

논문 2008-03-08

# 다관절 머니플레이터의 사용자 중심 제어기 설계

(User-Oriented Controller Design for Multi-Axis Manipulators)

손현석, 강대훈, 이장명\*

(HeonSuk Son, DaeHoon Kang, JangMyung Lee)

Abstract : This paper proposes a PC-based open architecture controller for a multi-axis robotic manipulator. The designed controller can be applied for various multi-axes robotic manipulators since the motion controller is implemented on a PC with its peripheral devices. The accuracy of the controller based on the computed torque method has been measured with the dynamic model of manipulator. Since the controller is implemented in the PC-based architecture, it is free from the user circumstances and the operating environment. Dynamics of the manipulator have been compensated by the feed forward path in the inner loop and the resulting linear outer loop has been controlled by PD algorithm. Using the specialized language, it can be more efficient in programming and in driving of the multi-axis robot. Unlike the conventional controller that is used to control only a specific robot, this controller can be easily changed for various types of robots. This paper proposes a PC-based controller that has a simple architecture with its simple interface circuits than general commercial controllers. The maintenance and the performance of the controller can be easily improved for a specific robot. In fact, using a Samsung multi-axis robot, AT1, the controller performance and convenience of the PC-based controller have been verified by comparing to the commercial one.

Keywords : Computed torque method, Robotic manipulator, PC-based control, Dynamic control

## 1. 서론

산업의 대형화, 정밀화, 자동화가 급격히 이루어지면서 현재 산업현장에서 로봇의 존재는 필수적인 것이 되고 있다. 간단한 링크를 가진 로봇으로부터 용접 및 도장 그리고 정밀 부품의 조립에 쓰이는 고정밀/다관절 로봇에 이르기까지, 산업현장에서 로봇의 종류와 그 적용 범위는 다양하게 되었다 [1-3]. 이런 로봇의 정밀한 제어를 위해서는 그 업무를 수행하기 위한 제어 알고리즘과 또한 여러가지 파라미터 (parameter)로 표시되는 로봇에 대한 정보를 활용하는 최적화된 제어기가 필요하다. 현재까지의 로봇 제어기는 제조 회사와 로봇의 형태 등에 의해서 각각의 파라미터와 기능에 가장 적합하게 구성된 전용의 제어기를 사용하였으며 그것이 당연한

것으로 받아들여졌다.

그 제어기의 구조를 살펴보면, 일종의 산업용 PC 기반의 Windows와 RTOS (Real Time Operating System)의 운영체제를 가지며 그 기반 위에 다 축의 모터 제어를 위한 보드가 장착된다. 서보 드라이브가 로봇관절의 개수만큼 이루어져 있으며 최종적으로 티칭 펜던트 (teaching pendant)를 사용하여 프로그래밍의 편의성을 제공하는 구조를 가진다 [4-7].

본 논문에서는 이러한 구조를 탈피하여 일반적으로 사용되는 범용의 운영체제가 탑재된 PC에 PCI rev2.2 형태의 모션 제어 카드를 메인보드에 장착하고 터미널 보드를 통하여 서보 드라이브와 PC와의 통신을 하는 간단한 구조의 제어기를 구성한다. 사용자 환경은 간단한 명령어를 이용한 프로그래밍을 통해서 제어가 가능한 구조로 설정하였다. 제어에 사용된 명령어는 두 개 혹은 세 개의 아스키(ASCII) 문자들로 구성된다.

본론에서는 우선 본 논문에서 제어하고자 하는 제어 대상인 2축 다관절 로봇의 동역학과 기구학에 대한 기

\* 교신저자(Corresponding Author)

논문접수 : 2008. 4. 15, 채택확정 : 2008. 6. 16.

손현석, 강대훈, 이장명 : 부산대학교 전자공학과

※ 본 논문은 부산대학교 해외장기과건 지원금으로 연구했음.

본적인 사항에 대해서 언급하며, 그 동역학 성분들을 이용하여 로봇의 효율적인 제어를 위한 알고리즘인 계산토크법과 PD제어에 대해서 설명을 한다 [8-10]. 이러한 사항들을 바탕으로 시뮬레이션과 로봇의 실제 구동 실험을 통하여 제안하는 제어기의 성능을 검증하고 사용의 편의성을 보여주고자 한다. 마지막으로 결론부에서는 본 논문에서 제시한 시스템의 효율성에 대해 정리하여 미래의 연구방향을 제시한다.

## II. 로봇의 추종제어를 위한 동역학 모델링

### 1. 동역학 모델링 및 문제점

로봇의 추종제어는 상위제어기에서 계산, 지령한 목표 위치 및 속도 지령을 최소한의 지연시간에 정확하게 모터의 토크 명령을 계산하는 것을 목표로 한다. 지연시간을 최소화하여 정확한 추종명령을 생성하기 위해서는 로봇의 동역학적 모델링이 기반이 되어야 하며 고려되지 않은 불확실한 파라미터와 외부적인 외란에 대해서 강인해야 한다.

