

## 아아크 용접 자동화를 위한 태스크 레벨 자동 프로그래밍 시스템 개발

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

°박 현 자, 이 범 회

서울대학교 전기공학부(Tel : 02-880-7316, Fax : 02-885-6620, E-mail : phj@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 is a tool 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 날로 그 필요성이 증가하고 있는 공정 자동화의 일환인 용접 자동화에 관한 것이다. 공정 자동화가 급속히 진행됨에 따라, 점점 더 다양하고 복잡한 로봇의 기능이 요구되고 있고, 그에 따라 로봇을 프로그래밍하는 로봇 자동 프로그래밍 시스템에 관한 연구가 활발하게 진행되고 있다. 그리고 작업 중심 언어의 개념을 이용한 태스크 레벨 자동 프로그래밍 시스템은, 구체적인 문제는 시스템 내부에서 해결하고 개괄적인 작업만을 지정하는 이상적인 프로그래밍 방법이므로 작업 환경을 지능적으로 관리하는데 필수적이라고 할 수 있겠다.

이 연구의 목적은 로봇에 의해 아아크 용접 자동화를 하는 것으로, 용접이나 로봇 프로그래밍의 비전문가가 효율적으로 아아크 용접용 로봇 프로그램을 작성할 수 있도록 자동 프로그래밍 시스템을 개발하는 것이다. 그 첫 단계로서 로봇 프로그래밍 언어들에 관해 조사하고, 태스크 레벨 언어의 특징 및 태스크 레벨 아아크 용접 자동 프로그래밍 시스템의 작업 계획 시스템을 구성한다. 다음 단계로서 작업 계획 시스템을 아아크 용접 자동 프로그래밍 시스템과 통합하여 태스크 레벨 자동 프로그래밍 시스템을 완성한다.

## 2. 작업 중심 언어와 태스크 플래너

### 2.1 로봇 언어의 분류 및 특징

로봇 언어는 크게 마이크로 컴퓨터 단계, 점간 단계, 원시 운동 단계, 구조 프로그래밍 단계, 작업 중심 단계로 나뉜다.

마이크로 컴퓨터 단계(Micro computer level) 언어는 관절의 이동 각도나 거리를 모터 구동을 위한 힘, 토크로 바꾸는 로봇 제어의 가장 기본적인 단계이며, 로봇의 구조에 따라 결정적이다.

점간 단계(Point-to-point level)는 교시를 기본으로 한 가장 사용하기 쉬운 방법으로 단순 반복 작업에 유용하다. 조건 분기 기능이나 센서 정보의 이용이 어렵고, 오프 라인 프로그래밍 시스템으로 확장할 수 없으며, 모든 위치 정보는 로봇 각관절 각도의 조합으로 나타내지는 관절 좌표 시스템(Joint Coordinate System)으로 표현된다.

원시 운동 단계(Primitive motion level)는 로봇 자체의 이동을 기술하는 데에 중점을 둔 언어로서, 각 관절을 따로 구동할 수 있으므로 낮은 단계의 하드웨어 및 소프트웨어를 실험하고 평가할 수 있어 주로 개발의 도구로 쓰이며 사용법이 비교적 간단하다. 대부분 어셈블리를 기초로 실현되어 간단한 조건 분기와 서브 루틴이 이용되며, 센서의 이용 능력도 향상되었으나 주로 ON/OFF의 이산 센싱 정도이다. 또한 병렬 처리와 프레임의 이용도 도입되었으나 매우 기본적인 수준이며, 좌표계는 위치 정보가 센서 및 로봇에 공동된 절대 좌표축에 따라 나타내지는 직각 좌표 시스템이다.

