

직접 교시 작업을 위한 로봇 작업 정보 편집 및 재생산 기법

Techniques of Editing and Reproducing Robot Operation Data for Direct Teaching

김한준¹, 왕영진¹, 김진오^{2,✉}, 백주훈²
Han-Joon Kim¹, Young-Jin Wang¹, Jin-Oh Kim^{2,✉}, and Ju-Hoon Back²

¹ 광운대학교 제어계측공학과 (Department of Control and Instrumentation Engineering, Kwangwoon Univ.)

² 광운대학교 로봇학부 (School of Robotics, Kwangwoon Univ.)

✉ Corresponding author: jokim@kw.ac.kr, Tel: +82-2-940-5158

Manuscript received: 2012.2.6 / Revised: 2012.9.1 / Accepted: 2012.11.22

Study of human-robot Interaction gets more and more attention to expand the robot application for tasks difficult by robot alone. Developed countries are preparing for a new market by introducing the concept of 'Co-Robot' model of human-robot Interaction. Our research of direct teaching is a way to instruct robot's trajectory by human's handling of its end device. This method is more intuitive than other existing methods. The benefit of this approach includes easy and fast teaching even by non-professional workers. And it can enhance utilization of robots in small and medium-sized enterprises for small quantity batch production. In this study, we developed the algorithms for creating accurate trajectory from repeated inaccurate direct teaching and GUI for the direct teaching. We also propose the basic framework for direct teaching.

Key Words: Human-robot Interaction (인간-로봇 상호작용), Direct Teaching (직접교시), Robot Operation Data (로봇 작업 정보), Robot Trajectory (로봇 궤적)

1. 서론

로봇의 개발 및 보급을 넘어 '인간과 로봇의 공존' 차원으로 로봇산업의 패러다임 전환이 일어나는 현 시점에서 인간-로봇 상호작용의 연구는 매우 중요한 의미를 갖는다. 미국, EU 등 선진국들은 인간-로봇 간의 협업 모델인 'Co-Robot' 개념을 도입하여 향후 로봇 신시장 선점에 대비하고 있다.¹ 이러한 인간-로봇 상호작용 기술의 대표적인 예로 로봇의 말단 장치를 직접 손으로 잡고 로봇의 이동 경로를 지시하여 반복 작업을 수행하게 해주는 직접교시가 있다. 로봇을 교시하는 방법에는 프로그래밍, 펜던트 등의 보조장비를 활용하는 방법, 센서를 통한 방법, CAD 파일을 이용한 OLP 기법 등이 있는데, 프로그래밍을

사용하는 방법은 사용자에게 직관적이지 못하므로 별도의 전문인력을 요구하고, 센서를 이용한 기법은 활용분야가 제한적이라는 단점이 있다. CAD를 활용한 기법은 정밀하고 빠르지만, 사전에 숙련된 작업자의 디자인 작업이 필요하며, 궤적을 생성해주는 전용의 소프트웨어가 필요하다.² 위와 같은 단점을 보완할 수 있는 직접교시는 직관적으로 로봇을 활용할 수 있기 때문에 로봇 전문인력이 필요하지 않고, 다품종 소량 생산을 하는 중소기업의 로봇 활용을 촉진시키며, 열악한 작업 환경에서 반복 작업을 하는 작업자들의 작업 환경개선과 생산 효율성 증대에 일조할 것으로 기대된다.³ 직접교시에는 힘/토크 센서 등을 사용하는 접촉식과 음성, 영상 신호 인식을 통한 비접촉식으로 분류된다. 비접촉식 교시 방법은 사용자

의 안전성을 보장할 수는 있지만, 환경적인 영향에 따라 정밀도가 떨어진다는 단점이 있고, 접촉식 교시 방법은 직관적이고 경로 생성 시간이 짧다는 장점이 있으나, 정밀도의 한계와 안전사고와 같은 단점이 있다.⁴

본 연구에서는 로봇 전문인력이 없는 중소기업에서의 저정밀 단순작업을 효율적으로 하기 위한 직접교시용 프로그램의 기본적인 프레임워크를 제시한다. 접촉식 직접교시의 정밀도를 보완하고 교시 데이터를 효율적으로 사용하기 위하여 궤적의 생성 및 저장/편집 알고리즘 기능을 구현하였다. 본 논문에서는 궤적 데이터를 중점적으로 다뤘으나, 접촉식 교시에서는 작업 궤적의 교시뿐만 아니라 작업자의 의도 감지 및 환경과의 접촉시 반력 및 충돌 등의 판별을 위한 작업력의 교시 또한 매우 중요하다.⁵ 개발된 프로그램에서는 경로 데이터뿐만 아니라 작업력의 데이터도 처리하도록 구현되어 있으나, 1 차적으로 궤적 생성과 편집기능 개발을 목표로 작업력은 기본 값을 입력하여 지시하도록 되어 있다. 작업력의 처리에 대해서는 향후 연구에서 추가로 다루고자 한다.

2. 관련 연구

2.1 직접교시 관련 연구

로봇이 수행해야 할 작업을 인간이 직접 교시하는 방식^{9,10}은 작업 경로 생성을 위한 로봇 프로그래밍이 필수적이지 않고, 손쉽고 직관적인 교시 방법이기 때문에 다양한 종류의 제품을 다루는 중소기업의 경우 그 활용도가 매우 높기에 세계 여러 곳에서 직접교시에 관련된 연구가 진행되고 있다.

