論 文
51D-2-3

CAN 통신을 이용한 다중모터 위치제어기 구현

An Implementation of The Position Controller for Multiple Motors Using CAN

李 健 寧*
(Keon Young Yi)

Abstract - This paper presents a controller for the multiple DC motors using the CAN (Controller Area Network). The controller has a benefit of reducing the cable connections and making the controller boards compact through the network including expansibility. CAN, among the field buses, is a serial communication methodology which has the physical layer and the data link layer in the ISO's OSI (Open System Interconnect) 7 layered reference model. It provides the user with many powerful features including multi-master functionality and the ability to broadcast / multicast telegrams. When we use a microprocessor chip embedding the CAN function, the system becomes more economical and reliable to react shortly in the data transmission. The controller, we proposed, is composed of two main controllers and a sub controller, which have built with a one-chip microprocessor having CAN function. The sub controller is plugged into the Pentium PC to perform a CAN communication, and connected to the main controllers via the CAN. Main controllers are responsible for controlling two motors respectively. Totally four motors, actuators for the biped robot in our laboratory, are controlled in the experiment. We show that the four motors are controlled properly to actuate the biped robot through the network in real time.

Key Words : CAN, DC Motor control, Distributed control, Communication based control, Real-time control

1. 서 론

산업현장에서 쓰이는 각종 장치가 고기능화 됨에 따라 이를 위한 제어장치도 점점 복잡해지고 지능화 되어왔다. 이러한 장치들을 연계하여 제어하기 위하여 네트워크 기반의 실시간 제어의 중요성도 주목 받아왔으며, 이에 관한 연구 및 응용이 다양한 제어시스템에 적용되어왔다. 1980년대 초에는 각종 제어기의 효율적인 통신 연계를 위하여 네트워크 표준화, 즉 Minimap과 Fieldbus 같은 MAP(Manufacturing Automation Protocol)에 관한 많은 노력도 이루어졌다. 필드버스는 산업 자동화나 다른 실시간 제어 응용 분야에 적합하도록 구성된 지역통신망(LAN)의 특수분야로 간주할 수 있다. 이는 센서, 구동장치 및 제어기 등 현장 장치들을 기존의 일대일(Point-to-Point) 연결방식의 통신보다 저가이며 간단하게 연결이 가능한 특징이 있다. 이와 관련하여 CAN, Fieldbus[1], MAP[2] 등을 이용한 여러 구성방식이 소개되었으며, 이 중 CAN은 1980년대에 실시간 응용을 목표로 Robert Bosch에 위해 개발되었다.

CAN은 저가이며 사용이 쉬워서 자동차 산업 및 자동화 분야에서 널리 이용되고 있다. 이는 두 선만으로 여러 장치를 모두 연결하여 정보교환이 가능하므로 선 연결의 복잡성을 획기적으로 단순화시킬 수 있기 때문이다. 특히, Siemens, Philips, NEC, National Semiconductors, Motorola, Hitachi, Fujitsu 및Intel 등 대부분의 반도체 제조업체들이 CAN 관련 부품을 생산하여 가격 또한 저렴하며, OSI 모델 중 최하위 두 계층만으로 통신 구현이 가능하여 시스템이 간단하며, 통신지연이 적어 여타의 MAP 시스템에 비하여 우수한 특성을 갖고 있다. 이러한 이유로 많은 센서와 구동장치를 필요로 하는 이동로봇[3], 자동차[4,5], 농작물 관리 시스템[6], 자동제어 장치 시스템[7], 생화학 공정[8] 및 에너지 절약형 주택[9] 등 다양한 분야에서 CAN의 활용이 널리 확산되고 있다.

본 연구에서도 CAN 기능을 갖는 단일 칩 마이크로프로세서(87C196CA)를 이용하여 경제적이고 효율적인 제어기를 설계 및 제작하여 연구실에서 개발중인 보행로봇[10]에 적용하고자 하는데 로봇에 CAN을 적용한 예는 본 연구 외에도 다른 연구[11]에서도 볼 수 있다. 그러나, 대부분의 연구는 상용 통신장치나 제어기를 사용하고 있어 본 연구의 최종 목표인 이동로봇에 적용하기에는 그 크기 및 비용 면에서 적합하지 않다. 따라서 본 연구에서는 CAN 기능을 갖는 별도의 제어보드를 설계 제작하여 다음과 같은 제어시스템을 구성하였다. 즉, 두 개의 모터 제어 및 4개의 위치센서를 감지하기 위한 보드 (이하 주 제어기) 2개와 모터의 궤적 산출을 위한 궤적 발생기가 실행되는 펜티엄 컴퓨터, 그리고 이들을 연계하기 위한 보조 CAN 보드(이하 보조 제어기)로 구성하였으며, 이들의 역할 및 상호 연결은 본문에서 설명한다. 이러한 제어기 구성으로 앞서 언급한 바와 같이 로봇