구조 프로그래밍 단계(Structured programming level)는 다양한 센서의 이용, 복잡한 제어 구조, 다양한 데이터 구조에 따른 실시간 계산이 가능하다. 또한 상태 변수들을 정의하여 사용자의 효과적인 프로그램을 돕고, 작업 중심 언어의 환경 모델링 기능을 갖기도 한다. 이들 언어의 좌표계는 상대 좌표에 따른 프레임을 정의하여, 외부 센서에 의한 위치 계산 및 조립 공정의 진행에 따른 물체상 프레임들의 연속적이고 복잡한 변화를 효과적으로 처리하는 구조적 직각 좌표 시스템이 이용된다.

선술한 언어들에 비해 로봇의 동작을 중점적으로 기술하는데 반하여 작업 중심 단계(Task-oriented level)의 언어는 로봇의 동작이 작업 공간 내에 미치는 영향을 기술하므로, 로봇보다는 작업 공간 내의 물체의 관점에서 명령 체계가 이루어진다. 센서나 좌표축의 이동 등을 내부적으로 해결하고 개괄적인 작업만을 지정하는 이상적인 프로그래밍 방법인 이 단계의 언어는, 자신의 작업 환경 모델링 능력과 작업 분석 및 이해 능력, 실행 가능한 로봇 프로그램의 생성 능력이 요구된다. 요구하는 기능은 World Modelling, Task specification, Collision-Free Path Planning, Automatic Grasping, Compliant

Motion, Error Checking 등이다.

## 2.2 태스크 플래너

태스크 플래닝(Task Planning)이란 주어진 태스크를 수행하기 위한 일련의 모션 프리미티브(Motion Primitives)를 생성하는 과정을 말하며, 크게 두 가지 접근 방법이 있다. State-Space Approach는 초기 상태에서부터 목표하는 상태까지의 경로를 찾고 정의하는 방법을 말하고, Action-Ordering Approach는 최초의 High-Level Plan(Skeleton Plan)이 주어졌을 때 목표를 완수하기 위한 세부 계획들을 생성해내는 방법을 말한다. 이 두 방법 중에 Action-Ordering Approach가 State-Space Approach에 비해 복잡한 로봇 매니플레이션에 적합하므로 Action-Ordering Approach를 이용하기로 한다.

사용자가 Weld\_Object와 같은 태스크 레벨 명령을 내리면 태스크 플래너는 환경 모델러와 경로 계획기로부터 이동 경로를 생성한다. 이동 경로를 생성하기 위해서 환경 모델러와 경로 계획기는 사용자로부터 작업 환경에 대한 정보 - 구체적으로 말해서 로봇의 위치, 작업대의 위치, 용접 대상물의 위치, 장애물의 위치 등 - 를 입력받아 용접 대상물까지의 충돌 회피 경로를 계산해 낸다.

로봇 매니플레이터가 용접 대상물까지 이동해 가면, 사용자로부터 입력받은 용접 대상물의 재질과 두께, 이음의 종류, 홈의 종류와, 용접 자세 등 용접 대상물에 대한 정보를 가지고, 데이터 베이스를 이용하여 용접 전압, 전류, 용접봉 공급 속도, 토오치의 속도 등 용접 조건들을 생성한다. 생성된 용접 조건을 이용하여 용접 작업을 수행하는 용접 프로그램을 생성해 내고 용접 작업을 수행한다.

## 3. 작업 계획 시스템의 설계

### 3.1 환경 모델러(World Modeller)

환경 모델링은 로봇 뿐만 아니라 작업 환경 내의 센서, 주변 기기 및 각종 물체들에 공통된 하나의 좌표계를 두어 이들 상호간의 결합 관계를 나타내는 작업으로, 작업 환경에 동일성을 부여하고, 보다 지능적인 작업을 가능하게 한다. 일반적으로 작업이 진행됨에 따라 이들 대상물에 관한 정보는 계속적으로 변화하게 되는데, 이러한 변화가 로봇 시스템에 전달되지 않으면, 다음 작업이 제대로 수행되지 못한다. 이러한 문제점은 모델링을 어렵게 하는 요소로 지적되고 있는데, 본 시스템에서는 작업의 진행 상황을 내부적으로 시뮬레이션하여 변화된 작업 공간의 정보를 계속적으로 보완함으로써 이 문제를 해결한다.

