

## 인터넷을 이용한 원격 로봇 제어기의 개발

### The development of the remote robot controller using the internet

\*임재환\*, 이종수\*\*, 최경삼\*\*\*

\*홍익대학교 대학원 전기제어공학과(Tel : +82-2-320-1493)

\*\*홍익대학교 전자전기공학부(Tel : +82-2-320-1669)

\*\*\*홍익대학교 전자전기공학부(Tel : +82-2-320-1120)

**Abstracts** We propose a remote controller for a SCARA typed direct drive manipulator with two degrees-of-freedom(DOF). A remote controller system for SCARA robot of DDA is designed using a 2 DSP (TMS320c31) board and Winsock(Internet program class library supplied by Microsoft). The design objective of the system is to implement real time dynamic control algorithms which have been tested only by simulations so far and remote control regardless of the distance between user and robot. Because this system runs on Win95, we developed a VxD program to communicate with DSP controller.

**Keywords** DSP, Winsock, Internet, remote, VxD.

#### 1. 서론

로봇은 자동화 산업 사회를 이끌어 나가는데 있어서 아주 중요한 역할을 담당하고 있으며, 공정 자동화의 확산에 따라 산업용 로봇은 다양한 분야에서 널리 사용되고 있다. 현재 로봇의 운영은 작업자가 로봇이 사용되는 생산 현장에 상주하며 관리하는 형태로 되어 있다. 그러나 이러한 거리의 제약은 실제 자동화의 실현에 큰 장애가 된다. 현재 우리는 컴퓨터 네트워크의 급속한 발달로 인터넷을 통해 거리에 상관 없이 세계 어디와도 데이터를 주고 받을 수 있다. 이러한 기술을 로봇 제어에 응용함으로써 거리의 제약을 극복하여 능률적 공장 자동화를 이루어 생산 현장의 효율성을 증가시킬 수 있다.

또한, 로봇 제어기는 점차 고정밀화, 고속화를 추구하고 있다. 하지만 현재의 산업용 로봇은 운동방식식이 비선형이며, 마찰력의 영향 및 작업중의 부하변동과 같은 외란이 존재하므로, 이들 비선형 성분을 보상하고, 파라미터의 변동에 대응하지 않는 한 그 성능도 개선될 수 없다. 이러한 문제에 대해 지금까지 많은 연구가 진행되어 왔고 이에 따라 고도의 일을 수행하기 위한 많은 알고리즘이 개발되고 컴퓨터 시뮬레이션을 통해 증명되었다. 그러나 이러한 우수한 알고리즘이 개발되어 왔음에도 불구하고 실제 산업용 로봇에 적용된 예는 적다. 이러한 이유는 산업용 로봇이 큰 기어비와 관절의 마찰 등으로 정확한 로봇의 동적운동방정식을 알아내기 어렵고 또한 알고리즘을 실제 로봇에 적용할 하드웨어 시스템이 발전하지 못함에 있다.

이에 본 논문에서는 위 두 가지 문제를 해결하기 위해

Winsock class 라이브러리를 이용하여 Internet 프로그램을 구현하였고 제어기에 실시간 계산 능력을 부여하기 위해 2대의 DSP board로 제어기를 구성하여 제어 알고리즘을 병렬 처리 하도록 하여 시스템의 성능을 향상 시켰다.

#### 2. 직접 구동형 로봇 구성

##### 2.1. 지원

본 연구에서 소개하는 로봇 매니퓰레이터의 기구학적 형태는 그림1과 같다. 이 로봇 매니퓰레이터는 2개의 NSK Mega Torque Motor, 2개의 Motor 구동 Unit, 2개의 듀랄루민 관절로 구성되어 있다. 그리고, NSK Mega Torque Motor의 사양은 표 1, 각각의 관절의 재원은 표2와 같다.