$N$  자유도를 갖는 로봇의 동역학 방정식은 일반적으로 다음과 같이 표시된다.

$$\tau = M(q)\ddot{q} + N(q, \dot{q}) \quad (1)$$

여기서  $N$ 은 다음과 같다.

$$N(q, \dot{q}) = C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) \quad (2)$$

이 식에서  $\tau$ 는 로봇관절의 토크 벡터를 의미하며,  $q, \dot{q}, \ddot{q}$ 는 관절의 위치, 속도, 가속도 벡터 성분을 나타내고,  $M(q)$ 는  $n \times n$ 관성행렬,  $C(q, \dot{q})$ 은 구심력과 Coriolis 효과를 포함하는  $n \times n$ 행렬이며,  $g(q)$ 는 중력을 나타내며  $n \times 1$ 벡터,  $f(\dot{q})$ 는 시스템에 주어지는 점성과 Coulomb마찰력을 나타내는  $n \times 1$ 벡터이다.

로봇 제어는 한 관절의 제어를 위하여 토크를 계산하는 과정에 시스템의 다른 축에 관한 운동 정보들이 다 같이 포함되는 비선형 다 변수 시스템이다. 또한 관성 등의 동역학 파라미터들은 로봇 시스템을 구성하는 다양한 기계 부품들의 형상, 중량, 위치 등에 의해서 결정되는 복잡한 값들이다. 이 파라미터들의 해석 과정이 복잡하고 계산시간이 오래 걸리므로 일반적으로 간략화된 동역학 모델을 구하여 제어를 실현한다.

### 2. 계산토크법

계산토크법은 비선형 시스템을 선형화하는 대표적인 제어 방식으로 강인제어(robust control), 적응제어(adaptive control) 등 다양한 제어 알고리즘에 기본적으로 사용된다. 로봇의 동역학 파라미터인  $M(q)$ ,  $N(q, \dot{q})$ 에 기반한 계산토크법 제어기는 다음과 같이 설계될 수 있다.

$$\begin{aligned} \tau &= M(q)\ddot{q}_d + N(q, \dot{q}) \\ &= M(q)(\ddot{q}_d + k_v\dot{e} + k_p e) + N(q, \dot{q}) \end{aligned} \quad (3)$$

여기서  $q_d$ 는 기준 경로,  $e = q_d - q$ 는 추종 에러를 나타내며,  $k_p, k_v$ 는 외부의 피드백 루프의 이득 값으로 각각 상수인 대각 행렬을 의미한다. 식 (3)으로 표시되는 제어기를 사용하게 되면 로봇 모델에서의 비선형성과 마찰력 등이 모두 보상되며, 최종적으로 다음과 같은 시스템의 오차방정식을 얻어서 원하는 목표치를 추종하게 된다.

$$M(q)(\ddot{e} + k_v\dot{e} + k_p e) = 0 \quad (4)$$

계산토크법의 결과에 의해 표시되는 식 (4)의 시스템은 선형시스템으로 PD제어기를 추가하면 추종에러를 지수 함수적으로 줄어든게 설계할 수 있다. 본 연구에서는 계산토크법과 PD 제어기를 결합한 제어기를 실제 산업용 로봇에 적용하여 로봇의 추종 오차를 최소화하여 제어 대상을 목표치에 수렴하도록 제어한다.

## III. PC 기반 제어기 설계

### 1. PC 기반 제어기의 필요성

현대의 생산 시스템은 제품에 대한 소비자들의 다양한 요구를 만족시키고, 생산기술의 변화와 PC의 급속한 발전 속도에 신속한 적응이 가능하도록 하기 위해 유연성(flexibility), 통합성(integration), 동시성(concurrency)을 만족시킬 수 있는 개방형 구조로의 전환이 요구된다. 산업용 로봇은 다양한 작업 기능과 사용자 프로그래밍이 가능한 구조로 되어있으나, 제작사마다 자사의 고유한 언어 구조의 제어시스템인 폐쇄형 구조(closed architecture)로 되어있다. 이러한 제어기는 제어대상의 모든 요건을 정확히 파악하고 제어한다는 점에서 장점이 있지만, 다른 로봇들의 디바이스들과의 호환이 어려우며, 또한 고장이 발생하는 경우 유지보수가 힘든 문제점을 안고 있다. 이런 문제점들을 극복하고 경제적이며 사용자의 편의성이 우수한 PC기반의 제어기를 필요로 한다.

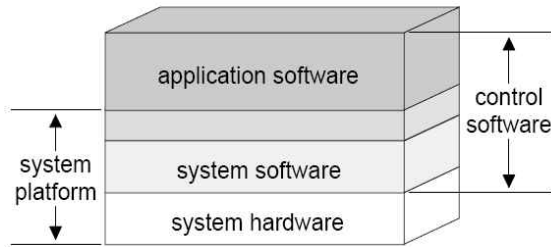


그림 1. PC 기반 제어기의 일반적 구조  
Fig. 1. Basic architecture of PC based controller.

PC 기반의 제어기를 사용할 경우, 다양한 로봇을 정확하게 제어할 수 있으며, 광범위하게 적용 분야를 넓힐 수 있다는 장점이 있다. 그림 1은 PC 기반 제어기의 일반적 구조를 보여준다. 만약 로봇 특성의 파악이 미흡할 경우에는 불확실한 파라미터들에 의한 비선형 성분들을 보상할 수 있는 제어 알고리즘인 신경망과 퍼지제어 등의 제어이론의 추가적인 적용으로 성능을 향상시킬 수 있다.

