

홍익 직접 구동팔의 동적 제어기 개발

이재완, 이종수, 최경삼
홍익대학교 공과대학 전기공학과

Design and Implementation of a Dynamic Controller for Hink-Ik Direct Drive Arm

Jae-Whan Lee, Jong-Soo Lee, and Kyung-Sam Choi
Department of Electrical & Control Engineering, Hong-Ik University

ABSTRACT

A scara type Direct Drive Arm(DDA) with two degrees-of-freedom is designed and implemented. The direct drive motor is used to furnish large torque to reduce the modeling error by the gear and chains. To control the DDA, a multiprocessor control structure with multirate dynamic control algorithm is designed. In the control algorithm, the dynamics of system is used to calculate the nominal control torque and the feedback controls are calculated with a parallel processing algorithm for each joint. The laboratory experiments on Hong-Ik DDA by dynamic control algorithm are presented and compared to that of PID control algorithm. This result shows that the proposed controller guarantees small trajectory error and stability. With this reserch, Hong-Ik DDA is extected to be utilized as a basic tool for robotics and control engineering..

1. 서론

산업 자동화의 여러 분야에서 적용되고 있는 산업용 로봇은 로봇 팔을 움직이기 위해 다양한 동력 전달 장치를 이용하므로 관절간의 상호 간섭 토오크가 감쇄 된다. 따라서 관절간의 기구학만을 고려하여 각 관절별로 독립적인 PID 위치 속도 제어를 하고 있다. 이러한 동력전달 방법은 마찰 및 백 래시 현상으로 인하여 모델링 오차가 커지므로 정밀한 고속의 운동을 제어하기 어렵게 된다. 그래서 홍익 대학교에서는 이러한 단점을 보완하여 정밀하고 고속의 작업이 가능하도록 하기 위해 동력전달 장치를 통하지 않고 모터가 로봇 팔을 직접 구동하는 방식의 직접 구동팔을 개발하였다. 직접 구동팔은 관절간의 상호 작용 토오크 등이 감쇄없이 구동 모터에 전달되므로 이를 고려하여 정확한 위치 제어를 할 수 있는 동적 제어 알고리즘이 필요하다. 많은 동적 제어 알고리즘을 실시간 제어가 가능하도록 실현하기 위하여, 동력학 계산을 중심으로 하는

주 제어기와 각 관절의 모터를 구동 제어하는 부 제어를 갖는 분산처리 구조의 하드웨어를 설계하였다. 사용한 제어 방식은 가변구조 시스템의 활동모드 제어 방식의 단점인 활동 표면에서의 매우 높은 주파수로의 입력결환으로 인한 떨림 현상을 개선한 "변형된 가변 구조 방식" 이다.

제어기에 사용된 프로세서는 intel 8086 으로서 부동소수점 연산을 위하여 MAX 모드를 이용하여 수치연산 프로세서 8087 을 사용하였다. 주 메모리는 128KB 이고 각 제어기간의 제어 명령과 정보 교환은 병렬 인터페이스 I/O 포트(8255)의 인터럽트 기능을 이용하여 공유 메모리 16KB 통해 전달 되도록 하였다.

설계된 제어기를 이용하여 "변형된 가변 구조 방식"에 의한 경로 오차와 "PID 제어 방식"에 의한 경로 오차를 비교 하였다.

2. 동적 제어기 구성

직접 구동 팔에 사용될 제어 시스템 구성은 측정된 전동기 정보와 동적 제어 알고리즘을 이용하여 원하는 토오크를 발생시키도록 하는 토오크 궤환 제어기 형태로 구성한다. 또한 직접 구동 팔에 대한 동적 제어 알고리즘을 효율적으로 구현하기 위하여 다중 프로세서 구조의 하드웨어를 설계하였다.

직접 구동 팔을 제어하는 시스템의 구성은 그림 1 에서와 같이 크게 세 부분으로 나누어진다. 즉, 경로 계획을 담당하는 사용자 관리 시스템(Host Computer : IBM AT 호환기종)과 근사적으로 규정 경로를 보장하도록 동력학 계산에 의해 규정 토오크를 계산하는 주 제어기, 그리고 각각의 관절별로 관절의 위치, 속도 오차에 의해 보상토오크를 계산하는 부 제어기로 구성된다.

