

論文98-35S-2-7

# DSP를 이용한 SCARA 로봇의 강인한 동적 제어 시스템 실현

## (Implementation of a Robust Dynamic Control System for SCARA Robot Using DSPs)

李章明\*, 朴興仁\*

(Jang-Myung Lee and Heung-In Park)

## 요 약

강인한 동적제어 알고리즘을 수행하기 위한 스카라 로봇 제어를 설계하였다. 본 연구는 관절제어기에서의 DSP 사용, 호스트 컴퓨터(Host Computer)와 네 개의 관절제어기의 인터페이스 그리고 관절제어기와 네 개의 서보 드라이브를 연결하는 인터페이스의 설계에 초점을 둔다. 스카라 로봇 몸체와 서보 드라이브는 상품화된 제품을 사용하였으며 각각의 관절제어기는 마더 보드(Mother Board)를 통해 하드웨어적으로 연결되고 공유메모리를 통해 소프트웨어적으로 연결된다. 마더 보드는 PC/104 버스 구조의 슬롯을 통해 보드상에서 관절제어기들을 연결하도록 설계되었으며, 공유메모리는 관절제어기들과 호스트 컴퓨터가 공유할 수 있도록 하여 가상적으로 전체시스템을 하나로 묶어준다. 시스템의 수행 능력과 성능을 검증하기 위해, 보다 빠르고 정확한 제어를 위한 강인한 동적제어 알고리즘을 제안하고 실험을 통해 확인하였다. 강인한 동적제어 알고리즘은 기본적으로 역 동적 알고리즘과 PID 보상기의 결합으로 이루어진다. 유도된 SCARA 로봇의 동력학 방정식에 근거한 역 동적 알고리즘은 본 시스템에서 0.3 msec 이내에 수행되며, 이 알고리즘은 모델링 에러와 시변 인자 때문에 고속 고정밀의 작업 환경에서는 정확도가 떨어짐을 알 수 있었다. 따라서, 제어수행 능력의 강인함을 위해 가변 PID 알고리즘과 역 동적 알고리즘을 결합한 강인한 동적제어 알고리즘을 고안하였다. 제안된 알고리즘에 의한 실험 데이터가 주어졌으며 PID와 역 동적 알고리즘으로 부터 얻어진 결과와 비교 분석하였다.

## Abstract

A control system for SCARA robot is designed for implementing a robust dynamic control algorithm. This study focuses on the use of DSPs in the design of joint controllers and interfaces in between the host controller and four joint controllers and in between the joint controllers and four servo drives. The mechanical body of SCARA robot and the servo drives are selected from the commercially available ones. The four joint controllers, assigned to each joint one by one, are combined into a common system through a mother board hardwarewise and through the global memory softwarewise. The mother board is designed to connect joint controllers onto the board through the slots adopting PC/104 bus structures. And, the global memory stores the common data which can be shared by joint controllers and the host computer directly, which virtually combines the whole system into one. To demonstrate the performance and efficiency of the system, a robust inverse dynamic algorithm is proposed and implemented for a faster and more precise control. The robust inverse dynamic algorithm is basically derived from an inverse dynamic algorithm and a PID compensator. Based upon the derived dynamic equations of SCARA robot, the inverse dynamic algorithm is initially implemented within 0.3 msec of the control cycle in this system. The algorithm is found to be not accurate enough for the high speed and precision tasks due to inherent modelling errors and time-varying factors. Therefore, a variable PID algorithm is combined with the inverse dynamic algorithm to support robustness of control performance. Experimental data for the proposed algorithm are presented and compared with the results obtained from PID and inverse dynamic algorithms.

\* 正會員, 釜山大學校 電子工學科  
(Dept. of Elec. Eng., Pusan National Univ.)

接受日字: 1997年5月29日, 수정완료일: 1998年1月9日

I. 서 론

II. 시스템의 개요

산업계는 인력난과 더불어 생산성 증대를 위하여 로봇의 사용을 지속적으로 추진하고 있다. 로봇의 활용을 증대시키기 위해서 신뢰성이 높으며 환경변화에 강인하고 고속, 고정밀의 작업능력을 갖춘 로봇 매니플레이터의 개발이 필요할 것이다. 산업용 로봇은 말단 공구장치의 움직임을 제어하여 대상 물체의 운반, 조립, 페인팅 또는 용접과 같은 다양한 작업환경에서 응용되고 있다. 이러한 작업을 효율적으로 정밀하게 수행하기 위해서는 우수한 동적 제어 알고리즘과 고속 고정도의 로봇 제어시스템을 필요로 한다. 현재까지 우수한 성능을 갖는 많은 제어 알고리즘들이 개발되었으나<sup>[2,4-6]</sup>, 대부분 복잡한 계산을 요구하는 동적 제어 알고리즘에 기반을 두고 있어 하드웨어로 실현하는 기술이 충분히 뒤따르지 못하여 그 알고리즘의 타당성을 시뮬레이션을 통하여 검증하는 정도로 그치는 경우가 일반적이었다<sup>[3,4]</sup>.

본 연구에서는 실시간에 다량의 계산이 요구되어지는 제어 알고리즘들을 수행할 수 있는 로봇 매니플레이터의 제어기를 설계 및 제작하였다. 각 축 제어를 위한 관절제어기에 DSP(Digital Signal Processor)를 사용하여 알고리즘의 수행 속도를 높였으며, 호스트 컴퓨터를 사용하여 다공정의 산업현장에서 로봇의 제어 환경을 가상적으로 실현하는 OLP(Off Line Programming) 소프트웨어와 제어기를 인터페이스하여 온 라인으로 직접 로봇을 제어할 수 있도록 제어시스템을 구축하였다.

본 시스템의 성능 검증 및 평가를 위해서 고전적 제어 이론인 PID, 역 동적 알고리즘(Inverse Dynamic Algorithm) 그리고 파라미터의 최적화를 위해 전문가 시스템에 바탕을 둔 퍼지이론과 정밀 동조 기법을 도입한 강인한 동적제어 알고리즘(Robust Dynamic Control Algorithm)을 실현하였다<sup>[4,14]</sup>. 강인한 동적제어 알고리즘은 역 동적 알고리즘에 PID 보상기를 결합시켜 모델링 오류, 플랜트 파라미터의 변화 그리고 실제의 시스템에 존재하는 외란과 같은 불확정성 요인들을 실시간에 보상하도록 하였다. 그리고 각각의 알고리즘을 이용한 실험 결과와 적용 알고리즘들의 PID 제어에 대한 개선치를 제시하였다.

1. 전체 시스템

본 연구의 제어시스템 구성은 그림 1과 같다. 전체 시스템은 호스트 컴퓨터(586-PC), DSP를 사용하여 제작된 4개의 관절제어기, 호스트 컴퓨터와 관절제어기 그리고 관절제어기 사이의 인터페이스를 담당하는 마더보드, 관절제어기와 서보 드라이브 사이의 인터페이스 보드, 서보 드라이브 그리고 스카라 로봇으로 이루어진다. 관절제어기의 프로세서로는 계산속도가 빠른 DSP를 사용하였다. DSP는 Floating-point 계열보다는 경제적으로 유리한 Fixed-point 계열로 선택하였고 그 중에서 가격대비 성능이 뛰어난 TMS320C50을 사용하였다<sup>[8-10]</sup>. 로봇의 각 축을 독립적으로 고속 제어하기 위하여 각 축별의 관절제어기에 독립적으로 DSP를 사용하였고, 각 축들의 공유데이터 저장과 통신을 위해서 16비트 버스를 통하여 병렬로 연결된 공유메모리(Global Memory)를 사용하였다. 각 관절제어기의 DSP가 제어할 데이터를 외부에서 입력받고, 제어한 결과를 출력하기 위하여 공유메모리는 16비트 버스로 호스트 컴퓨터에 연결되었다. 또한, 호스트 컴퓨터는 각 관절제어기의 이상 상태나 기타의 통신을 위해 각 관절제어기와 직렬 통신이 가능하도록 하였다. 서보 드라이브는 디지털형의 FARA CSD를 사용하였다. 로봇 매니플레이터는 스카라형의 FARA-SM5를 사용하였다.

