

Distributed Control of a Two Axis Convey Table Using Real-time Micro-Kernel

李 健 寧*
(Keon Young Yi)

Abstract - In this paper, we propose a PC based distributed controller for a two axis convey table using real-time micro-kernel. PC, Windows program, gives an easy way to implement wealthy GUI and micro-kernel, $\mu\text{C}/\text{OS-II}$, provides a real-time capability to control devices. We built a real-time distributed control system using $\mu\text{C}/\text{OS-II}$ kernel which needs to process the tasks for two motors within the desired time to synchronize the motion. We used both semaphore and message mail box for synchronization. Unlike the previous study where we used step motors for the actuator of two axes convey table, we rebuilt the convey table with DC motors and the dedicated position servo which had built in out lab, and then we implemented a realtime distributed control system by putting the micro-kernel into between PC and position servo. Moreover we developed the PC based graphic user interfaces for generating planar drawing image control. Experimental results also presented to show the proposed control system is useful.

Key Words : 분산제어, DC 모터 제어, 마이크로 커널, PC 기반 제어기

1. 서 론

개인용 컴퓨터 혹은 산업용 PC를 이용한 각종 제어기 구현은 여러 분야에서 볼 수 있다[1,2]. 특히, 독자적인 처리능력을 갖는 각종 센서 모듈이나 구동 모듈이 사용자 접속기능을 제공하는 주 컴퓨터와 통신망을 통하여 연결되는 분산처리 방식이 최근 널리 이용되는 제어기 구성 형태이다. 그러나 이들은 실시간 동작을 보장해야 하는 문제로 인하여 실시간 지원이 가능한 Lonworks[3]이나 CAN[4] 등의 통신망과 별도의 운영체제를 이용하여야 하는 전문 기술을 요구하고 있다. 따라서 개인용 컴퓨터 운영체제인 윈도우가 다양한 사용자 접속 기능과 편리한 개발 환경을 제공하고 있지만 실시간 기능을 지원하지 않으므로 이를 제어기 구현에 적용한다는 것은 그리 간단한 것이 아니다. 일부 실험용 기구에서는 일반 윈도우 환경에서 실행되는 제어기를 제공하고 있으나 다양한 화면구성과 편리한 사용자 접속기능(GUI)에도 불구하고 그 성능과 신뢰성 측면에서 매우 낙후된 수준에 머물러있다.

실시간 커널은 일반 운영체제가 하던 작업 관리에 시간 개념을 더한 것으로 주어진 시간 내에 작업이 완료되는 것을 최우선 과제로 하여 동작하게 되는데, 시스템의 논리적인 정확성뿐만 아니라 시간적인 정확성이 맞지 않으면 심각한 결과를 초래하는 특징을 가진다. 따라서 커널은 설계된 각

각의 작업들이 원하는 시간 내에 이루어질 수 있도록 서비스를 제공하여야 한다. 실시간이 요구되는 시스템을 설계할 경우 실시간 커널을 이용함으로써 인터럽트 처리나 제어 알고리즘 수행시간 계산 등 실시간 보장을 위한 여러 가지 작업을 피할 수 있는 장점이 있다. 따라서 전체 시스템의 성능은 이 커널의 동작 속도와 효율성에 의존하게 되며, 적절한 타스크의 설계와 더불어 타스크 사이의 통신 방식이 중요한 설계 요인이 될 것이다.[5,6]

본 연구에서는 윈도우 환경에서 제공하는 편리한 개발 환경과 다양한 사용자 접속 기능을 실시간 분산 제어기 설계에 활용하는 예를 보여주고자 한다. 즉, 2축 반송 테이블을 대상으로 하여 Visual Basic으로 사용자 접속기능을 구성하고, 상용화된 저가의 제어보드(Am188ES)에 매우 저가인 선점형 실시간 커널($\mu\text{C}/\text{OS-II}$ [7])을 구현하여 실시간 분산처리 시스템을 구현한다. 실시간 커널의 각 타스크(제어에 필요한 독립된 작업) 들은 세마포어와 메일 박스를 활용하여 동기화 되며, 이를 이용하여 윈도우 프로그램에서 전달되는 사용자 명령을 실시간으로 관리하여 두 개의 모터가 상호 연관된 움직임을 하도록 한다.