표 1 NSK Mega Torque Motor Specifications

Specification	Model RS 1410	Model RS 0608
Maximum Torque	245 N · m	39 N · m
Rotor Inertia	0.2675 Kg · m <sup>2</sup>	0.0075 Kg · m <sup>2</sup>
Maximum Speed	1.16 rev/s	1.16 rev/s
Mass	73.0 Kg	14.0 Kg
Resolver resolution	153600 pulse/rev	153600 pulse/rev

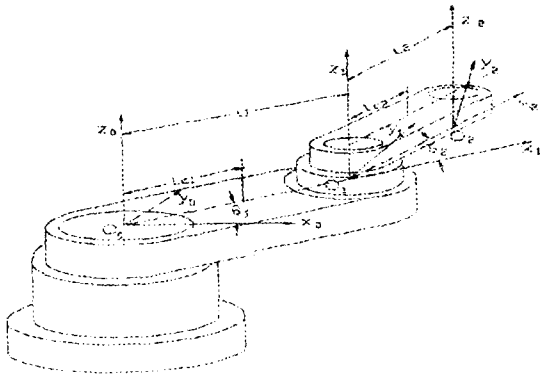


그림 1 Kinematic configuration of Hong Ik Manipulator with DH Coordinate Frame Assignment

표 2 Physical Parameters of Hong Ik Manipulator

Symbol	Value	Definition
$I_1$	0.3644755 [Kg m <sup>2</sup> ]	Lower Link Inertia
$I_2$	0.087443 [Kg m <sup>2</sup> ]	Upper Link Inertia
$I_3$	0.2675 [Kg m <sup>2</sup> ]	Rotor Inertia of the Lower Motor
$I_{4r}$	0.0075 [Kg m <sup>2</sup> ]	Rotor Inertia of the Upper Motor
$I_{4s}$	0.040 [Kg m <sup>2</sup> ]	Stator and Housing Inertia of the Upper Motor
$I_p$		Payload Inertia
$M_1$	13.02 [Kg]	Lower Link Mass
$M_2$	6.37366 [Kg]	Upper Link Mass
$M_3$	73.0 [Kg]	Lower Motor Mass
$M_4$	14.0 [Kg]	Upper Motor Mass
$M_p$		Payload Mass
$L_1$	0.37 [m]	Lower Link Length
$L_2$	0.25 [m]	Upper Link Length
$L_{e1}$	0.1632 [m]	Distance from c.g. of the Lower Link to the Lower Motor Axis
$L_{e2}$	0.125 [m]	Distance from c.g. of the Upper Link to the Upper Motor Axis

## 2.2. 기구학

로봇 매니퓰레이터의 기구학에 일반적으로 이용되는 Denavit Hartenberg 표시법으로 기구학을 구하면 다음과 같다.

$$\begin{aligned} X &= L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \\ Y &= L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \end{aligned} \quad (1)$$

(x,y)좌표가 주어질 때, 각 관절의 각도는 구하는 역기구학은 다음과 같다.

$$\begin{aligned} \theta_2 &= \text{atan2}(D, C) \\ \theta_1 &= \text{atan2}(y, x) - \text{atan2}(L_2 \sin \theta_2, L_1 + L_2 \cos \theta_2) \end{aligned} \quad (2)$$

여기서,  $C = \frac{r^2 - L_1^2 - L_2^2}{L_1 + L_2 \cos \theta_2}$ ,  $D = \pm \sqrt{1 - C^2}$ ,  $r^2 = x^2 + y^2$ 이다.

## 2.3. 동역학

이 로봇 매니퓰레이터의 병진 에너지와 운동에너지를 구하여 Lagrangian을 취하면 다음과 같은 동적 운동 방정식을 구할 수 있다.

$$\bar{M}(\theta(t)) \dot{\omega}(t) + \bar{v}(\theta(t), \omega(t)) = \bar{\tau}(t) - \bar{f}(t, \theta(t), \omega(t), \tau(t)) \quad (3)$$

여기서  $\bar{M}(\theta(t))$ 는 대칭형 positive definite 일반화된 관성 행렬,  $\bar{v}(\theta(t), \omega(t))$ 는 코리올리 가속도와 원심 가속도에 기인한 벡터,  $\bar{\tau}(t)$ 는 액츄에이터에 의하여 가해지는 입력 Torque,  $\bar{f}(t, \theta(t), \omega(t), \tau(t))$ 는 마찰 Torque 벡터이다. 식(1)에서의 각 항의 요소들은 다음과 같다.