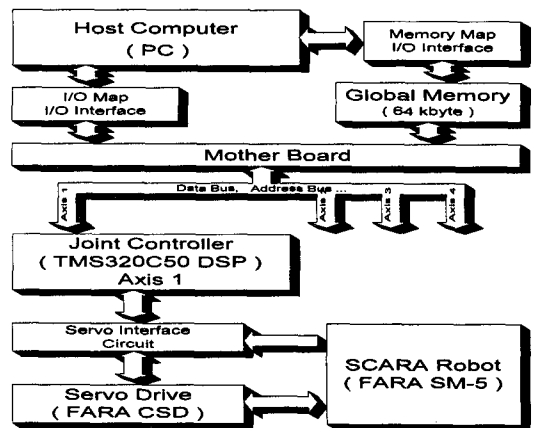


그림 1. 전체 시스템의 블록도  
Fig. 1. Schematic diagram of overall system.

2. 관절제어기

관절제어기는 그림 2에 나타난 것처럼 프로세서로는 TI사의 Fixed-point DSP인 TMS320C50을 사용하였다. TMS320C50은 40 MHz 클럭 사용시 하나의 Fixed-point Instruction을 매 50 nsec(20MIPS) 마다 수행하며 64k×16비트의 프로그램과 데이터 영역을 각각 어드레싱할 수 있고 64k의 I/O를 가질 수 있다. 그리고 내부에 9k×16비트의 프로그램/데이터 RAM과 2k×16비트의 Boot ROM을 가지고 있다. 내부의 레지스터는 32비트로 이루어지며 16×16비트 병렬 곱셈을 한 사이클에 수행한다.

관절제어기에 사용하고 있는 메모리는 프로그램 메모리로 액세스(Access) 시간이 20 nsec인 32k×16비트 SRAM을 사용하며, 데이터 메모리는 64k 중에서 공유메모리의 영역을 뺀 32k×16비트를 사용하는데 역시 액세스 시간이 20 nsec인 32k×16비트 SRAM을 사용한다. 위와 같이 구성된 메모리는 빠른 액세스로 DSP가 지연시간이 없이 데이터를 읽고 쓸 수 있도록 해 준다.

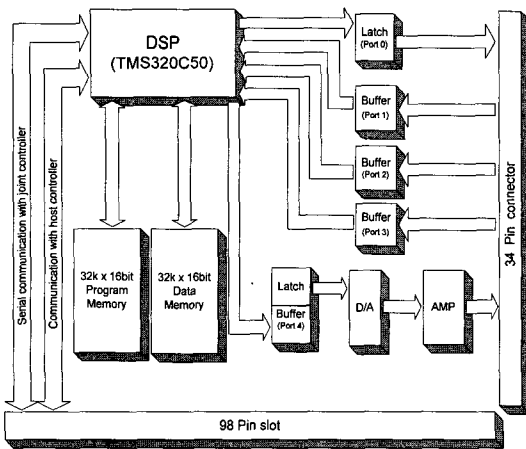


그림 2. 관절 제어기의 블록도  
Fig. 2. Schematic diagram of joint controller.

I/O 영역에는 분해능이 12비트인 A/D, D/A 변환기를 내장하고 있으며, 디지털 입출력이 34핀 커넥터를 통해 서보 드라이브와 연결된다. D/A 변환기로는 제어입력을 아날로그로 변환하여 서보 드라이브로 출력한다. 디지털 출력 포트로는 서보 on/off, 서보 리셋의 신호를 출력하고, 입력 포트로는 위치의 피드백값, 서보의 이상발생신호, 로봇 링크의 리밋(Limit) 신호를 입력으로 받아들인다.

3. 마더보드 (Mother Board)

4개의 관절제어기와 호스트 컴퓨터와의 통신을 관리하기 위해 마더보드가 존재한다. 그림 3은 호스트 컴퓨터와 관절제어기의 인터페이스를 나타내며, 마더보드의 역할을 보여준다. 마더 보드는 공유메모리와 조정회로, 직렬통신 조정회로, EPROM으로 구성된다. 공유메모리는 32k×16비트의 SRAM으로 액세스시간은 20 nsec이다. 이 공유메모리는 주로 호스트 컴퓨터가 오프라인(Off-Line)에서 학습 및 교시(Learning/Teaching)한 데이터를 받아 관절제어기에 전달하고 관절제어기에서 보내온 제어 결과를 호스트 컴퓨터에 보낼 수 있도록 저장하는 역할 뿐만 아니라 관절제어기 사이의 데이터 공유를 위한 목적으로 사용된다. 호스트 컴퓨터와는 50핀 커넥터를 통하여 연결되는데 이것은 16비트 데이터 버스, 16비트 어드레스 버스 (Address Bus), 그리고 제어신호(Chip Select Signal, Bus High Enable Signal, Memory Read/Write Signal) 등으로 구성된다. 호스트 컴퓨터에서는 메모리 맵 방식으로 공유메모리를 액세스한다.

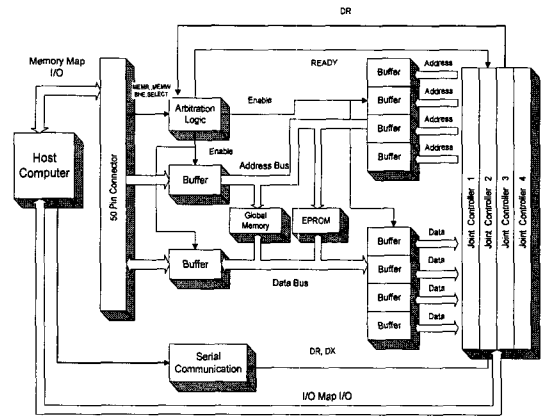


그림 3. 마더 보드의 블록도  
Fig. 3. Schematic diagram of mother board.

조정회로(Arbitration Circuit)는 호스트 컴퓨터와 4개의 관절제어기가 공유메모리를 액세스할 때 어드레스와 데이터 그리고 제어신호들의 충돌을 조정하기 위한 회로이다. 이 조정회로는 호스트 컴퓨터에 우선 순위를 부여하고 나머지 1축에서 4축까지의 관절제어기에 순서대로 우선 순위를 준다.

직렬통신 조정회로는 RS-232C 통신이 충돌없이 이루어지도록 멀티플렉스/디멀티플렉스를 이용하여 조정

하는 부분이다. 이 직렬통신은 PC가 Polling 방식으로 관절제어기와 통신을 하게 한다. EPROM은 로봇을 제어하기 위한 관절제어기의 프로그램들을 전원을 넣는 순간에 다운로드(Down Loading)해 주기 위한 것이다.

4. 관절제어기와 서보 드라이브의 인터페이스

관절제어기와 서보 드라이브를 연결하는 인터페이스 보드는 그림 4와 같으며 관절제어기의 34핀 커넥터와 서보 드라이브의 36핀 커넥터를 연결한다. 16비트 병렬 up/down 카운터, 서보 on/off 스위치, 서보 리셋 스위치, 릴레이 그리고 리밋 센서 인식을 위한 회로로 구성된다<sup>[11-13]</sup>.