앞선 연구에서는 2축 반송 테이블의 구동부로 스텝모터를 사용하였고 Am188ES 보드를 사용하여 스텝모터를 제어하였으나 DOS 모드 실행으로 편리한 사용자 접속기능 제공이 어려우며, 스텝모터의 정밀도 문제로 인하여 매끄러운 직선을 구현하지 못하였다[8]. 본 논문에서는 이러한 문제를 해결하기 위하여 스텝모터 대신에 DC 서보 모터를 사용하였고, Am188ES 보드에서 DC 모터 제어까지 담당하기에는 무리가 있어 87c196CA를 사용한 DC 모터 컨트롤 보드를 추가로 사용하였다. Am188ES 보드에서 다른 상용 컨트롤러 보드에서도 모터 동기화 문제를 해결할 수 있도록 PC에서

* 正 會 員 : 光云大學 電氣學科 副教授 · 工博
接受日字 : 2004年 1月 14日
最終完了 : 2004年 2月 10日

받은 이동 각도를 몇 개의 작은 각도로 분할하여 전송할 수 있는 기능을 추가하였다. PC 측의 사용자 인터페이스 프로그램은 Visual Basic을 사용하여 작성하였으며 반송 테이블의 궤적을 정의하고 기구학을 계산하는 기능을 수행하게 하였다.

2. 본 론

본 장에서는 전체 시스템의 구성과 실시간 처리를 위한 운영 시스템과 TASK, PC에서 실행되는 사용자 접속 프로그램, 반송테이블의 기구학 및 실험에 대하여 기술한다.

2.1 시스템 구성

시스템의 구성은 그림 1에서와 같이 마스터(Master)로 사용되는 PC와 슬레이브(Slave)로 사용되는 Am188ES 보드, 서보 모터 컨트롤러로 사용되는 87c196CA 보드, 반송 테이블의 기구부로 구성된다.

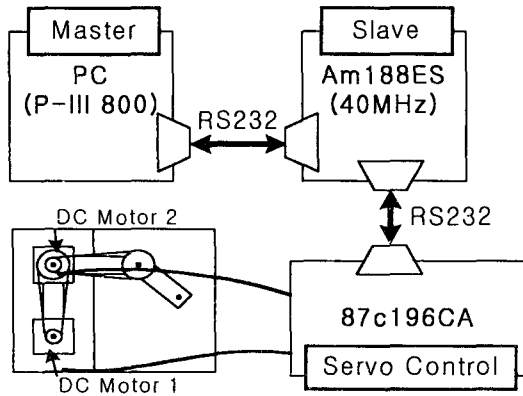


그림 1 실시간 마이크로 분산 제어 시스템의 구성도
Fig. 1 Diagram of real-time micro-distributed control system

마스터 제어기인 PC는 Visual Basic으로 제작한 사용자 인터페이스 프로그램을 이용하여 2축 반송 테이블의 이동을 위한 명령을 받아 기구학을 계산하여 각 모터의 이동각도를 구하여 슬레이브로 전달하며, 슬레이브로부터 각 구동부의 상태를 입력받아 사용자 화면에 표시해준다. 이 경우 전달하는 값이나 받아들이는 상태 값은 윈도우 프로그램 특성상 실시간 처리가 되지는 않는다. 물론 이전 연구[8]에서와 같이 마스터 PC에서 uC/OS-II 기반으로 시스템을 구현하여 실시간 처리가 되도록 할 수도 있으나 이는 DOS모드 실행을 전제로 해야 하며 이 경우 윈도우 환경의 GUI는 사용이 불가능하다. 이는 Windows 2000 이상의 OS의 경우 도스 자체를 에뮬레이션하고 있어 인터럽트 벡터를 수정한다 할지라도 실시간을 보장 받을 수 없음에 기인한다. 따라서 여기서는 PC 측에서는 GUI 환경의 그래픽 인터페이스를 제작하여 사용자의 편의를 도모하였으며, 아울러 모터의 제어를 위한 기구학 계산을 수행하도록 설계하였다. 즉, 사용자는 그래픽 환경에서 마우스를 이용하여 반송 테이블의 이동 궤적을 정의하고, PC 제어 프로그램은 사용자가 그린 도형

의 궤적에 따라 기구학을 계산하여, 각 모터의 이동 각도를 슬레이브 보드로 전송한다.