$$\bar{\tau}(t) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \end{bmatrix} \quad (4)$$

관성 행렬 :

$$\bar{M}(\theta(t)) = \begin{bmatrix} p_1 + 2p_3 \cos \theta_2 & p_2 + p_3 \cos \theta_2 \\ p_2 + p_3 \cos \theta_2 & p_2 \end{bmatrix} \quad (5)$$

여기서,  $p_1, p_2, p_3$  항은 다음과 같이 주어진다.

$$p_2 = I_2 + I_{4r} + I_p + L_{e1}^2 M_2 + L_{e2}^2 M_p$$

$$p_1 = I_1 + I_3 + I_{4s} + L_{e1}^2 M_2 + L_{e2}^2 (M_2 + M_4 + M_p) + p_2$$

$$p_3 = L_1 L_{e2} M_2 + L_1 L_2 M_f$$

코리올리 벡터 :

$$\bar{b}(t) = \begin{bmatrix} -\omega_2 (2\omega_1 + \omega_2) p_3 \sin \theta_2 \\ \omega_1^2 p_3 \sin \theta_2 \end{bmatrix} \quad (6)$$

마찰항의 행은 다음과 같다.

$$\bar{f} = \begin{bmatrix} f_{r1} \\ f_{r2} \end{bmatrix} \quad (7)$$

$$f_{ri} = \begin{cases} Th_i \text{sign}[\omega_i(t)] & \text{if } |\omega_i(t)| > 0 \\ Th_i \text{sign}[\tau_i(t)] & \text{if } |\omega_i(t)| = 0 \text{ and } |\tau_i(t)| > Th_i \\ \tau_i(t) & \text{if } |\omega_i(t)| = 0 \text{ and } |\tau_i(t)| \leq Th_i \end{cases}$$

여기서,  $Th_i$  쿨롱 마찰력 torque의 threshold 값이다.

## 3. 원격 구동팔 제어기

### 3.1. 원격 구동팔 제어기의 구성

전체 시스템의 구조는 그림 1과 같다.

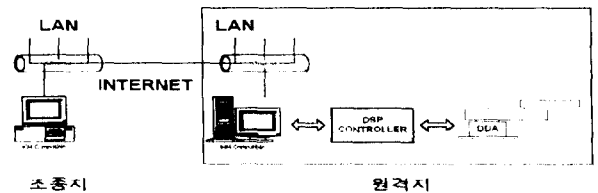


그림 2. 전체 Block Diagram

DSP Controller는 보다 빠른 연산 처리 능력 위해 그림 2와 같이 2개의 DSP Chip TMS320c31-50MHz을 이용하여 병렬 구조로 제작하였다.

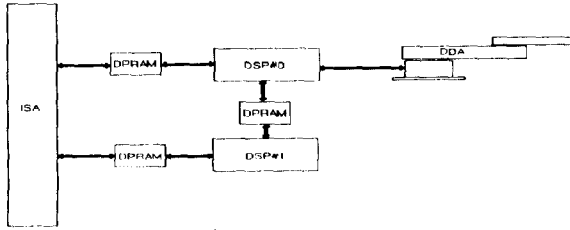


그림 3. 전체 DSP Controller Block Diagram

### 3.1.1 DSP Board의 구성

DSP#0 board의 구성은 표 3과 같다.

표 3. 제어기의 주요 사양

항목	세부사항
DSP	TMS320c31-50MHz
MEMORY	32K*32 ROM& 32K*32 SRAM& 2K*16 DPRAM&
# of COUNTER	2 Cannel (32bits)
# of Digital I/O	16 Inputs & 8 Outputs
Serial port	

DSP#1 board의 구성은 표 4와 같다.