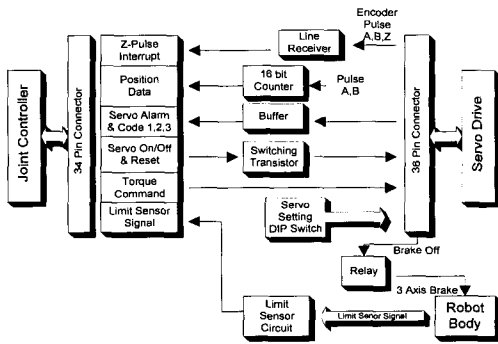


그림 4. 관절제어기와 서보드라이브의 인터페이스  
Fig. 4. Interface in between a joint controller and a servo drive.

16비트 병렬 up/down 카운터는 모터의 엔코드 센서에서 검출되는 90° 위상 차가 있는 A, B상을 이용하여 모터의 회전각도를 검출한다. 그 값은 관절제어기에 피드백되어 로봇 제어를 가능하게 한다. 서보 on/off와 서보 리셋 스위치는 인터페이스 보드를 통해 서보 드라이브를 동작 가능하게 한다. 즉, 호스트 컴퓨터는 관절제어기를 통해 서보 드라이브로 on/off와 리셋 명령을 보낼 수 있도록 하였다. 리밋 센서 회로는 각 로봇 팔의 영점 복귀를 위해 로봇에 부착되어 있는 포토(photo) 인터럽터의 신호를 인식한다. 또한 영점 복귀를 위해 각 축 모터의 Z상 신호를 관절제어기의 인터럽터 3에 연결하였다. 그 외에 모터의 비례 제어, 역회전 구동 금지, 정회전 구동 금지, 역방향 전류 제한, 정방향 전류제어를 할 수 있는 스위치가 인터페이스 보드에 존재한다. 표 1은 관절제어기와 서보 드라이브 사이의 신호에 대한 포트 어드레스, 신호 비트

수 그리고 입 출력 관계를 나타낸다.

표 1. 관절제어기와 서보 드라이브간의 통신 신호

Table 1. Communication signals between a joint controller and a servo drive.

Address	Bit	Signal	In/Out (DSP)
50H	1	Servo Reset	Output
50H	1	Servo On/Off	Output
51H	3	Limit Sensor Signal	Input
51H	4	Servo Alarm	Input
52H	16	Feedback Position	Input
	7	Reserved, Ground	Input/Output
54H	Analog	Current Command	Output
INT3	1	Z-Pulse Interrupt	Input

III. 관절제어기의 제어 흐름

그림 5는 관절제어기의 제어흐름을 나타내고 있다. 그림에서와 같이 초기 구동은 제어 프로그램을 호스트 컴퓨터에서 관절제어기로 옮기기 위한 리셋으로 이루어진다. 초기화를 하기 전에 호스트 컴퓨터가 공유메모리에 관절제어기용 제어 알고리즘을 다운로드해 놓아야 한다. 관절제어기의 DSP를 리셋하면 Boot Loader가 동작하여 공유메모리에 있는 관절제어기용 프로그램을 관절제어기의 프로그램 메모리로 옮긴다. 이 과정이 끝나면 DSP에서는 프로그램을 시작하므로, 스타트 신호를 기다리는 일정한 대기루프가 있다. 이 신호는 호스트 컴퓨터에서 4개의 관절제어기를 동시에 시작시키기 위한 것으로, DSP의 Branch control input의 동작을 행하는 BIO 단자에 입력된다.

로봇 제어를 위한 관절제어기의 샘플링 주기는 1 msec 이며 타이머(timer) 인터럽터에 의해 조정된다. 관절제어기와 호스트 컴퓨터와의 통신은 1 msec 주기의 반복문을 계수하여 16번을 계수하였을 때마다, 즉 16 msec 마다 이루어진다. 공유메모리의 일정영역에 통신 파라미터를 설정하여, 호스트 컴퓨터와 관절제어기가 정보를 주고 받을 수 있도록 하였다. 통신 프로그램과 함께 서보 실행프로그램과 궤적 보간 프로그램도 16 msec 주기로 실행된다. 서보 실행프로그램은 호스트 컴퓨터가 내린 서보 명령에 따라 로봇제어기를 동작시키며, 궤적 보간 프로그램은 16 msec 간격의 위치명령을 1 msec 간격의 위치계적으로 보간하는 것

이다. 보간 방식은 선형 보간을 사용한다. 영점복귀 작업은 호스트 컴퓨터로부터 영점복귀 명령이 내려지면 시작된다. 영점복귀작업은 온라인으로 제어를 수행할 때와는 달리, 복귀 제어를 스스로 생성하여 영점으로 복귀하게 한다.

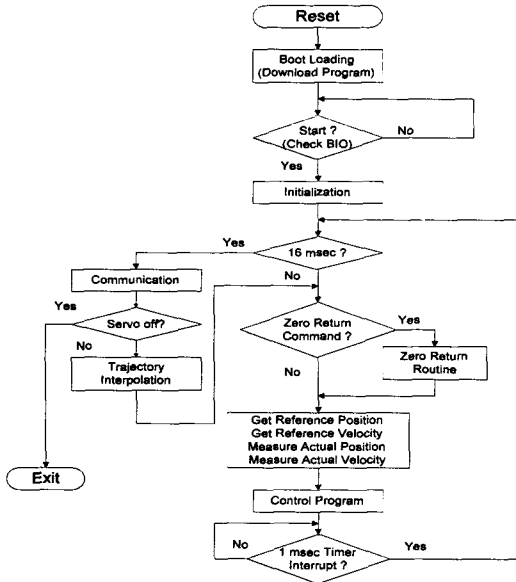


그림 5. 관절 제어기의 제어 흐름  
Fig. 5. Control flowchart of joint controller.

IV. 스카라 로봇의 동역학

스카라 로봇의 외형과 D-H(Denavit-Hartenberg)에 의한 좌표계는 그림 6과 같다. 링크 파라미터들은 표 2와 같다<sup>[7]</sup>.

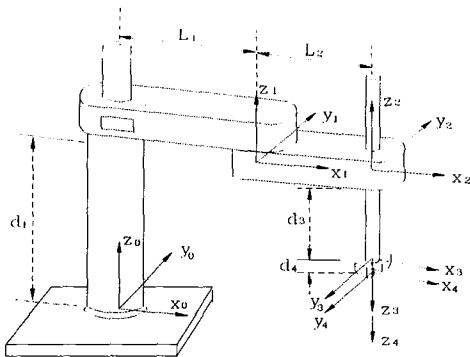


그림 6. 스카라 로봇  
Fig. 6. SCARA robot.

표 2. 링크 파라미터  
Table 2. Link parameters.

Joint	$a_i(\text{mm})$	$\alpha_i(^{\circ})$	$d_i(\text{mm})$	$q_i$
1	350	0	505	$q_1$
2	260	0	0	$q_2$
3	0	180	$d_3$	0
4	0	0	20	$q_4$

표2를 이용한 동차 변환 행렬  $H_0^4$ 를 구하면 식 (1)과 같다. 이에따라 매니플레이터의 운동방정식을 유도하면 식(2)와 같다.

$$H_0^4 = \begin{bmatrix} c_{1+2-4} & s_{1+2-4} & 0 & L_1c_1 + L_2c_{1+2} \\ s_{1+2-4} & -c_{1+2-4} & 0 & L_1s_1 + L_2s_{1+2} \\ 0 & 0 & -1 & d_1 + d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} L_1s_1 - L_2s_{12} & -L_2s_{12} & 0 & 0 \\ L_1c_1 + L_2c_{12} & L_2c_{12} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (2)$$

여기서  $c_1 = \cos \theta_1$ ,  $s_1 = \sin \theta_1$ ,  $c_{1+2-4} = \cos(\theta_1 + \theta_2 - \theta_4)$ ,  $s_{12} = \sin \theta_1 \sin \theta_2$  이다.