이와 같이 구성된 제어 시스템을 통한 정보전달 과정에서 사용자 관리 시스템의 역할은 경로 계획으로서, 주어진 위치를 이용하여 부드러운 위치 곡선을 계산하고 이에 따른 중간 위치, 속도, 가속도를 계산하여 매 표본 구간마다 주 제어기에 전달한다.

" 이 논문은 1992년도 홍익대학교 교내학술 연구조성비에 의하여 연구되었음"

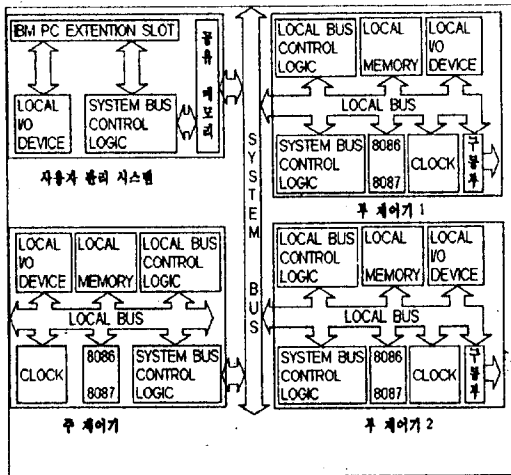


그림 1. 시스템의 블록 구성도(용익 직접 구성도)는 2 자유도이므로 2 개의 부 제어기가 사용됨

본 논문에서 사용한 동적 제어 알고리즘은 주 제어기가 로봇 모델을 이용한 동력학 계산을 처리하여 부 제어기로 규정 토크 데이터를 전달하는 반면, 부 제어기는 각 관절이 독립적으로 자신만의 경로오차에 의해 간단히 보상토크를 계산하도록 되어 있다. 따라서, 주 제어기와 부 제어기는 계산량에 큰 차이가 있으며 서로 다른 표본 구간을 가져야 한다. 그러므로, 본 논문에서 사용한 제어 시스템의 하드웨어는 이러한 멀티-레이트 제어방식을 구현할 수 있도록 하드웨어를 설계하였다.

위와 같은 정보 전달 과정을 도식적으로 표현하면 그림 2 와 같다.

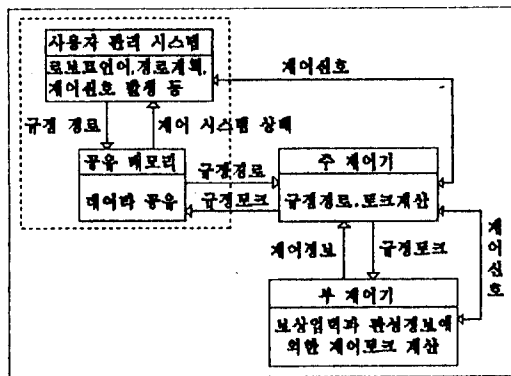


그림 2. 로봇 운영시스템의 정보전달 과정

2.1 하드웨어의 구성

2.1.1 사용자 관리 시스템에의 접속 모듈

주 제어기와 부 제어기간의 데이터 전송과 정보 교환에 이용되는 사용자 관리 시스템에 접속되는 전체적인 모듈의 구성도는 그림 3 에서 보여주고 있다.

일정한 시간 간격으로 주 제어기에 경로 계획의 결과를 전송하기 위하여 타이머 컨트롤러에 의한 인터럽

트 기능을 이용하였고, 병렬 인터페이스 포트의 인터럽트의 기능을 이용하여 주 제어기에서 전체 시스템의 동작 상태 및 전동기의 응급 상태 등의 정보를 사용자 관리 시스템에 전달할 수 있도록 하였다. 또한, 사용자 관리 시스템, 주 제어기, 부 제어기 상호간의 동작 명령, 위치 정보, 제어 정보 등의 교환은 공유 메모리를 이용하였다. 공유 메모리 이용 우선 순위는 부 제어기(1,2), 주 제어기, 사용자 관리 시스템 순으로서 우선 순위가 낮은 것은 우선 순위가 높은 제어기가 공유 메모리를 완료할 때까지 기다린후 이용하도록 설계하였다.