* 正 會 員 : 光云大 電子情報大 電氣工學科 副教授
接受日字 : 2002年 1月 14日
最終完了 : 2002年 2月 17日

제어장치의 소형화 및 각종 선 연결을 간략화 하였으며, 통신망을 통하여 모터의 실시간 제어 여부를 고찰해 봄으로써 실시간 제어가 요구되는 로봇제어로의 적용여부를 검토하며, 네트워크를 이용한 분산제어를 효율적 운영방안을 알아본다.

다음절에서는 본 연구에서 채택한 통신 방식인 CAN의 소개 및 시간지연 요소를 검토하며, 3절에서는 구현한 통신 기반 위치제어기에 대한 설명을 하고, 4절 및 마지막 절에서는 실험 결과 및 결론을 기술한다.

2. CAN(Controller Area Network)

ISO 국제규격(ISO11898)[13]에 등록되어 있는 CAN은 1980년에 독일의 자동차 회사인 Bosch에 의하여 개발[12]되었다. 이는 직렬통신을 이용하여 정보교환을 하기 때문에 전기 결선이 많은 자동차 부품간의 통신 연결을 획기적으로 줄여줄 수 있다. 이 통신기법은 OSI(Open System Interconnect) 7 계층 구조 중 최하위 두 계층인 물리 층(Physical Layer)과 데이터 연계 층(Data Link Layer) 만을 사용한다.

물리 층은 실제 신호가 어떻게 전달되는지를 정의하는 층으로, 데이터 비트의 타이밍(Bit Timing), 비트 부호화(Bit Encoding) 및 동기화(Synchronization)를 다루고 있으나 송수신 장치의 특성은 정의하지 않아 응용 목적에 적절한 송신 매체 선택의 폭을 넓혀주고 있다. CAN 버스는 입력선(RXCAN)과 출력선(TXCAN)으로만 구성되어 장치간의 연결선을 최소화 할 수 있다. 뿐만 아니라 데이터 연계 층의 오류 검사와 수정 기능으로 데이터 신뢰도가 높아 자동차, 항공기, 철도 차량 및 산업 자동화 부문과 같이 전기적 잡음이 심한 환경에서도 널리 이용된다.

데이터 연계 층은 데이터 묶음 제어(Framing), 우선순위 결정(Arbitration), 오류 검사(Error Checking), 오류보고 및 격리 등 전송 프로토콜에 관한 정의를 하고 있다. 이 프로토콜의 커널은 매체 접근 제어(Medium Access Control)의 보조 층과 논리 연계 제어(Logical Link Control)의 보조 층으로 구성된다. CAN 프로토콜에서는 식별자(Identifier)와 데이터를 동시에 보내는 메시지 경합기반(Contention based) 어드레싱 기법이 이용되어 노드를 추가할 경우에도 별도의 하드웨어나 소프트웨어 추가가 필요치 않다. 이 방법에서는 최상위 비트를 기준으로 우세 메시지(논리 0)를 보낸 쪽이 열세 메시지(논리 1)에 우위를 갖는 방식으로 버스 사용 순위를 정하게 된다. 즉, 여러 개의 장치에서 동시에 메시지를 보내는 경우 각각의 메시지는 버스에서 충돌을 하게 되는데, 일정시간 후에 식별자에 의한 비트단위의 우위 결정되고 (논리 0과 1이 충돌하면 0으로 결정됨) 각 장치들은 충돌된 식별자를 확인하여 자신의 식별자가 아니면 메시지 전송을 중단하는 방식이다. 이 방식은 CAN 칩에서 하드웨어에 의해 실시간으로 이루어진다.

