

DSP를 이용한 원격 로봇의 제어 시스템 구현

노 칠 래 정 명 진
한국과학기술원 전기및 전자공학과

Implementation of a Control System for a Telerobot using DSP.

C. L. Roh M. J. Chung
Dept. of Electrical Engineering, KAIST.

ABSTRACT

A high speed control system for a telerobot using DSP is developed. The system is designed to resolve computational burden in advanced algorithms. The design is assumed to have no specific algorithm and robot configuration. The system is composed of a teaching box, a DSP board, a set of servo drivers and 16 bit microcomputer system. The teachign box is designed as a man-machine interface, which has two joysticks with three degrees of freedom for velocity generation in Cartesian space. The DSP board, i.e. DSP56000ADS based on a 10.25MIPS digital signal processor, DSP56001, computes the inverse Jacobian matrix which transforms Cartesian velocity into joint velocity. A resolved motion rate control algorithm for a 5 degrees of freedom manipulator was implemented. About 100 Hz sampling rate was achieved in this system.

1. 서론

20세기 들어 산업의 급속한 발달은 작업의 높은 생산성을 요구하게 되었고 또한 인간의 작업 영역도 기존의 공장 내부에서의 작업 뿐 아니라 인간이 직접 접근하기 어려운 영역까지 확대 되었다. 생산성 향상에 큰 역할을 해 온 산업용 로봇은 대부분 정해진 작업장 내에서 미리 프로그램된 작업을 반복적으로 행한다. 반면에 원격 로봇(telerobot)은 인간이 접근하기 용이하지 않거나 위험한 곳, 예를 들면 우주 공간, 심해, 방사능 지역 등에서 인간을 대신하여 비반복적인 작업을 행하는 로봇을 말한다.[1]

원격 로봇은 작업자와 멀리 떨어진 곳에 위치하면서 작업을 수행함에 있어서 작업자의 감독과 명령을 필요로 하는 로봇이다. 작업자는 정상적인 제어 루프(control loop)에 직접 개입하여 인간의 감각 기관을 통하여 원격 지점의 작업 상태를 감지하면서 전체 작업을 감독하게 된다. 우주 공간에서의 작업과 같이 지구상의 작업자와 원격 로봇간의 거리가 멀어서 시간 지연이 문제가 될 경우에는 작업자는 원격 로봇에 효과적인 작업 지시를 내릴 수가 없게 되며 이 경우에는 관리 제어 방식을 도입

하여 작업을 수행하도록 하고 있다. 이 방식은 원격 지점에 원격 컴퓨터를 두어서 그 스스로 간단하고 평범한 결정을 내리도록 하며 작업자는 감독자로서 수시로 원격 로봇을 위한 상위 레벨(high level)의 동작 순서를 결정하고 판단을 내려서 원격 컴퓨터의 제한된 결정 능력을 보완해 주는 형식이다. 일반적으로 관리 제어 방식은 인간이 작업의 매 순간마다 개입할 수 없기 때문에 예측할 수 없는 상황이나 정교한 작업에는 부적합한 단점이 있다.