이 경우 두 모터로 전송되는 명령이 동기가 이루어지지 않으면 원하는 테이블의 움직임을 얻을 수 없는데, 이는 Am188ES를 사용한 슬레이브 보드에서 마스터 PC에서 전송된 이동 각도 명령을 여러 개의 작은 각도로 분할한 후 동기화하여 서보 컨트롤 보드로 전송하도록 하여 해결한다. 슬레이브 보드의 또 다른 역할로는 서보 컨트롤 보드의 진행 상황을 실시간으로 검사하여 마스터 PC 측에 다음 명령을 전송하도록 요구하거나 대기하도록 지시하는 것이며, 각 보드는 RS-232 방식을 통하여 19200[bps]의 속도로 명령을 주고받는다.

서보 컨트롤 보드는 CAN 통신이 가능한 보드로, 87c196CA 마이크로프로세서와 LMD18200 모터 드라이버를 이용하여 설계된 2개의 모터 제어용 PID 제어기이다. 2축 반송 테이블은 이전 연구[8]에서의 시스템을 개조(스텝모터를 직류모터로 교체)한 것으로 각 축의 길이는 120mm, 100mm 이며, 각 관절이 1:5의 기어 비를 갖는다.

2.2 uC/OS-II와 TASK의 설계

Am188ES보드에 포팅하여 사용한 실시간 커널은 uC/OS-II이며 교육용으로 사용 제한이 없는 저가의 상용 실시간 커널이므로 실험에서와 같은 작은 분산시스템을 구축하는데 적합하다. uC/OS-II 커널은 멀티타스킹, 세마포어 관리, TASK 관리, 메모리 관리, 시간 관리, 메일박스(Mail-box), 큐 등의 서비스를 제공하며[7]. 실행 코드 크기를 작게 하여 마이크로컨트롤러 뿐만 아니라 여러 마이크로 프로세서에 이식할 수 있다. 실제로 각 프로세서에 포팅된 많은 예제가 uC/OS 홈페이지에 제공된다.

uC/OS-II에서 다양한 서비스를 제공하여 CPU의 사용을 쉽게 해 주는 반면에 실시간 OS에서는 TASK의 효율적인 설계에 의해 시스템의 성능이 좌우된다[1]. 본 실험에서는 시리얼 통신을 관리하고, 명령의 분석 및 모터의 각도 분할 등의 작업을 위하여 표 1과 같이 3개의 TASK를 설계하였다.

표 1 Am188ES의 uC/OS-II 응용 TASK

Table 1 uC/OS-II application task of Am188ES

TASK 명	주요 임무
Control	<ul style="list-style-type: none"> 이동 명령을 n개로 재분할 다음 명령 전송 Monitor Task의 세마포어를 관리, 동기화
Receive	<ul style="list-style-type: none"> 시리얼 통신 관리. 명령 해석, Control Task로 MailBox를 전송
Monitor	<ul style="list-style-type: none"> 확인 명령('Q') 전송(500ms 마다)

각 TASK 간의 구조는 그림 2와 같다. 제어 흐름을 간략히 설명하면 다음과 같다. 먼저, PC 측에서 사용자가 반송 테이블의 이동 경로를 정의하면, Visual Basic으로 작성한

사용자 인터페이스 프로그램은 기구학을 계산하여 Am188ES 보드로 보낼 명령을 생성한다. 생성된 명령 ($K: \theta_1 \theta_2$)을 Am188ES 보드로 전송하면, 우선 수신 타스크(Receive Task)에서 명령어를 전달받는다. 이어서, 시리얼 통신을 관리하는 수신 타스크는 Am188ES 보드 측에서 처리할 명령이나 이동 명령, 87c196CA 보드로 직접 보내는 명령을 분류한다. 이동 명령일 경우 메일 박스를 이용하여 제어 타스크(Control Task)로 전송하고, 87c196CA 보드로 직접 보내는 명령일 경우는 여과 없이 통과시킨다. Am188ES의 내부 설정에 관한 명령일 경우는 다른 타스크의 도움 없이 직접 설정을 변경한다. Am188ES의 내부 설정은 다음의 표 2와 같다. 이동 명령일 경우는 메일 박스를 이용하여 제어 타스크를 활성화시키고, 각 축이 이동하여야 할 각도 θ_1, θ_2 를 전달한다.