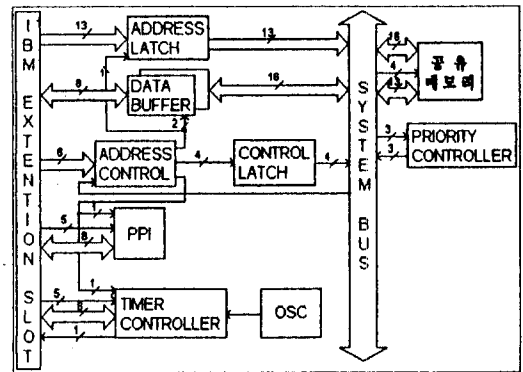


그림 3. 사용자 관리 시스템에의 접속 모듈 구성도

2.1.2 주 제어기

가. 기능

이 제어기의 특성은 경로 계획의 결과로부터 받은 정보와 측정에 의한 로봇 모델을 이용하여 규정 토크를 계산한다. 이 결과는 근사적으로 로봇 관절간의 비선형 결합 토크를 보상하며, 규정 경로를 운동하도록 하는 근사적인 전향 제어 토크이다. 또한, 사용자 관리 시스템과 부 제어기와의 정보 교환(POWER 상태, 전동기 구동 상태, 응급 상황 등)을 위한 중간 역할을 한다. 주 제어기의 전체 모듈 구성은 그림 4 와 같다.

나. 하드웨어 구성

주요 기능별 모듈의 특성을 살펴보면 다음과 같다.

a. MAIN PROCESSOR

중앙 연산 장치로서 INTEL-8086(5MHz) 을 사용하였다.

b. MATH PROCESSOR

로봇 모델은 2차 비선형 미분 방정식의 형태로 나타나며 로봇의 동력학 정보를 계산 하기 위해서 많은 부동소수점 연산이 요구된다. 따라서 빠른 계산 처리를 위해 INTEL-8086의 MAX 모드를 이용하여 수치 연산 프로세서(INTEL-8087)를 사용 하였다.

c. 메모리

주 메모리는 128KB(ROM: 64KB, RAM: 64KB)이고 공유 메모리의 사용 우선순위에 의하여 공유 메모리 16KB를 사용

한다.

· 사용자 관리 시스템에서 발생하는 규정경로(위치, 속도, 가속도등)는 공유 메모리를 통하여 받으며, 규정토크 계산에 의해 발생한 앞먹임 보상 정보와 관성 행렬 값의 전송은 공유 메모리에 저장함으로써, 각각의 부 제어기가 필요한 정보를 전송 받도록 하였다.

d. 병렬 I/O 포트

또한, 사용자 관리 시스템으로 주 제어기와 부 제어기에 대한 각종 정보를 주기 위하여 병렬 인터페이스 I/O 포트로 8255A 모드 1 의 인터럽트 기능을 이용하였다.

e. 인터럽트

i8259 프로그래머블 인터럽트 컨트롤러를 사용하였으며 각 채널의 사용은 다음과 같다.

번호	기능
0	PROGRAM ERROR
1	COPROCESSOR
2	PPI
3	TIME TICK 발생
4	부 제어기 1
5	부 제어기 2
6	하드웨어 점검
7	UNUSED

f. 시리얼 인터페이스

i8251A 프로그래머블 커뮤니케이션 컨트롤러를 사용하였으며 전송 속도는 2400 - 19600 bps 로 가변할수 있도록 하였으며 직렬 통신은 RS-232C 방식을 사용하고 있다.

g. 버스 컨트롤러

로컬 메모리와 공유 메모리를 이용함에 있어서 버스 충돌이 발생함이 없이 버스를 원활히 사용할 수 있도록 i8288 버스 컨트롤러와 우선순위의 판단에 따라 시스템 버스의 신호를 제어하여 공유 메모리에서의 버스 충돌을 방지하는 역할을 담당하는 버스 중계기로 구성하였다.

h. 타이머 컨트롤러

i8254 프로그래머블 타이머 컨트롤러를 사용하여 인터럽트 발생에 의한 감시(watchdog) 기능과 시리얼 커뮤니케이션의 전송속도를 프로그램할 수 있는 기능을 담당하도록 하였다.