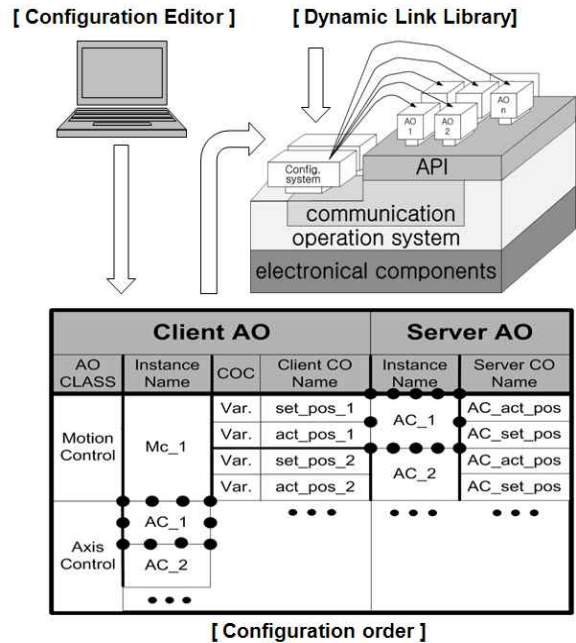
2. 제안된 사용자 중심 PC 기반 제어기의 실체

PC 기반 제어기는 머니플레이터의 객체지향형의 구조와 더불어 적절히 활용된다. 그림 2는 이미 OSACA라는 이름으로 독일의 Stuttgart대학의 연구소에서 개발된 산업 자동화를 위한 개방형 제어시스템을 나타낸 것으로 본 연구에서 기준 모델로 활용하여 제어기를 설계하여 구현하고자 한다. 그림을 통해 알 수 있듯이 이 모델은 전자적 구성요소, 운영시스템 모듈, 그리고 사용자 환경과 제어 알고리즘을 포함하는 응용 소프트웨어로서 구성되어 있다.

가. 하드웨어 플랫폼

일반적으로 이 부분은 계산을 담당하는 프로세서, 다른 외부장치로 입출력을 담당하는 I/O장치, 데이터의 저장과 로딩을 맡는 메모리, 그리고 제어 시스템을 사용하는 사용자의 편의를 제공하기 위한 주변 장치로서 보조 입력장치 혹은 출력 장치들로 이루어진다. 이런 하드웨어 플랫폼의 구성요소들은 일반적으로 사용되는 PC의 각 부분으로서 유지, 보수 등이 간단하며, 타사의 제품과 호환이 가능하도록 표준화가 되어 있다. 또한 다양한 장치의 사용이 원활하도록 하드웨어 드라이버가 충분히 제공된다. 본 연구에서 사용된 하드웨어 플랫폼 PC의 사양은 다음과 같다. CPU는 Intel사의 Pentium IV 1.2G 프로세서, PCI 카드 모션 제어기가 장착된 Gigabyte 메인보드, RAM은 512 Mbyte DDR 메모리를 사용하였다. 입력장치는 마우스와 키보드를 사용하고 하드웨어의 구성에 대한 추가가 필요하다면 그 변경 및

추가가 용이하도록 구성하였다.



AO : Architecture Object.  
API : Application Program Interface

그림 2. OSACA 표준 기준 모델

Fig. 2. Proposed reference model for the OSACA

나. 운영 시스템(Operating System)

운영 시스템은 제어 시스템에 있어서 하드웨어와 응용 소프트웨어, API (Application Programming Interface)등에 대해 개발자가 개발한 주변장치 및 소프트웨어에 대한 활용이 가능하도록 유효성(Availability)을 제공해야 한다. 하나의 메인 프로세서를 이용한 PC 기반의 제어기를 사용하여 산업용 대형 다관절 머니플레이터 제어를 할 때, 추종 제어를 위해서 특성 해석 및 궤적 계획에 연산시간이 많이 소요되므로 정해진 시간 내에 작업을 마무리 할 수 있도록 실시간 운영시스템 (Real Time Operating System: RTOS)이 반드시 필요하다. 이러한 실시간 운영시스템과 일반적인 운영체제의 중요한 차이는 CPU의 시간관리방법, 메모리의 할당, 인터럽트에 대한 응답과 기계장치를 구동하는 처리과정에서 자원공유방법 등이다.

본 논문에서는 실시간 운영체제를 적용하지 않았으며, 그 이유는 로봇의 2축에 대한 파라미터 값들을 비교적 정확하게 알고 있으므로, 로봇 특성 분석에 따른 연산을 줄일 수 있었기 때문이다. 또한 비교적 간단한 궤적 추종제어 실험을 통하여 성능을 검증하므로, 범용의 운영체제인 Microsoft사의 Window XP를 사용하였다. 관절수의 추가에 의한 영향과 궤적계획 등, 연산의 실시간성

이 보장되어야 할 경우는 실시간 운영체제를 도입하여 설치할 수 있을 것이다. 이와 같이 개방형 제어기의 경우 운영체제 역시 제어 성능에 맞도록 사용자가 구축할 수 있다는 장점을 지닌다. 고성능의 운영체제가 필요치 않다면 Linux와 같은 가격이 저렴한 운영체제의 사용으로 경제적 효율성을 기대할 수 있다.

