

# 공정 지향적인 프로그래밍 시스템

박 홍 석  
울산대학교 생산기계공학과

## A Task-oriented programming system

Hong-Seok Park  
University of Ulsan Department of Production Engineering

### Abstract

This paper presents an algorithmic approach used in the development of a task-level off-line programming system for the efficient application of robot.

In the method, robot tasks are graphically described with manipulation functions. By applying robot language these graphic robot tasks are converted into commands for the robot. A programming example demonstrates the potentiality of task-oriented robot programming.

**Key Words :** Explicit programming (명시적인 프로그래밍), Implicit programming (함축적인 프로그래밍), Robot task planning (로봇 공정 계획), World model(주변 모델), Manipulation function(조작기능)

### 1. 서론

산업 로봇의 응용에서 경제성을 증가시키기 위한 중요한 측면은 프로그램의 비용을 절감시키는 것이다. 프로그램 생성, 시험 및 초기 운전에서 기계 정지 시간을 줄이기 위해서 지난 몇 년전부터 Off-line-programming 방법[1,2]이 적용되어 왔다. 이것을 가지고 프로그램들이 로봇과 떨어져서 생성되고, 운전 가능한 프로그램이 되어 로봇 제어에 전송된다. 여기에서 문제는 프로그래밍의 방법은 여전히 명시적인 형태를 취하고 있는 것이다. 이 방법은 프로그래머에게 복잡한 로봇언어의 지식과 로봇 동작 및 움직임의 목표에 대한 자세한 기술을 요구한다. 작은 로봇응용과제라도 프로그램에 많은 어려움이 있다.

사용자에게 편리한 프로그래밍의 방향으로서 본 연구에서는 로봇 응용을 위한 함축적인(Implicit) 프로그래밍[3-5], 일반적으로 Task-Level 로봇 프로그래밍으로 알려진 시스템을 CATIA 환경에서 개발하고자 한다. 이 시스템에서는 로봇이 작업을 어떻게 수행하는가 아니라 무엇을 행하는가로, 즉 공정으로써 “피스톤 링을 피스톤에 삽입” 프로그램이 된다. 이를 위해 공정의 수행에 요구되는 로봇동작을 분석하여 기능 함수로 기술하였다. 기능함수로 표시된 각 동작들을 해당 로봇언어로 전환시킴으로써 계획된 공정이 실행 되도록 하였다. 이것을 통해서 프로그래밍의 단순화뿐만 아니라 생산의 유연성 및 경제성과 자동화에 크게 기여할 것이다.

### 2. 공정지향적인 프로그래밍 시스템

#### 2.1 프로그래밍 시스템 구성

로봇 프로그램의 생성은 계획자로 부터 조립순서, 조립에 요구되는 도구들과 그들 상호호적인 위치를 나타내는 Layout등에 대한 복잡한 지식을 요구한다. 이들에 대한 정보는 조립과제의 기술과 주변 모델의 모델링과정을 통해서 알 수 있기 때문에 로봇 프로그램을 생성하는 것이 가능하다(Fig. 1).