실제 데이터 전송은 메시지를 통하여 이루어지는데, 이 메시지는 데이터 프레임(Data frame), 원격 프레임(Remote frame), 오류 프레임(Error frame), 그리고 과부하 프레임(Over load frame)의 4가지 형태로 구분된다. 각각의 프레임 내부는 그림 1과 같이 식별자 길이에 따라 두 가지 형태로 구성된다. 표준 형식은(그림 1의 위쪽) 11-비트의 식별자가 사용되어 최대 2048개의 장치 수용 이 가능하다. 확장 형식(그림1의 아래쪽)의 경우는 29-비트의 식별자가 이용된다. 데이터 프레임과 원격 프레임만이 표준 및 확장 형식 사용이 가능하며, 한번에 보낼 수 있는 최대 데이터 량은 8 byte로 한정된다. 단, 데이터 전송을 요구하는 원격 프레임은 데이터 영역이 존재하지 않는다. 이 장치 구분을 위한 식별자는 경우에 따라서 동일 장치에서 여러 개를 사용할 수 있다. 즉, 동일 장치에서 여러 메시지를 수용할 수 도 있는데, 이 경우 각 메시지를 별도의 명령으로 처리하여 명령 해석 시간을 단축시키며 8-byte의 데이터를 모두 파라미터로 사용하여 정보처리 성능을 개선할 수 있다.

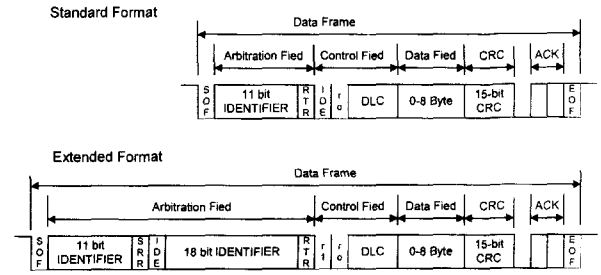

그림 1 CAN 메시지 프레임

Fig. 1 CAN Message Frames

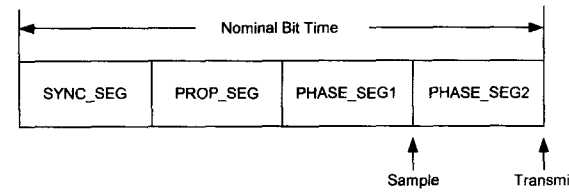

그림 2 CAN 프로토콜에서 정의된 비트 시간

Fig. 2 A bit time as specified by the CAN protocol

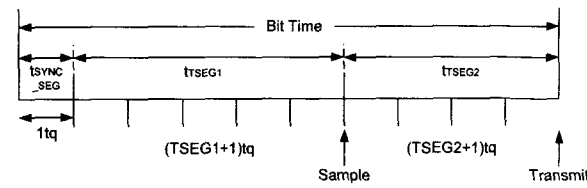

그림 3 CAN 제어기에 구현된 비트 시간

Fig. 3 A bit time as implemented in the CAN controller

CAN 프로토콜에서는 그림 2와 같이 동기 지연(SYNC_SEG), 전송 지연(PROP_SEG), 위상 지연 1(PHASE_SEG1), 그리고 위상 지연 2(PHASE_SEG2)의 4단계로 구분된 비트 전송 시간 구획을 정의하고 있다. 그림 3은 본 연구에서 채택한 87C196CA에서 구현된 비트타임을 보여주고 있다. 그림에서 1tq는 500[ns]가 되도록 설정하였으며, 전체 케이블 길이가 7[m]임으로 전송속도를 낮추어서 (TSEG1=12, TSEG2=1) 125[Kbps]로 하였다. 그러나, 전송거리와 속도에 관한 연구[12]를 참조하면 1[Mbps]의 경우