#### 다. 응용 소프트웨어(Application software)

제안된 형태의 제어시스템에서 응용 소프트웨어 모듈은 사용자 또는 제어 대상의 문제에 의해 기능의 추가, 삭제 혹은 변경이 허용되고, 각 구성 요소 간에 정보 교환이 가능한 형태인 객체 지향적 형태를 따라야 한다. 이와 같은 특성을 만족하기 위해서 응용 소프트웨어 모듈은 전체적 관리, API, 제어객체 등으로 분류하여 생각할 수 있다.

##### (1) 관리

제안된 형태의 제어시스템의 가장 중요한 특징은 여러 요인에 따라 구현된 환경의 재설정이 가능하다는 것이다. 여기서 관리는 이러한 형태의 제어시스템의 재설정 역할을 담당하여 수행하게 되는 부분을 말한다. 환경의 재설정을 위해서 먼저 사용자는 미리 제어시스템에 관한 기능과 성능을 포함한 다양한 것에 대한 분석을 한다. 이를 바탕으로 On-line 상에서 소프트웨어의 추가, 편집을 하게 되고, 이를 통하여 새로운 제어 시스템의 형태로서 재설정이 가능하다. 이렇게 설정된 제어기를 활용하여 제어시스템의 동작의 이상 유무를 확인한다.

##### (2) API (Application Programming Interface)

산업용 머니플레이터는 생산제품의 형태 혹은 생산 공정의 변화에 따라 작업에 이용되는 Tool 혹은 여러 가지 디바이스들의 재구성 또는 설치를 하게 된다. 이러한 구조적, 기능적인 변화에 따라 제어시스템의 구성요소도 재조정이 되어야만 필요한 성능을 최적의 형태로 구현해 낼 수 있을 것이다. 이와 같은 제어시스템의 유연성이나 확장성에 대한 문제점은 머니플레이터의 제어기를 구성하는데 많은 어려움을 유발하게 된다. 이런 어려움을 해결하는 방안으로 하드웨어나 소프트웨어 측면에서 기존의 하드웨어나 소프트웨어와 공통된 인터페이스 환경을 구축할 수 있는 범용 API 모듈을 시스템에 포함시킨다. 이 API 모듈을 활용하면 제어시스템의 모든 구성 요소가 쉽게 확장 또는 변경될 수 있다. 본 연구에서는 모든 제어시스템의 구현을 Microsoft사의 Visual C++ 프로그래밍으로 하였다. 머니플레이터에 정밀한 제어 환경을 구현 하기 위해서 다양한 장치들을 추가할 경우, Feedback과 정보처리에 관한 Visual

C++의 Programming에 의한 소스 코드(Source Code)의 수정이 불가피하다. 현재 구현된 API는 머니플레이터 각축의 모터와 함께 있는 절대치 인코더의 값을 되 받아서, 오차 등에 대한 보상을 통하여 로봇 파라미터의 특성에 맞는 제어 알고리즘인 계산토크법에 따라 토크의 계산이 가능하도록 구현되어 있다. 그러나 추가적인 제어 성능의 개선이 필요하다면, 이 역시 API 프로그램의 수정을 요한다. 응용 소프트웨어에서 가장 중요한 부분의 하나라고 할 수 있는 사용자 환경은 Visual C++ MFC를 통하여 그래픽 기반으로 개발하는 것이 통상적이다.

본 연구에서는 MFC를 이용하여 GUI를 구축하였으며, 모션제어기와와 상호 인터페이스 환경을 제공함으로써 사용자의 명령을 모션제어기로 전달하고 또한 모션제어기로부터의 다양한 측정값을 표시하여, 사용자가 머니플레이터의 동작 및 상태를 파악할 수 있도록 하였다. 그리고 모션제어기와와 연결 상태와 모터의 종류를 알 수 있도록 하였다. 이런 사용자 환경의 구성을 위해서 모션제어기가 어떤 기능이 가능한지 알아보고, 그에 따른 원하는 환경을 구현하기 위해 DLL(Dynamic Link Library)의 매뉴얼을 분석하였다. DLL은 하드웨어간의 인터페이스 환경을 구현하기 위해 활용되지만, 하드웨어와 소프트웨어의 상호 작용을 돕고 하드웨어에 할당된 기능을 효율적으로 활용하도록 프로그래밍하기 위하여 필요하다. 대부분의 하드웨어는 DLL형태로 인터페이스할 수 있도록 되어있다.

## IV. 시뮬레이션 및 실험

### 1. 계산토크법은 이용한 제어 성능 시뮬레이션

우선 계산토크법을 이용한 제어기의 실제 머니플레이터의 구동 실험에 앞서 Matlab 시뮬레이션을 통해 그 성능을 검증하고자 한다. 그림 3은 계산토크법을 활용하여 제어를 하였을 경우에 머니플레이터의 추종 오차가 감소하고 있음을 보이고 있다. 첫 번째 그림은 제어 플랜트 모델과 출력과의 추종 여부를 나타내고 있는 것으로 일정 시간이 경과함에 따라서 둘의 출력이 거의 일치하게 추종하고 있음을 나타낸다. 두 번째 그림은 오차량이 시간의 흐름에 따라 감소하여 최종적으로 오차량이 0으로 수렴하고 있음을 나타낸다.