독일의 프라운 호프 IPA 를 중심으로 유럽에서 진행 중인 중소기업용 산업용 로봇 개발 프로젝트인 유럽의 SME(Small and Medium Enterprise) Robot 에서는 음성인식, 영상인식, 힘제어 등의 다양한 방식을 활용해 직관적인 교시방법을 제안한다.^{11,12} 그 중에 BMW 의 자동차 생산공정에 투입된 파워메이트(powermate robot)의 경우, 로봇을 이용하여 사용자가 손쉽게 무거운 작업물을 다룰 수 있게 돕는 인간-로봇 협업 시스템을 제공한다.¹³ 또 독일의 DLR 연구소에서는 자체적으로 DLR 이 보유한 세계 최고수준의 힘제어 기술이 적용되어 부드럽고 안전한 접촉작업이 가능한 7 자유도 로봇팔

LWR 을 개발하여 이것을 이용해 직접교시 및 제현기술을 구현하였다.¹⁴ 그리고 미국의 Texas Austin 대학 RRG 에서는 HAT(Human Augmentation Technology)를 통해 사람이 로봇의 엔드이펙터를 잡고 직접 교시할 수 있는 기술을 개발하였다.¹⁵

앞서 살펴본 바와 같이 다양한 방법으로 사용자에게 직관성을 제시해주는 몇몇 연구가 수행되고 있지만, 직접교시의 경우 사용자가 로봇을 직접 교시하는 과정에서 위치 오차 및 불확실성이 발생하고, 로봇에게 사용자의 정확한 의도를 전달하기 어렵다는 단점이 있다.⁷

2.2 산업용 로봇 프로그램

기존 산업용 로봇에도 GUI 방식의 작업경로 학습 프로그램들이 있다. ABB 의 Robotstudio 와 KUKA 의 CADROB 이 대표적인 예이다. Robotstudio 는 로봇제작사인 ABB 가 제공하는 소프트웨어로 하드웨어에 대한 호환성이 높아 가장 널리 쓰이고 있다.²

CADROB 은 KUKAKorea 에서 디버링의 자동화를 위해 개발한 프로그램으로 CAD 정보를 읽어와 복잡한 궤적도 빠르고 정밀하게 작업할 수 있다.⁶ 이들 프로그램의 주 기능은 원거리에서 로봇의 현 상태 모니터링, 자동경로생성 및 최적화 충돌회피, 경로 데이터 표시, 작업 시뮬레이션 등이다. 하지만 이들 프로그램은 자사의 로봇에 맞춘 CAD 기반의 프로그램으로 CAD 작업을 요구하지 않는 저정밀도의 단순한 작업에는 적합하지 않다. 본 연구에서 개발하는 프로그램은 위와 같은 기능을 단순한 작업에서도 활용할 수 있도록 하면서 교시궤적 데이터의 편집과 관리를 위한 직접교시 로봇용 프로그램 개발을 목표로 한다.

2.3 궤적 수정 알고리즘

직접교시는 작업자의 교시과정에서 발생하는 위치 오차와 불확실성으로 인해 의도를 파악하기 위한 신호처리와 실시간 보상이 매우 중요하다. 한 가지 방법으로 오차 및 위치의 불확실성을 보상하기 위해 Douglas-Peucker algorithm (DPA)를 적용한 연구가 있다.^{7,8} DPA 는 교시 경로의 시점과 중점을 연결하고, 시점과 중점을 제외한 교시점 중에서 시점과 중점을 연결한 선분 상에서 수직으로 거리가 최대인 점을 찾아 기준 정밀도보다 거리가 작은 점이 나올 때 까지 점을 추가하여 교시데이터를 압축하여 불필요한 데이터를 줄이는 방법이다.⁷ 본

연구에서는 계산의 부하를 줄이면서 DPA 보다 다양한 기능을 갖추기 위해 단순한 기본 도형과 평면을 기준으로 궤적을 근사화하여 오차를 줄이는 알고리즘을 제안하였다.

3. 작업정보 편집 및 재생산 기술

작업정보의 효과적인 생산, 재생산을 위해서는 몇 가지 요소들이 필요하다. 먼저 작업 편집 및 재생산을 위한 작업 데이터를 생성할 수 있어야 하며, 편집/불러오기/저장이 가능해야 한다. 다음으로 작업자가 작업정보를 판단하고 편집할 수 있으며, 결과를 확인할 수 있어야 한다.

3.1 작업 정보 편집

3.1.1 순기구학-역기구학

산업용 로봇의 작업 정보는 로봇 관절 정보 및 센서 계측 정보로 구성된다. 작업정보를 직관적으로 편집/관리하기 위해서는 작업공간에서 로봇의 말단장치의 궤적을 지속적으로 인지하며 작업하는 것이 필요하다. 따라서 로봇 정보를 작업공간에서의 각종 정보로 변환시키는 작업이 필요하며 또한 작업공간 상에서 편집한 내용을 사용하여 로봇 궤적을 생성해내는 것이 필요하다. 이 과정에서 순기구학-역기구학이 사용된다. 역기구학 해는 Newton-Raphson 기법을 사용하며 이는 아래 식과 같다. 이때 T 는 동차변환행렬이고 q 는 관절 벡터이며 테일러 전개를 사용한다.

$${}^0T_n(q^*) = {}^0T_n(q + \delta q) \approx {}^0T_n(q) + \sum_{i=1}^n \frac{\partial {}^0T_n}{\partial q_i} \delta q_i \quad (1)$$