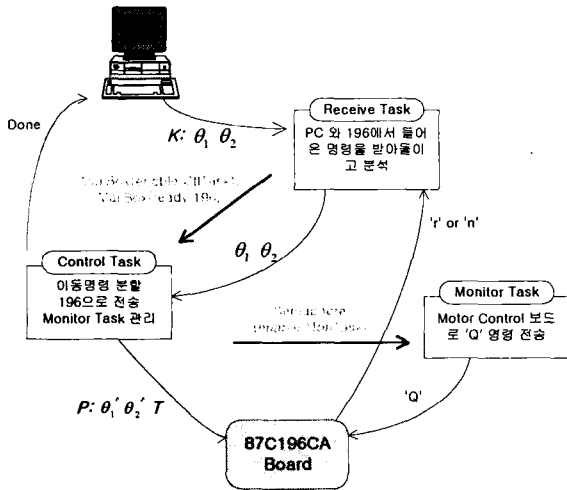


그림 2 uC/OS-II 응용 타스크의 구성도
Fig. 2 Diagram of uC/OS-II application task

표 2 Am188ES 보드 명령
Table 2 Am188ES board commands

명령 형식	기능
K: $\theta_1 \theta_2$	서보 컨트롤 보드로 전송할 이동 명령, 분할/동기화 작업 등의 과정을 거쳐 P: 명령으로 번역되어 전송된다.
N: num	서보 컨트롤 보드로 전송할 기본 STEP 값. 초기치는 500으로 설정된다.
D: num	동기화가 필요할 때, 각도 분할 횟수. 초기치는 1로 설정된다.
B	싱글 스텝 모드로 설정. 분할된 각도가 한번에 실행되지 않고 스텝별로 동작시킬 때 사용.
>	싱글 스텝 모드에서 다음 스텝 이동 명령.
C	Am188ES보드의 현재 설정 값 확인

제어 타스크는 전달받은 이동 명령을 미리 설정된 값에

따라 각도를 분할한다. 동시에 87c196CA 보드가 명령을 수행할 준비가 되었는지를 확인하는 타스크인 감시 타스크(Monitor Task)를 활성화시켜 다음 명령을 받을 준비가 되었는지를 확인한다. 이때, 87c196CA 보드는 다음 명령을 수행할 수 있다면, 수신 타스크로 준비(ready) 신호를 보낸다. 현재 이전 명령을 수행하고 있어서 명령을 받을 준비가 안 되어 있을 경우, 수신 타스크로 준비되지 않음(non-ready) 신호를 보내고 수신 타스크는 메일 박스를 이용하여 제어 타스크를 대기하도록 한다.

87c196CA보드가 이전 명령의 수행을 끝내고 다시 대기 상태에서 벗어나면, 제어 타스크는 다음 명령을 전송하고 수신 타스크의 다음 명령 전송 허가 메일을 전달 받을 때까지 대기 상태가 된다. 이와 같은 과정이 반복되어 PC로부터 전달받아 분할한 명령을 모두 전송하면, 작업이 완료되었음을 알린다.

표 3 응용 타스크의 메일 박스와 세마포어
Table 3 Mail-box and Semaphore of application task

명칭	기능
MboxCtlTask	컨트롤 타스크를 동기시킴
MboxRdy196	196 보드가 다음 명령을 실행할 준비가 되었음을 컨트롤 보드에 알림
SemMonTask	Monitor Task의 동기화
SemComm1	시리얼 포트1의 공유사용을 위한 세마포어
SemComm2	시리얼 포트2의 공유사용을 위한 세마포어
MboxBreak	스텝명령 구현 (디버그용)

uC/OS-II에서는 동기화를 위해서 세마포어 및 메일 박스, 큐 등의 서비스를 사용할 수 있고 본 실험에서는 1개의 세마포어와 2개의 메일 박스를 사용하였다. 추가로 시리얼 포트의 공유 사용을 위해 2개의 세마포어가 사용되었으며, 디버그를 위한 스텝명령 전송 및 타스크의 브레이크 신호로 메일 박스가 1개 사용된다.

메일박스 및 세마포어의 사용 용도는 표 3과 같다.