그러나, 원자력 발전소 내부에서의 작업이나 고장난 우주선의 수리 작업과 같이 작업자와 원격 로봇간의 거리가 충분히 가까워서 시간 지연이 문제가 되지 않을 때에는 작업자는 유리창을 통하여 또는 모니터를 보면서 온 라인(on-line)으로 원격 로봇을 조종할 수 있게 된다. 이 경우에 작업자와 원격 로봇간의 인터페이스가 중요하게 된다. 지금까지 원격 로봇을 조종하기 위한 입력 장치로는 대체로 스위치에 의한 방식, 마스터 암에 의한 방식, 조이스틱에 의한 방식등이 있었다. 스위치에 의한 방식은 로봇의 각 관절마다 대응하는 스위치를 두어서 조종하는 방식으로 일련의 작업을 지시하는데 있어 번거로운 여러번의 스위치 조작이 필요하고 또 정교한 작업을 지시하는데도 어려움이 있다. 마스터 암을 사용하는 방식은 마스터-슬레이브 제어 방식(master-slave control)이라 하며 원자력 발전소 등에서 빠른 위치 이동이 필요한 작업에서 널리 사용되고 있다. 이 방법은 작업자가 있는 지점에 마스터 암을 두어서 마스터를 조종함에 따라 슬레이브 매니플레이터가 동작되는 방식이다. 그러나 이 방식은 마스터 암이 슬레이브 암과 같은 기구학적 특성을 가져야 한다는 단점이 있다. 조이스틱에 의한 방식은 암의 길이가 길고 이동이 느린 우주 공간에서의 작업 등에서 잘 쓰이는 방식이며 조이스틱의 변위에 비례하는 명령에 의해서 로봇을 조종하는 방식이다. 이 때 두개의 3차원 조이스틱을 이용하여 기존 직교 좌표상에서 본 로봇의 선속도 및 각속도를 지시하면 이를 로봇의 각 관절에 해당하는 관절 속도로 분해하여 제어하는 것이 가능하며 이 방식을 분해 속도 제어(Resolved Motion Rate Control) 방식이라 한다. 이 방식은 간단하고 조종이 용이하며 작업자는 관절 좌표계보다 직교 좌표계에 더 친숙해 있기 때문에 일찍부터 원격 로봇의 조종 방식으로 제안되어 왔다.[2] 그러나

이 방식은 직교 좌표상에서 본 로봇의 선속도 및 각속도를 관절 속도로 변환할 때 계산량이 많은 자코비안 행렬(Jacobian matrix)과 이의 역행렬을 계산해야 하는 문제가 있어서 이를 실시간으로 처리하는데 많은 어려움이 있었다.

본 연구에서는 이러한 점을 극복하기 위하여 현재 신호 처리용으로 각광을 받고 있는 고속의 디지털 신호 처리기(DSP)를 메인 CPU로 하는 원격 로봇 제어 시스템의 제작[3][4]과 이를 원격 로봇에 적용하여 본 시스템의 효용성을 보이고자 한다. 로봇 제어 시스템의 구성은 모토롤라(Motorola)사의 DSP56001을 메인 CPU로 하는 DSP56000ADS(Application Development System)을 메인 보드(board)로 사용하였다. 여기에 D/A 보드, 엔코더 보드, 작업을 지시하기 위한 교시 상자(teaching box)와 이들을 DSP에 연결시켜주는 인터페이스 보드, 그리고 각 관절마다 하나씩의 서어보 증폭기로 구성된다. 특히 교시 상자는 두개의 3차원 조이스틱을 사용함으로써 작업자가 편리하게 주어진 작업을 지시할 수 있도록 하였다. 본 연구에서 사용한 원격 로봇은 무브마스터(Movemaster) RV-M1이며 자유도(d.o.f.)가 5인 레볼류트(revolve)형의 로봇이다. 로봇의 각 관절의 액츄에이터는 DC 모터이며 엔코더에 의해서 현재 위치를 감지한다. 이렇게 구성된 전체 시스템을 이용하여 원격 로봇을 분해 속도 제어 방식으로 제어하기 위한 알고리즘을 어셈블리 언어를 사용하여 작성하였다. 본 논문의 구성은 1장의 서론에 이어 2장에서는 분해 속도 제어의 개념과 자코비안 행렬에 대하여 설명하고 3장에서는 본 연구에서 사용한 DSP의 특성과 간단한 제어 시스템의 구성을 4장에서는 실험 방법 및 결과분석에 이어 5장에서는 결론 및 추후 과제에 대하여 언급하고자 한다.

2. RMRC의 개념 및 자코비안 행렬.

2.1 RMRC의 개념.

분해 속도 제어(RMRC)란, 기준 직교 좌표계의 관점에서 본 로봇 손의 임의의 움직임에 이를 로봇 각 관절의 움직임이 결합된 것으로 생각하여, 이를 분해하여 각 관절의 움직임으로 변환함으로써 각 관절이 동시에 서로 다른 시변율로 움직여 임의의 원하는 움직임을 얻고자 하는 방식이다.