역 동적 알고리즘을 적용하기 위해서는 로봇 몸체의 동력학 방정식과 모터의 동력학 방정식이 요구된다. 먼저 일반적인 n-link 로봇의 동력학 모델은 Lagrange-Euler 동력학 방정식에서 다음과 같이 정리할 수 있다.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3)$$

여기서  $D(q)$ 는  $n \times n$  관성 행렬,  $D(q) = [d_{ij}(q)]_{n \times n}$ ,  $d_{ij}(q) = d_{ji}(q)$ ,  $\ddot{q}$ 는  $n \times 1$  가속도 벡터,  $C(q, \dot{q})$ 는 centrifugal 및 Coriolis 항,  $\dot{q}$ 는  $n \times 1$  속도 벡터,  $q$ 는  $n \times 1$  자세 벡터,  $G(q)$ 는  $n \times 1$  중력 벡터 그리고  $\tau$ 는  $n \times 1$  토크 벡터를 나타낸다.

$N(q, \dot{q}) = C(q, \dot{q}) + G(q)$  및  $N(q, \dot{q}) = [n_i(q, \dot{q})]_{n \times 1}$ 로 정의하면,  $D(q)$ ,  $N(q, \dot{q})$ 은 로봇의 기구학적인 파라미터들을 사용하여 다음과 같이 표시된다<sup>[1]</sup>.

$$d_{11} = (I_{z1} + I_{z2} + I_{z3} + I_{z4}) + \left(\frac{m_1}{4} + 2m_3 + m_4\right)L_1^2 + \left(\frac{m_2}{2} + 3m_3 + m_4\right)L_1L_2\cos q_2 + \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2$$

$$d_{12} = (I_{z2} + I_{z3} + I_{z4}) + \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2 + (2m_3 + m_4)L_1L_2\cos q_2$$

$$\begin{aligned}
 d_{13} &= 0, \quad d_{14} = I_{z21}, \\
 d_{22} &= (I_{z22} + I_{z23} + I_{z24}) + \left(\frac{m_2}{4} + m_3 + m_4\right)L_2^2, \\
 d_{23} &= 0, \quad d_{24} = I_{z21}, \\
 d_{33} &= m_3 + m_4, \quad d_{34} = 0, \quad d_{44} = I_{z21}, \\
 n_1 &= \left(\frac{m_2}{2} - m_3\right)L_1L_2\dot{q}_1^2 - (4m_3 + 2m_4)L_1L_2\dot{q}_1\dot{q}_2, \\
 n_2 &= \left(\frac{m_2}{2} + m_3 + m_4\right)L_1L_2\sin q_2\dot{q}_1^2, \\
 n_3 &= (m_3 + m_4)g, \quad n_4 = 0, \\
 m_1 &= 3 \text{ kg}, \quad m_2 = 1.5 \text{ kg}, \quad m_3 = 1.5 \text{ kg}, \quad m_4 = 1.5 \text{ kg}, \\
 I_1 &= 0.821 \text{ kgm}^2, \quad I_2 = 0.136 \text{ kgm}^2, \\
 I_3 &= 0.008 \text{ kgm}^2, \quad I_4 = 0.0014 \text{ kgm}^2, \quad \text{and } g = 9.8 \text{ m/sec}^2.
 \end{aligned}$$

여기서  $C_i = \cos q_i$ ,  $L_i$  은 로봇 링크  $i$  의 길이,  $m_i$  는 로봇 링크  $i$  의 질량(kg) 그리고  $I_{zzi}$  는  $z$  축에 대한 로봇 링크  $i$  의 관성 모멘트(kg · m<sup>2</sup>)이다.

### V. 강인한 동적제어 알고리즘

산업용 로봇의 대부분은 선형시스템의 제어를 위해 이론적으로 잘 정립된 PID 제어 알고리즘을 사용하고 있다<sup>[7,15]</sup>. 이와 같은 알고리즘은 무 부하나 저속의 동작에서는 안정화되어 있고 만족스럽게 작업을 수행할 수 있지만, 부하가 있을 때나 고속의 동작에서는 관성, centrifugal, Coriolis 및 중력항이 활성화되어 정밀하지 않게 된다. 이러한 비선형성, 동역학적 Coupling의 영향과 마찰력, 외부의 조건 변화 및 동적 모델링의 오차로 인한 외란을 극복하는 강인한 제어를 만들기 위해서 역 동적 제어 알고리즘에 가변 PID 보상기를 결합한 강인한 동적제어 알고리즘을 제안한다.

매니플레이터의 동력학 방정식은 개개의 모터 측면에서는 외란으로 간주될 수 있다. 실제로 SCARA 로봇의 토크  $\tau$  는 감쇄기어를 통하여 모터로부터 전달되므로 모터의 동력학 방정식은 모터의 관성 및 Viscous friction을 고려하여 다음과 같이 표시된다.

$$J_m \ddot{q}_m + B_m \dot{q}_m = \tau_m - R\tau \tag{4}$$

여기서  $\dot{q}_m$ ,  $\ddot{q}_m \in R^4$  은 각속도와 각가속도를 각각 나타내고,  $J_m = \text{diag}\{ J_{mi}, i=1 \cdot \cdot 4\}$ ,  $B_m = \text{diag}\{ B_{mi}, i=1 \cdot \cdot 4\}$ ,  $R = \text{diag}\{ r_i, i=1 \cdot \cdot 4\}$

를 나타내며,  $r_i$  는  $i$  번째 관절의 기어비를 나타낸다. 그리고  $\tau_m$  은 모터에서 제공되는 토크를 나타낸다.

(4) 식의  $\tau$  에 (3)식을 대입하고  $q = R q_m$  인 관계식을 사용하여 로봇의 동력학을 모터의 변수  $q_m$  으로 나타내면 다음과 같다<sup>[5]</sup>.

$$[J_m + D_m(q_m)] \ddot{q}_m + [C_m(q_m, \dot{q}_m) + B_m] \dot{q}_m + G_m(q_m) = \tau_m \tag{5}$$

여기서  $D_m(q_m) = R D(R q_m) R$ ,  $C_m(q_m, \dot{q}_m) = R C(R q_m, R \dot{q}_m) R$ ,  $G_m(q_m) = R G(R q_m)$  이다.

비선형 성분들을 보상하고 모델링 오차 및 Coupling 항들을 외란으로 처리하여 선형의 오차방정식을 유도하기 위하여 역 동적 알고리즘과 PID 알고리즘의 결합에 의한 제어칙  $u$  는 다음과 같이 표시된다.

$$\begin{aligned}
 u = & [J_m + D_m(q_m)] \ddot{q}_m^d + [C_m(q_m, \dot{q}_m) + \\
 & B_m] \dot{q}_m + g_m(q_m) + K_P e + K_I \int e dt + K_D \dot{e}
 \end{aligned} \tag{6}$$

여기서  $K_P = \text{diag}\{K_{P1}, \dots, K_{P4}\}$ ,  $K_I = \text{diag}\{K_{I1}, \dots, K_{I4}\}$ ,  $K_D = \text{diag}\{K_{D1}, \dots, K_{D4}\}$  를 나타내고  $e = q_m^d - q_m$  으로 정의된다.