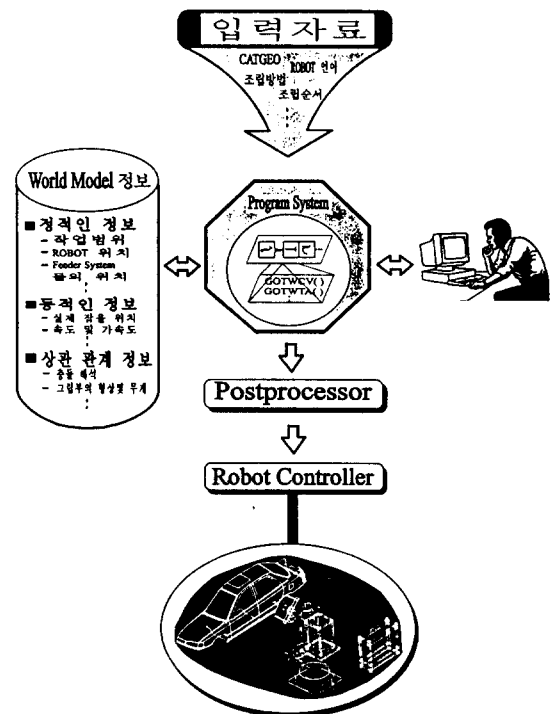


Fig. 1 Structure of the programming system for generation of robot program

프로그래밍의 초기자료는 조립순서, 조립을 위한 접합방법과 로봇 프로그램언어이다. 프로그램 시스템은 이들 자료와 생성된 주변 모델 정보의 도움으로 조립과제를 로봇 제어를 위한 언어로 전환한다. 로봇의 응용에서 주변 모델은 전 작업범위내에서 기술적인 데이터의 기술을 포함한다. 로봇 외에도 이 모

델로부터 도구, 조립물 등에 대한 정적인 정보뿐만 아니라 프로그램 진행동안 로봇과 관계되는 동적인 정보들도 얻을 수 있다. 이와 같이 프로그래밍 시스템에서 중요한 역할을 하는 주변 모델을 CATIA System을 이용하여 모델링하였다.

시스템의 작업방법은 먼저 조립을 위한 로봇의 운동은 기능에 따라 분류되어 VDI - 표준 2860[6]의 조작 기능(Manipulation function)들로 기술된다. 다음 단계에서 조작 기능들은 그에 상응하는 로봇 프로그래밍 언어로 전환된다. 사용자는 프로그래밍 시스템과의 대화를 통하여 조립과제의 변경으로 인한 데이터의 수정을 주변 모델에 근거하여 할 수 있다.

로봇 프로그램은 현장에 적용되는 특정 로봇의 제어를 위한 언어로 직접 프로그래밍 가능하므로 원칙상 기존의 로봇 프로그래밍 시스템들의 Postprocessor는 필요치 않으나, 현장의 다양한 로봇에 응용을 위해 그 기능을 첨가시키는 것이 바람직하다고 본다. 본 시스템은 특정 기업의 특성에 따라 CATIA 시스템과의 호환성을 유지해야 하므로 개발환경으로 Fortran언어의 형태를 취한 CATIA-IUA를 이용하였다. 로봇운동의 제어를 위한 로봇 언어로는 CATGEO를 사용하였다. 개발된 프로그램 시스템에 의해 생성된 프로그램은 로봇 Controller에 down load 되기 전에 Postprocessor를 걸쳐서 Controller가 인식할 수 있는 언어로 전환된다. 제시된 시스템에서는 모기업의 특정 로봇제어에 이용하였으므로 로봇 제작사로부터 개발되어 사용되고 있는 Postprocessor를 이용하였다.

## 2.2 로봇 동작 기술을 위한 모델

로봇 운동의 묘사는 조작 기능(Manipulation function)에 의존한다. 움직임의 각 동작은 정의된 기능을 가지고 기술될 수 있다. 조작 기능들은 VDI 2860 취급 분야의 일부인 Moving과 Securing에 속한다. Moving 중에서도 Arrange 와 Feeding은 고려치 않았다. 왜냐하면 이 기능들은 공급(Feeder) 시스템의 기술에 관여되고 로봇 운동에는 무관하기 때문이다. 기능 Securing은 Grip 운동의 묘사에 이용된다. Tensing과 Relaxing은 Jaw의 움직임에 의한 마찰력으로 물건을 지지하는 Grip에 사용된다. Holding과 Detaching은 자력(Magnetic)과 진공(Vacuum) Grip의 작용을 묘사하는데 있어서 중요하다. 여기에 사용된 개념과 정의를 가지고 조립과제 해결에 필요한 각각의 Handling과정과 조립과정들이 해석되어질 수 있다. 이러한 동작행위를(이하 조립함수라 명한다) 로봇제어에 관련해서 기능적으로 각각 상세히 기술하면 Fig. 2에 보여진 조작기능들의 순서로 나타난다. 실제 접합이 일어나는 과정을 표시하기 위해 기호(Symbol) Jointing을 첨가했다.

