 따라 제어된다.

본 연구를 통해 우리는 RS-232대신에 CAN Bus를 이용한 네트워크를 구성하고 현재의 시스템에 제어기와 모터를 추가하여 다중 직류 모터제어를 보다 경제적이 며 편리하게 제어할 수 있음을 예측할 수 있다...

3.1 전체 시스템의 구성

전체 시스템의 구성은 그림 2와 같다. PC에서 생성된 궤적은 RS-232를 통해 제어기에 전송되고, 이 데이터는 다시 모터에 전송된다. 모터를 제어하는 제어기의 CPU는 CAN통신 기능을 내장한 인텔의 16비트 마이크로 프로세서인 87C196CA를 채택하였다.

이 제어기는 두 개의 PWM 출력신호와 직류모터의 궤적을 얻기 위한 엔코더 입력카운터가 있다. CPU에서는 고속의 제어로 PWM을 출력하기 위한 EPA(Event Processor Array)를 사용해 CPU의 오버헤드를 줄였다. 엔코더는 펄스 값을 읽어내기 위해 DC모터에 연결된 HP의 HCTL-2020을 사용했다. PC는 궤적 생성을 위해 사용되었고, 비주얼 베이직으로 프로그래밍한 윈도우 환경의 궤적 생성 함수를 만들어 주는 역할을 한다.

본 시스템에서는 주 제어기인 PC로부터 모터제어기로 데이터를 전송하고 있다. 다시 모터제어기로부터 나오는

신호를 RS-232를 통해 PC가 받고, 그 데이터에 따라 궤적을 형성한다. 또한 주 제어기의 명령에 따라 모터가 생성하는 궤적을 엔코더로부터 읽어들이어 모니터 상에 디스플레이 한다.

87C196CA는 유연한 인터럽트 조정체계인 인터럽트와 PTS(Peripheral Transaction server)을 가지고 있다. 본 논문에서는 문제점을 가지고 있는 PTS방식 대신에 소프트웨어 방식의 인터럽트를 이용해 PWM신호를 발생해 모터를 제어했다.

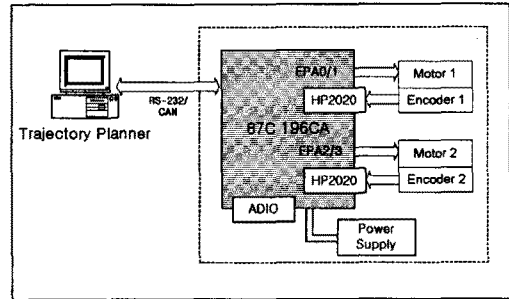


그림 2. 시스템의 전체 구성도
Fig. 2. The block diagram of the system

3.2 소프트웨어의 구성

제어할 두 개의 직류모터의 속도를 연속하도록 엔코더에서 출력된 단위시간당 펄스의 개수를 카운팅하고 모터에 가해지는 전압을 계산한다. PWM 신호 파형의 on time을 결정하면 펄스 폭이 변경된 PWM 신호 파형이 발생하게 돼 모터의 속도가 바뀌게 된다.

본 논문에서 사용한 87C196CA를 사용하여 PWM 신호를 발생하기 위해서는 CPU의 PWM 출력포트를 초기화하고 EPA의 레지스터들을 compare mode로 설정하며 핀의 출력형태는 토글모드로 설정한다.

이곳에서 EPA time 레지스터를 이용해 on time을 설정하고, EPA의 기능을 결정하기 위한 COMP 레지스터 값을 토글모드로 설정한다.

시스템의 클럭과 타이머의 카운팅의 방향을 정해줘서 타이머레지스터를 1step당 2(ms)의 시간으로 초기화한다. EPA mask를 설정한 후 인터럽트가 동작 가능하도록 초기화를 해서 EPA가 실행하게 되면 PWM 출력 값을 update한 후에 COMP인터럽트 동작을 설정한다. COMP인터럽트 서비스 동작 중 PWM 펄스 폭을 설정할 경우 다시 EPA인터럽트 서비스를 시작 해서 PWM 신호 파형을 발생한다.

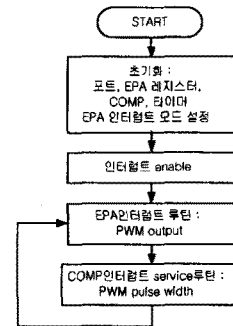


그림 3. 제어기 동작 순서도
Fig. 3. Flow char of controller

직류모터의 위치 데이터는 외부로 출력하고 모터의 이동목표 위치 값을 입력받을 수 있도록 프로그래밍구현을 하였다. 위치를 제어하기 위해서 쓰인 제어기로 PID제

여기를 이용했으며, 모터의 회전시작과 끝 부분에 속도를 가속, 등속, 감속구간으로 나누어서 모터의 회전을 부드럽게 이루어지도록 하였다. 가속구간은 1 step이 10(ms)로 지정하여 10 step 동안 가속을 한다. 등속구간 30 step, 감속구간 10 step의 간격으로 속도제어 루틴을 사용하였다. [8]

PC는 비주얼 베이직을 이용해 제어기와 인터페이싱했으며, RS-232통신 에뮬레이터와 타이머 컨트롤을 이용하여 모터의 시작과 정지 위치, 타이머의 시간 간격을 설정하고, 궤적의 값을 제어기에 입력하여 그 값을 다시 출력할 수 있도록 하였다. 통신 에뮬레이터는 접속포트를 선택해서 아스키화일을 전송할 수 있고, 궤적을 생성하여 전송할 수 있도록 구성되었다..