위의 제어 규칙을 블록 선도로 나타내면 그림 7과 같다. 제어칙  $u$  와  $\tau_m$  을 동일하게 두면 (5)식과 (6) 식으로 부터 오차 방정식은 다음과 같이 구해진다.

$$[J_m + D_m(q_m)] \ddot{e} + K_P e + K_I \int e dt + K_D \dot{e} = 0 \tag{7}$$

순서적으로  $\ddot{q}_m = \ddot{q}_m^d$  라고 가정하면  $\ddot{e} = 0$  이 되어

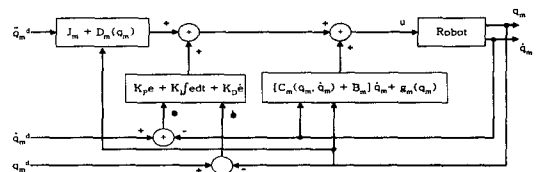


그림 7. 강인한 동적제어 알고리즘  
Fig. 7. Robust dynamic control algorithm.

$$K_P e + K_I \int e dt + K_D \dot{e} = 0 \tag{8}$$

의 식이 얻어지며, 이 오차 방정식은  $K_P$ ,  $K_I$ ,  $K_D$ 가 대각 행렬이므로 오차 방정식이 각각의 모터변수 ( $q_m$  및  $q_m^d$ )에 대해 독립적인 2차의 오차방정식으로 구해지며,  $K_{P_i}$ ,  $K_{I_i}$  및  $K_{D_i}$ 를 적절히 조절하여 안정도와 오차 특성을 만족시킬 수 있다.

동력학 방정식의 모델링 오차 및 외란 등의 특성으로 인하여 설계된 PID 이득 값이 (8)식의 오차 방정식을 만족시키지 못한다. 따라서, 가변 PID 제어를 퍼지법칙에 의해 설계하여 독립적인 모터제어의 실현을 통하여 로봇의 작용점에서의 궤적을 정확히 추종하도록 하였다. (7)식에서  $\ddot{e} = 0$ 로 가정하여 (8)식을 구하는 대신에  $\| \dot{e} \| = \epsilon$ 으로 가정한다. 그러면 (7)식은 다음과 같이 변환된다.

$$K_P e + K_I \int e dt + K_D \dot{e} = \eta \tag{9}$$

여기서  $\eta \triangleq -[J_m + \overline{D_m}(q_m)]^{-1} \epsilon$ ,  $\epsilon \triangleq [\epsilon_1 \ \epsilon_2 \ \epsilon_3 \ \epsilon_4]^T$ 이며,  $\overline{D_m}(q_m)$ 은  $D_m(q_m)$ 의 대각요소로 구성된 행렬이다.  $J_m + \overline{D_m}(q_m)$ 은 대각 행렬이며  $D_m(q_m)$ 의 비 대각 행렬 항은  $\epsilon$ 에 포함된다는 것을 주의해야 한다.

1. Fuzzy 제어규칙

개념적으로 퍼지 제어 시스템은 실험으로 유도된 퍼지 규칙을 사용하는 전문가 시스템이다. 본 연구에서는 역 동적 알고리즘에 강인성을 부여하기 위해,  $K_P$ ,  $K_I$  및  $K_D$ 의 값이 실시간에 동조되도록 하였다. 과도 응답의 개선을 위해서 PD 보상기를 적용하였고, 정상 상태 오차를 줄이기 위해서는 PI 보상기를 적용한다. 이러한 개념은 2차계 시스템과 VSPID<sup>[14]</sup> 기법을 위해 적용되어졌다. 로봇의 동작 중에는 비선형 동특성으로 인하여 위치와 속도 오차가 크게되고 정상상태가 되면서 점점 작아지는 사실에 따라 위치와 속도 오차를 퍼지 논리를 위한 입력 변수로 사용하였다. 상하 운동을 하는 3축의 경우에는 중력을 고려해서 운동의 방향을 퍼지 입력 변수로 사용하였다. 이렇게 새로운 이름으로 명명된 강인한 동적제어 알고리즘은 전문가의 경험과 지식으로 유도된 퍼지 규칙을 사용하여 PID 계수를 조정하는 과정이 수행된다. 본 연구에서는 퍼지 추론을 위해서 Mamdani<sup>[16]</sup>에 의해 제안된 max-min 방법이 사용되었고, 매 샘플링 순간의 최종

의  $K_D$ 와  $K_I$ 를 얻기 위한 퍼지 출력의 비퍼지화를 위해서는 mean-of-centers 방법<sup>[14]</sup>이 사용되었다.

표 3. 퍼지 규칙  
Table 3. Fuzzy rule.

위치 Error \ 속도 Error	LO	ME	HI
LO	LL	LM	LH
ME	ML	MM	MH
HI	HL	HM	HH

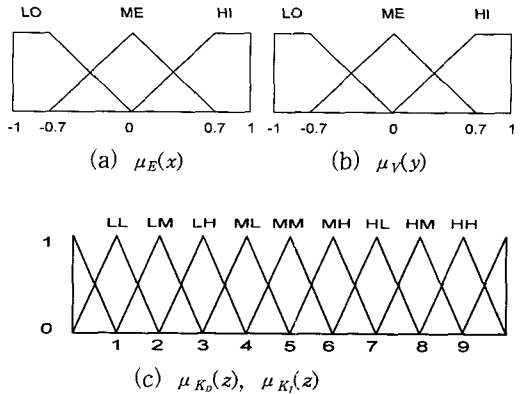


그림 8. 1, 2, 4축의 퍼지 입 출력 귀속함수  
Fig. 8. Fuzzy input/output membership functions for axis 1, 2, and 4.

이후 개의 언어 변수를 사용한 퍼지 규칙이 PID 계수의 조정을 위해 사용되었고, 표 3에 나타나 있다. LO는 low, ME는 medium 그리고 HI는 high를 각각 나타낸다. 매 샘플링 순간 마다 주어진 위치와 속도 오차에 대하여 이후 개의 퍼지 규칙 가운데서 네 개만이 항상 적용된다.

1, 2, 4축 일 때 퍼지 출력값으로 PID 계수를 얻기 위하여 그림 8에서 처럼  $\mu_E(x)$ ,  $\mu_V(y)$  그리고  $\mu_{K_D}(z)$ 를 퍼지 입력 출력 변수를 위한 귀속함수로 정의한다. 그러면 전진부의 적합함수는 다음과 같이 나타내어진다.

$$\lambda_i = \mu_{E_i}(x_0) \wedge \mu_{V_i}(y_0) \tag{10}$$

여기서  $\wedge$ 는 conjunctive AND에 해당하는 triangular-norm의 일종인 Min 연산을 나타낸다. Mamdani의 최소 퍼지 추론을 따르는 이후 개의 퍼지 규칙에 대한 퍼지 추론의 결과는 다음과 같이 얻어진다.

$$\mu_{K_D}(z_0) = \bigcup_{i=1}^9 \{ \lambda_i \wedge \mu_{K_{Di}}(z_0) \} \quad (11)$$

여기서  $\cup$ 는 disjunctive OR에 해당하는 triangular-conorm의 일종인 Max 연산이다.

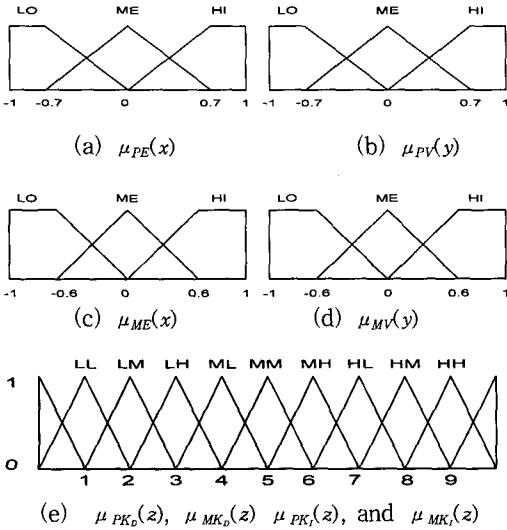


그림 9. 3축의 퍼지 입 출력 귀속함수  
Fig. 9. Fuzzy input/output membership functions for axis 3.