이와 같이 시뮬레이션 결과에서 알 수 있듯이 계산토크법은 동역학 파라미터들로부터 계산된 토크를 바탕으로 비선형의 성질을 가지는 제어 대상을 선형화하여 오차 성분을 감소시킨다는 것을 알 수 있다. 계산토크법을 실제 머니플레이터에 적용시킬 경우, 어떠한 성능을 나타내는지, 다음의 실험을 통해서 그 결과를 고찰하고자

한다.

그림 4는 본 연구에서 사용한 제어시스템의 구성을 보여준다. 로봇은 상용의 AT1 본체의 2축과 3축만을 제어대상으로 활용 한다.

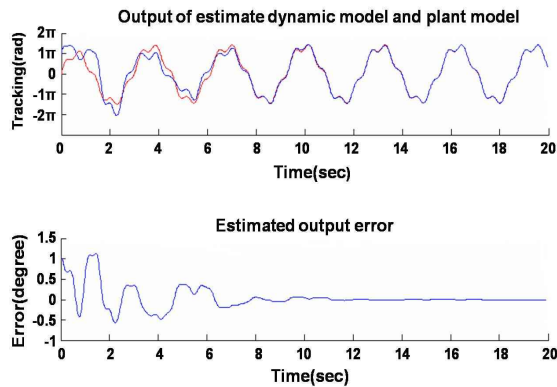


그림 3. 계산토크법을 이용한 오차 개선  
Fig. 3. Tracking performance by computed torque method

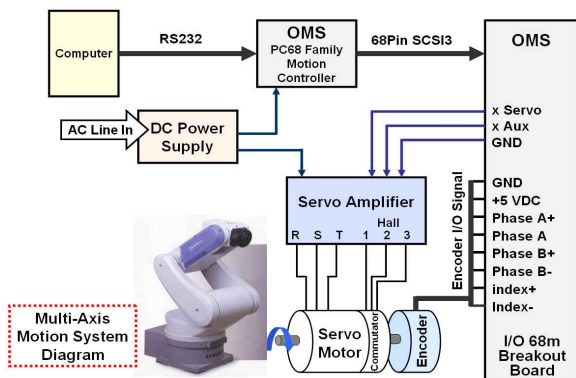


그림 4. 제안된 제어기의 구성도  
Fig. 4. Configuration of the proposed control system

2. A11 로봇을 이용한 제어기 성능 평가

본 논문에서 제안하고자 하는 제어시스템은 제어부와 구동부, 그리고 제어 대상이 되는 머니플레이터로 크게 세 부분으로 분류할 수 있다.

가장 핵심이 되는 제어부는 일반적으로 사용되는 PC 메인보드에 장착되는 PCI 카드 형태의 모션제어기이다. 모션제어기는 일반적으로 다 축의 모터를 제어하기 위한 제어 장치이다. 본 연구에 사용된 모션제어기는 2축과 3축 제어만을 위해 사용하지만 필요할 경우 최대 8개의 모터를 확장하여 제어할 수 있는 성능을 가진다.

실험은 모션제어기에서 제공하는 모든 기능을 가장 손쉽게 사용하기 위해서 EZ-Command 라는 모션 제어

기 전용의 명령어를 입력 받아 사용한다. MFC를 이용하여 사용자 환경을 구성 하였으며, 입력 창을 통하여 머니플레이터의 원하는 축의 구동 명령을 프로그래밍하여 지령한다. 토크 계산법과 PD 제어기의 결합된 형태의 제어기 구현을 위해서 Visual C++을 이용하여 소스 코드를 프로그래밍 한다. 그리고 모션 제어기의 DLL과 일을 이용하여 모션제어기와의 소프트웨어를 통한 인터페이스가 가능하도록 환경을 제작하였다 [11].

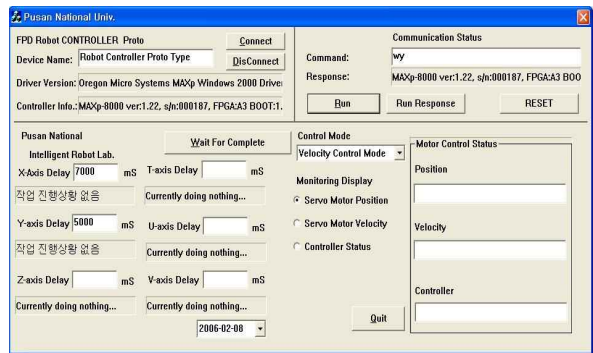
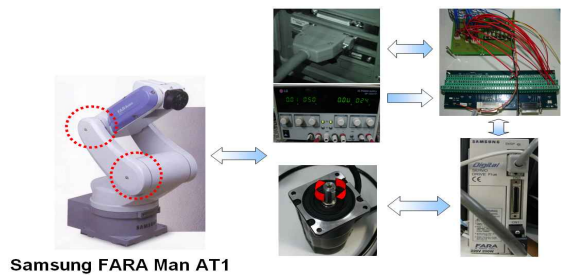


그림 5. 제안된 PC 기반 제어기의 실제 구조  
Fig. 5. Proposed PC based controller structure

서보 드라이브는 정격 용량이 200W인 삼성의 CSDJ Plus Series를 사용하였으며 머니플레이터는 앞서 말한 바와 같이 삼성전자의 Faraman AT1을 사용하여 시스템을 구성하였다. 제어대상은 두개의 축으로 한정하여 실험을 진행하였으며 AT1의 2, 3축을 대상으로 평면 2관절 머니플레이터의 형태로 제어한다.