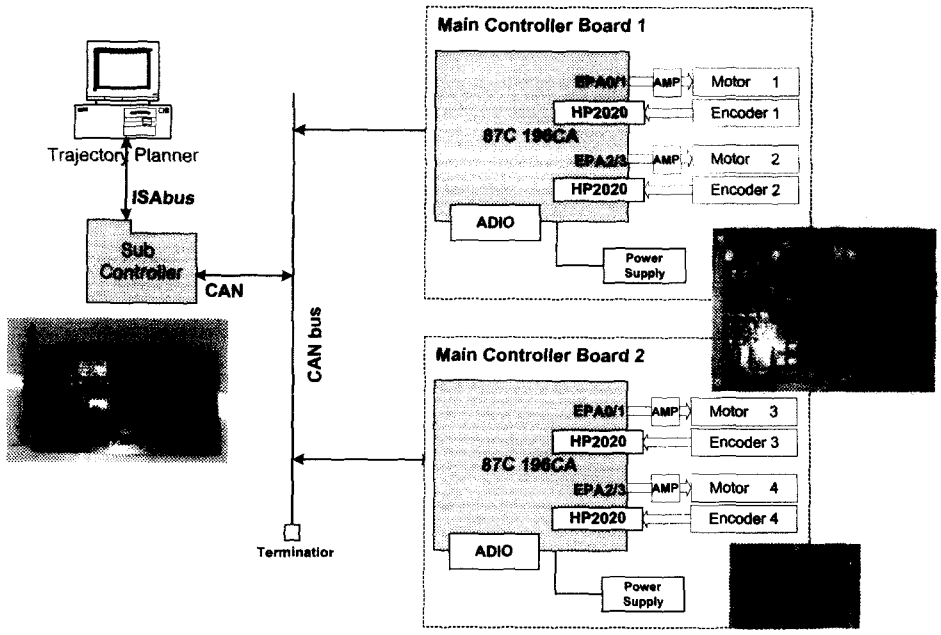

그림 4 시스템 구성도

Fig. 4 The configuration of the system

40[m]까지, 40[Kbps]의 경우 1100[m]까지 전송됨을 감안할 때 좀더 빠른 전송도 가능하다. 실제 데이터 전송을 관찰해 보면 오류 발생을 방지하기 위한 스터프(Stuff) 비트(5개의 연속되는 비트가 전송되면 자동적으로 반대 비트를 삽입하여 보내는 기능)로 인하여 보내는 데이터 유형에 따라 전체적인 속도가 저하됨을 알 수 있는데, 본 연구에서 이용된 CAN의 실질적인 전송속도는 약 100[Kbps]로 간주해도 무방하며, 이는 본 연구의 명령어 전송량 및 상태 관측 량을 감안할 때 충분한 속도이다.

3. 다중 모터 제어

본 연구에서는 하나의 컴퓨터에서 통신망을 통하여 연결되는 4개의 모터를 동시에 제어하는 것을 목표로 하고 있다. 전체 시스템은 1개의 주 컴퓨터, 1개의 보조 제어기, 2개의 주 제어기, 그리고 4개의 모터 및 센서로 구성되며, 이 절에서는 이들 각각의 역할 및 연관관계를 설명한다.

3.1 제어 시스템의 구성

전체 시스템 구성은 그림 4와 같다. 그림의 컴퓨터는 윈도우환경에서 Visual Basic^{tm}으로 작성된 궤적 발생기를 실행하게된다. 이는 사용자의 명령을 받아 각 모터의 기준궤적을 산출하여 보조제어기를 통하여 주제어기로 보내며, 아울러 주제어기로부터 센서 정보를 받아서 화면에 표시하게 된다. 궤적 발생기에서 생성된 모터의 기준 궤적은 컴퓨터에 내장된 보조 제어기를 통하여 주 제어기로 전달된다. 이 보조 제어기는 컴퓨터에서 주제어기별로 구분하여 직접 명령을 전달하는 경우 실시간이 보장되지 않는 문제(윈도우의 동작상태에 따라 두 제어기에 명령어 도달 시간이 달라 모터 동작을 동기 시킬 수 없음)를 방지하는 역할을 한다. 궤적 발생기와 보조 제어기와의 통신은 ISA 버스에 위치한 I/O 포트를 통하여 이루어진다. 보조 제어기에서 수신한 기준 궤적은 다시 제어 대상인 4개의 모터를 제어하는 두 개의 주제어기 단위로 분리된 후CAN 버스를 통하여 전달된다.

그림의 주 제어기 및 보조 제어기(사진 참조)는 인텔사에서 제작한 16MHz로 동작하는 CAN을 내장한 87C196CA 16-bit 마이크로프로세서를 사용하여 제작하였다. 주 제어기는 두 개의 모터를 구동하기 위한 PWM 신호 발생기(LMD18200을 통하여 증폭됨)와 엔코더 신호를 받아 모터의 위치를 인식하기 위한 포트 (HCTL-2020을 통하여 펄스를 계수 함)를 갖고 있다. 이 제어기에 대한 세부사항은 지면관계상 생략하였다. 모터 구동을 위한 고속의 PWM신호를 구현은 CPU 내부 기능인 EPA(Event Processor Array)를 사용하였다. CPU 설명서에 의하면 PTS(Peripheral Transaction Server) 모드로 프로그램 하면 CPU의 부담을 최소화 하고 좀더 고속의 PWM신호를 만들 수 있다고 하였으나 인터럽트 처리과정에서 다소의 문제점이 있어 비정상적인 출력을 발생하는 경우가 있어 본 연구에서는 채택하지 않았다. 이에 관한 자세한 사항은 인텔사의 기술보고(CPU Bug List)를 참조하면 된다.