함수 이송(Transport)은 로봇가 정해진 경로없이 한 점에서 다른 점으로 이동하는 것을 의미한다. 이 기능으로 물체는 멀거나 가까운 거리로 이동되어진다. 이때 운동속도는 빠르고, 충돌을 피하기 위해 특별히 지정되는 경우를 제외하고는 운동경로는 자유롭다. 회전(Turning)에서는 물체가 이송중이나 또는 정지상태에서 돌려진다. 나머지 함수들은 물체의 조작, 예를 들면 잡음(Picking)과 놓음(Placing), 여러가지 접합방법들로 구성된다. Fig. 2의 잡음과 놓음의 아래 부분은 자력 Grip이나 진공 Grip의 사용시를 나타낸다. 로

봇의 움직이는 과정이 같기 때문에 접합방법 기호(Insertion)등은 함수 놓음으로 기술될 수 있다. 대개의 조립 셀에서는 로봇이 수행하기 불가능하거나 어려운 접합방법, 예로써 억지끼움(Pressing in), 나사(Screwing)및 리벳(riveting) 등은 전문적인 작업장(Station)에서 이행되어진다. 나사 접합에서 조작 기능들의 순서는 로봇이 접합할 물체를 나사 작업장으로 옮겨서, 그 곳에서 나사 접합이 행해지는 것을 나타낸다.

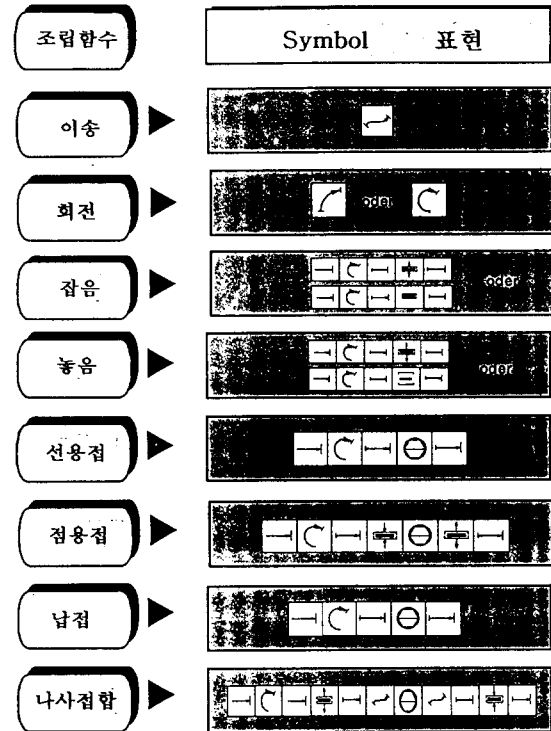


Fig. 2 Symbolic representation of assembly function

위에 기호로 나타낸 조립함수의 운동순서는 조립과제나 공구의 특수성에 의존하여 벗어날 수 있다. 그러나 많은 조립문제에 대한 로봇의 운동특성을 해석하여 유도하였으므로 대개의 조립운동에 유효하다.

이와 같은 조작기능 기호로 조립 과정 및 Handling 과정들의 해석과 기술을 통해서 아래의 장점및 가능성을 얻을 수 있었다.

- 문제에 관련된 모든 요구사항을 고려하여 복잡한 로봇의 움직임을 간단하고, 명료하게 나타낼 수 있다.
- 적합하지 않거나 불필요한 동작과 과정은 쉽게 파악되어 제거할 수 있다.
- 부가적인 기능, 예를 들면 Vision Sensor에 의한 감시 및 시험기능등을 첨가함으로써 쉽게 확장가능하다.

각 조작기능들에 의존해서 조립함수의 실제 실행을 위해서는 수행해야할 기능 외에도 도달해야할 위치, 이동 속도, 잡는 힘, 접합 시간과 접착제에 의한 접합에서 경로 등의 정보가 필요하다. 로봇의 응용에서 이러한 데이터는 그때의 작업환경을 나타내는 주변 모델에서 얻는다. 얻어진 데이터는 Databank에 의해 관리되어진다.

