

인간 학습을 이용한 산업용 로봇의 효율적 프로그래밍 방안

Industrial Robot Programming Method Utilizing the Human Learning Capability

김 성 수, 이 종 태
동국대학교 산업공학과

ABSTRACT

Nowadays, most shop floors using industrial robots have many problems such as constructing robot workcell, generating robot arm moving trajectory, etc.. In the case of programming robot-arms for a specific task, shop operator commonly use the teach pendant to record the target position and determine the moving trajectory. However, such a teaching process may result in an inefficient trajectory in the sense of moving distance and joint angle fluctuation. Moreover, shop operators who have little knowledge about robot programming process need a lot of learning time and cost.

The purpose of this paper is to propose a user friendly robot programming method to program robot-arms easily and efficiently for shop operator so that the programming time is reduced and a short and stable trajectory is obtained.

1. 서론

급속히 발전하고 있는 현대 산업사회에 있어서 로봇의 활용은 공장 자동화를 이룩하고 나아가 무인공장의 핵심부분의 하나인 자율적인 로봇(Autonomous Robot) 시스템을 구축하기 위한 필수적인 요소이다. 현재의 산업용 로봇은 지속적인 개발 단계에 있고 사용되는 범위가 점차 확산되어 가고 있지만, 기술적인 측면이나 경제적인 부담으로 인하여 산업 전반에 걸쳐 보편화되고 있지는 못하다.

현재, 생산현장에서 산업용 로봇을 이용하는 데에는 로봇 작업영역(robot workcell)을 구축하고 이동경로를 결정하는 등에 따르는 프로그래밍의 부담과 같은 문제점이 있다. 특정한 작업에 대해 산업용 로봇-팔을 프로그래밍(programming) 할 경우, 현장프로그래머(user or shop operator)들은 일반적으로 티치 펜던트(Teach Pendant)를 사용하여 작업목표점을 입력하고 입력된 목표점(target point)을 기초로하여 이동경로를 설정하고 있다.

그러나, 이러한 교시과정(Teaching Process)은 로봇-팔의 이동경로를 설정하는데 있어서 경로상의 불필요한 움직임을 초래할 수 있으며, 관절각의 과도한 변이를 유발할 수도 있다. 또한, 로봇에 대한 기초지식이 미비한 현장의 작업자들을 교육시켜 프로그래머로 양성하는 데에는 많은 시간적인 손실과 경제적인 부담을 가져온다.

따라서, 본 논문의 목적은 현장프로그래머들이 로봇-팔을 프로그래밍하는데 있어서 좀 더 손쉽게 효율적인 이동경로를 설정할 수 있게 하는데 있으며, 프로그래밍 시간을 단축하고 최적 이동경로를 생성할 수 있는 인간학습을 이용한 로봇-팔 프로그래밍 방법을 제안하는데 있다.

본 연구는 현장 프로그래머들을 대상으로 하고 있으며, 보다 상세한 현장의 프로그래밍 방법과 문제점들을 알아보기 위하여 현장 프로그래머들에 대한 설문조사 결과를 바탕으로 본 연구에서 수행할 실험들에 대한 실험절차와 실험방법을 결정하였다.

실험에서는 미리 결정한 3가지 유형의 작업을 4명의 피실험자가 수행하는 동안 로봇-팔의 1축과 2축의 이동각도에 대한 총 누적치(input key의 횟수)를 계산하여 이동경로에 대한 각 관절각 변화를 측정하였고, 이에 따라 소요되는 시간도 측정하여 비교해 보았다.

실험에는 수평왕복 운동을 하면서 조립작업에 이용되는 2관절 삼축 SCARA형 로봇(SM2)를 모델로 사용하였고, 로봇-팔의 이동경로를 설정하기 위한 실험이므로 gripper에 대한 작업은 배제하였다.

컴퓨터를 이용한 시뮬레이션용 로봇-팔은 Turbo C language와 Turbo C++를 이용하여 작성하였고, 인간학습용 로봇-팔을 이용한 실험결과 데이터를 PC상으로 받아들이기 위하여 Versatile I/O card라는 interface card와 Rotary encoder라는 센서를 사용하였다. 또한, 두가지 실험을 통하여 얻어진 데이터를 분석하는 데에는 한글 EXCEL Version 5.0을 사용하였다.