3.2 궤적 발생기 및 GUI

사용자의 명령을 입력받아 모터의 기준궤적을 만들어내는 Visual Basic^{tm}으로 작성된 그래픽 기반 사용자 접속 프로그램은 그림 5와 같다. 모터의 기준위치 및 이동속도를 정하기 위한 선택 창이 화면 맨 위쪽에 있으며, 이어서 기준궤적과 현재 위치가 표시되는 창이 있다. 또한 모터의 전원인가

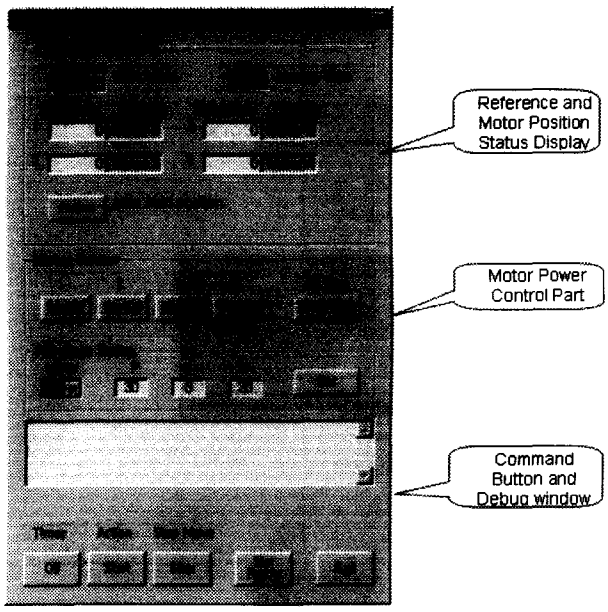

그림 5 궤적 발생기를 위한 사용자 접속

Fig. 5 User interface for the trajectory planner

또는 비상시 차단을 위한 단추와, 각 모터의 PID 이득을 설정하기 위한 창이 있다. 화면 하단부에는 수동으로 명령을 입력하거나 각종 디버그 메시지를 출력하기 위한 창과 자동으로 명령을 전송하거나 수신된 상태 데이터를 파일로 기록하기 위한 단추가 있다.

제어기로 전송되는 모터 이동 명령어는 "P 300 100 300 -100 400" 의 형식을 취한다. "P" 는 이동 명령을 나타내며, 이어서 표시된 4개의 숫자는 사용자가 미리 정한 모터의 위치 값(PPx, x = 1...8)을 나타내고 (10배수된 각도 값), 마지막 숫자는 이동 속도(400 스텝, 1스텝은 2ms)를 나타낸다. 명령어로는 상태 요구를 위한 "S", 모터의 전원 제어를 위한 "M" 이 있으나 자세한 설명은 생략하였다.

3.3 제어기 프로그램 구성

앞서 설명한 궤적발생기 외에도 80C196CA로 구성된 보조 제어기 및 주 제어기 프로그램이 있다. 컴퓨터와 모터 제어를 담당하는 주제어기의 통신 중계를 담당하는 보조 제어기는 앞서 설명한 모터 이동 명령을 주 제어기 단위로 분리하여 전송하는 역할을 하며, 각각의 주 제어기에서 올라오는 상태 데이터를 모터 순서대로 재정리하여 "S 300 100 300 -100 1234" 의 형태로 전송한다. 여기서 각 숫자는 모터의 위치(10배수 된 각도)를, 마지막 숫자는 현재의 스텝번호(시간에 해당)를 의미한다. 이 방식은 새로운 주제어기가 추가되어도 보조 제어기 프로그램의 일부(제어기 숫자)만 변경하면 되는 장점을 갖는다. 주제어기와의 통신 데이터는 "P1 300 100 400", "P2 300 -100 400", "S1 300 100", "S2 300 -100" 의 형태를 취한다. 이 보조 제어기는 컴퓨터와 주제어기 사이의 통신 연계뿐만 아니라 실시간 보장이라는 중요한 역할을 한다. 컴퓨터에서 주제어기 별로 명령을 직접 보내는 경우 윈도우의 동작상태에 따라 두 제어기에 명령어 전송 시간이 달라 모터 동작을 동기 시킬 수 없는 문제가 발생한다. 따라서, 보조제어기는 컴퓨터로부터 통합된 주 제어기로의 명령을 받아 정해진 순번대로 전송하여 항상 일정한 주기의 명령어 전달을 보장하는 것이다.