2.1.3 부 제어기

가. 기능

부 제어기는 직렬 통신에 의해 전동기 구동부를 초기화하고 병렬 토크 제어값을 계산하는 것이 주요 기능이다. 부 제어기는

① 전동기 구동부에서 전송받는 위치, 속도 정보와,

② 주 제어기가 공유메모리에 저장한 관성행렬 정보와 규정 토크 정보, 그리고,

③ 사용자 관리 시스템이 공유 메모리에 저장한 규정경로를 읽어 들여 위치 제어를 위한 제어 토크를 계산한다.

나. 하드웨어 구조

부 제어기의 하드웨어는 전동기 구동부로부터 현재 위치를 받아들일 수 있는 하드웨어, 구동부의 아날로그 속도 정보를 A/D 변환하기 위한 회로, 그리고 전동기 구동부에 병렬로 명령을 내릴 수 있도록 I/O 포트를 주 제어기에 추가한 구조를 하고 있다. 이와 같이 추가된 부분은 그림 4 에서 * 표시가 된 부분이다.

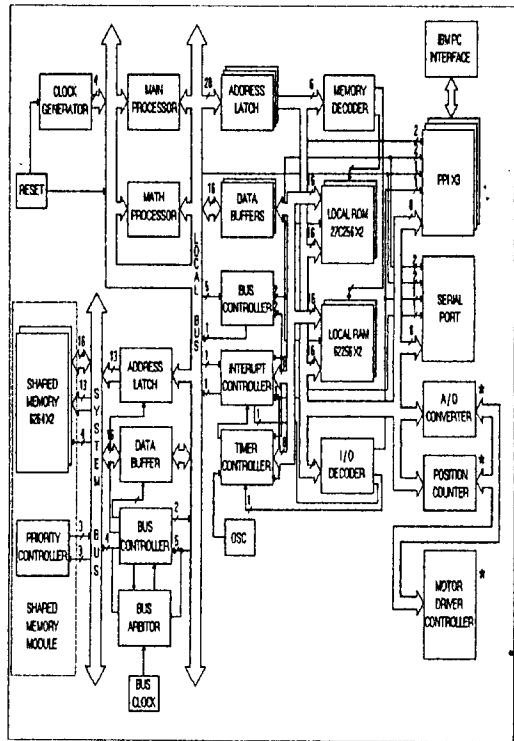


그림 4. 주 제어기 및 부 제어기의 모듈 구성도(* 표시는 부 제어기에 해당되는 회로이다.)

2.2 궤환 제어 토크 알고리즘

n 개의 관절을 갖는 로봇 매뉴플레이터의 Lagrange - Euler 방식을 이용한 동적 운동 방정식은 다음과 같이 표현된다.

$$D(q(t))\ddot{q}(t) + h(q(t), \dot{q}(t)) = \tau(t) \quad (1)$$

윗 식에서 $\tau(t) \in R^n$ 는 관절에 가해지는 일반된 토크 벡터, $q(t) \in R^n$ 는 로봇 관절의 변수 벡터, $\dot{q}(t) \in R^n$ 는 관절 속도 벡터, $\ddot{q}(t) \in R^n$ 는 관절 가속도 벡터이다. 그리고 $D(q(t)) \in R^n$ 는 대칭 관성행렬, $h(q(t), \dot{q}(t)) \in R^n$ 는 비선형 코리올리스 힘과 구심

력, 중력 등을 포함하는 비선형 상호 작용 토오크 벡터이다.