그림 9는 3축에 대한 귀속 함수를 나타낸다. 그림에서 (a), (b)의  $\mu_{PE}(x)$ ,  $\mu_{PV}(y)$ 는 Z 축 방향이 양일때의 위치오차와 속도오차의 입력 귀속함수를 나타내며, (c), (d)의  $\mu_{ME}(x)$ ,  $\mu_{MV}(y)$ 는 Z 축 방향이 음일때의 위치오차와 속도오차의 입력 귀속함수를 나타낸다. (e)의  $\mu_{PK_D}(z)$ ,  $\mu_{MK_D}(z)$ ,  $\mu_{PK_I}(z)$ ,  $\mu_{MK_I}(z)$ 는 Z 축 방향이 양일때와 음일때의  $K_D$ ,  $K_I$ 의 출력 귀속함수를 나타내고 있다. 식 (11)에서 얻어진 결과는 mean-of-centers 방법을 사용하여 개별적인 PID 계수를 얻기 위해 (12)식과 같이 비퍼지화된다.

$$K_{Di} \text{ (or } K_{Di}) = \frac{\lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 + \lambda_4 y_4}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} \quad (12)$$

여기서  $\lambda_1 \sim \lambda_4$ 는 식 (10)에서 선택된 입력 귀속함수를 위한 비중을 나타내고,  $y_i = \mu_{K_{Di}}^{-1}(\lambda_i)$ 는 추론된 출력을 나타낸다.

2. 외란 보상을 위한 미세조정

퍼지제어칙에 의해  $\hat{K}_{Pi}$ ,  $\hat{K}_{Ii}$  및  $\hat{K}_{Di}$ 는 정의되어 실제 수행 시의 오차방정식 (9)는 각 모터별로

구해진다. 즉

$$K_P \hat{e} + K_I \int \hat{e} dt + K_D \dot{\hat{e}} = \hat{\eta} \quad (13)$$

여기서,  $\hat{\cdot}$ 는 매 제어사이클 마다 측정되어지는 값을 나타낸다. (13)식의 오차방정식을 2차계에서 설계된  $K_P$ ,  $K_D$ , 및  $K_I$  값의 특성을 갖도록 하기위해  $\hat{\eta}_i = 0$  라는 가정하에 조정한다. 즉,

$$\overline{K_{Pi}} = K_{Pi} + \frac{\hat{\eta}_i}{\hat{e}_i} \quad (14)$$

로 값을 조정한다. 이 미세 조정과정을 거치고 나면 오차방정식은 2차의 동차방정식(Homogeneous Equation)이 되어 안정도 점점과 더불어 설계된 오차 특성을 추종할 수 있도록 된다. 그림 10은 강인한 동적제어 알고리즘의 제어 흐름도를 나타내고 있다. 그림의 알고리즘은 각 관절제어기에서 수행되는 과정을 나타내고 있으며, 강인한 동적제어 알고리즘의 실행을 위해서 공유메모리로부터 각각의 관절의 위치 및 속도값을 저장 및 판독하는 과정이 포함된다.

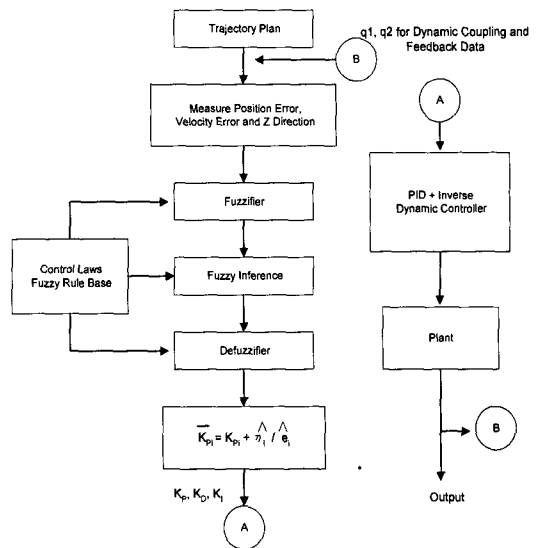


그림 10. 강인한 동적제어의 제어흐름도  
Fig. 10. Control flowchart of robust dynamic control algorithm.

VI. 실험 및 제어기 성능 평가

1. 실험

본 제어시스템의 우수성을 검증하기 위하여 종래의



PID 알고리즘, 역 동적 알고리즘을 각각 실현하고 최종적으로 강인한 동적제어 알고리즘의 실현을 통하여 시스템의 검증과 더불어 제어 알고리즘의 성능평가를 행하였다.

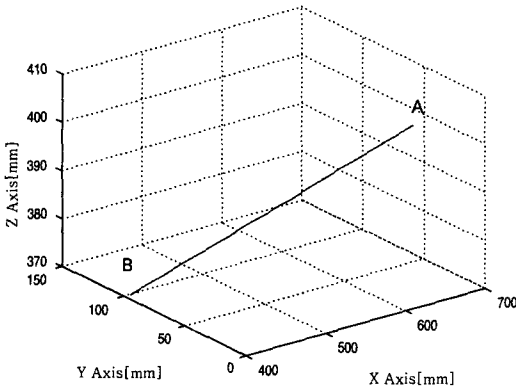


그림 11. 실험 궤적 계획  
Fig. 11. Experimental trajectory.

로 행해졌으며 실험을 통하여 최적의 이득 값들을 설정하였다. 역 동적 알고리즘 및 강인한 동적제어 알고리즘을 실현하기 위해서는 1 msec의 제어사이클 마다 각 모터의 각도 및 속도값이 공유메모리에 저장되어져 각 관절제어기에서 동력학 항들을 계산하도록 하였다. 역 동적 알고리즘에서도 외란을 보상하기 위해 고정된 이득의 PID 루프를 사용하였으며, 실험적으로 최적으로 선택된 이득의 값은 표 4에 주어져 있다.

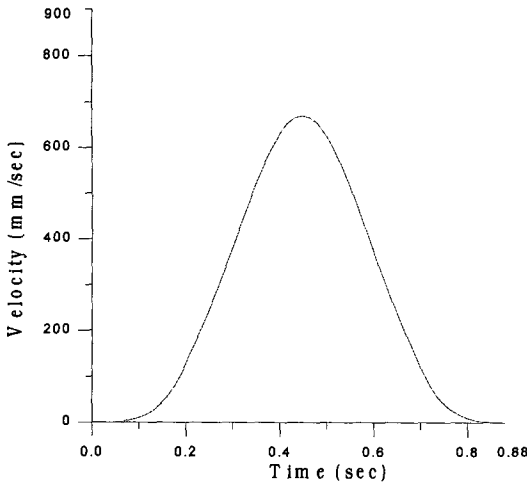


그림 12. 저속 운동시의 속도 그래프  
Fig. 12. Velocity profile for lower speed motion.

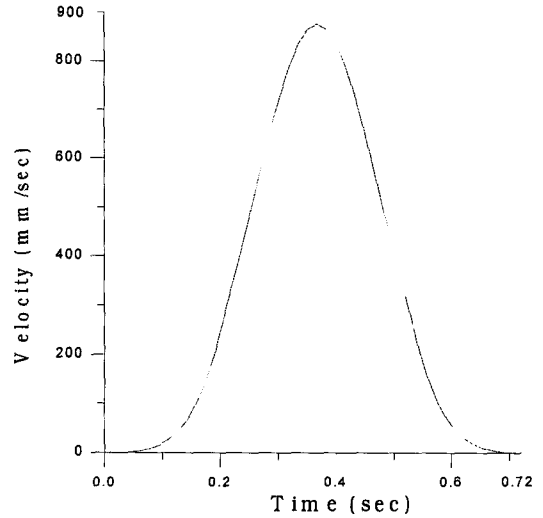


그림 13. 고속 운동시의 속도 그래프  
Fig. 13. Velocity profile for higher speed.