2.3 조작 기능들의 프로그램 언어화  
조립함수들이 조작 기능들의 기호로 기술된 후 로봇 프로그램 생성을 위해 각 기호들은 그에 상응하는 로봇 프로그램 언어로 전환되어야 한다. 이 전환은 로봇 제어를 위한 프로그래밍 언어의 문장론과 의미론에 대한 지식을 가지고 이루어진다. Fig. 3에 일 예로써 앞면에 소개된 조작 기능들을 수행하기 위한 명령어를 CATGEO로 나타내었다.

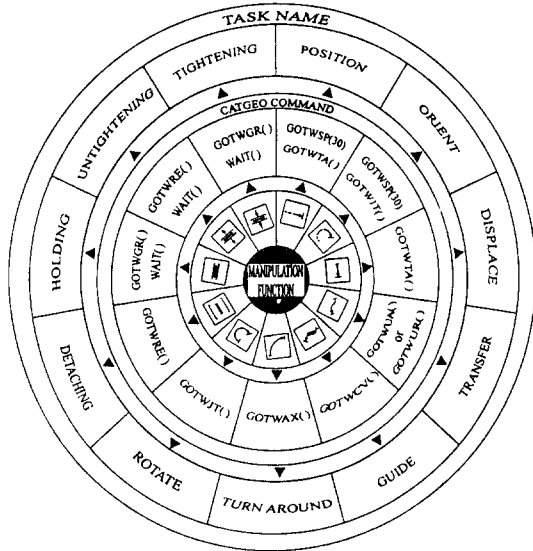


Fig. 3 Translation of the manipulation functions into the CATIA-CATGEO commands

Position과 Orient를 가지고 Robot는 정확한 위치에 도달해야 하므로 속도를 감속해야 한다. 이를 위해 GOTWSP의 명령어를 사용했다. Transfer에서 GOTWUA는 Robot의 절대좌표계에 관련된 운동이고, GOTWUR는 상대좌표계의 값에 따라 움직인다. 기능 "Rotate"와 "Turn around"는 외부에서 공급되는 에너지, 즉 공압등에 의해서 구동되는 Grip의 운동으로 대상물이 돌려지는 경우이다. Rotate는 정지된 상태에서의 방향 전환이고, Turn around에서는 로봇이 대상물을 이동하면서 회전한다. "Tightening"과 "Untightening"은 Grip운동에서 기계적인 지연 때문에 명령어 WAIT가 요구된다.

조립을 위한 로봇의 동작이 이 조작 기능들의 조합으로 구성되므로 각각이 프로그램상에서 고유의 이름을 갖는 독립적인 단위가 되도록 하였다. 이렇게 함으로써 조립 특수성에 의하여 조립함수의 로봇 운동의 특성이 바뀔 경우 이들의 재조합으로 새로운 변화에 쉽게 적응되도록 하였다.

프로그래밍을 위해서는 소개된 명령어 외에 프로그램 흐름을 제어하기 위한 구조적인 요소도 요구된다. 여기에 소개하면 일반 프로그램 언어 등에서 전형적인 제어문장인 IF.....THEN.....ELSE 등을 들 수 있다.

#### 2.4 시스템 구현

로봇 프로그램 생성 과정은 Fig. 4의 층 모델로 표현되어진다.

로봇 프로그램의 자동생성은 두단계로 나누어진다. 기능층에서는 조립에 요구되는 조작기능들의 순서를 기호(Symbol)로 자세히 기술한다. 언어층에서는 이들을 해당 로봇의

특징에 따라 로봇 제어 관련된 운동명령어로 전환시킨다. 각 단계에서의 작업은 앞절에서 소개된 법칙에 따라 행해진다. 이것으로 법칙에 의해 수행되는 전문가 시스템의 작업방법을 취하여 로봇 프로그램이 자동으로 생성되게 한다.