2.3 사용자 접속 프로그램

사용자 인터페이스는 그림 3과 같이 구성하였다. 사용자가 원하는 좌표에 도형을 그리면 다음의 2.4장에 소개되는 알고리즘에 의하여 각 좌표를 관절의 이동 각도로 변환한다. 변환된 이동 각도는 RS-232 통신을 통하여 Am188ES 보드로 전달되며 모터가 주어진 이동 각도만큼 이동하고 나면 다음 이동 각도를 자동으로 보낸다. 기구학을 계산하기 위하여 기구부의 물리적인 파라미터들(관절의 길이, 기어비, 정밀도) 설정 메뉴를 이용하여 손쉽게 바꿀 수 있게 하여, 다양한 반송 테이블에 적용할 수 있도록 하였다.

PC 제어 소프트웨어는 4개의 자식(Child) 윈도우로 구성되며, 각 윈도우(그림3의 주화면, 하단의 2개와 우측 1개)는 다음의 표 4와 같은 기능을 갖는다.

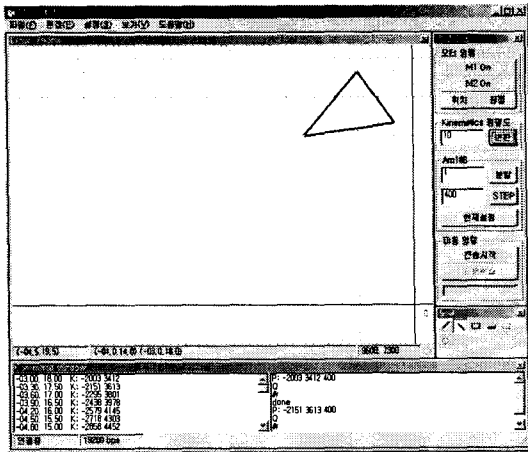


그림 3 PC 제어 소프트웨어
Fig. 3 PC control software

표 4 자식 윈도우의 기능

Table 4 The function of child windows

Window 명칭	기능
Draw window	도형을 그리는 윈도우. 마우스를 드래그 하여 도형을 정의.
Status window	전송해야 할 버퍼의 내용 표시 시리얼 통신으로 받은 데이터 표시 연결 상태 표시
Tool window	Draw window에서 작업 할 도구 선택
Communication window	모터 on / off 명령 전송 시작 / 전송 취소 현재 설정 확인 명령 기구학 계산 명령

사용자가 그림 3에서와 삼각형 궤적을 정의하기 위해서는 도구 윈도우(Tool window)에서 직선 그리기를 선택한 후, 그리기 윈도우(Draw window)에 마우스를 클릭하여 세 개의 직선을 생성한다. 같은 방식으로 사각형 모양의 도형도 그릴 수 있고, 다른 방법으로는 도구 윈도우(Tool window)에서 사각형 그리기를 선택하면 보다 쉽게 사각형 궤적을 정의할 수 있다. 궤적의 정의를 마치면, 통신 윈도우(Communication window)의 '변환 버튼'을 통해 기구학 계산을 계산하고, Am188ES 명령을 생성한다. 명령이 생성되면 '전송 시작' 버튼이 활성화되고 반송 테이블이 궤적과 같은 모양으로 움직이게 된다.

그림 3에서 보이는 삼각형 모양의 궤적 입력에 대한 명령어 및 실행 결과는 표 6과 그림 9에서 확인할 수 있다.

2.4 반송로봇 기구학

관절 각도 θ_1 과 θ_2 가 주어지면 순기구학으로부터 말단 장치의 좌표 (x, y) 를 결정할 수 있다. 거꾸로 Q의 좌표 (x, y) 가 주어졌을 때, θ_1 과 θ_2 를 알기 위해서 다음과 같은 역기구학을 이용한다.

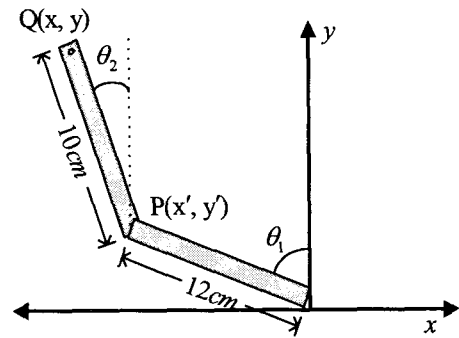


그림 4 2축 반송 테이블의 단면도
Fig. 4 Diagram of two axes convey table

그림 4 에서와 같이 두 축의 길이가 각각 $a_1=12(cm)$, $a_2=10(cm)$ 일 때, 코사인 제 2법칙으로부터 각도 θ_2 는 다음과 같이 주어진다.

