아이크 옵점 자동화를 위한 테스크 레벨 자동 프로그래밍 시스템 개발

Development of a Task Level Automatic Programming System for Arc Welding Automation

박현자, 이범희
서울대학교 전기공학부 (Tel: 02-880-7316, Fax: 02-885-6620, E-mail: phi@robot1.snu.ac.kr)

Abstracts With the progress in process automation, it becomes necessary that a robot should have various sophisticated capabilities. A robot programming language that can give a robot such capabilities without any change in robot architecture. Especially a task level automatic programming system enables a robot able to perform a job intelligently. Therefore anyone who is not an expert on welding or robot programming can easily use it. In this research, basic automatic welding program is combined with workspace information, which makes users do an arc welding job automatically.

Keywords welding automation, task-oriented language, world modeller, path planner, task level automatic programming system

1. 서론
본 연구는 납로 그 필요성이 증가하고 있는 공정 자동화의 일반
인 용접 자동화에 관한 것이다. 용접 자동화가 급속히 진행됨에 따
라, 점점 더 다양하고 복잡한 로봇의 기능이 요구되고 있고, 그에
따라 로봇을 프로그래밍하는 로봇 자동 프로그래밍 시스템에 관한
연구가 활발하게 진행되고 있다. 그중 작업 중심 언어의 개념을
이용한 테스크 레벨 자동 프로그래밍 시스템은, 구체적인 문제는 시
스템 내부에서 해결하고 개별적인 작업만을 지정하는 이상적인 프
로그래밍 방식으로 작업 환경을 지능적으로 관리하는데 필수적이
다고 할 수 있었다.

이 연구의 목적은 로봇에 의해 아크용접 자동화체계를 하는 것으
로, 용접이나 로봇 프로그래밍의 미비한점이 있지만, 아크용
접용 로봇 프로그래밍을 작성할 수 있도록 자동 프로그래밍 시스템을
개발하는 것이다. 그 첫 단계로서 일반언어로 테스크 레벨 언어들에
해당 조사하고, 테스크 레벨 언어의 특성 및 테스크 레벨 아크용 접 자
동 프로그래밍 시스템의 작성 지침 시스템을 구성한다. 다음 단계로
시작 계획 시스템을 이용한 용접 자동 프로그래밍 시스템과 통
합하여 테스크 레벨 자동 프로그래밍 시스템을 완성한다.

2. 작업 중심 언어와 테스크 플레더

2.1 로봇 언어의 분류 및 특성
로봇 언어는 크게 아크로 컴퓨터 단계, 점간 단계, 원시 통도
단계, 구조 프로그래밍 단계, 작업 중심 단계로 나뉜다.
아크로 컴퓨터 단계는 프로그래밍의 단계의 커다란 반도
적도 거리로 가는 작업 중의 방법으로, 테스크로 바꾸는 로봇인
어의 가장 기본적인 단계이며, 로봇의 구조에 따라 결정적이다.

점간 단계(Point-to-point level)는 교시를 기본으로 한 가장 이용
하기 쉬운 방법으로 단순 반복 작업에 유용하다. 조건 분기 기능이
나 센서 정보의 이용이 어렵고, 로봇 라인 프로그래밍 시스템으로
확장할 수 없으며, 모든 위치 정보는 로봇 각각의 각도의 조합으로
남아내지는 관절 좌표 시스템(Joint Coordinate System)으로 표현된
다.
원시 운동 단계(Primitive motion level)는 로봇 자체의 이동을
기술하는 데에 중점을 둔 언어로서, 각 관절을 따로 구동할 수 있
으며 이거는 단계의 한정성 및 소프트웨어 설계하기가 편리할 수
있어 주로 개발의 도구로 보이며 사용법이 비교적 간단하다. 대부분
어셈블리 언어로 설계하여 간단한 단계 전처리와 소프트웨어가
이용되며, 센서의 이용 능력도 현명함으로서 주로 ON/OFF의 이산 태신
정도이다. 또한 별도 처리 프레임의 이용도 보였으나 매우 기
본적인 수준이며, 프레임은 위치 정보가 센서 및 로봇에 공통된 점
에 대해 좌표계면에 따라 나타내지는 각도 좌표 시스템이다.
구조 프로그래밍 단계(Structured programming level)는 다양한
센서의 이용, 통합된 제어 구조, 다양한 데이터 구조에 대한 실제
계산이 가능하다. 또한 상태 변수를 정의하여 사용자의 효과적인 프로그램을
도출, 작업 중심 언어의 정확 모델링 기능을 갖고도 한다.
이들 언어의 특성은, 상대 좌표에 따른 프레임을 정의하며, 외
부 센서에 의한 위치 계산 및 조립 공정의 진행에 따른 문제의 프
로그램의 연속적이고 복잡한 변화를 효과적으로 처리하는 구조적
적 좌표 시스템이 이용된다.
전술한 언어들이 로봇의 동작을 결정적으로 반하여 작업
중심 단계(Task-oriented level)의 언어는 로봇의 동작이 작업
공간 내에 미치는 영향을 기술하자, 로봇보다는 작업 공간 내의
문제의 점에서 명확하게 나타난다. 센서나 좌표계의 이동 등
을 내부적으로 해결하고 개별적인 작업만을 지정하는 이상적인 프
로그램 방식이 이 단계의 언어는, 자신의 작업 환경 모델링 능력과
작업 복합체의 능력, 실험 가능한 로봇 프로그램의 생성 능력이
요구된다. 요구하는 기능은 World Modelling, Task specification,
Collision Free Path Planning, Automatic Grasping, Compliant
22.2 테스크 패들러