실험에 사용된 궤적은 그림 11과 같은 3차원상의 직선 궤적을 그림 12~13과 같이 정의된 속도 특성을 만족시키면서 추종하도록 하였다. 즉 저속 운동시는 0.88 sec, 고속 운동시는 0.72 sec 만에 A에서 B로 그림 12~13과 같은 속도궤적을 갖고 움직이도록 하였다.

PID 알고리즘의 경우에는 각축의 제어가 독립적으

표 4. 최적의 PID 상수값들  
Table 4. Optimal PID Parameters.

Gain	PID	Inverse Dynamic
$K_{P1}$	22	20
$K_{P2}$	13	15
$K_{P3}$	2	2
$K_{P4}$	2	2
$K_I1$	0.0032	0.0029
$K_I2$	0.0015	0.0013
$K_I3$	0.0015	0.0015
$K_I4$	0.0012	0.0012
$K_D1$	64	61
$K_D2$	48	45
$K_D3$	48	32
$K_D4$	32	32

강인한 동적제어 알고리즘에서는 역 동적 알고리즘에서 사용한 PID 상수값들을 기준으로 위치오차, 이

동방향 및 속도오차를 고려하여  $K_{Pi}$ ,  $K_{Ii}$  및  $K_{Di}$  를 퍼지 제어칙에 의해 결정하였으며  $K_{Pi}$  의 값은 외란을 보상하기 위해 미세조정되었다. 실험 결과는 로봇의 작용점의 이동 및 위치의 정도에 초점을 맞추어  $\text{Error} \triangleq (\sqrt{\delta x^2 + \delta y^2 + \delta z^2})$ 로 정의하여 표시한다. 또한, 저속 무부하 운동시의 고속 부하 운동시를 비교함으로써 제어 알고리즘의 성능비교를 하고자 한다. 실험 결과는 그림 14~17에서 나타나 있다.

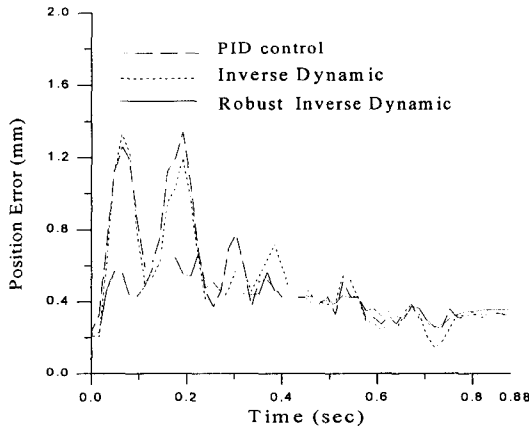


그림 14. 저속 운동시 위치 오차 그래프  
Fig. 14. Time history of position error at low speed.

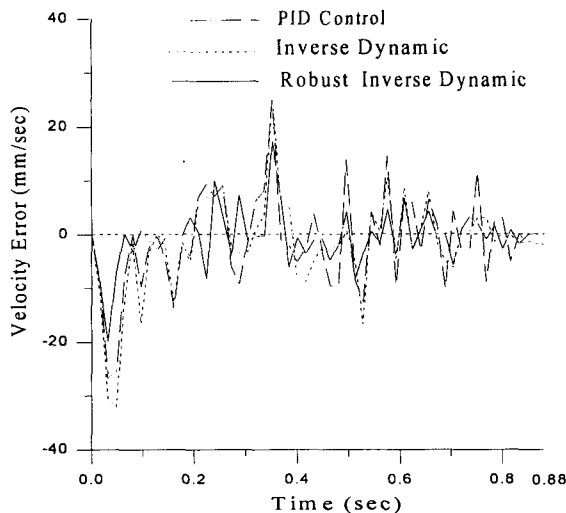


그림 15. 저속 운동시 속도 오차 그래프  
Fig. 15. Time history of Velocity error at low speed.

그림 14, 15에서 나타난 것처럼 저속 무부하 운동시에

는 PID 제어보다 역 동적 알고리즘이 뚜렷한 성능향상을 보이지 않고 있다. 이것은 저속운동 시에는 관성, centrifugal, Coriolis 그리고 중력항 등의 비선형의 외란이 크게 작용하지 않기 때문에 역 동적 알고리즘에서 사용된 동력학적 모델링이 제어의 성능에 미미하게 작용하기 때문이다.

강인한 동적제어 알고리즘에서는 위치오차, 속도오차 그리고 이동방향이 퍼지 모델링되어 PID 이득을 동적으로 보정해 준 결과 뚜렷한 성능 향상을 보임을 알 수 있다. 특히 속도가 급하게 변하는 0.1~0.2 sec 부분에서 고정된 PID 이득을 사용하는 역 동적 알고리즘보다 월등한 오차 개선 효과를 가져옴을 알 수 있다. 속도오차 부분도 강인한 동적제어 알고리즘에서는 상당히 개선됨을 볼 수 있다.

고속의 운동과 부하로 야기되는 비선형의 외란 때문에 PID 제어보다 역 동적 알고리즘과 강인한 동적제어 알고리즘이 보다 우수한 성능을 가짐을 그림 16, 17에서 확인할 수 있다. 그리고 역 동적 알고리즘에 의하여 오차를 줄일 수 있었지만 퍼지 모델링과 외란을 보상하는 미세  $Kp$  조정이 추가됨으로 오차를 현저히 줄일 수 있음을 확인하였다.

PID 제어의 최대 위치 오차를 100%로 두었을 때 역 동적 알고리즘과 강인한 동적 제어 알고리즘의 최대 위치 오차를 퍼센트로 계산하여 비교한 최대 오차 개선율은 그림 18에 나타나 있다. 제어기의 성능 평가를 위해서, 세 개의 알고리즘의 제어 주기는 1 msec로 고정하였다.

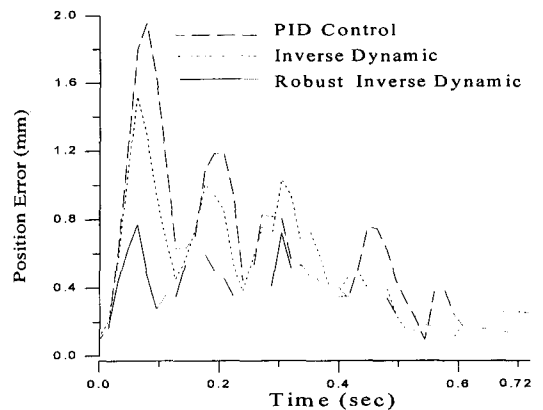


그림 16. 고속 운동시 위치 오차 그래프  
Fig. 16. Time history of position error at high speed.

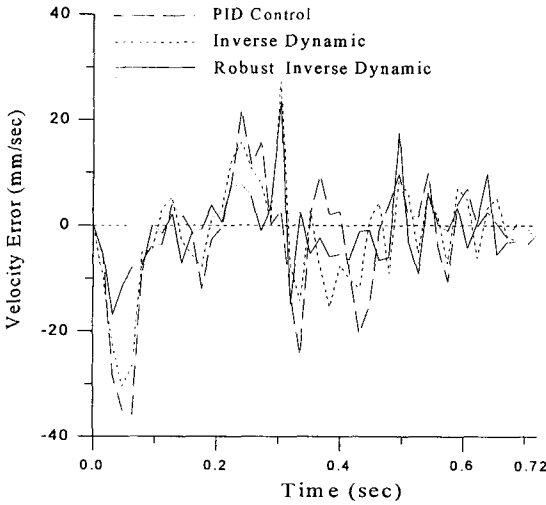


그림 17. 고속 운동시 속도 오차 그래프  
 Fig. 17. Time history of velocity error at high speed.