2. 현장의 로봇 프로그래밍 방법 및 문제점

산업현장에서 일평균 로봇을 가동하는 시간은

부서나 작업의 종류에 따라 다소 차이는 있었지만 거의 10시간 정도이며, 한번 생산라인에 투입된 로봇은 5년을 기준으로 새로운 형태로 교체되어지고 있었다. 그러므로 로봇 워크셀(Robot Workcell)을 구축하는 초기에는 합리적인 작업장 설계가 필요하며, 로봇-팔을 효율적으로 프로그래밍하여 잘못된 경로에 의한 많은 시간과 비용의 손실을 방지해야 한다.

로봇-팔에 대한 프로그래머를 양성하는 경우 현장의 작업자들을 적게는 3일 정도에서 많게는 6개월 이상 교육을 시켜 수행하고 있다. 교육된 프로그래머가 주어진 작업에 대해 로봇-팔을 프로그래밍 할 때에는 주로 티치 펜던트(Teach Pendant)를 이용하여 목표점들을 입력하고 PC상에서 로봇 고유언어로 프로그램을 작성한 후 컨트롤러로 download하여 사용하고 있다. 이러한 프로그래머들에 대한 교육과정이나 프로그래밍 과정은 많은 경제적 투자를 요구하고 있다.

또한, 현장에 있는 현장프로그래머들은 기존에 발표된 로봇틱스에 관련된 이론이나 문헌에 대하여 큰 관심을 나타내지 못하고 있는데, 이는 이들은 지극히 수학적이고 복잡한 계산이나 알고리즘을 원한다기 보다는 사용자 위주로 프로그래밍이 가능한 편리한 접근방법을 원하고 있음을 나타낸다.

3. 인간학습에 의한 로봇-팔 프로그래밍 방안

3.1 제안된 프로그래밍 방법의 구성

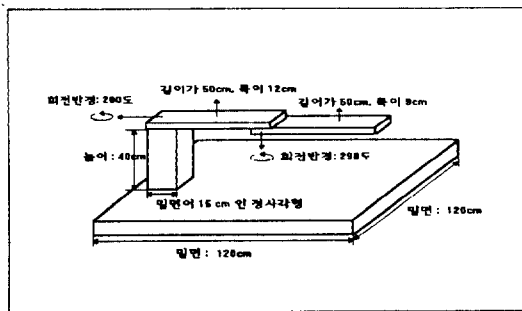
본 논문에서 제시하는 프로그래밍 방안은 총 3단계로 구성되어져 있으며, 그 내용은 다음과 같다.

- 1단계 : 실제 로봇 운영환경을 물리적인 경로 생성 장치로 전환하여 구축한다.
- 2단계 : 설치된 로봇-팔 모형을 이용하여 인간 조작자가 주어진 작업에 대한 경로를 생성한다.
- 3단계 : 얻어진 인간학습 데이터를 평가하여 PC 시뮬레이션 프로그램과 비교한다.

3.2 로봇-팔 모형의 제작

본 실험에서는 수평왕복운동을 하는 조립용 2관절 삼성 SCARA 로봇-팔 모델을 사용하였다. 삼성 SCARA 로봇은 자유도 4를 갖고 있는데, 1축과 2축은 수평왕복운동을 할 수 있고 운동영역범위는 각축이 $\pm 200^\circ$ 와 $\pm 290^\circ$ 로 제한된다. 3축과 4축은 gripper 부분으로써, Z축과 θ 축으로도 불리우며, 수직운동과 회전운동이 가능하다.

다음의 [그림 1]은 제작된 로봇-팔 모형의 사양을 나타낸다.



[그림 1] 제작된 로봇-팔의 사양

본 실험에서 제작된 로봇-팔 모형은 최대한 실제 로봇-팔의 자유도와 사양에 충실하도록 제작되었고, 로봇 본체와 작업장은 나무를 이용하여 구축되었다. 또한 본실험의 목적은 로봇-팔의 이동경로를 분석하기 위한 실험이므로 gripper(3축과 4축)는 부착하지 않고 다만 목표점에 로봇-팔의 끝부분이 도달하였다는 것을 피실험자에게 인지시키기 위하여 모형의 끝에 기준축을 달아주었다.