3.1.2 작업정보의 편집 기술

이미 얻어진 로봇의 궤적을 이동, 회전, 축소/확대, 평면으로의 투영, 직선화, 원호화 등의 기능으로 궤적을 편집 하여 새로운 궤적을 생성하는 요소로 사용할 수 있다. 이는 직접교시의 특성상 사람의 교시로 인한 부정확성에서 발생하는 오차를 줄이는 데 효과적으로 사용될 수 있다. 복잡한 형상의 고정밀도의 작업을 편집하는 것은 시간적 소모가 크고 무리가 있겠지만, 단순한 형상의 저정밀도 작업에서는 직접 교시한 궤적을 정확한 평면이나 원호, 직선으로 사용자의 본래 의도에 맞게 편집하는 기술은 매우 유용하리라 생각된다.

3.2 편집 알고리즘

3.2.1 기본 알고리즘

기본 알고리즘으로 Cut, Translation, Rotation, Scaling 기능이 있다. Cut(Fig. 1(b))은 궤적 파일에서 일부 구간을 선택하여 저장하는 기능이다. Translation, Rotation, Scaling 기능은 Homogeneous transform 을 사용하여 기존의 궤적에 적용하여 궤적을 공간좌표 상에서 이동(Fig. 1(c)), 회전(Fig. 1(d)), K 배(Fig. 1(e)) 할 수 있는 기능이다.

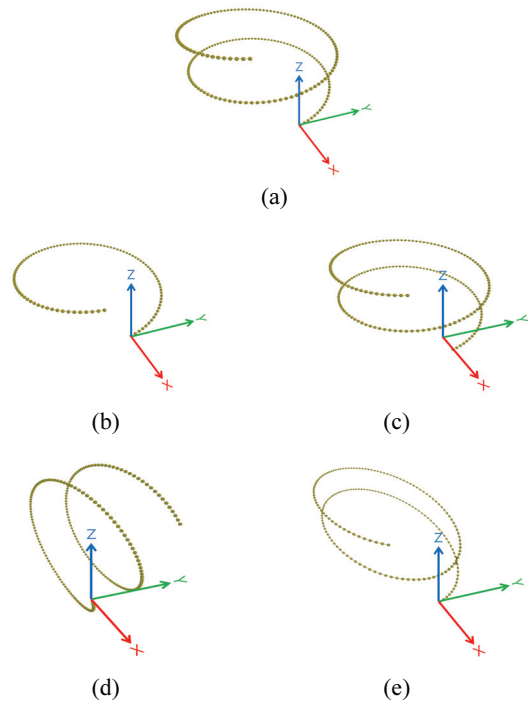


Fig. 1 (a) Reference (b) Cut (c) Translation (d) Rotation (e) Scaling

3.2.2 선택부분 직선화

궤적 위의 두 점을 선택하여, 두 점을 잇는 원하는 직선을 정의하고, 직선화할 구간 안에서 궤적 위의 점과 앞서 정의한 직선 사이의 거리를 구하여 점을 정의한 직선 위로 이동시킨다. 궤적이 왕복운동과 같은 전/후로 움직이는 것을 고려하여 편집 후에도 궤적의 운동이 반영되도록 정의된 직선위로 내적을 취하여 궤적 위의 점을 이동시키는 방법을 사용했다. 그 과정은 아래와 같다. Fig. 2 는 위의 방법을 그림으로 표현한 것이다.

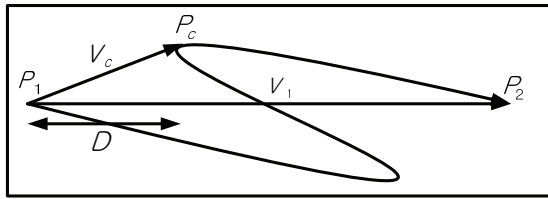


Fig. 2 Linearization

1) 궤적 위의 두 점 P_1 (시작점), P_2 (끝 점)를 선택하여 두 점을 잇는 직선 $V_1 = P_2 - P_1$ 을 정의한다.

2) 현재 궤적 위의 점을 P_c 라 하고, P_1 (시작점)에서 궤적 위의 점 P_c 까지의 직선을 $V_c = P_c - P_1$ 라고 정의한다.

3) 이 때 P_c 와 V_1 사이의 거리는 V_c 와 V_1 의 단위 벡터 \hat{V}_1 를 내적하여 $D = V_c \cdot \hat{V}_1$ 라고 계산할 수 있다.

4) 거리 D 를 단위 벡터 \hat{V}_1 에 곱한 벡터 $V_t = D\hat{V}_1$ 를 사용하여 변환 행렬을 구성한다.

5) 변환행렬을 원래 궤적에 곱하여 편집된 궤적을 계산하여 위의 과정을 구간 내 모든 점에 반복한다.

Fig. 3은 $P_1(483.1, 0.0, 234.8)$, $P_2(283.2, 13.9, 263.0)$ 를 선택하여 선택구간을 직선화 한 것이다.

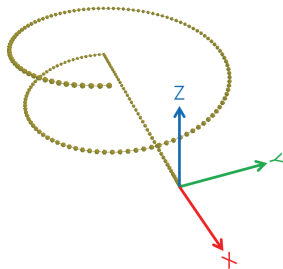


Fig. 3 Linearization

3.2.3 XY/YZ/XZ 평면으로의 정사영

궤적을 XY, YZ, XZ 평면에 투영하기 위한 변환 행렬은 각각 순서대로 아래 식과 같이 쓸 수 있다.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

그림 Fig. 4는 XY 평면으로 투영한 것이다.