기준 직교 좌표계에서 본 로봇 손의 위치를 각각 P_x, P_y, P_z 그리고 방향을 각각 요(yaw) 각 a , 피치(pitch) 각 b , 롤(roll) 각 θ 라 하면 자유도가 6인 로봇손의 기준 직교 좌표계와 관절 좌표계 사이에는 다음과 같은 비선형 함수 관계가 성립한다.

$$x(t) = f(q(t)) \quad (1)$$

여기서 $f(q)$ 는 6×1 벡터 값 함수이고 $q(t)$ 는 관절 좌표값 $(q_1, q_2, \dots, q_6)^T$ 이다. 일반적으로 매니퓰레이터의 자유도가 m 이고, 기준 좌표계에서 관심을 두는 차원을 n 이라 하면 (1)식을 시간에 대하여 미분하면

$$\frac{dx(t)}{dt} = \dot{x}(t) = J(q)\dot{q}(t) \quad (2)$$

이고 여기서 $J(q)$ 는 $q(t)$ 에 대한 자코비안 행렬이며

$$J_{ij} = \frac{f_i}{q_j} \quad 1 \leq i \leq n, 1 \leq j \leq m \quad (3)$$

이다. 따라서 (2)식은 변화율, 즉 미소 변위의 관점에서 두 좌표 간에는 선형관계가 성립된다. 만약 로봇의 자유도와 기준 좌표

계에서 관심을 두는 $x(t)$ 의 차원이 같고 $q(t)$ 에 대하여 자코비안 행렬이 아니면 역행렬이 존재하고

$$\dot{q}(t) = J^{-1}(q)\dot{x}(t) \quad (4)$$

이 되어, 직교 좌표상에서의 움직임에 얻기 위한 관절 좌표상에서의 각 관절의 속도 명령을 생성해 낼 수 있다. 그러므로 기준 직교 좌표상에서 임의의 방향으로 로봇 손의 선속도 및 각속도가 주어 진다면 이를 각각의 관절 속도로 분해하여 제어하는 방식이 가능해진다. 이러한 제어 개념을 분해 속도 제어(RMRC)라 하며 이를 블록 다이어그램으로 나타내면 다음과 같다. [5]

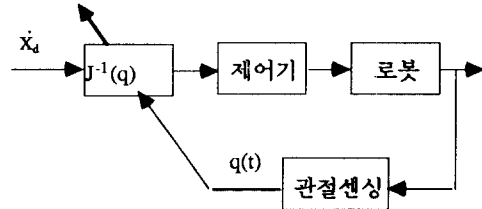


그림 1. 분해 속도 제어 블록 다이어그램

분해 속도 제어의 구현시 가장 문제가 되는 부분이 자코비안 행렬의 역행렬을 구하는 것이다. 특히 어느 특정한 지점에서 자코비안 행렬이 랭크의 갯수를 잃어서 선형 독립인 행이 나 설의 갯수가 줄어들게 되고, 특정 관절에만 과도한 명령이 가해지면서 작업자가 원하는 작업을 할 수 없게 된다. 이처럼 자코비안 행렬이 랭크의 갯수를 잃는 지점을 특이점이라 하며 그 정의는 다음과 같다.[6]

정의 1. $J(q)$ 를 로봇 암의 $6 \times n$ 자코비안 행렬이라 하면, " \vec{q} 가 로봇 암의 특이점이다."의 필요 충분 조건은 다음과 같다.

$$\text{rank}[J(q)] < \min\{6, n\}.$$

따라서 특이점은 자코비안 행렬이 랭크의 갯수를 잃게 되는 점이다.