4. 실험목적 및 실험방법

4.1 실험의 목적

본 실험의 목적은 현장 프로그래머들이 현재 널리 사용하고 있는 티치 펜던트(Teach Pendant)를 이용한 로봇-팔 이동경로에 대한 교시(Teaching) 방법과 본 논문에서 제시하는 로봇-팔 모형을 이용한 인간학습 교시방법을 비교하여 교시과정(Teaching Process)에 있어서의 프로그래머들간의 특정 작업에 걸리는 로봇-팔 이동거리(총 key input의 횟수)의 차이와 프로그램을 하는데 걸리는 프로그램 set-up time을 조사하는 것이다.

이를 통하여 똑같은 작업이 주어졌을 경우 이동경로를 설정하는데 사용자에게 따라 얼마만한 이동경로의 차이를 나타내고 있으며, 학습횟수에 따른 학습효과가 어떻게 변화하는가를 파악한다. 또한, 경로설정을 하는데 걸리는 경로생성시간도 평가한다.

4.2 실험의 가정

현실 여건상 기존에 생산된 SCARA 로봇(SM2)와 기계적으로 똑같은 인간학습용 로봇-팔 모델을 제작한다는 것은 많은 문제점이 있으므로, PC상에서 SCARA 로봇을 시뮬레이션용 로봇-팔로 프로그래밍하여 사용하였다. 또한, 산업용 로봇-팔의 이동경로에 대한 프로그래머들 간의 작업 경로설정의 차이와 결정경로에 대한 총 input key 횟수(각도 누적치)의 차이를 비교하기 위한 실험이므로 그리퍼(gripper)에 대한 작업은 배제하였고, 작업에 소요되는 작업시간은 각 작업마다 동일한 시간을 갖는다고 가정하였다.

4.3 수행작업의 결정

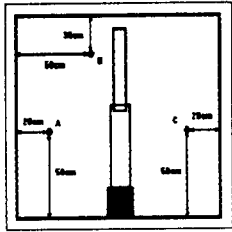
본 실험에서는 수행되는 작업을 총 3가지의 형태로 결정하였다. 첫번째 작업은 목표점이 총 3점인데, 목표점 B에서 추제품을 조립한다고 가정한다. 이동경로는 A에서 부품을 들어올려 제품이 있는 B로 이동하여 작업을 한 후 다시 C로 이동하여 부품을 가져와 최종적으로 B에 있는 제품에 조립을 마치는 공정을 수행한다. 두번째 작업은 총 5개의 목표점을 가지고 있다. 이 실험에서 최종 목표점은 C가 되는데 그 이동경로는 A→C→B→C→D→C→E→C와 같이 결정하였다.

마지막 실험은 첫번째 실험과 같은 경로 A→B→C→B를 거치지만 목표점 A와 B사이 장애물이 있어서 그 장애물을 피하면서 작업을 수행해야만 한다.

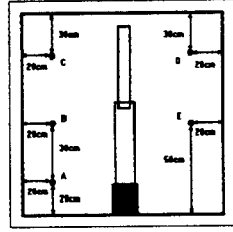
본 실험에서 작업의 형태를 세가지로 결정한 이유는 목표점의 갯수가 다르고 또한 난이도의 차이가 있을 경우 4명의 피실험자들이 생성하는 경로상의 차이점과 더불어 그에따라 소요되는 경로결정 시간을 비교하기 위해서이다. 3가지 작업형태는 인간학습을 이용한 경로생성 실험과 PC상의 시뮬

레이션 프로그램에 동일하게 적용된다.

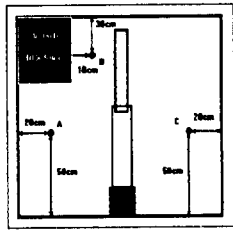
[그림 2]과 [그림 3], 그리고 [그림 4]는 목표점이 각각 다른 세가지 유형의 작업형태에 있어서 작업 목표점의 배치와 실제 치수를 나타낸다.