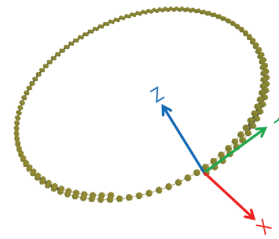


Fig. 4 XY Projection

3.2.4 사용자가 선택한 평면으로의 정사영

사용자가 평면을 선택하여 정의하기 위해서 3 점을 선택하고, 선택한 3 점을 지나는 평면의 방정식을 유도하여 궤적 위의 점과 정의한 평면 사이의 거리를 구한다. 이 때 궤적이 평면을 기준으로 나뉘게 되어 이동시킬 방향이 2 가지가 나오는 문제가 생긴다. 이것은 거리 값에 절대 값을 취하여 해결하였다. 앞의 과정을 거쳐 구한 거리를 사용하여 궤적 위의 점을 평면으로 이동시킨다. 그 과정은 아래와 같다. Fig. 5는 위의 방법을 그림으로 표현한 것이다.

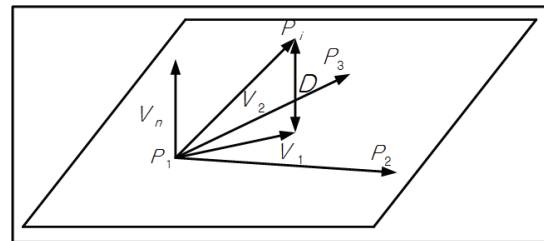


Fig. 5 Projection to Plane

1) 궤적 위의 3 점 (P_1, P_2, P_3)를 선택하고 평면을 평면의 방정식 $L = aX + bY + cZ + d$ 에 따라 정의한다.

2) 평면의 방정식의 법선 벡터 V_n 을 하기 위해 직선 $V_1 = P_2 - P_1$ (7), $V_2 = P_3 - P_1$ 를 정의하고 두 직선을 외적 하여 평면의 법선 벡터를 구한다. V_n 의 각 성분은 평면의 방정식의 계수와 같다. $V_n = (a, b, c)$

3) 남은 계수는 다음과 같이 한 점을 대입하여 구할 수 있다. $d = -ax_1 - by_1 - cz_1$

* $P_1 = (x_1, y_1, z_1)$

4) 궤적 위의 점을 $P_i = [x_i, y_i, z_i]^T$ 라 하고, 궤

적 위의 점에서 평면까지의 거리 D 를 다음과 같이 구한다.

$$D = -\frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}} \quad (3)$$

- 5) 평면 위의 점은 $P_i - DV_n$ 라 계산할 수 있다.
- 6) 거리 D 를 평면의 법선 벡터 V_n 에 곱한 벡터 $V_i = DV_n$ 를 사용하여 변환 행렬을 구성한다.
- 7) 변환행렬을 원래 궤적에 곱하여 편집된 궤적을 계산하여 위의 과정을 구간 내 모든 점에 반복한다.

Fig. 6 은 세 점 $P_1(483.1, 0.0, 234.8)$, $P_2(392.6, 95.7, 248.8)$, $P_3(283.2, 13.9, 263.0)$ 을 선택하여 정의된 평면으로 궤적을 투영한 것이다. 투영되었는지 확인할 수 있도록, 정의된 평면을 화면에 표시하였다.

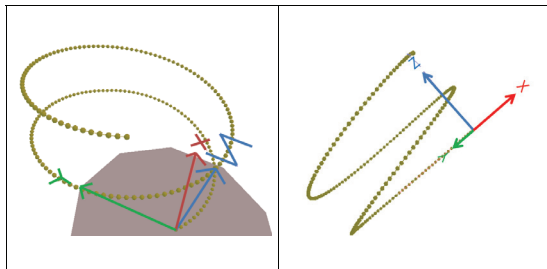


Fig. 6 Projection to Plane

3.2.5 최소자승법으로 정의한 평면으로의 정사영

궤적 내의 3 점으로 평면을 정의할 수 없을 때 선택 구간에서 가장 근접한 평면을 정의하기 위한 최소자승법을 적용한 투영 기능이다. 최소자승법은 본래 2 차원에서 임의의 점들의 나열에 가장 근접한 직선을 찾는 기법이지만 이를 3 차원에 적용하기 위해 본래 2 차원에서의 최소자승법 공식을 3 차원으로 확장하였다. 그 과정은 아래와 같다.

- 1) 최소자승법을 적용할 점의 개수 i 를 선택한다.
- 2) 평면의 방정식은 $z = ax + by + c$, 각 점과 근접한 평면 사이의 오차를 ϵ 라 한다.
- 3) 2)의 정의를 갖고 오차를 포함한 평면의 방정식을 $z_i = ax_i + by_i + c + \epsilon_i$ 라 한다.

4) 3)의 식에서 각 점과 평면 사이의 오차를 $\epsilon_i = z_i - ax_i - by_i - c$ 라 한다.

5) 최소자승법을 위한 오차의 제곱의 합인 $S = \sum_{i=1}^n \epsilon_i^2$ (14)를 고려한다.