한편, $n = 6$ 일때 자코비안 행렬이 특이 행렬이면 $n \times n$ 행렬 $J^T(q)J(q)$ 도 특이 행렬이 되고 특이점 근처에서는 로봇을 제어하기가 어렵게 된다. 따라서 이 경우에는 매니퓰레이터 기민성(Manipulator Measure)의 척도를 사용한다.

$$\text{dex}(q) = \det[J^T(q)J(q)] \quad n \leq 6 \quad (5)$$

따라서 $\text{dex}(q) = 0$ 이면 그 점은 특이점이고, 특이점일 경우에는 $\text{dex}(q) = 0$ 이 된다. 더우기 관절값이 특이점으로 가까워짐에 따라 기민성 척도 $\text{dex}(q)$ 는 점점 0으로 접근한다. 특이점은 두가지 종류가 있으며 경계선 특이점과 내부 특이점으로 나뉘며 내부 특이점이 중요한 문제가 된다. 본 연구에서는 기민성 척도를 사용하여 임의의 관절에 과도한 명령이 가해지지 않도록 하였다. 또한 본 연구에서 사용한 무브마스터 로봇의 경우 자유도가 5이고 자코비안 행렬은 6×5 행렬이 되어 역행렬을 구하는 것이 용이하지 않게 된다. 이러한 경우에는 일반화된 역행렬(generalized inverse)을 사용하며 그 정의는 다음과 같다.

정의 2. A 를 $m \times n$ 행렬이라고 하자. 그 때 " $n \times m$ 행렬 X 는 행렬 A 의 일반화된 역행렬이다."의 필요 충분 조건은 다음 4가지 특성 중에서 적어도 특성 1 또는 2를 만족시켜야 한다.

1. $AXA = A$
2. $XAX = X$
3. $(AX)^T = AX$
4. $(XA)^T = XA$

이 정의로부터 $m \times n$ 행렬 A 의 일반화된 역행렬은 다음처럼 구할 수 있다.

$$A^+ = \begin{cases} A^T[AA^T]^{-1} & m \leq n \\ A^{-1} & m = n \\ [A^T A]^{-1} A^T & m \geq n \end{cases} \quad (6)$$

따라서, 본 연구에서 사용한 로봇의 경우 자유도가 5인 로봇이므로 $x(t)$ 에 대한 관절 좌표상에서의 궤적은 다음과 같이 구할 수 있다.

$$q = [J^T(q)J(q)]^{-1}J^T(q)x \quad (7)$$

2.2 무브메스터 로봇의 자코비안 행렬.

자코비안 행렬을 구하기 위한 로봇의 링크 좌표와 변수들은 다음과 같다.[7]

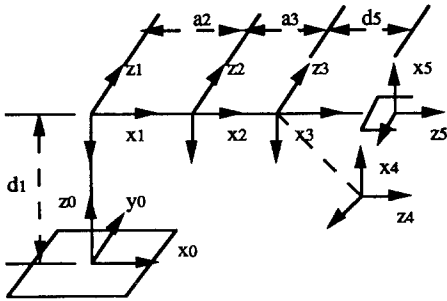


그림 2. 원거리로봇의 링크 좌표계.

관절	q	a	a	d	범위
1	0	-90	0	300	-150 ~ 150
2	0	0	250	0	-100 ~ 30
3	0	0	160	0	0 ~ 110
4	-90	-90	0	0	-160 ~ 0
5	0	0	0	150	-180 ~ 180

표 1. 원거리로봇의 링크 변수.

기준 직교 좌표계 (x_0, y_0, z_0) 에서 본 로봇 손의 선속도 및 각속도는

$$\begin{bmatrix} V(t) \\ W(t) \end{bmatrix} = J(q)q(t) = [J_1, J_2, \dots, J_5]q(t)$$

이고 여기서 $J(q)$ 는 6×5 자코비안 행렬이고 각 열벡터 $J_i(q)$ 는 관절 i 가 회전할때 다음과 같다.

$$J_i(q) = \begin{bmatrix} z_{i-1} \times p_i^{-1} \\ z_i \end{bmatrix} \quad (8)$$

p_i^{-1} 는 기준 직교 좌표계로 표현된 관절 좌표계 $(i-1)$ 의 원점으로 부터 손좌표계 원점까지의 위치 벡터이다. z_i 는 기준 직교 좌표계로 표현된 i 번째 관절의 운동 축방향의 단위 벡터이다. (8)식에 의하여 구한 무브메스터 로봇의 자코비안 행렬은 다음과 같다.