작업 공간 내의 물체는 다양한 형태를 가지므로, 이들을 체계적으로 표현하는 규칙이 마련되어야 하는데, 본 시스템에서는 작업 공간내의 모든 물체를 직육면체로 근사하여 나타내기로 하겠다. 사용자로부터 근사된 직육면체의 기준점과, 가로, 세로, 높이를 나타내는 세 점을 입력받아 데이터로 사용한다.

이러한 데이터 외에 로봇 손 끝의 위치, tool의 잡는 위치와 크기, 용접 대상물의 용접 위치, 장애물의 개수, 용접 대상물에 대한 정보들을 입력받아 사용한다.

### 3.2 경로 계획기(Path Planner)

작업 공간의 상태에서부터 접근 위치, 후퇴 위치 및 충돌이 일어나지 않기 위한 중간 경위점 등을 선택한다. 충돌 회피를 위한 알고리즘은 작업 중심 언어의 중요한 연구 분야의 하나로, 최전 관철을 갖는 로봇에 대한 것으로는 크게, 장애물을 로봇의 관절 좌표에서 근사시키는 방법과, 장애물 경계선의 정확한 수식을 이용하는 접근 방법으로 양분된다. 현재의 충돌 회피 방법은 지극히 간단한 형태로 Z축 방향으로의 경로 변환만이 허용되며, 여기에 사용된 가정은 다

음과 같다.

- ① 로봇 손 끝의 충돌만을 고려한다.
- ② 충돌이 일어나지 않는 최소 시간 경로가 아니라, 직각 좌표계에서의 최단 거리 경로를 찾는다.
- ③ 경로 변경은 Z축 방향으로만 허용한다.
- ④ 이동시 로봇이 잡고 있는 물체의 방향은 변경하지 않는다.
- ⑤ 작업 공간 내의 고정 물체에 대한 충돌만을 고려한다.

최단 경로는 이 다각형들의 꼭지점 중 몇 개를 경위하게 된다는 특징에 근거를 두고 있다.

충돌본 시스템에서 채택한 모델은 x-y 평면에 수직인 어떤 평면으로 절단하든지 모든 물체가 직사각형으로 표현되는 특징이 있다. 이와 같은 특징은 3차원의 문제를 2차원 문제로 분할하여 고려할 수 있게 하므로 간단한 알고리즘을 이용한 경로 선택을 가능하게 한다.

따라서 경로 선택 방법도 두 개의 2차원 문제로 분할하여 접근한다. 이 충돌 회피 알고리즘은 2차원 평면에서 다각형의 물체를 피하여 이동하는 회피를 위한 중간 경로를 선택하는 방법은 다음과 같다.