6) 5)에서 구한 오차 제곱의 합이 최소가 되는 평면을 구하기 위해 오차를 포함한 평면의 방정식을 행렬식으로 $Z = AX + E$ 로 표현한다. 여기서 (Z, A, X, E) 는 다음과 같다.

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}, A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}, X = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, E = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad (4)$$

7) 평면의 방정식의 상수 a, b, c 를 다음과 같이 구한다. $X = (A^T A)^{-1} A^T Z$

8) 7)에서 구한 평면의 방정식으로 궤적과의 거리를 구하여 변환행렬을 구한다.

9) 원래 궤적에 변환 행렬을 곱하여 편집된 궤적을 계산하여 구간 내 모든 점에 반복한다.

Fig. 7 는 시작점 $P_1(483.1, 0.0, 234.8)$ 에서부터 P_2 끝점(283.2, 13.9, 263.0)까지의 100 개의 점을 선택하여 정의된 평면으로 궤적을 투영한 것이다. 투영되었는지 확인할 수 있도록, 정의된 평면을 화면에 표시했다.

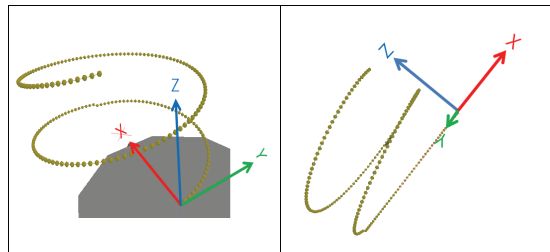


Fig. 7 Projection to Plane using least-square

3.2.6 선택부분 원호화

사용자가 선택한 구간을 원하는 반지름 R 을 갖는 원호로 변환하는 기능이다. 두 점을 지나는 원호를 만들기 위해서 원호가 존재하는 평면을 정의할 필요가 있기 때문에 3 점으로 평면을 정의한다. 여기서 3 점 가운데 중앙의 점은 평면을 생성하는 것에만 사용한다. 첫 점과 끝점을 지나

고 반지름이 R 인 원호를 생성한다. 정의한 평면 위에는 선택한 두 점을 지나는 반지름이 R 인 원호가 아래의 Fig. 8 에서처럼 두 개가 존재 한다. 우리는 직접 교시한 궤적에 가까운 하나의 원을 선택하기 위해 두 원호의 중심을 구하고 현재 궤적과 각각의 생성된 원호와의 거리를 구하여 근접한 원호를 갖는 중심을 선택하도록 하였다. 궤적의 운동이 편집 후에 모두 반영되어 왕복운동과 같은 반복 운동도 그대로 편집 후에 적용되도록 하였다. 그 과정은 아래와 같다. Fig. 8 은 그림으로 표현한 것이다.

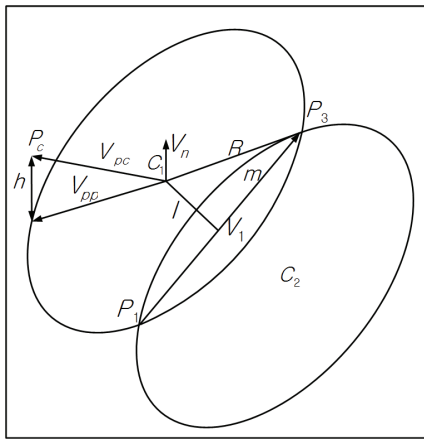


Fig. 8 Algorithm to select the center of the circle

1) 원호를 정의하기 위하여 첫 점 P_1 , 중간 점 P_2 , 끝 점 P_3 , 원하는 반지름 R 을 정의한다. 이렇게 정의하면 P_1 과 P_3 를 지나면서 반지름이 R 인 원은 P_1, P_2, P_3 를 지나는 평면에서 두 개를 정의할 수 있다.

2) 시작점에서 끝점까지의 벡터를 $V_1 = P_3 - P_1$ 이라 정의하고, 벡터의 크기의 반을 $m = 1/2 \times |V_1|$ 라고 정의한다. 정의한 R 과 m 을 이용해 원호의 중심에서 V_1 벡터까지 수직인 선분 $l = \sqrt{R^2 - m^2}$ 을 정의한다.

3) 원호가 그려지는 평면을 정의하기 위해 P_1, P_2, P_3 를 지나는 평면의 법선벡터를 V_n 이라 하고, V_n 의 단위 벡터를 \hat{V}_n , V_1 의 단위 벡터를 \hat{V}_1 이라 할 때 두 벡터의 외적인 $V_i = \hat{V}_n \times \hat{V}_1$ (19)를 구한다. V_i 의 단위벡터 \hat{V}_i 는 l 의 방향 벡터이다.

4) 이제까지 구한 값으로 앞에서 정의 한 두 개의 중심을 $C = (P_1 + m \times \hat{V}_i) + (l \times \pm \hat{V}_i)$ 로 표현할 수 있다.

5) 현재 궤적 값을 P_c 라 할 때 중심에서 현재의 궤적 값까지의 거리를 $V_{pc} = P_c - C$ 라 할 수 있고, 그 수직거리는 $h = V_{pc} \cdot \hat{V}_n$, 수평거리는 $V_{pp} = V_{pc} - h \times \hat{V}_n$ 가 된다.

6) 이제까지 구한 값으로 $V_i = C + \hat{V}_{pp} \times R$ 를 구하고 변환행렬을 구성한다.