$$J_1(q) = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23} - d_5s_{234}) \\ c_1(a_2c_2 + a_3c_{23} - d_5s_{234}) \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$J_2(q) = \begin{bmatrix} -c_1(a_2s_2 + a_3s_{23} + d_5s_{234}) \\ -s_1(a_2s_2 + a_3s_{23} + d_5s_{234}) \\ -a_2c_2 - a_3c_{23} + d_5s_{234} \\ -s_1 \\ c_1 \\ 0 \end{bmatrix}$$

$$J_3(q) = \begin{bmatrix} -c_1(a_3s_{23} + d_5c_{234}) \\ -s_1(a_3s_{23} + d_5c_{234}) \\ -a_3c_{23} + d_5s_{234} \\ -s_1 \\ c_1 \\ 0 \end{bmatrix}$$

$$J_4(q) = \begin{bmatrix} -d_5c_1c_{234} \\ -d_5s_1c_{234} \\ d_5s_{234} \\ -s_1 \\ c_1 \\ 0 \end{bmatrix}$$

$$J_5(q) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -c_1s_{234} \\ -s_1s_{234} \\ -c_{234} \end{bmatrix}$$

여기서 $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{ij} = \cos(q_i + q_j)$ 그리고 $s_{ij} = \sin(q_i + q_j)$ 를 의미한다. 원거리로봇을 손 좌표계의 관점에서 제어하고자 할 때에는 자코비안 행렬에 회전 행렬(rotation matrix)을 곱하여, 손 좌표계에서의 명령으로 변환하여 제어할 수 있다.

$$\begin{bmatrix} V_0^5(t) \\ W_0^5(t) \end{bmatrix} = \begin{bmatrix} R_0^5 & 0 \\ 0 & R_0^5 \end{bmatrix} [J(q)] q(t)$$

3. 제어 시스템의 구성

3.1 개요

본 연구에서 구성한 전체 시스템은 다음과 같다.[8]

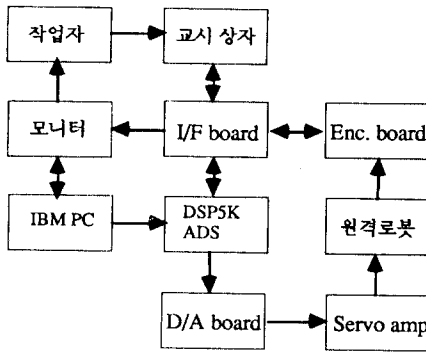


그림 3. 제어 시스템 구성도.

작업자는 IBM PC/AT 상에서 원격 로봇을 제어하기 위한 제어 알고리즘을 프로그램하여 DSP에 다운로드 한다. 작업자는 주 컴퓨터인 IBMPC의 모니터를 보면서 원격 로봇의 현재 위치와, 작업 환경의 정보, 또한 작업자가 지시하는 정보를 얻는다. 작업자는 조이스틱 두개로 구성되는 교시상자를 조종하고, 모니터나 유리창을 통하여 원격 지점의 상태를 보아가면서 작업을 수행할 수 있다. DSP56KADS는 원격 로봇의 제어 및 작업에 필요한 모든 정보를 받아서 이를 빠른 시간내에 처리하여 제어기에 필요한 명령을 만들어 내는 일을 한다. 즉 DSP는 교시 상자로 지시되는 원격 로봇의 선속도 및 각속도를 받아서 이를 관절 속도로 변환하여 주는 일을 한다. D/A 보드는 DSP에서 계산한 제어 명령을 아날로그 값으로 변환시켜 주고 다음 단의 서어보 증폭기에 적합한 입력 레벨을 조정해 준다. D/A는 12 비트를 사용하였으며 채널을 7개로 하였다. 서어보 증폭기는 지령 전압에 비례하는 전류를 출력하도록 되어 있으며, 모터의 용량에 맞추어서 실장하였다. 원격 로봇으로 사용한 무브메스터 로봇은 원래