#### 3.2.1 장애물 모델의 근사

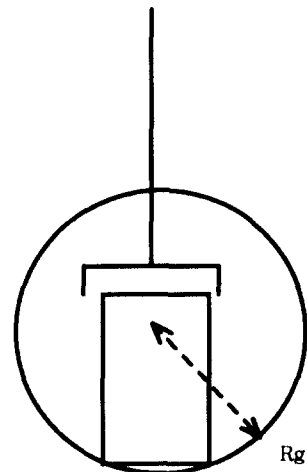


그림 3.1 로봇 손 끝의 근사 과정  
Fig.3.1 Approximation of robot end effector

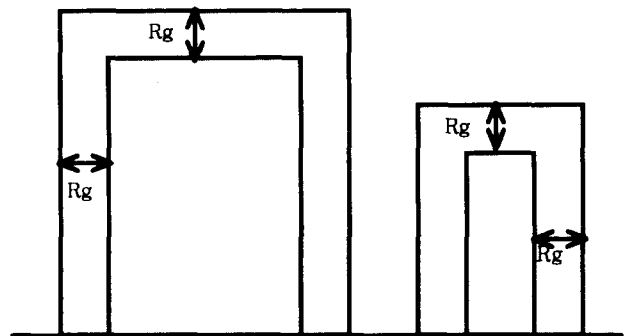


그림 3.2 장애물 근사 과정  
Fig. 3.2 Approximation of Obstacles

그림 3.1과 그림3.2에 나타나 있는 것처럼 손에 잡고 있는 물체의 크기에 따라 물체의 방향 변화에 따른 영향이 무시될 만큼 충분히 큰 반지름을 갖는 구로, 로봇의 손끝을 근사시키고, 손끝을 점으로 다룰 수 있도록 모든 장애물의 크기를 구의 반지름만큼 증가시켜 조정한다.

### 3.2.2 수평면 확인

이동 경로의 시작점과 끝점에 의해 이루어진 직선과 작업 공간내의 모든 장애물을 x-y평면으로 투영시켜 2차원에서 이들의 교점을 구함으로써 충돌 가능성이 있는 물체들을 찾아낸다.

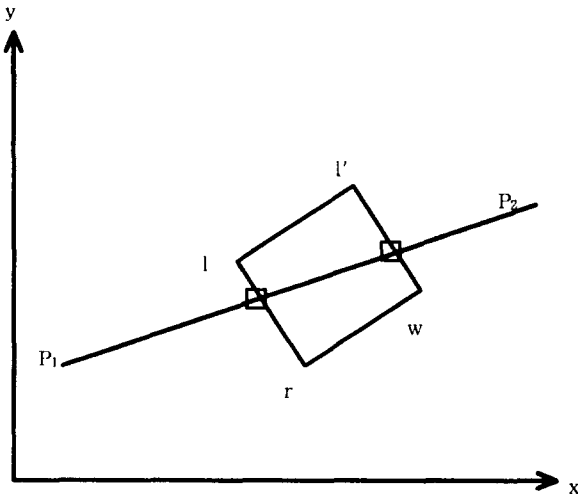


그림 3.3 수평면 확인  
Fig. 3.3 Checking horizontal plane

### 3.2.3 수직면 확인

앞에서 구한 물체들을 이동 경로가 이루는 직선이 x-y평면으로 투영될 때 이루는 면으로 절단하고, 그 면에서 이동 경로의 시점과 종점이 이루는 직선의 기울기와, 시점 또는 종점과 앞에서 구한 교점이 이루는 직선의 기울기를 비교함으로써, 충돌이 일어나지 않도록 적절한 교점을 포함하여 경로를 변경한다.

그림 3.4에서는  $P_1A_1$ ,  $P_1A_2$ ,  $P_1A_3$ ,  $P_1P_2$ 의 기울기 중에서  $P_1A_1$ 의 기울기가 가장 크므로 그 끝점  $A_1$ 을 중간 경위점  $O_1$ 으로 잡고,  $A_1A_2$ ,  $A_1A_3$ ,  $A_1P_2$ 의 기울기 중에서  $A_1A_3$ 의 기울기가 가장 크므로 그 끝점  $A_3$ 를 중간 경위점  $O_2$ 로 잡는다.

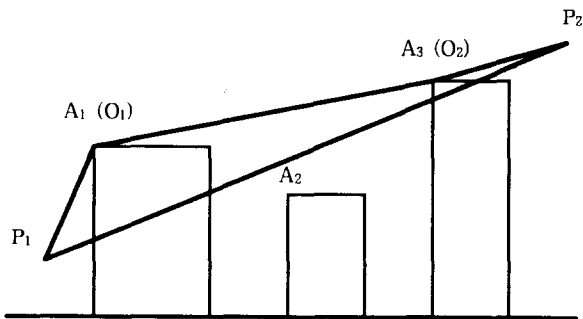


그림 3.4 수직면 확인  
Fig. 3.4 Checking vertical plane

## 3.3 작업 공간 관리자(Workspace Manager)

로봇에 알맞은 역기구학(Inverse Kinematics)을 풀어 그 결과를 작업 공간 모델과 비교하여, 도달 불가능한 지점이 프로그램되는 것을 방지한다.