이용 가능한 로보트 모델식은

$$\hat{D}(q(t))\ddot{q}(t) + \hat{h}(q(t), \dot{q}(t)) = \tau(t) \quad (2)$$

그리고 로보트의 원하는 규정 경로를 $q_d(t), \dot{q}_d(t), \ddot{q}_d(t)$ 이라 하고, 실제 운동경로를 $q(t), \dot{q}(t), \ddot{q}(t)$ 이라 할 때, 활동 표면 벡터 $s(t) \in R^n$ 을

$$s(t) = \dot{e}(t) + k \cdot e(t) \quad (3)$$

라고 정의한다.

다음과 같은 제어 입력을 고려하자.

$$\tau(t) = \tau_f(t) + \tau_c(t) \quad (4)$$

여기서 $\tau_f(t)$ 이산 앞먹임 보상 성분으로서

$$\tau_f(t) = \hat{D}(\bar{q}_d(t))\ddot{\bar{q}}_d(t) + \hat{h}(\bar{q}_d(t), \dot{\bar{q}}_d(t)) \quad (5)$$

으로 주어지며, 궤환 제어 성분 $\tau_c(t)$ 는 활동 모드 제어를 변형한 형태로서

$$\tau_c(t) = \hat{D}(\bar{q}_d(t)) \cdot u(t) \quad (6)$$

이다.

위의 제어 입력 중 앞먹임 보상 성분 $\tau_f(t)$ 는 로보트 관절간의 비선형 결합 토오크를 근사적으로 보상하는 이산 계단 함수로서, 식 (2) 를 이용하여 계산하며, 표본 주기는 동역학 계산시간을 고려하여 충분히 크게 선택하여야 한다.

또한, 궤환 제어 성분 식(6)의 $u(t)$ 는

$$u(t) = -k\dot{e}(t) - k_p e(t) - k_{pp}(t) \quad (7)$$

으로 주어진다.

여기서, 벡터 $o(t) = [o_1(t), \dots, o_n(t)]^T \in R^n$ 는

$$o_i(t) = \frac{s_i(t)}{|s_i(t)| + \Delta} \quad (8)$$

으로 정의된다.

여기서, Δ 가 0 이면 가변구조 제어방식과 유사하게 되며 $o_i(t)$ 는 불연속 스위칭을 하게 된다. 따라서, Δ 의 역할은 이를 연속함수로 만드는 것이다.

이 제어 방식이 제한된 경로 오차를 보장함에 대하여는 참고문헌 [14]를 참조하기 바란다.

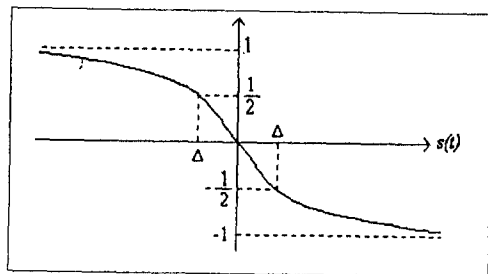


그림 5. 연속 보간 함수

2.3 소프트웨어

로보트 운영 시스템으로부터 발생한 정보는 공유 메모리에 저장되며 주 제어기와 각각의 하부 제어기에 순차적으로 전달하여야 한다. 이러한 정보 전송의 한 방

법으로 인텔 멀티 버스 구조를 이용한 인터럽트를 사용하여 매 표본 시간마다 정보를 전달하도록 하며, 주 제어기에서 전송되는 정보는 로보트 시스템의 근사적인 모델을 계산하여야 하므로 긴 표본시간에 전달된다. 이때 하부 제어기는 주 제어기에서 전달된 정보와 로보트 운영 시스템에서 전송된 정보를 이용하여 토오크 궤환 제어를 수행하도록 하였다.

시스템 정보의 전체적인 전달과정은 다음과 같다.