는 산업용으로 개발된 로봇이나, 본 연구에서는 이를 원격 로봇으로 가정하고 사용하였다. 이는 근본적으로 원격 로봇과 산업용 로봇 사이에 하드웨어에는 거의 차이가 없기 때문에 가능하였다. 로봇의 각 관절의 액츄에이터는 DC 모터이며 모터의 용량은 1, 2, 3축이 30W 정도이고 4, 5축이 11W 정도이다. 로봇을 효과적으로 제어하는데는 궤환 입력이 중요하며 정확한 위치 센싱을 위해서 본 로봇에는 각 관절 축마다 증분형 엔코더가 부착되어 있으므로 이를 위해서 엔코더 보드를 제작하였다. 이 보드는 엔코더로부터 출력되는 진상과 지상의 펄스 신호를 이용하여 로봇의 회전 방향을 알 수 있도록 상승/하강 신호를 만들고 이를 카운트하여 15 비트의 엔코더 보드를 제작하였다. 이렇게 전체 시스템이 구성되면 작업자는 제어 알고리즘을 DSP에 다운로드 한 후 교시 상자를 이용하여 로봇 손이 원하는 궤적을 갖도록 속도 명령을 내려서 작업을 지시할 수 있다.

3.2 DSP의 특성.

본 연구에서 사용한 DSP 칩 DSP56001은 모토롤라 사에서 통신, 고속 제어, 수치 연산, 컴퓨터 응용 및 오디오 응용 등 많은 연산을 필요로 하는 작업을 빠른 시간 내에 처리하기 위한 목적으로 개발되었다. DSP56001은 10.25 MIPS의 연산 속도를 가지며 데이터 버스는 24비트이며 4개의 데이터 버스 와 3개의 어드레스 버스 구조를 가지면서 고속의 데이터 이동이 가능하도록 설계되었다. 메모리는 1개의 512 워드(wod)의 프로그램 메모리와 2개의 데이터 메모리 X,Y로 구성된다. 이들은 외부 메모리를 접속하여 확장이 가능하게 되어 있다. 또한 MAC 연산(곱하기 + 더하기)과 메모리 포인터 연산을 한번의 명령어 수행 시간 내에 실행함으로써 배터/행렬 연산이 용이하도록 되어 있다. 이외에도 데이터 입출력을 위한 다수의 입출력 포트(port)와 호스트 인터페이스 포트가 있다. DSP56001의 구조는 다음과 같다. DSP56001은 고정 소수점 연산을 하므로 부동 소수점 연산을 위해서는 별도의 부동 소수점 연산용 서브루틴을 사용하여야 하며

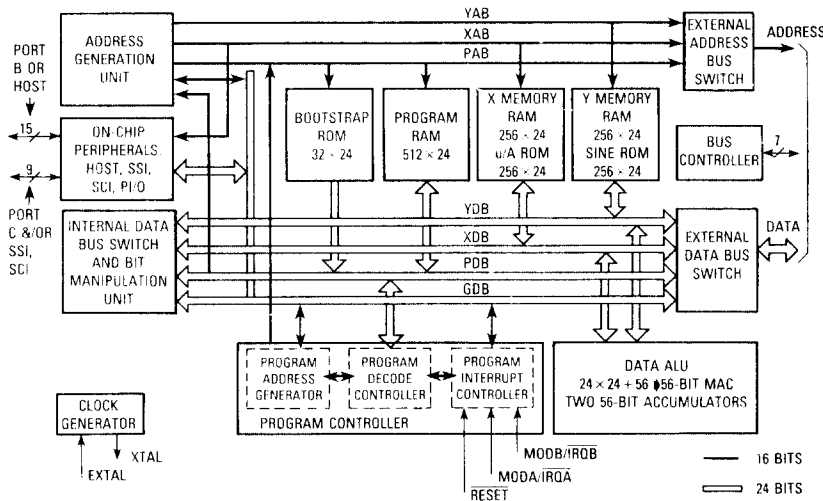


그림 4. DSP56001의 구조.