테스크 패들러(Task Planning)이란 주어진 테스크를 수행하기 위한 일련의 모션 프리미티브(Motion Primitives)를 생성하는 과정을 말하며, 크게 두 가지 방법이 있다. State-Space Approach는 초기 상태로부터 목표상태까지의 경로를 찾고 재정의하는 방법을 말하고, Action-Ordering Approach는 최초의 High-Level Plan(Skeleton Plan)을 추계한 뒤 목표를 완수하기 위한 세부계획을 생성해내는 방법을 말한다. 이 두 방법 중에 Action-Ordering Approach가 State-Space Approach에 비해 복잡한 로봇 맵데이터에 적합하므로 Action-Ordering Approach를 이용하기로 한다.

사용자가 Weld Object와 같은 테스크 레벨 명령을 내리면 테스크 플래너는 해당 모델에 따라 적절한 경로 계획이 이루어진다. 이동 경로를 계획하기 위해서는 모델의 경로 계획이 이루어진다. 사용자의 작업 경로에 대한 정보를 얻기 위해 모델의 위치, 작업내의 위치, 움직임 대상물의 위치, 장애물의 위치 등을 활용한다. 움직임 대상물의 위치를 이용해서 움직임 진행 방향을 계산한다.

로봇 맵데이터는 움직임 대상물까지 이동해야 가먼, 사용자로부터 입력받은 움직임 대상물의 좌표와 두께, 이름의 종류, 종류의 종류, 움직임 시작 및 움직임 대상물에 대한 정보를 가지고, 데이터베이스를 이용하여 움직임 진행 방향, 전류, 움직임 방향, 토픽의 속도 등 움직임 조건을 생성한다. 생성된 움직임 조건을 이용하여 움직임 작업을 수행하는 움직임 프로그램을 생성해내고 움직임 작업을 수행한다.

3. 작업 계획 시스템의 설계

3.1 환경 모델러(World Modeller)

환경 모델러는 로봇이 어떤 작업을 하는 환경 내의 센서, 주변 기기 및 각종 플랫폼들에 공동으로 사용하는 하나의 표준을 두어 이를 상호간의 통신을 할 수 있게 하는 작업으로 일반적으로 작업이 진행될 때마다 이들 대상물에 관한 정보는 계속적으로 변경하게 되는데, 이러한 변경이 로봇 시스템에 전달되지 않으면, 작업이 정확하지 못하게 된다. 이러한 문제가 표준을 이해하는 복잡한 시스템에서 발생하게 되며, 본 시스템에서는 작업의 진행 상황을 대로적으로 시스템 리더에게 전달하여 환경에 따른 작업 공간의 정보를 계속적으로 보완함으로써 이 문제를 해결한다.

작업 공간 내의 물체는 다양한 형태를 가지며, 이를 예제적으로 표현하는 규칙이 마련되어야 하는데, 본 시스템에서는 작업 공간 내의 모든 물체를 적절하게 표현하여 나타내기로 하였다. 사용자로부터 구현된 작업내의 기준점과, 가로, 세로, 높이를 나타내는 세 점을 입력받아 데이터로 사용한다.