4. 실험

본 실험에서 두 개의 모터에 입력되는 샘플링시간은 500(Hz)이고, PWM파형의 주기는 20(kHz)로 설정하였고 PWM펄스 출력비율을 최대 90(%)에서 최저 10(%)로 설정할 수 있다. 모터가 1회전할 때 엔코더에서 발생하는 펄스 수는 2채널 400펄스이다. 두 개의 모터는 Maxon사의 5(W)모터와 15(W)모터를 사용하였다. 각각의 모터에 달린 엔코더는 2채널, 100(count/turn) 와 3채널, 500(count/turn)이고 4.4:1인 기어헤드와 71:1인 기어헤드가 장착되어 있다. 따라서 모터 1이 1회전하는데 필요한 펄스는 1760 펄스가 되고 모터 2가 1회전하는데 필요한 펄스는 142000펄스를 요구하게 된다.

모터의 위치를 결정하기 위해서 궤적생성기인 PC에서 한 지점까지 일정한 값을 동일한 시간 간격으로 입력한 후 다음 두 개의 모터가 지정된 위치에 동시에 도착함을 보이고 출력 값을 그래프로 디스플레이하여 두 모터의 위치 값을 비교하였다

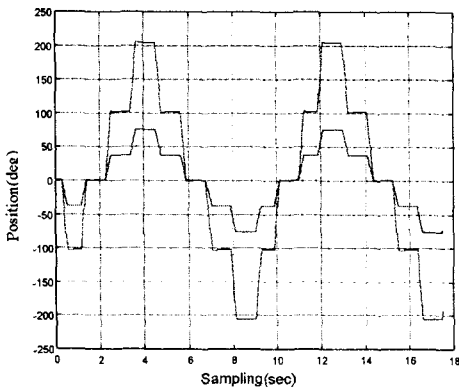


그림 4. 두 모터의 위치 궤적

Fig 4. The trajectory of two motors

그림 4는 실험에서 두 개의 드라이브에 일정한 간격의 위치제어 신호를 주어 각기 다른 속도로 주어진 구간 내에서 어느 시점에 두 개의 모터가 거리와 속도의 비가 일정하게 이동하게 됨을 보였다. 이동시 처음 가속구간과 목표점 도달시 감속구간은 2(ms/step)의 각 10 step에 걸쳐 진행되고 전체 샘플링 타임은 50 step으로 설정되었다. 현재 시스템에서 실험한 결과는 이동주기를 280(ms)로 했을 때 두 모터의 목표점에 동시에 도달함을 알 수 있다.

그림 5의 결과는 이동주기를 100ms로 하였고, 두 모터의 궤적에서 목표도달 지점에서 오버슈트가 나타났다. 이런 문제점은 PID제어기의 gain값을 조정하여 줄이는 방법과 가속시간과 감속시간을 늘려주어 안정한 상태로 도달하도록 할 수 있다. 한편 전체주기가 길어질수록 두

모터가 도달지점에 동시에 정확한 도달을 하였다.

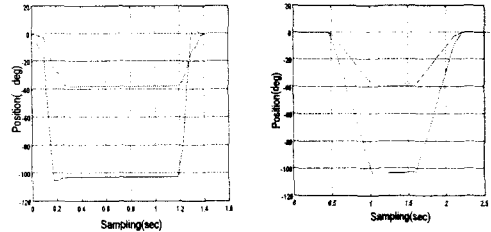


그림 5. 100ms와 560ms주기의 궤적 비교

Fig. 5. The trajectory of 100ms and 560ms

5. 결론

본 논문에서는 CAN통신 기능을 갖는 단일 마이크로 프로세서를 이용하여 두 개의 직류 모터를 동시에 제어할 수 있는 제어기를 개발하였다. 이 제어기는 시리얼 통신 또는 CAN통신을 이용하여 주 컴퓨터로부터 각 모터의 기준궤적을 입력받을 수 있어, 또 다른 제어기를 네트워크로 연결하면 다중 모터의 위치제어를 쉽게 구현할 수 있다. 현재 각 모터의 기준궤적은 PC에서 비주얼 프로그램에 의하여 만들어지며, 시리얼 통신을 통하여 위치제어기로 전달된다. 전달된 각 모터의 기준궤적이 PID제어기에 의하여 효과적으로 제어됨을 보여주었으며, 향후 두 개 이상의 위치제어기를 CAN통신으로 연결하여 보행로봇 자세제어를 위한 다중모터 제어기로 확장 적용할 예정이다.

(참고 문헌)

- [1] ISO 11898 : Road vehicles, Interchange of digital information - controller Area Network for high-speed communication, 1992
- [2] K.G. Shin, "Real-time communications in a computer-controlled workcell," IEEE Trans. Robot. Automat, vol. 7, pp. 105-113, Feb. 1991
- [3] CAN Specification Version 2.0, Robert Bosch GmbH, 1991
- [4] K. M. Zuberi and K. G. Shin, "Scheduling messages on CAN for real-time CIM applications," IEEE Trans. Robotics and Automation, vol. 13, no. 2 pp. 310-314, 1997
- [5] 최치권, 조영조, 유범재, 오상록, 윤태웅, "고속전철을 위한 CAN(Controller Area Network)의 응용층 설계 및 성능 해석," KACC, October pp. 883-866, 1997
- [6] 박세용, 김동준, 김갑일, 김대원, 김홍석, 이호길, "필드버스를 이용한 로봇셀의 구현과 자료운용," KACC, October pp. 1249-1252, 1996
- [7] 김기암, 홍승호, 김지용, 고성준, "생산자동화 시스템에서 필드버스 네트워크 구축 기술 연구," KACC, October pp. 1020-1023, 1996
- [8] David M. Auslander and Cheng H. Tham, "Real-time Software for Control," prentice hall, pp. 93-95, pp. 110-111, 1990