이들의 연산 시간은 다음과 같다.

연산종류	연산	최고	최저
fadd_xy	$R = Y + X$	1.27usec	7.41usec
fsub_xy	$R = Y - X$	1.17usec	7.02usec
fmpy_xy	$R = X * Y$	1.85usec	6.24usec
fdiv_xy	$R = Y / X$	1.27usec	10.73usec
fmac_xya	$R = A + X * Y$	3.80usec	14.34usec

표 2. 주요 부동 소수점 연산 실행 시간

DSP를 사용하는데 있어서 큰 문제점은 대부분의 DSP가 그렇듯이 마이크로프로그래밍을 하는 것이라고 할 수 있으며 이를 위해서는 프로그래머가 DSP의 구조에 대한 충분한 이해와 깊은 주의가 필요하다.

DSP56KADS는 본 연구에서 구성한 전체 로봇 시스템의 중심부이며 작업자가 지시하는 명령과 로봇으로부터 감지되는 정보를 이용하여 다음 단계에 가져야 할 명령을 생성하고 제어하는 역할을 한다. 이는 DSP56001을 탑재한 DSP 전용 보드로 Motorola사에서 제공하는 것이다. 이 시스템은 DSP를 이용한 여러 타아켓 시스템의 개발을 위해 제작되었으며 그 구성은 다음과 같이 ADM(application development system), IBM PC 버스 인터페이스 카드, 그리고 PC 상에서 실행되는 유저 인터페이스 프로그램으로 구성된다.[10]

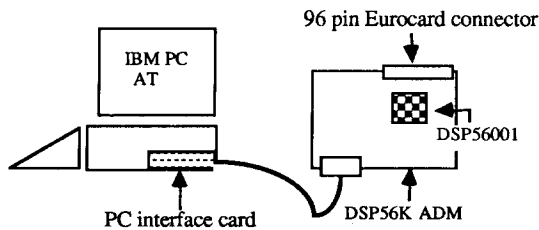


그림 5. DSP56000 ADS의 구성.

ADM 하드웨어는 20.48 Mhz에서 동작하도록 설계되어 있고 하나의 호스트에 대하여 동시에 여러 개의 ADM을 사용할 수 있다. 메모리는 3개이며 유저가 프로그램 개발을 위한 4K 워드의 프로그램 메모리와 데이터 저장을 위한 각 2K 워드의 X 데이터 메모리와 Y 데이터 메모리로 구분된다. 또한 직/병렬 데이터 전송을 위한 여러개의 단자부와 DSP의 모든 핀에 액세스할 수 있는 96 핀의 유로 카드 단자부가 있다. 본 연구에서는 이 유로 카드 단자부를 이용하여 DSP와 주변 제어 보드간의 인터페이스를 담당하도록 하였다. 유저 인터페이스 프로그램으로는 ADS56000.EXE이 있으며 이 프로그램을 이용하여 유저가 작성한 프로그램을 ADM에 다운 로딩할 수 있다. 프로그램의 작성은 DSP 어셈블리 언어를 사용하였다.

4. 실험 방법 및 결과

4.1 실험 방법

DSP에 프로그램을 하는 절차는 C언어와 같은 상위 레벨 언어로 작성한 프로그램을 크로스 컴파일러를 이용하여 어셈블리 언어로 번역하는 방법과 바로 어셈블리 언어로 구현하는 방법이 있다. 일반적으로 수행 속도가 문제가 될 때에는 어셈블리 언어로 바로 구현하는 것이 보통이고 본 연구에서도 이 방법을 택하였다. 또한 삼각 함수값이 필요하기 때문에 이를 위하여 사인 함수값을 90도를 900등분하여 Y 데이터 메모리에 다운 로딩한 후에 이를 사용하는 방법을 취하였다. RMR 알고리즘을 구현한 전체 순서도는 다음과 같다.