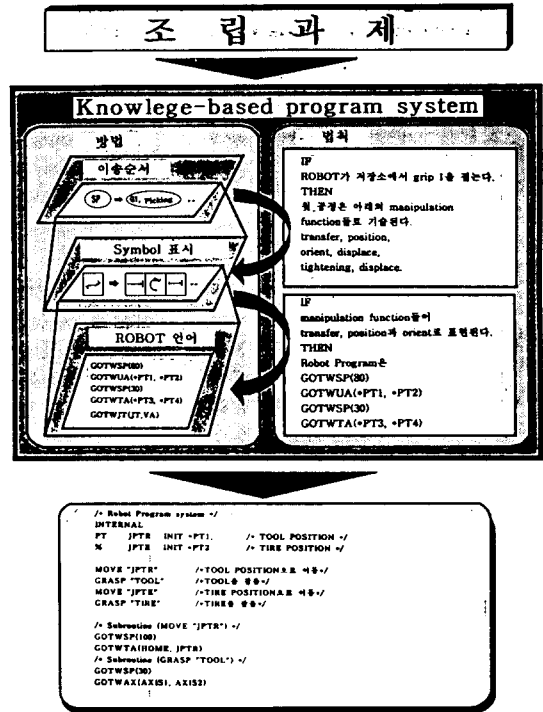


Fig. 4 Working method of the robot program system

Fig. 4의 아래 부분에 작성된 프로그램의 일부를 나타내었다. 프로그램의 구조는 세 개의 기본구조, 즉 입력정보(주변모델로 부터 얻어진 Data), 주 프로그램, 부 프로그램으로 구성되어 있다.

자주 반복되는 작업순서는 부 프로그램으로 구현되게 하였다. 이 같은 과제지향적인 기술을 통해서 명시적인 프로그램에서 사용자에게 요구되었던 과제해결에 필요한 지식, 즉 조립동작기술, 명령어, Parameter 와 도달점등을 개발된 프로그램 시스템에 전가시킬 수 있다. 이로써 프로그램에 대한 사용자의 부담을 줄이고 계획자는 단순한 일상업무로 부터 제외될 수 있다.

시스템의 확장은 조립순서에서 감시기능, 즉 시험 및 검사기능 삽입으로 이루어질 수 있다. 이에 의해 장래에 시스템은 지능 조립의 범위에서 센서의 지원하에 이루어지는 자동조립을 위한 프로그램도 생성할 수 있다.

#### 3. 응용사례

로봇 프로그래밍 범위에서 개발된 시스템을 차체, 타이어, 4개의 볼트를 가지고 차체에 타이어를 조립시키는 과제에 적용하여 보았다 (Fig. 5).

조립이 기존 시스템에서 수행되느냐 혹은 새로운 시스템의 설계가 요구되느냐의 결정에 따라 조립과제에 최적의 시스템은 CATIA 여러기능의 도움으로 주변 모델화 시킨다. 사용

자는 조립과제를 정확하게 파악하여 요구되는 공정과 작업순서로 부터 주변모델에서 조립에 요구되는 로봇의 운동경로를 유도한다. 이를 통해서 조립에 필요한 정적, 동적인 데이터 및 상관관계(충돌회피 등)의 정보들이 얻어진 후, 시스템내 데이터베이스에 저장된다. 타이머 조립순서에 따라 요구되는 로봇의 동작에 의존해서 사용자가 로봇프로그램 생성을 위해 자연어와 유사한 조립합수와 대상물을 기술하면 로봇 프로그램은 앞절에서 보인 것처럼 두 단계를 통해서 형성되어진다. 이 과정을 통해서 자연어처럼 기술하여 로봇 프로그램이 자동적으로 생성되도록 한다. 이것을 개발된 Program System이 맡아서 수행한다.