이러한 데이터 외에 로봇 손 꼭대의 위치, 바이너리의 좌표, 유형 대상물의 움직임 위치, 장애물의 개수, 움직임 대상물에 대한 정보를 입력받아 사용한다.

3.2 경로 계획기(Path Planner)

작업 공간의 상태로부터 작업 위치, 후속 위치 및 충돌이 일어나지 않도록 가장 간단한 경로를 선택한다. 충돌 회피로 일어난 알리카운트는 작업 중심의 영역을 보다 흥미로운 연구 분야의 하나로, 회전 관상의 접근는 로봇의 정확한 위치에서 근사치는 방법과, 장애물 경계선의 정확한 수신을 이용하는 접근 방법으로 얻어진다. 현재의 충돌 회피 방법은 주로 간단한 형태로 Z축 방향으로의 경로 선택만이 허용되며, 여기에서 사용된 가정은 다.

![그림 3.1 로봇 손 꼭대의 근사 과정](Fig.3.1 Approximation of robot end effector)

![그림 3.2 장애물 근사 과정](Fig. 3.2 Approximation of Obstacles)
그림 3.1과 그림 3.2에 나타나 있는 것처럼 손에 잡고 있는 물체의 크기에 따라 물체의 방향 변환에 따른 영향이 무시될 만큼 충분히 큰 반지름을 갖는 구조로, 로봇의 손끝을 근사시키고, 손끝을 정교로 다룰 수 있도록 모든 장애물의 크기를 구의 반지름만큼 증가시켜 조정한다.

3.2.2 수평면 확인
이동 경로의 시작점과 끝점에 의해 이루어진 직선과 직업 공간내의 모든 장애물을 x-y평면으로 두명시키 2차원에서 이들의 교점을 구함으로써 충돌 가능성이 없는 물체들을 찾아낸다.

로봇에 앞쪽은 역기구학(Inverse Kinematics)을 통해 그 결과를 직업 공간 모델과 비교하며, 도달 불가능한 지점이 프로그래밍되는 것을 방지한다.

이러한 기준은 정확한 모델의 정보를 통해 구현되며, 결과를 드리나는 시스템의 기능과는 무관한 부분이므로 사용자가 항상 논리적으로 합당한 위치 정보를 입력한다는 조건이 만족되면 불필요한 부분이 될 수도 있다. 그러나 안전한 프로그램이 작성되었을 경우, 첫 수준의 체계 시스템과 연결하여 이용하기 위해서는 반드시 필요한 기능이다. 이 모델은 작업 공간 내에서의 각 물체들의 위치를 관리하는 부분이다.

4. 자동 프로그래밍 시스템의 설계

4.1 Task Level 자동 프로그래밍 시스템의 구조와 기능
본 시스템은 운동 단계의 인터페이스가 갖는 단점을 보완하여 최소한의 노력으로 원하는 프로그램을 작성할 수 있게 하는 보조작업 도구로서 다음과 같은 기능을 구현한다.

1) 환경 모델링을 위한 자료의 위치 및 방향 정보 오자는 없는다.
2) 로봇의 동작에 의한 오차가 없이 실제 환경과 모델링된 환경은 항상 같다.

이 시스템의 구조는 다음과 같다. 사용자가 사용자 인터페이스를 통해 원하는 작업 또는 작업 환경에 대한 데이터를 입력하면, 정교 계획기, 작업 공간 관리자는 환경 모델의 정보를 바탕으로 물체 포지트를 통해 서로 정보를 교환하여, 매크로 레더 구동을 위한 프로그램 모델을 만들어 낸다.

이 과정에서 실제의 로봇 프로그램을 출력할 수도 있으며, 본 시스템은 여러 종류의 로봇 언어를 생성하기 위해 pseudo code의 개념을 이용하여 특정 로봇 언어와 관련한 부분은 프로그램 머임이라는 공통의 문법을 이용하여 실제의 로봇 프로그램을 작성할 수 있는 제한 정보를 생성한다. 이 장려는 프로그램 합성기에 의해 원하는 로봇 언어로 변환된다.