7) 두 개의 원호 중에 우리는 본래의 궤적에 근사한 원호를 찾아야 하기 때문에, 궤적과 변환된 궤적의 차이를 $\delta P_i = (\delta x_i^2 + \delta y_i^2 + \delta z_i^2)$ 라 하고, δP_i 의 제곱의 합인 식(5)가 더 작은 원호의 중심을 선택한다.

$$S = \sum_{i=P_1}^{i=P_3} (\delta x_i^2 + \delta y_i^2 + \delta z_i^2) \quad (5)$$

8) 선택한 원호로 위의 과정을 구간 내 모든 점에 반복한다.

Fig. 9 는 $P_1(483.1, 0.0, 234.8)$, $P_2(392.6, 95.7, 248.8)$, $P_3(283.2, 13.9, 263.0)$ 으로 선택하여 알고리즘을 적용하여 궤적을 편집한 것을 나타낸다.

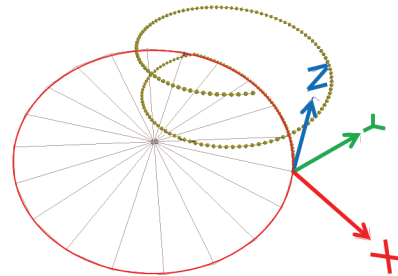


Fig. 9 Circular arcs

편집 기능으로 말단의 궤적을 수정하였으며, 역기구학을 풀이하여 새로운 관절 변위를 얻었다. 본문에서 제작한 GUI 프로그램은 역기구학의 적용을 선택할 수 있는데, 역기구학을 적용하면, 궤적의 편집으로 궤적이 로봇의 작업 반경을 벗어나지 않도록 프로그램에서 경고 및 제제를 할 수 있도록 되어있다. 즉 역기구학을 풀어서 궤적의 유효성을 검사할 수 있다. 역기구학을 적용하지 않으면, 로봇의 작업 반경의 제한을 받지 않고 궤적의 편집이 가능하다. 위에 언급한 궤적의 변환들은 작업 경로에 대하여 선정된 기준위치에 원점이 있는 좌표계에 대한 것임을 주의한다.

4. 결과 및 적용

4.1 Simulation

4.1.1 Model

본 연구는 직접교시 로봇으로 제안된 6 축 로봇을 모델로 진행하였다. 로봇은 6 개의 회전모듈로 구성되어 있고 각 관절은 위치제어를 수행하며, 로봇의 말단에는 힘/토크 센서가 부착되어 교시력, 작업력을 측정한다. 하지만 본 연구에서 개발한 프로그램에서는 교시력과 작업력의 반영은 진행단계에 있다. 로봇의 D-H parameter 는 Table 1 에 표시되어있다. Fig. 10 은 로봇과 프로그램이 연동된 초기 상태를 보여준다.

Table 1 D-H Parameter of Robot

	θ (°)	d (m)	a (m)	α (°)
1	θ_1	0.385	0.0	90.0
2	θ_2	0.0	0.36	0.0
3	θ_3	0.0	0.0	-90.0
4	θ_4	0.35	0.0	90.0
5	θ_5	0.0	0.0	0.0
6	θ_6	0.287	0.0	-90.0

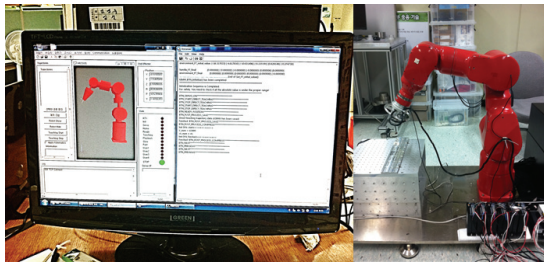


Fig. 10 HRI(Human Robot Interaction) Program initial state

작업 정보를 관리한다. 즉 CRobotDataStorage 는 하나의 작업에 해당하는 모든 데이터를 갖는다. 최상위 계층에는 CRobotDataStorageList 라는 개체가 있어 여러 작업을 리스트 형태로 관리한다. Fig. 11 은 개발된 프로그램의 구조이다. 로봇에 연결된 서버와 TCP/IP 로 통신하며 로봇의 각축의 데이터를 받아서 로봇의 움직임을 화면에서 보여준다. 궤적의 교시를 시작하면 로봇의 말단을 따라 궤적이 화면에 표시되면서 동시에 생성된다. 생성된 궤적을 저장하여 불러냄으로 편집 및 새로운 궤적의 생성이 가능하다. Fig. 12 는 본 프로그램의 기본화면이고, Fig. 13 은 궤적 편집창이다. 화면의 기준 좌표는 로봇의 첫 번째 축의 바닥 면을 원점으로 하고 있다.

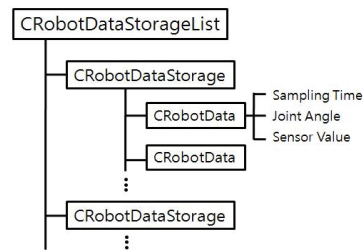


Fig. 11 Data structure and Classes

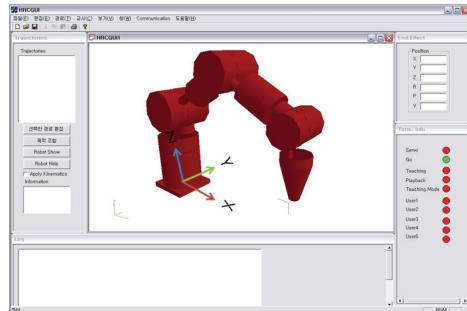


Fig. 12 HRI Program

4.1.2 Software

소프트웨어 개발환경은 Microsoft Visual Studio, 프로그래밍 언어는 C++을 사용하였다. 그래픽 라이브러리로 OpenGL 과 Chart Director 를 사용하였고 기구학과 및 관련된 연산을 위해 ROBOOP,¹⁶ NEWMAT¹⁷ 라이브러리를 사용하였다.