실험에 사용된 제어 시스템의 형태는 그림 5와 같다. 제안된 제어기를 사용하여 반복 정밀도 및 오차 범위에 대한 측정, 속도 오차, 계산토크법을 이용한 적합한 토크 지령의 확인, 3가지 항목의 실험을 실행하였다. 이 모든 측정에는 머니플레이터의 구동축에 사용된 AC 서보모터에 부착된 절대치 인코더를 이용하여 모터의 회전량에 대한 실험 데이터를 연산과정을 거쳐 결과 값으로 나타내었다. 또한 제어 지령에 관한 머니플레이터의 구동 과정을 모니터링 하여 실험결과를 도출하였다.

실험에서 사용된 속도는 포물선 속도 궤적으로, 모션 제어기는 그림 5의 메뉴에서 선형, Cosine, 포물선 속도

케적을 제공한다. 이들 중, 머니플레이터의 구동에 있어서 응답 속도가 빠르고 가 감속이 비교적 완만한 형태인 포물선 형태의 속도 케적을 사용하여 실험을 진행하였다 [12].

실험 1. 반복 정밀도

본 실험은 같은 케적을 반복하여 1000회를 움직였을 때 나타나는 오차 값에 대한 그래프이다. 실험을 통해 얻어진 결과 값을 쉽게 알아볼 수 있도록 하기 위해서 오차 량에 절대 값을 취하여 나타내었다. Y축은 오차 량으로 단위는 mm이다. 각축의 인코더 값을 얻고, AT1 로봇의 AC 서보 모터의 인코더는 1회전 당 2048의 펄스 신호를 발생하므로 각 축의 기어 비를 통하여 연산된 값을 거리 량으로 나타내었다.

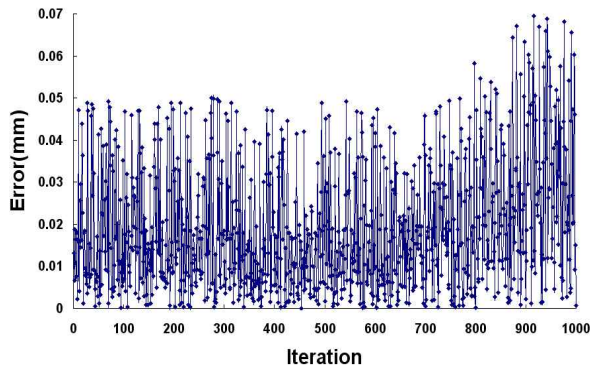


그림 6. AT1 Lower Arm의 반복 정밀도(2축)  
Fig. 6. Repeatability of AT1 lower arm (2nd axis)

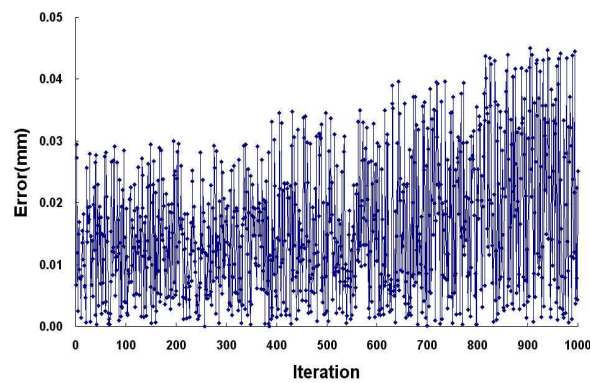


그림 7. AT1 Upper Arm의 반복 정밀도(3축)  
Fig. 7. Repeatability of AT1 upper arm (3rd axis)

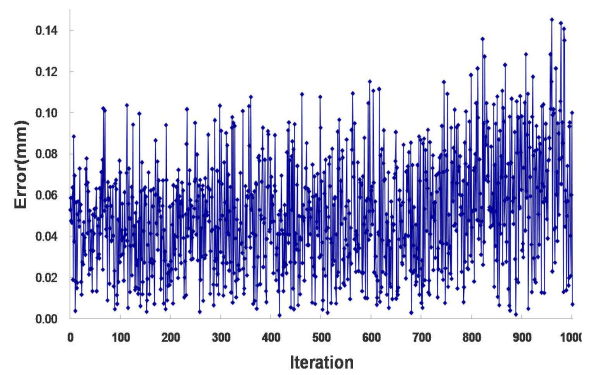


그림 8. AT1 Arm의 반복 정밀도(2축 및 3축 동시제어)

Fig. 8. Repeatability of AT1 arm (2nd and 3rd axis)