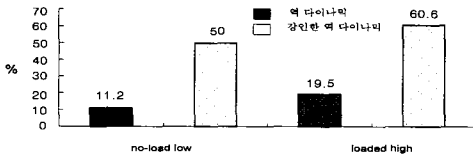


그림 18. PID 제어에 대한 역 동적 알고리즘과 강인한 동적 제어 알고리즘의 최대 오차 개선율  
 Fig. 18. Improvement rate of inverse dynamic and robust dynamic control algorithms against PID algorithm.

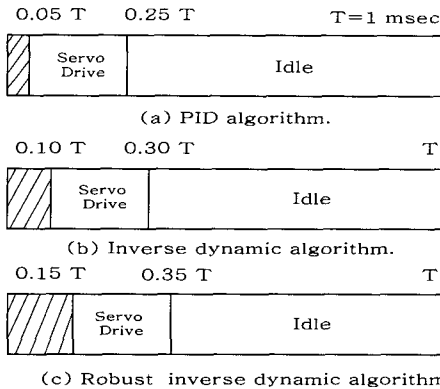


그림 19. 제어기에서 알고리즘 수행을 위한 소요 시간  
 Fig. 19. Time required for each algorithm in the designed controller.

그림 19에서 빗금친 부분은 어셈블리 코드를 사용한 알고리즘 수행시간을 나타내고 Servo Drive라고 표현한 부분은 서보 드라이브에서 소요되는 시간을 표시한다. 그리고 Idle은 CPU가 동작을 하지 않고 대기하고 있는 시간을 나타내고 있다. 그러므로 샘플링 주기 안에서 0.35msec의 가장 긴 처리시간을 요구하는 강인한 동적 제어 알고리즘을 수행하는데 충분함을 확인할 수 있다. 따라서 본 실험에서 사용된 알고리즘보다 학습 등의 목적으로 더욱 복잡해진 알고리즘도 실시간에 처리할 수 있는 충분한 여유시간이 있음을 알 수 있다. 또한, 본 실험에서 사용한 강인한 동적 제어 알고리즘의 수행 능력을 증가시키기 위해 제어 주기를 0.5 msec로 줄이는 것도 가능할 것이다.

VII. 결 론

본 논문에서는 계산량이 많은 제어 알고리즘들을 수행할 수 있도록 빠른 계산 속도를 가지고 있는 DSP를 이용하여 스카라 로봇의 제어기를 설계/제작하였다. 제작된 시스템은 병렬 통신망과 직렬통신망의 구현으로 그 응용 범위를 넓혔으며 각 축에 독립적인 제어기가 사용하여 로봇의 각 축을 각기 다른 알고리즘으로 제어할 수 있도록 하였다. 시스템의 성능 검증과 평가를 위하여 순수한 PID, 역 동적 알고리즘, 퍼지 이론으로  $K_D$ ,  $K_I$  계인을 동조하고 외란 보상을 위해  $K_F$  계인을 미세 조정하는 강인한 동적 제어 알고리즘을 실현시켰다. 강인한 동적 제어 알고리즘에 의해 각 축별로 독립제어를 수행한 결과 비선형, 시변특성 등의 외란을 극복하면서 우수한 성능을 가짐을 알 수 있었다. 실험 결과와 알고리즘 수행 속도의 분석을 통해, 본 연구에서 설계된 제어 시스템은 복잡한 연산을 요구하는 동적 제어 알고리즘의 수행을 위하여 적합함을 알 수 있었다.

앞으로의 과제는 외란을 예측하는 관측기 혹은 예측기를 설계하여 강인한 동적 제어 알고리즘의 성능을 향상시킨 제어시스템을 구현하는 것이다.

참 고 문 헌

[1] F. L. Lewis, C. T. Abdallak, and D. M. Dawson, *Control of Robot Manipulators*, Macmillan Publishing Company, 1993.

- [2] M. W. Spong, and R. Ortega, "On Adaptive Inverse Dynamics Control of Rigid Robots," *IEEE Trans.*, AC-39, pp. 1866-1871, 1994.
- [3] Y. D. Song, T. L. Mitchell, and H. Y. Lai, "Control of a Class of Nonlinear Uncertain Systems via Compensated Inverse Dynamics Approach," *IEEE Trans.*, AC-39, pp. 1866-1871, 1994.
- [4] K. K. Shyu, P. H. Chu and L. J. Shang, "Control of Rigid Robot Manipulators via Combination of Adaptive Sliding Mode Control and Compensated Inverse Dynamics Approach," *IEE Proc. Control Theory Appl.*, Vol. 143, No. 3, May 1996.
- [5] P. Rocco, "Stability of PID Control for Industrial Robot Arms," *IEEE Trans. Robo and Auto.*, Vol. 12, No. 4, August 1996.
- [6] R. Kelly and R. Salgado, "PD Control with Computed Feedforward of Robot Manipulators: A Design Procedure," *IEEE Trans. Robo. and Auto.*, Vol. 10, No. 4, August 1994.
- [7] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons, Inc., 1989.
- [8] *TMS320C5X User's Guide*, Texas Instruments Inc., 1993.
- [9] *Digital Signal Processing Applications with the TMS320 Family*, Volume 1, 2, 3, Texas Instruments Inc., 1989.
- [10] K. S. Lin, G. A. Frantz, and R. Simar, Jr., "The TMS320 Family of Digital Signal Processors," *IEEE Proc.*, Vol. 75, No. 9, Sep. 1987.
- [11] P. Pillay and R. Krishnan, "Modeling, Simulation, and Analysis of Permanent-Magnet Motor Drives, Part II: The Brushless DC Motor Drive," *IEEE Trans. Ind. Appl.*, Vol. 25, No. 2, Mar. 1989.
- [12] T. Maeno and M. Kobata, "AC Commutatorless and Brushless Motor," *IEEE Trans. Power Appl. Syst.*, Vol. PAS-91, pp. 1476-1484, Jul. 1972.
- [13] J. Zubek, A. Abbondanti, and C. J. Nordby, "Pulsewidth Modulated Inverter Motor Drives with Improved Modulation," *IEEE Trans. Ind. Appl.*, Vol. 1A-11, pp. 695-703, Nov. 1975.
- [14] C. L.Chen and F. Y.Chang, "Design and Analysis of Neural/Fuzzy Variable Structural PID Control System," *IEE Pro. Control Theory Appl.* Vol. 143, No. 2, pp. 200-208, Mar. 1996.
- [15] G. F. Franklin, J. D. Powell and A. Enami-Naeini: *Feedback Control of Dynamic System*, Addison-Wesley, 1986.
- [16] E. H. Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Systems," *IEEE Trans. Com.*, C-26, pp. 1182-1191, 1977.

---

 저 자 소 개
 

---



李章明(正會員)

1957년 11월 22일생. 1980년 2월 서울대학교 전자공학과 졸업(공학사). 1982년 2월 서울대학교 전자공학과 대학원 졸업(공학석사). 1990년 2월 USC(남가주 대학) 박사학위 취득(Computer - Engi-

neering 전공). 현재 부산대학교 공과대학 전자공학과 부교수, 부산대학교 컴퓨터 및 정보통신 연구소 전자기술 부장. 주관심분야는 다관절 로봇 시스템의 설계 및 제어.  $\mu$ -processor 응용시스템 설계, Motor 구동 방식 및 Sensor Fusion 등임



朴興仁(正會員)

1970년 12월 20일생. 1996년 2월 동의대학교 전자공학과 졸업(공학사). 1996년 3월 ~ 현재 부산대학교 대학원 전자공학과 석사과정. 주관심분야는 로봇 시스템 설계 및 제어, 지능 제어 시스템