. 경로 계획에 의한 원하는 정보

$q_{a1} = [q_{a1}, \dot{q}_{a1}, \ddot{q}_{a1}]$: 첫번째 관절에 필요한 정보

$q_{a2} = [q_{a2}, \dot{q}_{a2}, \ddot{q}_{a2}]$: 두번째 관절에 필요한 정보

. 구동부로부터 궤환 정보

$q_{m1} = [q_{m1}, \dot{q}_{m1}]$: 첫번째 관절에 필요한 정보

$q_{m2} = [q_{m2}, \dot{q}_{m2}]$: 두번째 관절에 필요한 정보

τ_{m1}, τ_{m2} : 전항 보상 토오크

τ_{c1}, τ_{c2} : 제어토오크

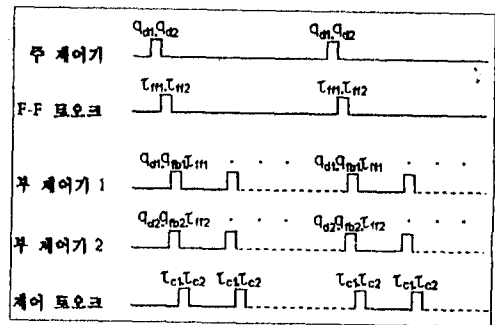


그림 6. 로보트 운영 시스템에서 전달되는 정보 전달과정

2.3.1 사용자 관리 시스템

하드웨어적으로 구성된 인터럽트 규약 및 공유 메모리 우선 순위에 따라 경로 계획의 전송 및 시스템의 에러에 따른 정지 명령, 동작의 개시와 종료에 대한 관리를 담당할 수 있도록 프로그램 되었다.

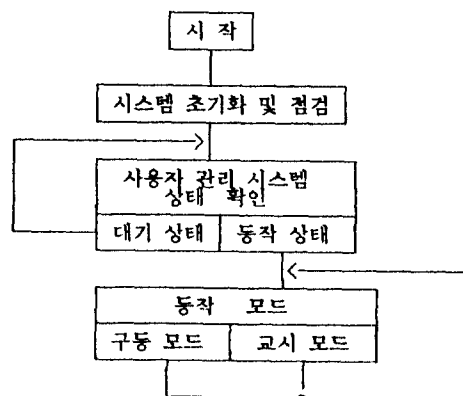


그림 7. 주 제어기 흐름도

2.3.2 주 제어기

주 제어기는 하드웨어 초기화와 동력학 계산을 하며 시스템 상태 점검 및 이의 내용을 사용자 관리 시스템에 전송하도록 프로그램 되었다.

이 프로그램의 흐름도는 그림 7 과 같다.

2.3.3 부 제어기

부 제어기는 하드웨어 초기화와 주 제어기에서 계산된 원하는 토크 값과 궤환 정보(위치, 속도)에 의해 제어 토크를 계산하여 전동기 구동부에 전송하게 되는 것과, 또한 전동기 구동부를 직렬 통신에 의해 초기화하도록 프로그램되었다.

이 프로그램의 흐름도는 그림 8 과 같다.