본 소프트웨어는 데이터 구조를 3 개의 계층으로 설계하였다. 최하위 계층은 샘플링 시간과 그에 해당하는 관절 정보 및 작업 정보를 저장하고 있고, 그 상위 계층인 RobotDataStorage 에서 작업의 전 영역에 해당하는 샘플링 시간과 관절 정보 및

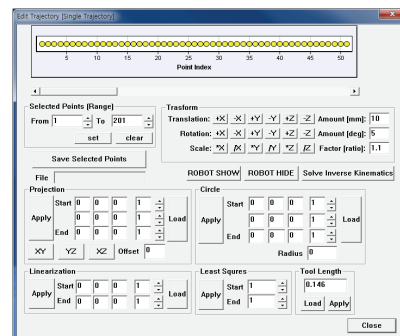


Fig. 13 The Interface for trajectory editing

4.2 편집도구를 활용한 재생산

시뮬레이션의 진행은 다음과 같다. 원형 궤적과 직사각형의 궤적을 직접교시 하여 얻은 교시 데이터 중 필요한 부분만을 잘라낸 후, 평면화 한 뒤에 원형궤적은 원호화를 직사각형 궤적은 직선화를 거쳐서 작업대상과 비교하여 오차를 측정한다. 궤적의 편집은 생성된 궤적 위의 필요한 점을 선택하여 각 점의 좌표값을 불러와 3 장의 알고리즘을 적용하는 방식이다. 기준 형상은 Fig. 14 와 같다.



Width: 15 mm Height: 50 mm Radius: 20 mm

Fig. 14 Target object

직접교시로 생성된 raw data 는 Fig. 15 와 같다.

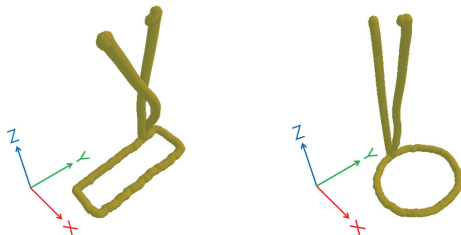


Fig. 15 Trajectory raw data

아래 Fig. 16 은 기본 알고리즘에서 언급한 Cut 기능을 사용하여 필요한 부분만 잘라낸 뒤, 최소자승법을 이용한 평면화를 거쳐, 직사각형 궤적은 직선화 알고리즘을 적용하고, 원형 궤적은 원호화 알고리즘을 적용하여 편집한 궤적이다.

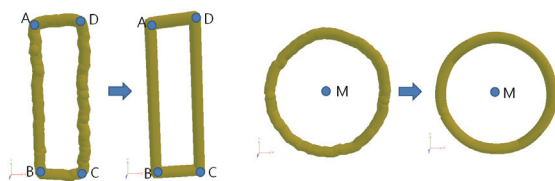


Fig. 16 Edit trajectory data

편집 전 후의 수치적 해석을 위해 직사각형 궤적의 꼭지점을 Fig. 16 과 같이 A(377.7, -12.2, 770.1),

B(393.0, -11.0, 770.1), C(391.9, 38.5, 769.7), D(377.3, 37.1, 769.7)로 표기하고 교시궤적과 편집궤적의 각 선분 AB, BC, CD, DA 의 직선과 궤적의 오차의 크기를 구하여 교시궤적과 편집궤적을 비교하였다 (Fig. 17). 원형 궤적은 중심을 M(385.5, 1.5, 769.1)으로 표기하고 중심으로부터 궤적까지의 거리의 값과 비교하여 오차를 구하고 교시궤적과 편집궤적을 비교하였다 (Fig. 18).

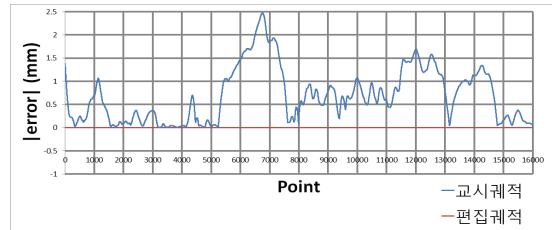


Fig. 17 Comparison graphs (rectangular)

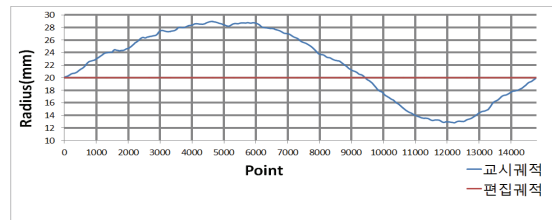


Fig. 18 Comparison graphs (circular)

Fig. 17 에서 보면 직사각형 궤적 교시 시에 오차의 최대값은 2.46 mm 이고, Fig. 18 에서 보면 원형 궤적 교시 시에 오차의 최대값은 8.92 mm 이다. 궤적 편집의 결과 원하는 평면으로의 투영과 직선화 및 원호화가 이루어진 것을 확인 할 수 있다.