$$\cos \theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} = D \quad (1)$$

$$\sin(\theta_2) = \pm\sqrt{1 - D^2} \quad (2)$$

따라서 두 관절각은 다음 식으로 구해진다.

$$\theta_2 = \tan^{-1} \frac{\pm\sqrt{1 - D^2}}{D} \quad (3)$$

$$\theta_1 = \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{a_2 \sin \theta_2}{a_1 + a_2 \cos \theta_2} \right) \quad (4)$$

이 때, 식(3)에서 양과 음의 부호를 선택함으로써, 팔꿈치 상향 해와 팔꿈치 하향 해를 모두 구할 수 있으나, 본 논문에서는 양의 값만을 취하여 실험을 하였다.

표 5 사각형을 그리기 위한 각 이동 좌표와 회전각, 명령
Table 5 Coordinates, rotation angle and command to draw a quadrangle

	P점	Q점	Axis1 회전각	Axis2 회전각	명령형식
출발	(0.0, 22.0)	(0.0, 12.0)	0.00	0.00	
A1	(-5.0, 18.0)	(-8.38, 8.59)	-44.30	64.06	K: -2215 3203
B1	(-2.5, 18.0)	(-7.52, 9.35)	-38.81	68.94	K: -1940 3447
A2	(0.0, 18.0)	(-6.29, 10.22)	-31.59	70.53	K: -1579 3526
B2	(0.0, 15.5)	(-7.74, 9.17)	-40.17	90.90	K: -2009 4545
A3	(0.0, 13.0)	(-8.77, 8.19)	-46.95	108.21	K: -2347 5411
B3	(-2.5, 13.0)	(-10.09, 6.49)	-57.25	106.65	K: -2862 5332
A4	(-5.0, 13.0)	(-10.93, 4.95)	-65.64	102.02	K: -3282 5101
B4	(-5.0, 15.5)	(-9.90, 6.78)	-55.58	84.92	K: -2779 4246
A1	(-5.0, 18.0)	(-8.38, 8.59)	-44.30	64.06	K: -2215 3203

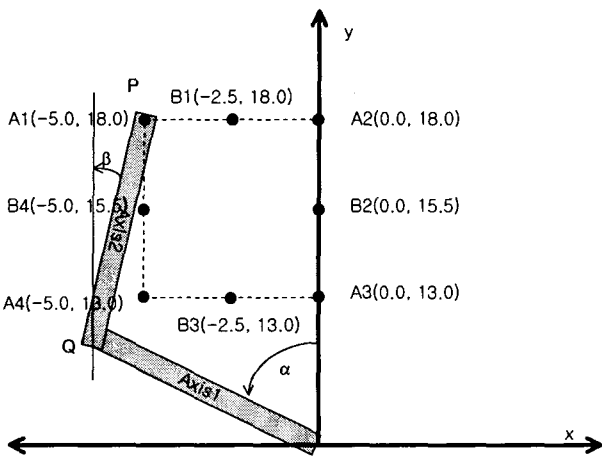


그림 5 사각형 경유 지점
Fig. 5 Passing point to draw a quadrangle

표 5는 실험에서 사용한 사각형을 그리기 위한 사용자 입력에 대한 제적으로, 그림 5의 표시된 지점을 경유하는 경우 (여러 개의 경유점을 추가할 수 있으나 본 실험에서는 하나만을 추가하였음)의 두 축의 이동 좌표와 회전각 및 Am188ES 보드로 전송되는 명령형식을 나타내었다. 표 5의 Am188ES 보드로 전송하는 명령형식($K: \theta_1 \theta_2$)에서 θ_1 과 θ_2 는 반송 테이블의 관절 기어비와 87c196CA 보드의 정밀도를 고려하여 Axis1 회전각, Axis2 회전각에 각각 50 배 한 값이다.

2.5 실험

그림 6는 본 실험 장치의 사진이다. 그림 하단에 슬레이브 보드(Am188ES), 우측에 서보 컨트롤 보드(87c196CA), 그리고 좌측에 DC 모터로 개조된 2축 반송 테이블을 보여 준다. PC는 생략 되었으며, 중앙의 작은 보드는 반송테이블과 제어기 연결을 위한 콘넥터 보드이다.