[그림 2] 목표점 3개



[그림 3] 목표점 5개



[그림 4] 목표점 3개, 장애물

4.4 Teach Pendant 방식을 응용한 시뮬레이션

현재 사용하고 있는 티치 펜던트에 의한 교시과정과 프로그래밍 과정을 PC상에서 실행하기 위하여 Turbo C language와 Turbo C++ language를 사용하여 수행작업의 형태에 따라 로봇트-팔 운영 환경을 그래픽화하였다.

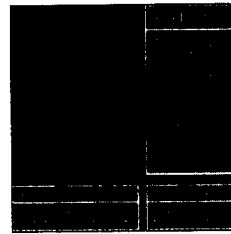
화면의 디스플레이(display)에서 우선 화면 왼쪽 상단에는 로봇트-팔의 본체 움직임을 알 수 있는 로봇트-팔 모형과 작업장 그리고 목표점을 나타냈으며 키보드 조작에 대한 안내화면이 화면의 오른쪽 상단에 나타나 있다. 그리고 오른쪽 하단에는 사용자가 현재의 로봇트-팔 위치를 알 수 있는 각도에 대한 변화값이 나타나 있고 본 실험의 측정기준인 총 key input값(각도의 누적)이 1, 2축이 나뉘어져 왼쪽 하단에 표시되었다. 그래픽상에 나타나는 로봇트-팔은 pixel단위이기 때문에 제작되어진 로봇트-팔 모형과 똑같은 비율로 그래픽 화면을 처리하기 위하여 실제 치수와 1:3 비율(실제 거리 1cm=그래픽 화면 3 pixel)로 시뮬레이션용 로봇트-팔을 표시하였다.

티치 펜던트 방식을 이용한 본 컴퓨터 시뮬레이션용 프로그램은 로봇트-팔이 RIGHT/LEFT나 UP/DOWN 키의 입력을 한번 받을 경우 2° 씩 움직이도록 설계되어져 있다. 키보드를 통해 입력되어진 목표점으로 프로그래머가 원하는 횟수만큼의 시뮬레이션을 수행하면서 하나의 작업사이클이 끝날때 입력되어지는 각각의 축들에 대한 키값의 총 누적치를 기록함으로써, 서로 다른 작업자들이 실험을 반복하면서 얻어지는 데이터의 변화값을 알 수 있다. 또한, 화면에 나타나는 로봇트-팔에 대해 기록된 이전의 기록을 삭제하면서 화면을 초기화시키는 Homming(원점복귀) 기능도 첨가하여 프로그래머가 실험을 반복하면서 새로운 작업을 수행할 수 있도록 프로그램을 제작하였다.

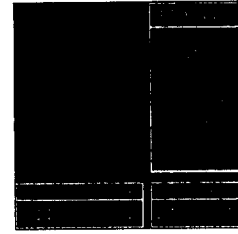
또한, 이동경로 보정을 위한 보정 프로그램을

추가하였는데 이것은 주어진 목표점을 입력하고 실행시키는 과정을 반복하는 동안 잘못 입력되는 불필요한 동작을 제외하면서 필요한 지점으로만 로봇트-팔을 이동시키는 보정작업이 필요하였다. 이 보정 프로그램은 주어진 점 A 와 B에 대하여 입력되어진 키값을 직선으로 fitting함으로써 PC상의 로봇트-팔을 수행하는데 효과적인 경로를 생성하도록 도와준다.

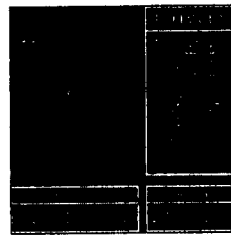
다음의 [그림 5]과 [그림 6] 그리고 [그림 7]은 작업형태에 따라 PC상의 시뮬레이션용 로봇트 운영 환경을 프로그램한 화면이다.