표 4. 제어기의 주요 사양

항목	세부사항
DSP	TMS320c31-50MHz
MEMORY	32K*32 ROM& 32K*32 SRAM& 2K*16 DPRAM&

PC와의 인터페이스 부분은 보다 빠른 통신을 위하여 직렬 통신을 사용하지 않고 Dual-port RAM을 이용하였고 TMS320c31에서 2개의 interrupt와 PC에서의 2개의 Service Requests를 할당하였다. 각 board의 로직은 Xilinx와 VHDL를 이용하여 설계했다.

### 3.1.2 VxD 프로그래밍

어떤 하드웨어를 Win95에서 동작 시키기 위해서는 그 하드웨어에 맞는 VxD를 제작하여야 한다. 하드웨어 제어의 4가지 기본적인 프로그램 요소. 즉, I/O 동작, 인터럽트 핸들링, 특정 메모리 접근, DMA를 Win95에서 구현하기 위해서는 VMM, VPICD, VDMA에서 제공하는 적절한 service 루틴을 이용해야 한다. 본 논문에서는 DDK와 VtoolsD를 사용하여 VxD를 만들었다.

### 3.1.2 Winsock 프로그래밍

Winsock은 마이크로 소프트사가 인터넷 통신 프로그램 위해 제공하는 class로 본 논문에서는 이 class를 사용하여 Server와

Client를 구성하였다. 사용 방법은 표 5와 같다.

표 5. Winsock 사용 순서

Server	Client
CSocket sockSrvr;	CSocket sockClient;
sockSrvr.Create(nPort);	sockClient.Create();
sockSrvr.Listen();	
	sockClient.Connect(strAddr,nPort);
sockSrvr.Accept(sockRecv);	

### 3.2 제어기의 운용

Client는 Server에 로봇의 작업 위치 정보를 주면 Server 컴퓨터는 DSP#1에 경로 계획에 필요한 정볼트 DPRAM를 통해 전달한다. DSP#1은 이 정보를 가지고 1ms마다 경로를 발생 시키고 이 결과를 DSP#0에 전달한다. DSP#0는 이 정보를 가지고 DSP#1의 계산에 무관하게 병렬로 로봇 제어를 실행하고 그 결과를 DPRAM를 통해 Server에 전달한다.

## 4. 결론

본 논문에서는 2대의 DSP와 병렬 처리, Winsock를 이용하여 원격 제어가 가능한 직접 구동 로봇 팔의 동작 제어기 개발하였고 성능 및 실용성을 보이기 위해 직접 구동형 로봇 팔에 응용하여 보았다. 이 제어기는 다른 상용의 제어기에 비해 보다 많은 Memory와 Host와의 통신을 DPRAM을 이용하여 보다 빠르게 할 수 있는 장점이 있다. 또한 Win95상에서 제어가 가능하도록 이 제어기에 맞는 VxD를 설계하였다.

이 제어기는 병렬 처리를 통해 얻어진 향상된 계산 능력은 실시간 제어가 필요한 여러 응용 분야에도 적용 된다.

이 제어기를 개발함에 있어서 현대 제어 이론에 바탕을 둔, 로봇의 동력학을 고려한 토오크 제어 알고리즘의 효율성을 실제로 평가하는 도구로 활용될 것을 기대한다.

## 5. 참고 문헌

- [1] J.J.Craig, 'Introduction to Robotics-Mechanics and Control', Addison Wesley, 1989.
- [2] K.S.Fu, R.C.Gonzalez, C.S.G.Lee, 'Robotics-Control, Sensing, Vision, and Intelligence', McGraw Hill, 1987.
- [3] Walter.Oney, 'Systems Programming for Windows95', MicroSoft Press, 1995.
- [4] 'TMS320C3x User's Guide', Texas Instruments, 1995.
- [5] 'TMS320C3x Floating-Point DSP Optimizing C Compiler User's Guide', Texas Instruments, 1995.
- [6] 이종수, "로봇의 하이브리드 제어기 설계에 관한 연구", 박사 학위논문, 서울 대학교,1987.

본 연구는 97년 홍익대학교 교내 연구비에 의해 연구된 결과임