주 제어기의 구성은 그림 6과 같이 1개의 메인 루틴과 3개의 인터럽트 서비스 루틴으로 이루어진다. 메인 루틴은 CAN 인터럽트 루틴의 도움을 받아CAN 버스를 통한 보조 제어기와의 통신을 담당한다. 즉, 위치명령을 수신하고 각 모터의 위치를 보내게 된다. CAN 인터럽트 루틴은 CAN 버스의 상태에 따라 버스상의 메시지를 받아들이거나 전송하는 역할을 하며, EPA 및 COMP 인터럽트 루틴은 각각 PWM 신호 발생(50?s) 및 제어입력 산출을 근거로 한 펄스 주기 변경(모터로 공급되는 전류 량을 조절하기 위한 제어 명령 반영: 2ms)하는 임무를 수행한다.

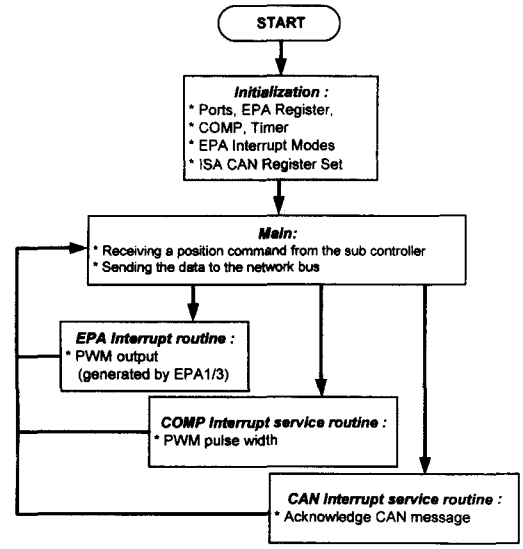

그림 6 주제어기의 제어 흐름도

Fig. 6 Control flow of the main controller

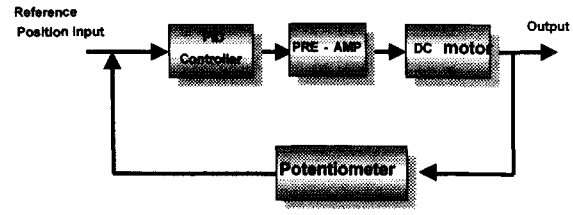

그림 7 주제어기 구조

Fig. 7 The controller structure

모터의 제어 명령 산출방법은 PID 제어기로 구성하였다. 컴퓨터로부터의 제어 명령은 모터의 최종 위치와 이동시간만 주어지기 때문에 주 제어기에서는 그림 8과 같이 각 스텝단위의 위치명령으로 재구성하여 모터가 목표 위치까지 부드럽게 이동하도록 한다. 즉, 주어진 전체 스텝에서 초기 10 스텝(20 ms)은 가속을, 마지막 10 스텝은 감속을, 나머지는 등속 운동으로 구분하여 모터의 위치를 제어한다.

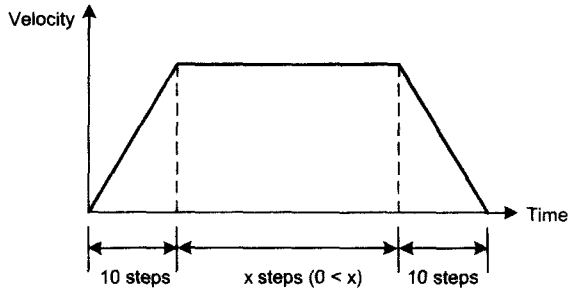

그림 8 모터의 속도 명령

Fig. 8 Velocity command for motor

현 시점에서는 연속되는 운동 명령을 부드럽게 하기 위한 Look-Ahead 알고리즘을 구현하지 않았다. 이는 본 연구의 목표가 통신으로 연결된 보행로봇의 모터를 동기 시켜 제어하는 것으로 운동 특성상 컴퓨터의 이동 명령이 구분동작(명령과 명령 사이에 순간 정지 상태가 있음)으로 이루어지기 때문이며, 가속 및 감속 구간의 스텝 수는 모터의 움직임을 육안으로 관측하여 실험적으로 정하였다.

4. 실 험

구성된 통신기반 제어기로 4개의 20[W]의 맥슨 모터를 제어하였다. 모터 구동을 위한 PWM 신호는 20[KHz], 제어기 샘플링 시간은 2 ms이다. 위치 인식을 위한 엔코더는 1회전 당 500 펄스 (이를 4채배하여 2000펄스로 사용함)를 만들어 내며, 66:1의 기어를 통하여 모터에 부착되어있다. 따라서 실제 위치 분해능은 0.003도 이다. 그러나 실제 보내지는 각도 명령이나 측정된 각도는 10배수된 값으로 약 0.1도의 정밀도를 보이고 있다.