[그림 5] 목표점 3개



[그림 6] 목표점 5개



[그림 7] 목표점 3개, 장애물

4.5 실험방법

4.5.1 Teach Pendant 방식을 응용한 시뮬레이션

선별되어진 4명의 피실험자들에게 목표점 3점과 목표점 5점, 그리고 목표점 3점과 장애물이 있는 세가지 작업유형을 숙지시킨 후 키보드 조작에 익숙해 질 수 있는 충분한 예비실험 과정을 가졌다. 키보드 조작과정은 일정한 횟수의 제한을 가지지 않고 피실험자들 자신이 각 작업에 대하여 로봇트-팔을 움직이는 4가지 키를 여러번 반복하면서 스스로 익숙해 졌다고 생각했을때 경로생성을 위한 실험을 시작하였다. 본 실험의 초기에는 각 작업에 대한 실험횟수에 대하여 제한을 두지 않았지만 10회에서 15회가 되면서 단순 작업을 반복하는데 따르는 피실험자들의 심리적인 요인 등으로 인하여 실험 데이터는 더이상의 학습효과를 거둘 수 없다고 생각하여 모든 실험의 횟수를 15회로 제한하여 실시하였다. 또한, 피실험자가 각 작업에 대한 경로를 결정하는 시간을 조사하기 위해 초시계를 사용하여 시간을 체크하였다.

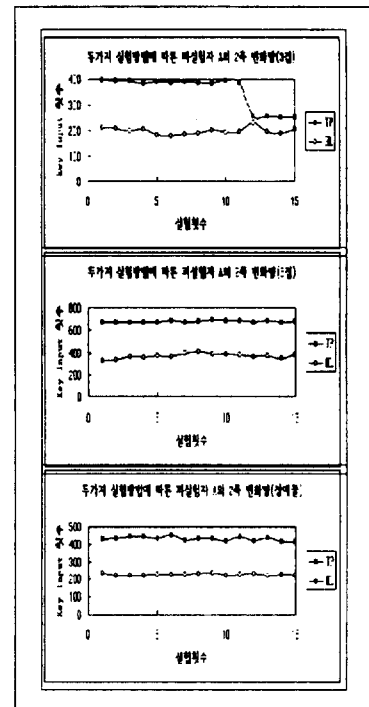
4.5.2 제작된 로봇트-팔 모형의 경로생성 실험

제작된 로봇트-팔 관절에 연결되어져 있는 센서는 매우 정밀한 것이며 로봇트-팔은 피실험자가 움직이는 데로 민감하게 이동경로를 생성하기 때문에 PC상에서 키보드를 조작하여 로봇트-팔을 움직일 때 보다 피실험자는 더 많은 주의가 필요하다. 이 실험 역시 4.1의 실험과 같이 횟수에 제한을 두지 않은 예비실험을 실시하였는데, 예비실험을 종료하

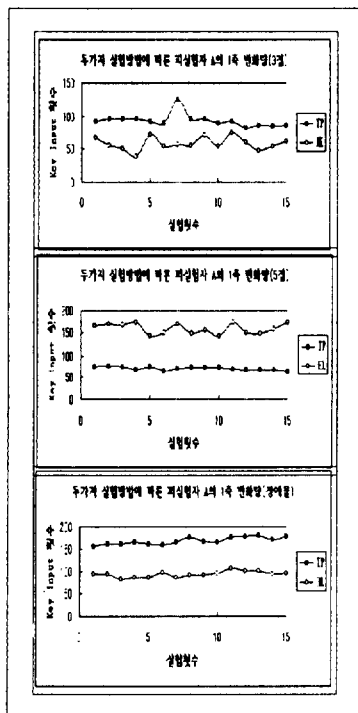
는 시간은 PC로 실행한 키보드 연습보다 더 빨랐다. 인간학습 경로생성 실험에서도 미리 결정한 3가지 작업유형에 따라 4명의 피실험자가 각각 15번의 본실험을 실시하였다. 피실험자는 로봇-팔 모형 끝에 장착된 엔드-이펙터(end-effector)를 직경 4cm의 작업목표점에 위치함으로써 작업위치에 도달하였음을 알 수 있고, 이동경로를 생성할 경우 피실험자 자신에게 가장 적합한 위치에서 로봇-팔 모형을 조작한다.