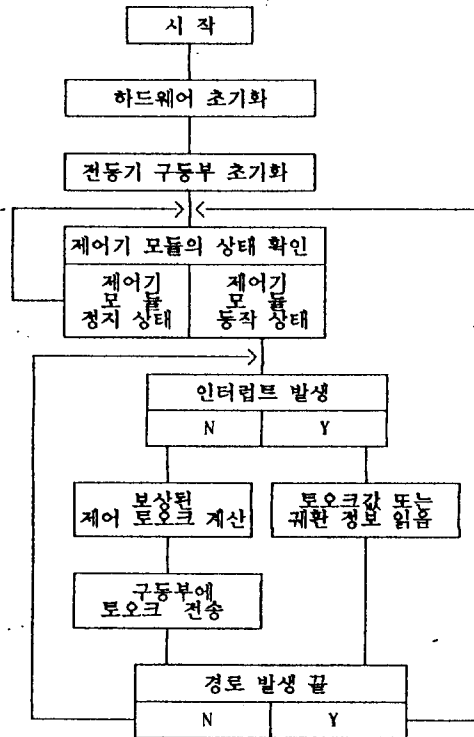


그림 8 부 제어기 흐름도

3. 결과 및 검토

그림 9 와 같은 규정 경로를 통하여 변형된 가변구조 제어방식을 통한 실험에서 이득 변화에 의한 위치 오차에 대한 실험 결과는 표1.~표9. 까지이며 PID 제어 방식과의 위치오차 비교는 그림 10 과 같다. 이득 k 와 k_s 가 증가하면 오차가 감소 하지만 $k=300$, 과 $k_s=70$ 이상이면 전동기의 심한 진동과 제어토크의 많은 변화가 있고 오차가 다시증가함을 확인하였다. 그리고 이득 Δ 와 k_0 의 변화에서 Δ 의 감소와 k_0 의 증가는 오차를 감소시킨다. 식 (4)와 (5)에서 보는 바와 같이 Δ 의 감소와 k_0 의 감소에 의해 시스템은 안정된다. 이것은 시스템 구조가 슬라이딩 모드 제어와 비슷

하다면 궤환제어는 보다 작은 이득을 갖게 됨을 의미한다.

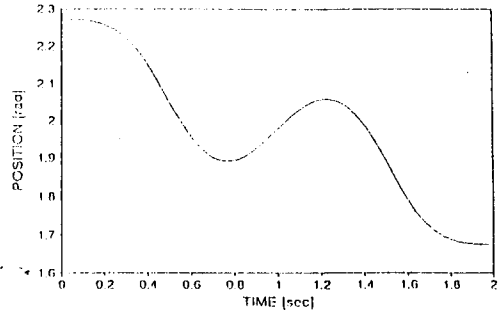


그림 9 규정 경로

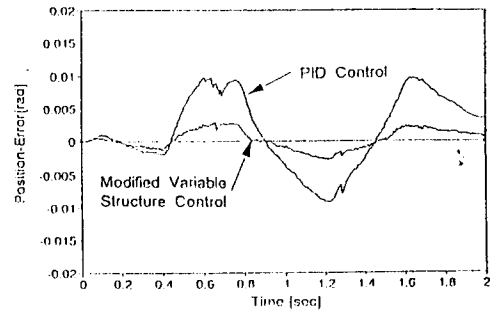


그림 10 위치 오차 비교

표1 ($\Delta = 0.1, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.133476	0.098110	0.084366	0.076441	0.071450
30	0.048968	0.035776	0.030499	0.027714	0.025591
50	0.031624	0.025111	0.023621	0.020025	0.017202
70	0.022114	0.021658	0.018266	0.014948	0.013807
90	0.023662	0.017562	0.016012	0.013290	0.02227

표2 ($\Delta = 0.1, k_0=30.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.108526	0.078568	0.069409	0.064946	0.061744
30	0.038076	0.029763	0.025754	0.023746	0.022357
50	0.020023	0.018912	0.016210	0.013901	0.011916
70	0.017150	0.015894	0.012512	0.010135	0.009429
90	0.015235	0.011674	0.008719	0.005283	0.003429

표3 ($\Delta = 0.1, k_0=60.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.069937	0.060815	0.05585	0.053165	0.052050
30	0.028474	0.023459	0.021132	0.020068	0.020326
50	0.019979	0.018727	0.017854	0.016422	0.015474
70	0.017790	0.013899	0.018553	0.018553	0.013213
90	0.019309	0.015812	0.012783	0.097190	0.045436

표4 ($\Delta = 1.0, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.133476	0.098110	0.084366	0.076411	0.071450
30	0.048968	0.035776	0.030499	0.027714	0.025591
50	0.031624	0.025111	0.023621	0.020025	0.017202
70	0.022114	0.021658	0.018266	0.014948	0.013807
90	0.023662	0.017562	0.016012	0.013290	0.02227

표5 ($\Delta = 1.0, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.108526	0.078568	0.069405	0.064946	0.061744
30	0.038076	0.029763	0.025754	0.023746	0.022237
50	0.023791	0.019203	0.017961	0.014948	0.013896
70	0.019425	0.017838	0.014237	0.012937	0.012554
90	0.017479	0.014008	0.014831	0.012254	0.012882

표6 ($\Delta = 0.1, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.069937	0.060814	0.055850	0.053165	0.052050
30	0.028474	0.023459	0.021132	0.020068	0.020326
50	0.019979	0.019876	0.014172	0.013695	0.011887
70	0.017790	0.013899	0.012417	0.018553	0.013213
90	0.019304	0.015812	0.01783	0.097170	0.045436