실험방법은 사용자가 정의한 8개의 기준점(PP1~PP8)을 두 제어기로 순차적으로 보내는 방법을 취하였다. 두 제어기의 동기 된 제어특성을 관찰하기 위하여 동일한 명령을 두 제어기에 보냈으며, 실험 결과는 지면 관계상 제어기 당 하나의 모터에 대한 내용만을 표시하였다.

그림 9(a)는 주 제어기 1의 모터 1의 궤적을 나타낸 것이며, 그림 9(b)는 주 제어기 2의 모터 3의 궤적을 표시한 것이다. 두 그림에서 앞선 점선은 컴퓨터에서 내려보낸 기준 궤적(점으로 표시된 부분이 명령어 전송 값임) 이며, 뒤의 실선이 실제 측정된 각도 값이다. 전송 명령은 0~5~10~15도로 이동하였다가 다시 15~10~5~0도로 이동하는 형태로 한 것이다. 그림에 각 구간의 스텝은 300~125~200~150, 300~150~125~200으로 한 주기가 총 1550 스텝(3.1초)이다.

명령이 내려간 후 제어가 시작되어 측정된 값이 400 스텝(0.8초)후에 표시되고 있다. 그림 9(a), (b)모두에서 보여주듯이 기준 궤적과는 달리 측정값에서 제어에 의한 가속 및 감속구간이 표시됨을 알 수 있다. 이 결과에서 주목할 것은 이 두 결과가 얼마나 부합되는가 하는 것으로 그림 9(c)에 표시하였다. 그림에서 점선은 모터 1과 3의 위치 차이를, 실선은 모터 2와 4의 위치 차를 표시한 것이다. 그림에서 알 수 있듯이 두 오차 모두 0.1도 이내에 있어 두 제어기가 잘 동기 되고 있음을 알 수 있다.

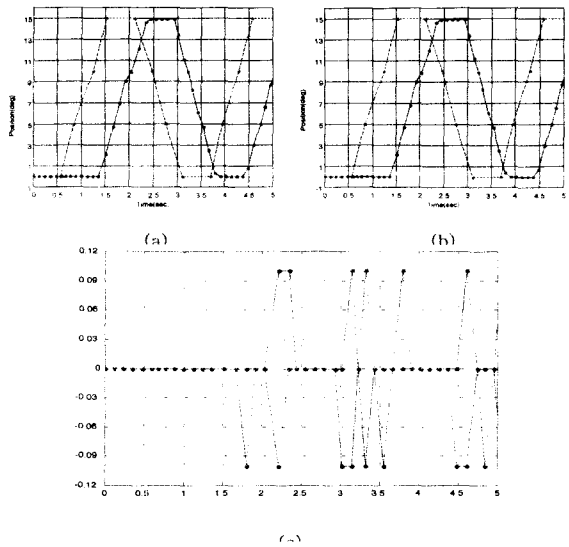

그림 9 모터 1과 3의 궤적(PP1 = 600ms)

Fig. 9 The trajectory of 1st and 3rd motor(PP1 = 600ms)

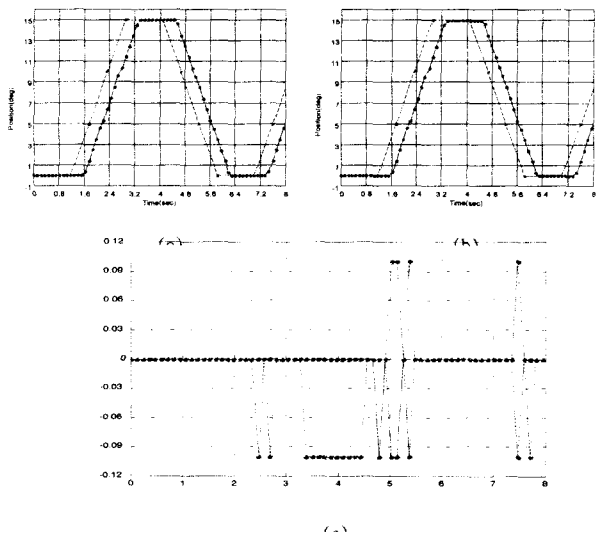

그림 10 모터 1과 3의 궤적(PP1 = 1200ms)

Fig. 10 The trajectory of 1st and 3rd motor(PP1 = 1200ms)

그림 10 은 앞에서와 같은 조건에서 이등 속도를 늦춘 경우의 실험 결과를 보여주고 있다. 이 경우에도 앞에서와 같이 두 제어기가 잘 동기 되고 있음을 알 수 있다. 다만 상단의 정지구간에서 백래쉬로 인하여 오차가 일정기간(3.3초~4.4초) 유지되고 있으나 이들 모두 0.1도 이내의 작은 값을 나타낸다.