이러한 기능은 정확한 모델의 정보를 통해 구현되는데, 겉으로 드러나는 시스템의 기능과는 무관한 부분이므로 사용자가 항상 논리적으로 합당한 위치 정보만을 입력한다는 조건이 만족되면 불필요한 부분이 될 수도 있다, 그러나 안전한 프로그램이 작성되고, 좀 더 수준 높은 계획 시스템과 연결하여 이용하기 위해서는 반드시 필요한 기능이다. 이 모듈은 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 언어가 갖는 단점을 보완하여 최소한의 노력으로 원하는 프로그램을 작성할 수 있게 하는 보조자적 도구로서, 다음과 같은 가정하에 구성된다.

- 환경 모델링을 위한 자료의 위치 및 방향 정보 오차는 없다.
- 로봇의 동작에 의한 오차가 없어 실제 환경과 모델링된 환경은 항상 같다.

시스템의 구조는 다음과 같다. 사용자가 사용자 인터페이스를 통해 원하는 작업 및 작업 환경에 대한 데이터를 입력하면, 경로 계획기, 작업 공간 관리자는 환경 모델의 정보를 바탕으로 블록 보드를 통해 서로 정보를 교환하여, 매니플레이터 구동을 위한 프로그램 모델을 만들어 낸다.

이 과정에서 실제의 로봇 프로그램을 출력할 수도 있으나, 본 시스템은 여러 종류의 로봇 언어를 생성하기 위해 pseudo code의 개념을 이용하여 특정 로봇 언어와 무관한 부분은 프로그램 모델이라는 공통의 문법을 이용하여 실제의 로봇 프로그램을 작성할 수 있는 제반 정보를 전달한다. 이 결과는 프로그램 합성기에 의해 원하는 로봇 언어로 변환된다.

### 4.2 사용자 인터페이스

사용자 인터페이스는 시스템과 사용자와의 의사 전달을 담당하는 부분으로 사용자가 프로그램의 경험이 없는 초보자임을 고려할 때, 그 중요성이 강조되는 부분이다. 프로그램 메뉴는 크게 세 부분으로 나누어지고 그 각각의 기능은 다음과 같다.

#### 4.2.1 파일 메뉴(File)

새로운 파일을 작성하거나, 이미 작성되어 있는 파일을 편집하거나, 작성한 파일을 저장하고자 할 때 선택하는 메뉴로 세부 메뉴로는 New, Load, Save, Save as, Exit이 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 위험 조건 데이터를 입력하고자 할 때 선택한다. 세부 메뉴에는 Workspace data와, Welding process variable, Welding & weaving condition data가 있다.

- Workspace data : 로봇 손 끝, 작업 도구, 장애물, 용접 대

상물의 위치 와 크기 등 작업 공간에 대한 모든 정보를 입력한다.

- Welding process variable : 사용자가 용접에 대해 잘 모를 경우 선택하며, 용접 대상물에 대한 정보 즉 용접 대상물의 재질, 두께, 이음의 종류, 홈의 종류, 홈의 거리, 용접 자세 등을 입력한다.
- Welding & weaving condition data : 사용자가 용접에 대해 잘 아는 경우 선택하며, 용접 조건 데이터 즉 용접 전류, 용접 전압 등과 위빙 조건 데이터 즉 위빙 모드, 위빙 속도, 위빙 폭 등을 입력한다.