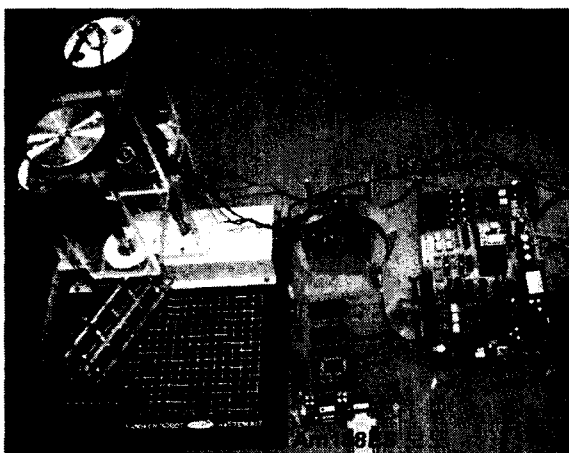


그림 6 2축 DC 모터 제어 시스템
Fig. 6 Control system of two axes DC servo motor

그림 3의 사용자 입력 창에서 삼각형 및 사각형 그리기 명령을 실행한 경우 서보 컨트롤러에 전달되는 최종 명령어는 표 6과 같은 형태로 변환되어 전송된다.

$$P: \theta_1 \theta_2 STEP$$

여기서 STEP은 서보 컨트롤 보드의 샘플링 시간으로 1 STEP은 2ms 이다. 따라서 4000은 한 변을 그리는 시간이 8초를 의미하며, 그림 B의 경우는 4초씩 구분하여 그리는 것을 의미한다.

표 6 사각형과 삼각형을 그리기 위한 87c196CA 보드 명령
Table 6 87c196CA board command to draw a quadrangle and a triangle

사각형을 그린 경우		삼각형을 그린 경우	
그림 7-A	그림 7-B	그림 8-A	그림 8-B
P: -2215 3203 4000	P: -2215 3203 2000	P: -2003 3412 4000	P: -2003 3412 2000
	P: -1940 3447 2000		P: -2718 4303 2000
P: -1579 3526 4000	P: -1579 3526 2000	P: -3417 4968 4000	P: -3417 4968 2000
	P: -2009 4545 2000		P: -2955 5095 2000
P: -2347 5411 4000	P: -2347 5411 2000	P: -2420 5065 4000	P: -2420 5065 2000
	P: -2862 5332 2000		P: -2268 4309 2000
P: -3282 5101 4000	P: -3282 5101 2000	P: -2003 3412 4000	P: -2003 3412 2000
	P: -2779 4246 2000		
P: -2215 3203 4000	P: -2215 3202 2000		

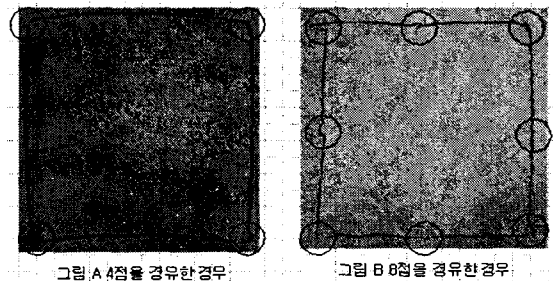


그림 7 2축 스텝모터로 그린 사각형
Fig. 7 Quadrangle drew by two axes step motor

그림 7은 위의 표 6의 좌표에 대한 이전 실험 결과를 나타낸 것으로, 스텝모터의 0.18도의 정밀도와 펄스 방식의 구동으로 인해 매끄러운 직선을 그리지 못하였다. 사각형의 중간점을 거쳐 가는 방식으로 그린 그림 7-B에서도 각 점에서 직선이 흔들리는 현상이 나타나고 있다.

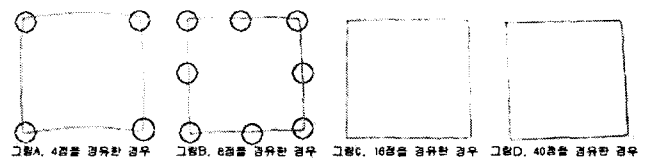


그림 8 2축 DC 모터로 그린 사각형
Fig. 8 Quadrangle drew by two axes DC servo motor

본 실험에서는 이러한 문제를 해결하기 위하여 기구부를 DC 모터로 교체하여 0.02의 정밀도를 갖게 하였으며, 이 경우 실험 결과는 그림 8과 같다. 실험 결과 그림에서 나타나는 바와 같이 매끄러운 직선으로 나타났으며 사각형의 중점을 거쳐 갈 경우 더욱 직선에 가까워지고 각 점에서 흔들리는 현상도 사라졌다.