위의 결과 그림 6과 그림 7, 그림 8을 통하여 본 논문에서 제안한 제어기의 반복 정밀도를 검증하였다. 기존의 AT1의 제어기의 성능은 0.05 mm의 반복 정밀도의 요건을 만족하지만 실험에서는 Lower Arm의 경우 대략 800회의 반복 운동 이후부터, 다소 오차 량이 많이 증가하여 약 0.07 mm의 오차가 발생하였다. 하지만 Upper Arm의 경우 반복운동이 늘어날수록 오차량이 증가하는 양상을 보이긴 하였지만 대체로 0.05 mm 내에서 반복 운동을 하고 있음을 알 수 있었다. Upper Arm과 Lower Arm을 동시제어 하였을 경우에는 대체로 0.1 mm 정도의 오차를 가지다가 800회의 반복 운동 이후부터 오차량이 증가하여 0.14 mm 내에서 반복 운동을 하였다. 이를 통하여 반복 정밀도는 모든 것이 최적화된 파라미터를 가진 전용의 제어기에 비해서 다소 미흡하였으나, 향후 머니플레이터 제어에 적합한 제어 알고리즘의 적용을 통하여 대부분 개선될 수 있으리라 판단한다. 전반적으로 반복 정밀도는 횡수의 반복에 따라 그 성능이 저하되긴 하였으나 급격히 변동하지 않는 것으로 보아, 반복 정밀도의 안정성을 검증할 수 있었다.

실험 2. 속도 추종 능력

머니플레이터는 허용된 가장 빠른 속도로 목표점에 도달할 때, 그 성능이 우수하며 활용에 따른 효율이 높다. 이번 실험은 제어기에서 주어진 속도지령에 얼마만큼의 오차를 가지는 가를 가늠하기 위한 실험으로서 실험 조건은 머니플레이터의 자중만이 가해진 무 부하 상태에서 2, 3축을 동시제어 하여 500 rpm의 속도를 얼마나 빠르게 추종하는 가를 나타낸다.

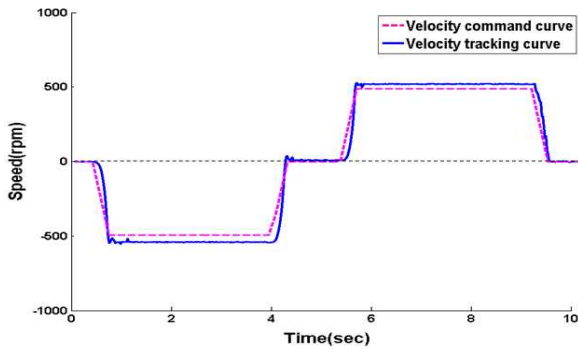


그림 9. 속도 지령에 따른 추종  
Fig. 9. Velocity tracking trajectory

실험 3. 계산토크법에 의한 제어 지령의 확인

본 연구에서 제안한 토크 계산법은 머니플레이터의 동역학 파라미터들을 분석하여 그것에 최적의 토크를 계산하여 원하는 궤적을 추종하기 위해 사용되었다. 이것을 확인하기 위하여 실험 2의 조건과 마찬가지로 왕복 운동을 하는 머니플레이터의 토크 지령을 관측하였다. 그 결과는 그림 10과 같다.

이 결과는 머니플레이터의 반복 운동 궤적에 따른 토크 지령을 나타낸 것으로 머니플레이터 동역학 성분의 영향을 상대적으로 많이 받는 Lower Arm에 내려지는 지령을 관찰한 것이다. 그림 10에서 토크가 증가하는 부분은 상대적으로 많은 토크가 필요한 머니플레이터가 왕복 운동을 시작하는 시점이며 감소하는 부분은 큰 토크가 필요하지 않은 정속 운동을 하는 구간과 머니플레이터가 하강 곡선을 그리는 부분을 나타낸다. 이것을 볼 때, 머니플레이터의 동역학에 기반한 계산된 토크 지령이 적절히 활용되고 있으며 이에 의한 제어가 이루어지고 있음을 알 수 있었다. 즉, 머니플레이터의 자세에 따라 모터에 가해지는 하중과 관성력에 따른 다른 크기의 토크 지령을 발생함으로써 주어진 속도 궤적에 적합한 가속도와 속도를 유지하며 안정적인 운동이 이루고 있음을 확인할 수 있다. 위의 실험들을 통해서 본 연구에서 제안하는 사용자 중심의 PC기반 제어기의 성능을 검증해 보았다.