이러한 결과로 작업하고자 하는 형상이 명확하다면, 고정밀의 작업이 아닌 정밀도가 낮은 단순 작업에서는 CAD/CAM 데이터나 프로그래밍을 사용하지 않고 직접교시로 간단히 궤적을 생성하고 편집함으로써 작업의 효율을 높일 수 있다는 것을 확인하였다.

5. 결론

본 연구에서는 CAD 정보를 이용하지 않고, 사람의 직접교시를 통하여 얻어진 궤적 정보를 관리하고 이를 편집하여 새로운 작업정보를 생성하는 기술을 개발하였다. 또한 개발된 알고리즘을 6 축 로봇에 적용하여 프로그래밍 지식이 없이 새로운

작업정보를 생성할 수 있다는 것과 고정밀도를 요구하지 않는 경우 직접교시로 생성된 궤적을 사용자의 의도대로 편집할 수 있음을 확인하였다. 이 기술은 고정밀도를 요구하지 않는 단순작업 시 직접교시를 활용하면, 전문인력이 아닌 사용자가 직관적으로 로봇을 움직여 궤적을 교시하고, 교시된 궤적을 쉽고 효율적으로 저장과 편집을 가능하게 해줄 것이라 생각된다. 전문인력이 없는 중소기업에서는 산업용 로봇을 좀 더 쉽게 활용할 수 있도록 도움을 줄 것이다. 본 프로그램은 고정밀도의 작업을 하기에는 한계가 있으며 이 한계를 극복하기 위해 보다 쉬운 작업이 가능하도록 직관적인 UI 구성과 더 자유로운 편집 방법의 개발이 필요하며, 교시력과 작업력의 반영이 필요하다. 또, 궤적의 편집 기능을 사용할 때 순기구학-역기구학 연산을 많이 수행하므로 관절이 많은 로봇 경우에는 처리 속도와 정밀도의 조절이 필요하며 정교한 작업을 위한 고속 알고리즘의 개발이 필요하다.

후 기

본 연구는 광운대학교 2011 년도 교내학술연구비 지원에 의해 연구 되었음.

참고문헌

1. Ministry of Knowledge Economy, "Government-wide, draw the next 10 years the big picture of the robot," 2012.
2. Sul, I. H., "Recent research trends on industrial robot programming," KOSSEN Expert Review, 2012.
3. Park, D. I., Kyung, J. H., and Jung, K. J., "Human-robot cooperation technologies research trends," Machinery and Materials, Vol. 22, No. 4, pp. 76-83, 2010.
4. Park, J. S., Ji, S. H., Nam, K. T., Lee, S. M., Lee, S. J., and Lim, M. T., "Pattern-based Path Planning Algorithm for Human-Robot Cooperation," Proc. of KSPE Autumn Conference, pp. 173-174, 2010.
5. Park, C. H., Kyung, J. H., Park, D. I., Park, K. T., Kim, D. H., and Gweon, D. G., "Direct Teaching Algorithm for a Manipulator in a Constraint Condition using the Teaching Force Shaping Method," Advanced Robotics, Vol. 24, pp. 1365-1384, 2010.
6. KUKA Korea, "Deburring solves the robot," Monthly Tools Journal, pp. 56-57, 2011.
7. Song, M. S., Lee, S. H., Li, C. J., Kim, D. H., and Han, C. S., "Direct Teaching and Playback Algorithm of Robot Manipulator," Proc. of KSPE Spring Conference, pp. 65-66, 2010.
8. Li, C. J., Park, C. H., Kyung, J. H., and Chung, G. J., "Motion control of the direct-teaching robot at teaching point using trajectory planning algorithm," Proc. of KSPE Spring Conference, pp. 311-312, 2011.
9. Asada, H. and Asari, Y., "The Direct Teaching of Tool Manipulation Skills Via the Impedance Identification Skills," Proc. of IEEE International Conference, Vol. 2, pp. 24-29, 1988.
10. Kushida, D., Nakamura, M., Goto, S., and Kyura, N., "Human Direct Teaching of industrial articulated robot arms based on force-free control," Artificial Life and Robotics, Vol. 5, pp. 26-32, 2001.
11. Meyer, C., "Intuitive programming of industrial robots," International Symposium on Robotics, pp. 69-74, 2006.
12. Pires, J. N., "Robot-by-voice: Experiments on commanding an industrial robot using human voice," Industrial Robot, An International Journal, Emerald Group Publishing Limited, Vol. 32, No. 6, pp. 505-511, 2005.
13. Schraft, R. D., "Powermate - A safe and intuitive robot assistant for handling and assembly tasks," Proc. of IEEE Int. Conf. in Robotics & Automation, pp. 4074-4078, 2005.
14. Hirzinger, G., "Cartesian impedance control for torque controlled light-weight robots," Proc. of IEEE Int. Conf. on Robotics & Automation, pp. 657-663, 2002.
15. Pholsiri, C., "Extended generalized impedance control for redundant manipulators," Proc. of 42nd IEEE Int. Conf. on Decision and Control, pp. 1255-1260, 2003.
16. Gourdeau, R., "Object oriented programming for robotic manipulators simulation," IEEE Robotics and Automation Magazine, Vol. 4, No. 3, pp. 21-29, 1997.
17. Davies, R., "New mat11," http://www.robertnz.net/nm_intro.htm