4.2 사용자 인터페이스
사용자 인터페이스는 시스템과 사용자간의 의사 전달을 담당하는 부분으로 사용자가 프로그래밍의 경험이 없는 초보자임을 고려할 때, 그 중요성이 강조되는 부분이다. 프로그램 메뉴는 크게 세 부분으로 나누어져 그 각각의 기능은 다음과 같다.

4.2.1 파일 메뉴(File)
새로운 파일을 작성하거나, 이미 작성되어 있는 파일을 열거나, 저장한 파일을 저장하고자 할 때 선택하는 메뉴로 세부 메뉴로는 New, Load, Save, Save as, Exit가 있다.

- New : 새로운 파일을 작성하거나, 새로운 데이터를 입력하여 운영 프로그램을 운용한다.
- Load : 이미 작성되어 있는 데이터 파일을 읽어들인다.
- Save : 작성한 파일을 저장한다.
- Save as : 작성한 파일을 다른 이름으로 저장한다.
- Exit : 운영 프로그램을 종료시킨다.

4.2.2 입력 메뉴(Input)
작업 공간에 대한 정보와, 용접 대상물에 대한 정보, 그리고 용접 조건 및 위치 조정 데이터를 입력하려 할 때 선택한다. 세부 메뉴에는 Workspace data와, Welding process variable, Welding & weaving condition data가 있다.
- Workspace data : 로봇 손 꼭, 작업 도구, 장애물, 용접 대
4.23 finished product (Make)

입력받은 작업 환경 데이터와, 용접 작업 변수, 그리고 용접 및
방위 조건 데이터로부터 충돌 회피 경로를, 용접 조건 데이터, 용접
작업 프로그램을 만드는 내고자 할 때 선택하며, 세부 메뉴는 Make
collision free path, Make welding condition data, Make
welding job program, Make all 등이 있다.

- Make collision free path: Input 메뉴의 workspace data에서
  입력받은 작업 환경 데이터를 이용하여 충돌 회피 경로
  를 생성한다.

- Make Welding condition data: 사용자가 용접에 대해서 참
  모로는 경우 입력한 welding process variable 즉 용접 재료,
  용접봉의 종류와 크기, 용접 속도, 용접봉 공급 속도 등을
  생성한다.

- Make welding job program: 사용자가 용접에 대해서 참
  모로는 경우 입력한 용접 조건 데이터와 명령 조건 데이터 또는
  용접 작업 변수를 이용하여 로봇이 용접을 할 수 있도록 용
  접 작업 프로그램을 생성한다.

- Make all: 입력받은 모든 정보를 이용하여 충돌 회피 경로,
  용접조건 데이터, 용접 작업 프로그램을 한꺼번에 모두 생성
  한다.

5. 결론

본 연구는 기존의 로봇 제어 및 연계 시스템을 그대로 이용
하여 주어진 작업을 수행할 수 있는 로봇 자동 프로그래밍 시스
템에 관하여 연구하였다. 작업 중심의 명령어를 아예 용접이라
는 한정된 분야로 제한시키고, 특정 로봇 언어의 문법을 약력하
할 필요없이 프로그래밍할 수 있게 하였으며, 몇 가지의 작업 중심
명령어를 실현하기 위한 기본적인 기능을 구현하며, 비논리적인
작업이 입력되는 가능성을 최소화하고, 최소한의 노력으로 원하
는 프로그램을 작성할 수 있게 하였다.

반면, 본 시스템은 프로그래머 필요한 많은 전문적인 의사 결
정 부분은 - 충돌 회피 경로 선택, 준비 위치의 선정 등 - 을
매우 단순하게 처리하고 있다. 이러한 의사 결정 부분들을 좀 더
지능화시켜 고유하게 보완하여, 시스템을 작업 중심 언어의 수
준으로 향상시키는 것이 앞으로 남은 중요한 과제로 하겠다. 즉,
충돌 회피를 위해 좀 더 효과적인 최단 경로 선택 알고리즘을
개발하여 구현하고, 더욱 다양한 문제를 도입하고 인식하여 적
절한 잡을 위치를 선택하도록 하며, 센서 정보 처리를 위한 지식
을 참가하여 보다 복잡한 용접 작업에 응용하고자 한다.

6. 관련된 연구문헌


New York: ASME, Nov. pp.53-64, 1982


[10] 전석수, "용접 자동화를 위한 아크 용접 자동 프로그래

학교 공학석사 학위논문, 1989.