#### 4.2.3 실행 메뉴(Make)

입력받은 작업 환경 데이터와, 용접 작업 변수, 그리고 용접 및 위빙 조건 데이터로부터 충돌 회피 경로, 용접 조건 데이터, 용접 작업 프로그램을 만들어 내고자 할 때 선택하며, 세부 메뉴로는 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 경로를 생성한다.
- Make Welding condition data : 사용자가 용접에 대해서 잘 모르는 경우 입력한 welding process variable 즉 용접 전류, 용접봉의 종류와 크기, 용접 속도, 용접봉 공급 속도 등을 생성한다.
- Make welding job program : 사용자가 용접에 대해 잘 아는 경우 입력한 용접 조건 데이터와 위빙 조건 데이터 또는 용접 작업 변수를 이용하여 로봇이 용접을 할 수 있도록 용접 작업 프로그램을 생성한다.
- Make all : 입력받은 모든 정보를 이용하여 충돌 회피 경로, 용접조건 데이터, 용접 작업 프로그램을 한꺼번에 모두 생성한다.

## 5. 결 론

본 연구는 기존의 로봇 제어기 및 언어 시스템을 그대로 이용하여 주어진 작업을 수행할 수 있는 로봇 자동 프로그래밍 시스템에 관하여 연구하였다. 작업 중심의 명령어를 아아크 용접이라는 한정된 분야로 제한시키고, 특정 로봇 언어의 문법을 익혀야 할 필요없이 프로그래밍할 수 있게 하였으며, 몇 가지의 작업 중심 명령어를 실현하기 위한 기본적인 기능을 구현하여, 비논리적인 작업이 입력되는 가능성을 최소화하고, 최소한의 노력으로 원하는 프로그램을 작성할 수 있게 하였다.

반면, 본 시스템은 프로그래머시 필요한 많은 전문적인 의사 결정 부분들 - 충돌 회피 경로 선택, 잡는 위치의 선정 등 - 을 매우 단순하게 처리하고 있다. 이러한 의사 결정 부분들을 좀 더 지능적으로 처리하도록 보완하여, 시스템을 작업 중심 언어의 수준으로 향상시키는 것이 앞으로 남은 중요한 과제라 하겠다. 즉, 충돌 회피를 위해 좀 더 효과적인 최단 경로 선택 알고리즘을 개발하여 구현하고, 더욱 다양한 물체를 모델링하고 인식하여 적절한 잡는 위치를 선택하도록 하며, 센서 정보 처리를 위한 지식을 첨가하여 보다 복잡한 용접 작업에 응용하고자 한다.

## 6. 관련된 인용문헌

[1] Dornfeld, D. A., Tomizuka, M., and Langan, G.,

"Modeling and adaptive control of arc welding process," *Meas. and Cont. for Batch Manufac.*, Hardt, D. E., Ed. New York: ASME, Nov. pp.53-64, 1982

- [2] Lindberg, R. A., and Braton, N. R., "Welding and other joining process," Boston: Allyn and Bacon Inc., 1979.
- [3] ChuXin Chen and Mohan M., "Task Planning and Action Coordination in Integrated Sensor-Based Robots," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 25, No. 4, 1995.
- [4] Christesen, N., Davies, V. de L., and Gjermundsen, "Distribution of temperatures in arc welding," *Brit. Welding J.*, vol. 54, no.2, pp.54-75, Feb. 1965,
- [5] K.S.Fu et al., *Robotics, control, sensing, vision, and intelligence*, McGraw-Hill, 1987.
- [6] C.S.G. Lee et al., *Tutorial on Robotics*, IEEE Computer Society, 1986.
- [7] Sicard, P., Levine, M. L., "An approach to an expert robot welding system," *IEEE Trans. on Systems, Man, and Cybernetics*, vol.18, no.2, pp.204-222, March/April 1988.
- [8] *Welding Handbook-Volume 1: Fundamentals of Welding*, 7th ed. New York: American Welding Society, 1976, pp.52-98.
- [9] 엄기원, *최신 용접 공학*, 동명사, 1988
- [10] 전 명수, "용접 자동화를 위한 아아크 용접 자동 프로그램 시스템 개발," 서울대학교 공학석사 학위논문, 1995.
- [11] 조 혜경, "로봇 자동 프로그래밍 시스템의 개발," 서울대학교 공학석사 학위논문, 1989.