5. 결 론

본 연구에서는 네트워크를 통하여 4개의 모터를 동기 시켜 제어하는 제어기를 구현하였다. 제어기 구현을 위하여 CAN기능을 갖는 단일 칩 마이크로프로세서를 이용하여 DC 모터를 제어하기 위한 주 제어기를 제작하였으며, 이 제어기

를 통신망을 통하여 실시간으로 제어할 수 있도록 컴퓨터에 장착되는 CAN 통신용 보조 제어기 또한 제작하였다. 모터의 위치제어는 가감속 궤적 발생기능과 PID 제어를 이용하여 제어기를 구현하였고, 그래픽 기반의 궤적 발생기를 컴퓨터에 구현하여 현재 개발중인 보행로봇으로 확장적용을 용이하게 하였다.

실험 결과를 토대로 연구결과를 분석해보면, CAN네트워크로 연결된 두 제어기의 모터가 동기 되어 잘 제어됨을 알 수 있었다. 이 연구 결과는 현재 개발중인 보행로봇으로 확장 적용할 예정이다.

감사의 글

본 연구는 2000년도 광운대학교 교내학술연구비 지원에 의하여 이루어진 연구로서, 관계부처에 감사드립니다.

참 고 문 헌

[1] 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

[2] David M. Auslander and Cheng H. Tham, "Real-time Software for Control," prentice hall, pp. 93-95, 1990

[3] M. Wargui and A. Rachid, "Application of Controller Area Network to Mobile Robots," Electrotechnical Conference, MELECON '96, 8th Meditarranean, Vol 1, pp. 205 -207, 1996

[4] Xing Wang, Chen Huiyan and Huarong Ding, "The application of controller area network on vehicle," Proceeding IEEE Vehicle Electronics Conference, (IVEC '99), Vol.1, pp. 455-458, 1999

[5] N. Papadoglou and E. Stipidis, "Short message service link for automatic vehicle location reporting," Electronics Letters, Vol.35, pp. 876-877, 1999

[6] C.M.J. Alves-Serodio, J.L. Monteiro and C.A.C. Couto, "An integrated network for agricultural management applications," Proceeding IEEE ISIE '98, International Symposium on Vol.2, pp. 679-683, 1998

[7] G. Cena and A. Valenzano, "An improved CAN fieldbus for industrial applications," Industrial Electronics, IEEE Transactions on Vol.44, pp. 553-564, 1997

[8] K. Gollmer and C. Posten, "Flexible automation of a biotechnological pilot-plant with real time network," Proceeding IEEE Conference on Control Applications, Vol.3, pp. 1941-1946, 1994

[9] J.C. Metrolho, C.M.J.A. Serodio and C.A.C.M Couto, "CAN based actuation system for greenhouse control" Proceeding IEEE ISIE '99. International Symposium on Vol. 2, pp 945-950, 1999

[10] K.Y. Yi and Yuan F. Zheng, "Biped Locomotion by Reduced Ankle Power," Proceeding IEEE ICRA 96, Vol. 1, pp. 584-589, 1996

[11] K.G. Shin, "Real-time communications in a computer-controlled workcell." IEEE Trans. Robotics and Automation, vol. 7, pp. 105-113, Feb. 1991

[12] CAN Specification Version 2.0, Robert Bosch GmbH, 1991

[13] ISO 11898 : Road vehicles, Interchange of digital information - Controller Area Network for high speed communication, 1992

[14] C. Serodio, J.B. Cunha, M. Cordeiro, A. Valente, R. Morais, P. Salgado, and C. Couto, "Mnet-DACS: Multi-level Network Data Acquisition and Control System," Industrial Electronics, 1997. ISIE '97., Proceedings of the IEEE International Symposium on Volume: 1, pp.39 43, 1997

저 자 소 개

이 건 영(李 健 寧)

1959년 8월 2일생.1982년 한양대 공대 전기공학과 졸업. 1984년 동 대학원 전기공학과 졸업(석사). 1993년 동 대학원 졸업(공박). 현재 광운대 전기공학과 부교수

Tel : 02) 940-5142

Fax : 02) 942-0107

E-mail : keonyi@daisy.gwu.ac.kr