표7 ($\Delta = 0.1, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.134079	0.100289	0.085644	0.077739	0.072135
30	0.049455	0.035776	0.031072	0.027956	0.023250
50	0.031416	0.022973	0.019817	0.017402	0.016269
70	0.022679	0.018508	0.017569	0.013984	0.056555
90	0.020144	0.016994	0.014174	0.015339	0.012696

표8 ($\Delta = 0.1, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.116738	0.088742	0.077463	0.069937	0.065917
30	0.045201	0.034010	0.029231	0.026075	0.024241
50	0.028692	0.021861	0.020129	0.016796	0.016140
70	0.023335	0.018916	0.015369	0.018731	0.016235
90	0.021715	0.017937	0.013362	0.012528	0.013822

표9 ($\Delta = 0.1, k_0=1.0$)

$k_s \backslash k$	100	150	200	250	300
10	0.107575	0.078966	0.069067	0.064199	0.059915
30	0.042913	0.031481	0.027309	0.024854	0.023163
50	0.036994	0.028745	0.024451	0.018457	0.014551
70	0.021493	0.018650	0.016527	0.014007	0.012737
90	0.019735	0.019499	0.014340	0.013927	0.013767

4. 결론

용익 직접 구동팔에 변형된 기변구조 방식의 동력학 알고리즘을 구현하여 위치 제어를 하였다. 또한,

그 결과를 이용하여 기존의 PID 제어 방식에 의한 결과와 비교하였다. 비교 결과 동적제어 방식이 PID 방식에 비해 현저히 작은 경로 오차를 보장하였으며 실시간 계산에 부담이 없었다. 또한 동적제어 알고리즘의 제어 계수를 조정함에 따라 나타나는 경로 오차 및 떨림 현상 등의 변화에 대한 결과를 정리 비교 하였다.

참고문헌

- [1] Tatasuo Arai, Eiji Nakano, Tomoaki Yano, Ryoichi Hashimoto and Ichiro Takeyama, DEVELOPMENT OF DIRECT-DRIVE HUMAN-LIKE MANIPULATOR *Proc. of 15th ISIR*, pp 447-454, 1985.
- [2] H. Asada and H. Yamamoto, Torque feedback control of MIT direct-drive robot, *Proc. of 13th ISIR*, 1983.
- [3] Donald Schmitz, Pradeep Khosla and Takeo Kanade, Development of CMU Direct-Drive Arm II, *Proc. of 15th ISIE*, pp. 471-478, 1985.
- [4] Takashi Suehiro and Kunikatsu Takase, DEVELOPMENT OF A DIRECT DRIVE MANIPULATOR : ETA-3 AND ENHANCEMENT OF SERVO STIFFNESS BY A SECOND ORDER DIGITAL FILTER, *Proc. of 15th ISIR*, pp. 479-486, 1985.
- [5] Pradeep K. Khosla and Takao Kanade, "EXPERIMENTAL EVALUATION OF THE FEEDFORWARD COMPENSATION AND COMPUTED-TORQUE CONTROL SCHEMES" *Proc. CDC*, pp. 790-798, 1987.
- [6] Chae H. An, Christopher G. Atkeson, John D. Griffiths and John M. Hollerbach, EXPERIMENTAL EVALUATION OF FEEDFORWARD AND COMPUTED TORQUE CONTROL, *Int. Cont. Robotics and Automation*, pp. 165-168, 1987.
- [7] Haruhiko Asada and Swee-Keng Lim, THE JOINT TORQUE FEEDBACK CONTROL OF A DIRECT-DRIVE ARM, *Robotics Research : 2nd Int Symposium*, MIT Press, pp. 238-243, 1987.
- [8] 이종수, 로봇의 하이브리드 제어기설계에 관한 연구, 박사학위논문, 서울 대학교, 1987.
- [9] J.S. LEE and W.H. KWON, A Hybrid Control Algorithm for Robotic Manipulators, *Robotica* volume 9, pp. 307-318, 1991.