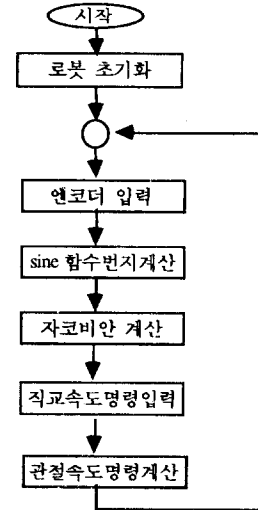


그림 6. 분해 속도 제어 알고리즘 순서도

4.2 실험 결과

본 연구에서의 분해 속도 제어에 필요한 연산의 횟수와 시간은 다음과 같다.

연산 종류	횟수	연산 시간	총 연산 시간
move	1506	0.1usec	150.6usec
add	3	0.1usec	0.3usec
mpy	15	0.1usec	0.15usec
mac	11	0.1usec	0.11usec
fmpy_xy	55	6.24usec	343.2usec
fsub_x	50	7.02usec	351usec
fdiv_xy	15	10.73usec	160.9usec
fmac_xy	175	14.34usec	2509.5usec
float_a	30	8.10usec	243usec

표 3. 속도 명령을 발생하는데 필요한 연산 횟수 및 시간

위 연산에 걸리는 시간은 약 3.75 msec이다. 본 연구에서는 제작한 시스템을 이용하여 분해 속도 제어 알고리즘을 적용하였을

때 약 10msec 이내의 샘플링 시간을 얻을 수 있었다.

5. 결론 및 추후 과제

본 연구에서는 DSP를 이용한 원격 로봇의 제어 시스템을 제작하고 여기에 분해 속도 제어 알고리즘을 적용하여 5축의 원격 로봇을 효과적으로 조종할 수 있었다. 본 시스템은 속도 면에서 다른 범용의 마이크로프로세서보다 월등히 뛰어난 DSP를 사용함으로써 일반적으로 계산량이 많은 로봇 제어에 효과적으로 사용될 수 있다는 것을 보였다. 단점으로는 일반적으로 어셈블리 언어의 구현이 상위 레벨 언어에 비하여 어렵고, 또한 DSP 자체의 하드웨어에 대한 깊은 이해가 필요하다는 것이다. 이러한 점들을 고려하여 추후 과제로는 다음과 같은 연구가 필요 되어진다.

1. 센서를 도입하여 더 성능이 향상된 원격 로봇 시스템과 이의 효율적인 제어를 위한 유지 인터페이스에 대한 연구가 필요하다.
2. 부동 소수점 연산이 가능하고 상위 레벨 언어의 지원이 가능한 DSP의 도입이 필요하다.
3. 범용 프로세서와의 인터페이스 기술이 필요하다.

참고 문헌

- [1] T.B Sheridan, "Telerobotics", Automatica, vol 25, No.4, 1989
- [2] D.E. Whitney, "Resolved Motion Rate Control of Manipulator and Human Prostheses", IEEE Trans. Man-Machine Systems, MMS-10(2), pp.47-53, June, 1969.
- [3] N. Takahashi, T. Ikeda, N. Tagawa, "A High Sample Rate Control System Using a DSP based Numerical Calculation Engine", Proc. IEEE conf. Robotics and Automation, 1989.
- [4] James H. Graham, "Special Computer Architecture for Robotics: Tutorial and Survey", IEEE Trans. Robotics and Automation, vol 5, No 5, Oct., 1989.
- [5] K.S. Fu, R.C. Gonzalez, C.S.G. Lee, "ROBOTICS - Control, Sensing, Vision and Intelligence. 2nd Ed. McGraw-Hill.
- [6] Robert J. Schilling, "Fundamentals of Robotics - analysis and control", 1990, Prentice-Hall.
- [7] Industrial Micro Robot System Model RV-M1 Instruction Manual, Mitsubishi Electric Corporation.
- [8] 노철래, "DSP를 이용한 원격 로봇의 제어 시스템 개발에 관한 연구", 1991년 석사학위논문, KAIST.
- [9] DSP56000/DSP56001 Digital Signal Processor User's Manual, Motorola.
- [10] DSP56000ADS User's manual, Motorola.