5. 실험결과

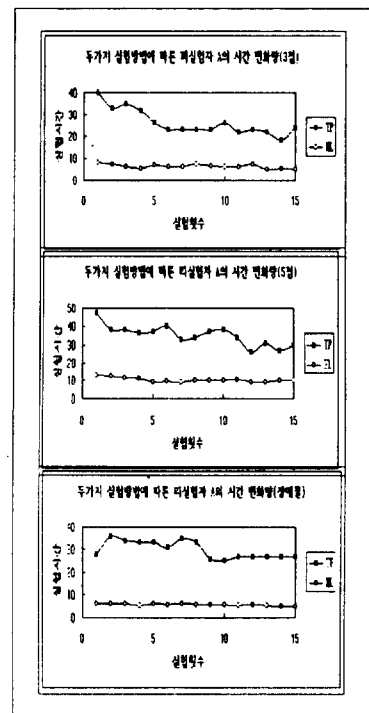
Teach Pendant 방식에 의한 시뮬레이션 실험과 본 논문에서 제시한 인간학습을 이용한 경로설정 실험에 의해 3가지 작업의 1축과 2축 그리고 시간에 대한 변화량을 조사해 보았다. 아래의 [그림 8]에서 [그림 10]까지는 두가지 방식에 따른 각 축들과 시간에 대한 변화량을 비교한 그래프인데, 그래프에 나타난 대로 인간학습을 이용한 경로설정 방법이 각 축들의 움직임이나 시간이 더 적은 범위내에서 이루어졌다. 또한 평균이동 거리의 평균값과 표준편차도 Teach Pendant를 이용한 방법보다 더 우수함이 판명되었다.



[그림 9] 2축에 대한 각 작업별 두가지 실험비교



[그림 8] 1축에 대한 각 작업별 두가지 실험비교



[그림 10] 시간에 대한 각 작업별 두가지 실험비교

6. 결론

본 논문에서 제시한 프로그래밍 방안에서 따라 로봇-팔에 대한 이동경로를 설정해 본 결과 현재 생산현장에서 사용하고 있는 Teach Pendant에 의한 교시방법보다 시간이나 로봇-팔이 움직이는 거리가 더 적게 나타났다. 기업에서 로봇을 생산하

는 경우, 보다 정밀한 센서를 부착하고 기계적인 문제점을 해결하여 생산되는 로봇과 동일한 로봇-팔 모형을 제작한다면 이동경로를 결정하는데 있어서 생산현장에 있는 현장 프로그래머들에게 좀

더 손쉬운 프로그래밍 tool을 제시할 수 있으리라
생각된다. 또한, 정확도에 중점을 둔 그리퍼의 작
업까지 통제할 수 있는 방안에 대한 연구도 필요할
것이다.

참고문헌

- [1] 김태윤, 김홍복, "메카트로닉스 산업용 로봇",
도서출판 생능 1991.
- [2] C. Chang, M.J. Chung, and Z.Bien, "Collison-
free motion planning for two articulated Ro-
bot arms using minimum distance function",
Robotica Volume 8, 1990
- [3] Chiu-Chi Wei, Ali K. Kamrani, Henry Wiebe,
"Animated simulation of the Robot Process
Capability", Computers & I · E Vol. 23, 1992
- [4] Jihong Lee and Zengnam Bien, "Collison-free
trajectory control for multiple Robot based on
neural optimization network", Robotica Volume
8, 1990
- [5] Mikell P.Groover, "AUTOMATION, PRO-
DUCTION SYSTEMS, AND COMPUTER
INTEGRATED MANUFACTURING",
Prentice-Hall 1987.
- [6] R. Bernhardt, R. Dillman, K. Hormann and
K. Tierney, "Integration of Robots into CIM",
Chapman & Hall 1992.
- [7] Tien-Chien Chang, Richard A.Wysk, Hsu-Pin
Wang, "Computer-Aided Manufacturing",
Prentice-Hall 1991.
- [8] "FARA SCARA ROBOT SM2/SS2 OPERA-
TOR GUIDE", 삼성전자 생산기술센터