그림의 C, D는 기구학을 정밀히 풀어 더 많은 좌표를 거쳐 가는 경우에 대한 실험 결과이다. 이때 한 번을 그리는데 걸리는 시간을 4000 스텝으로 앞서와 동일하게 그렸다. B의 경우와 큰 차이는 없으나 각 변이 좀더 직선의 형태를 보여주고 있다.

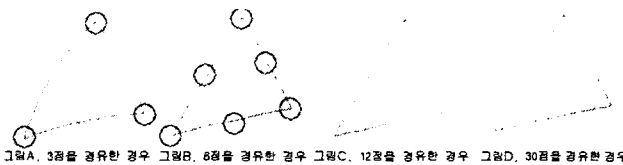


그림 9 2축 DC 모터로 그린 삼각형
Fig. 9 Triangle drew by two axes DC servo motor

그림 9는 사각형 도형보다 직선의 흰 정도의 관찰이 용이한 삼각형을 그려본 것이다. 그림의 좌측(A)에서 우측(D)의 그림으로 갈수록 더욱 깔끔한 직선이 그려져 각 변을 10번의 좌표로 나눠서 그린 그림 9-D의 경우는 거의 완벽한 직선에 가까운 그림이 그려져 만족할 만한 결과를 얻었다.

3. 결 론

본 논문에서는 PC를 마스터로 사용하고 실시간 커널을 포팅한 Am188ES을 슬레이브로, 87c196CA 보드를 서보 컨트롤러로 이용하여 2축 반송 테이블을 제어하는 마이크로 분산시스템을 구현하였다. 이는 PC 프로그램의 다양한 GUI와 편리한 개발 환경을 실시간 제어에 간편히 적용할 수 있음을 보여준 사례연구임에 의미를 부여할 수 있다.

또한, 본 실험에서는 PC에 uC/OS-II 커널을 직접 설치한 기존 제어기의 제어성능 향상 외에도 PC 기반의 GUI 프로그램을 추가하여 반송테이블의 궤적을 바꾸기 위해 별도의 프로그램 코드 수정 없이 명령 윈도우 상에서 마우스를 이용하여 그림을 그려 쉽게 바꿀 수 있게 하였다.

제시한 시스템의 구성은 다축 제어로 확장하는 경우에도 모터 제어기를 통신망으로 연결한 후 PC 프로그램의 기구 해석 프로그램만 수정하면 쉽게 적용이 가능한 강점을 가지고 있다.

감사의 글

본 연구는 2002년도 광운대학교 교내학술연구비 지원에 의하여 이루어진 연구로서, 관계부처에 감사드립니다.

참 고 문 헌

- [1] Ian D. McCammon, Stephen C. Jacobsen, "Communication and Control for Distributed Microsystems", IEEE Control Systems Magazine, pp. 48-50, 1990
- [2] Wook Hyun Kwon, Seong-Gyu Choi, "Real-Time Distributed Software-In-the-Loop Simulation for Distributed Control Systems", IEEE Int. Symp. on Computer Aided Control System Design, pp. 115-119, 1999
- [3] Samuel M. Smith, "An Approach to Intelligent Distributed Control for Autonomous Underwater Vehicles," Proceedings of the 1994 Symposium on AUV'94, pp. 105-111, 1994
- [4] Gerhard Gruhler, "CANopen based distributed control systems," IEE Colloquium on Industrial Automation and Control, pp. 4/1-4/3, 1998
- [5] Seung Ho Hong, "Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems", IEEE Trans. on Control Systems Technology, vol. 3, pp. 225-230, 1995
- [6] Jean J. Labrosse, "MicroC/OS-II The Real-Time Kernel", R&B Books, 2002
- [7] Jean J. Labrosse, "Embedded Systems Building Blocks", R&B Books, 2002
- [8] 김종걸, "실시간 커널에 의한 스텝 전동기의 마이크로 분산제어 시스템의 구현", 광운대학교 석사학위논문, 2001

저 자 소 개



이 건 영 (李 健 寧)

1959년 8월 2일생. 1982년 한양대 전기공학과 졸업. 1993년 동 대학원 전기공학과 졸업(공학). 1996년~현재 광운대 전기공학과 부교수

Tel : 02) 940-5142

Fax : 02) 942-0107

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