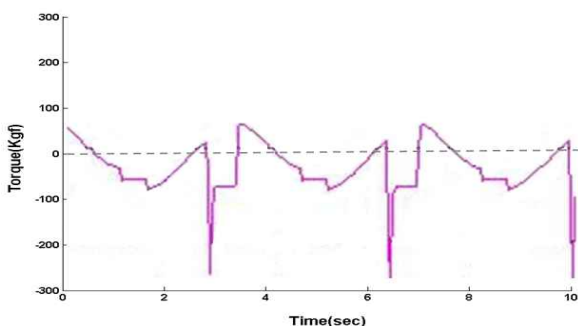


그림 10. 머니플레이터 왕복 운동에 따른 토크지령  
Fig. 10. Torque commands for iterative motion

V. 결론

본 논문에서는 머니플레이터를 보다 효율적이고 안정적으로 제어하기 위하여 PC기반의 제어기를 구현하였다. 제어 알고리즘은 잘 알려진 계산토크법을 사용하였고 실험을 위해서 머니플레이터의 동역학 성분을 모델링하여 그 값을 사용하였다. 또한 시뮬레이션을 통하여 계산토크법에 의한 추종 궤적의 오차가 지수 함수적으로 수렴하게 됨을 알 수 있었다. 상용인 AT1 로봇의 몸체에 본 연구에서 제안한 제어 알고리즘을 적용하여 반복정밀도, 속도 추종, 계산된 토크 지령을 확인함으로써, PC 기반 제어기 및 제어알고리즘의 성능을 최종적으로 검증 하였다. 뿐만 아니라 모션제어기에서 사용되는 전용 언어인 EZ-Command를 이용하여 모션제어기의 내부에서 구현된 다양한 기능을 필요에 따라 전부 활용할 수 있도록 사용자 환경을 구축하여 사용의 편의성을 검증하였다. 제안한 PC기반 제어기는 사용자의 다양한 요구에 따라 PC의 세부 사양부터 제어 시스템을 구성하는 장치에 이르기까지 모든 것을 원하는 대로 구성할 수 있으므로 확장성과 호환성이 우수하여 복잡하게 다양화 되고 있는 산업 현장에서 그 성능을 발휘할 수 있으리라 생각된다.

그리고 향후, 실제 산업용 로봇으로 사용되는 4축 이상의 완전한 PC기반 제어환경에서는 비선형 성분의 보상을 위한 강인제어 알고리즘 혹은 신경망과 퍼지 등 지능형 제어 알고리즘의 추가가 필요할 것이며, 다양한 통신 Protocol을 활용한 Network기반의 서버 드라이브와 모션 제어기에 관한 연구가 필요할 것이다.

참고 문헌

- [1] R. Gourdeau and H. M. Schwartz, "Optimal control of a robot manipulator using a weighted time energy cost function," Proc. IEEE Conf. Decision Control, pp. 1678-1681, Dec. 1989.
- [2] Kyung-Sook Bae, Jae-Yeon Lee, Keun-Chang Kwak, and Ho-Sub Yoon, "Control network design for multi-body robot based on HEEEB," Journal of JEMEK, Vol. 2, No. 1, pp. 21-26, 2007.
- [3] Bang-Hyun Cho, Byoung-Suk Choi, and Jang-Myeong Lee, "Time optimal trajectory planning for a robot system under torque and impulse constraints," International Journal of Control, Automation, and Systems, Vol. 4, No. 1, pp. 10-16, Feb. 2006.

- [1] B. K. Kim and K. G. Shin, "Suboptimal control of industrial manipulator with a weighted minimum time fuel criterion," IEEE Trans. Automatic Control, Vol. AC-30, No. 1, pp. 1-10, Jan. 1985.
- [2] E. G. Gilbert and I. J. Ha, "An approach to nonlinear feedback control with application to robotics," IEEE Trans. Sys. Man Cybern., Vol. SMC-11, No. 6, pp. 879-881, 1981.
- [3] L. Hunt, R. Su, and G. Meyer, "Global transformations of nonlinear systems," IEEE Trans. Autom. Control, Vol. AC-28, No. 1, pp. 24-31, Jan. 1983.
- [4] J. E. Bobrow, S. Dubowsky, and J.S. Gibson, "On the optimal control of robotics manipulator with actuator constraints," Proc. American Control Conf., pp. 782-787, Jun. 1983.
- [5] Z. Sifler and S. Dubowski, "Time optimal paths and acceleration lines of robotic manipulators," Proc. of the 30th Conf. Decision and Control, pp. 98-99, 1989.
- [6] 김성우, 이주승, "강화학습이 갖는 로봇 시스템의 학습 능력 향상의 의미," Journal of Control, Automation and Systems Engineering, Vol. 1, No. 2, pp. 88-90, Dec. 1990.
- [7] 김성우, 이주승, 홍금식, "A PC based open robot control system: PC ORC," Journal of Control, Automation and system Engineering, Vol. 6, No. 5, pp. 115-125, May 2000.
- [8] P. Lutz and W. Sperling, "OSACA - the vendor neutral Control Architecture," Proceedings of European Conference on Integration in Manufacturing (IM), pp. 128-133, Oct. 1997.
- [9] P. Lutz and W. Sperling, "Designing applications for an OSACA control," Proceedings of the International Mechanical Engineering Congress and Exposition (The ASME Winter Annual Meeting) Dallas, USA, pp. 16-21, Nov. 1997.

## 저 자 소 개

### 손 현 석



2007년 진주산업대학교  
전자공학과 졸업.

2007 ~ 현재, 부산대학교  
대학원 석사과정 재학 중.

관심분야 : AC서보드라이브, 다관절 로봇제어,  
마이크로프로세서 응용제어.

Email : [yankiss@pusan.ac.kr](mailto:yankiss@pusan.ac.kr)

### 강 대 훈



2003년 부경대학교 제어  
계측공학과 졸업.

2007년 부산대학교 메카  
트로닉스 석사과정 졸업.

2007년 ~ 현재, 현대 제철  
근무 중.

관심분야 : 로봇제어, 비선형제어.

Email : [dhkang@pusan.ac.kr](mailto:dhkang@pusan.ac.kr)

### 이 장 명



1980년 서울대학교 전자  
공학과 공학사.

1982년 서울대학교 전자  
공학과 공학석사.

1990년 University of  
Southern California 공학  
박사.

1992년~현재, 부산대학교 전자공학과 교수.

관심분야 : 다관절 로봇 시스템 설계 및 제  
어, 협동 제어, 이동 물체의 위치 추적, 마이  
크로프로세서 응용 제어.

Email : [jmlee@pusan.ac.kr](mailto:jmlee@pusan.ac.kr)