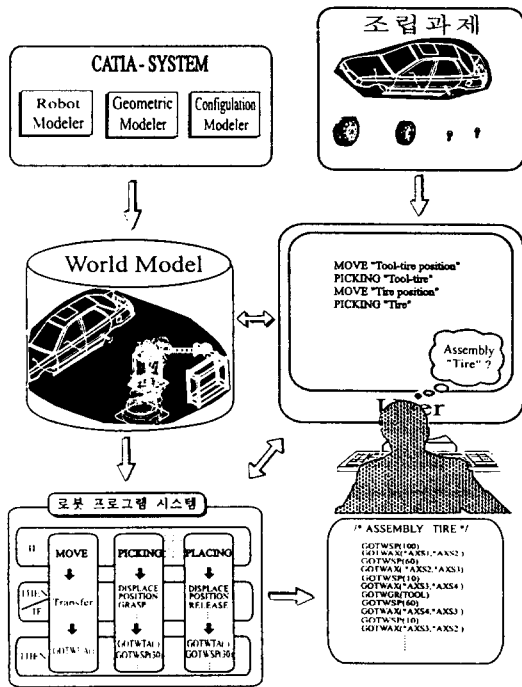


Fig. 5 Automatic generation of robot program

완성된 프로그램을 주변 모델에서 수행시켜 최종적으로 이상 유무를 판단한다. 이 단계에서 종합적으로 로봇의 작업을 관찰할 수 있고, 충돌문제에 관한 해석도 할 수 있다. 화면에 로봇 Joint의 현재 위치를 표시해주기 때문에 로봇의 각 Joint의 작업범위 한계에 너무 근접하는 동작은 피해서 프로그램을 수정할 수 있다. Joint의 최대값 근처에서 운전을 하면 위치 부정확도와 요소부품의 마모를 초래할 위험이 크다.

개발된 시스템 사용상의 잇점으로는 조립과제의 특수성에 의해 조립합수의 내용이 바뀔 경우에는 프로그램 시스템은 조작기능 Task명의 재조합으로 새로운 조립합수를 쉽게 만들어 프로그래밍에 이용되도록 하였다. 이렇게 조립합수의 확장과 축소를 자유롭게 함으로써 프로그램 시스템 유연성의 향상을 최대한으로 도모하였다.

#### 4. 결론 및 전망

소개된 공정지향적인 프로그래밍 시스템은 프로그래밍의 단순화와 사용자의 편의성을

추구한다. 이 시스템의 개발은 로봇 운동의 정확한 분석과 이의 체계화위에 이루어졌다.

개발된 프로그래밍 시스템의 수행능력은 차체에 타이머 조립과제에 적용되어 증명되었다. 이 현장응용력을 통하여 프로그래밍 시스템의 도움으로 로봇 프로그래밍에서 조립과제는 합축적인, 즉 공정으로 기술될 수 있다는 것을 보였다.

로봇 프로그래밍분야에서 진보적으로 추구하고자 하는 것은 조립과제와 로봇 주변환경의 완전한 기술로 자동적으로 조립을 위한 최적의 로봇 운동경로를 계획하고 생성하는 것이다. 이러한 지능적인 프로그래밍은 많은 센서로 지원되는 Task - Level 프로그래밍의 발전으로서 기대할 수 있다. 여기에 실현된 로봇 프로그래밍 시스템은 이와 같은 방향으로의 발전을 위한 기초를 다졌다.

#### [참고 문헌]

1. 조소형, 박강, 강성철, 김문상, "금형의 연마작업을 위한 로봇 오프라인 프로그래밍 시스템의 개발", 대한기계학회논문집, 제 15권 4호, pp.1387-1397, 1991.
2. Nagata, T., "Research Trends in the Off-Line Programming of Robots", J. Rob. Mech. Vol. 1, PP. 278-283, 1989
3. Freund, E., Heck, H., Krefst, K. and Mauve, C., "Osiris--An object-oriented system for implicit robot programming and simulation", Robotersysteme, Vol. 6, No. 3, pp. 185-192, Sep. 1990.
4. Bernhardt, R., Dillman, R. and Tierney, K., "Integration of Robots into CIM", Chapman & Hall, London, 1992.
5. Rondeau, J. M. and Elmaraghy, H. A., "Roboter programming and task planning", Manufacturing Review, Vol. 3, pp.245-250, 1990.
6. N. N., "VDI-Richtlinie 2860 : Handhabungsfunktionen, Handhabungseinrichtungen Begriffe, Definition, Symbole", VDI-Verlag, Duesseldorf, 1990.