

# 로봇 응용을 위한 공정 지향적인 프로그래밍 시스템 개발

박 홍 석\*

## A Development of Task-oriented Programming System for the Application of Robot

Hong-Seok Park\*

### ABSTRACT

Robot programming has been discussed in detail during the recent years. Numerous studies in particular presented relevance, solution concepts and implementation of off-line programming. In this paper a new user-friendly robot programming method is introduced, which permits the implicit description and programming of assembly process. On the functional level of programming, the assembly processes are described in terms of their operational functions. On the language level, the individual functions are then translated into commands for the robots.

**Key Words** : Explicit programming(명시적인 프로그래밍), Implicit programming(암묵적인 프로그래밍), World model(주변 모델), Manipulation function(조작기능), Assembly function(조립함수)

### 1. 서 론

짧아지는 제품의 수명과 다양한 고객의 요구에 따라 자주 변화되는 과제에서 조립시스템의 생산성과 경제성을 향상시키기 위해서 로봇을 효율적으로 이용하는 것이 필요합니다. 사용되어지는 프로그래밍 방법은 로봇 응용의 경제성에 중요한 영향을 끼친다.

로봇 프로그래밍은 지금까지 현장에서 대개 온라인 프로그래밍의 한 방법인 Teach-In으로 이루어져 왔다. 이것으로 인해 로봇의 실 가동률은 상당히 줄어들었다. 불

과 몇년전서부터 조립 분야에 적극적으로 적용되어 온 그래픽 시스템 지원 하의 오프라인 프로그래밍은 시스템의 가동률과 유연성을 높여 왔다.<sup>(1,4)</sup> 그러나 프로그래밍의 방법에서는 여전히 명시적인(Explicit) 형태를 취하고 있다. 이 방법은 사용자에게 복잡한 로봇 언어에 대한 높은 지식을 요구한다. 실제 현장의 사용자는 프로그램이나 로봇 기술에 특별한 교육을 받은 상태가 아니므로 많은 어려움이 있었다. 그 외에도 명시적인 로봇 프로그래밍은 프로그램의 작성에서 수행해야 될 기능, 예를 들면 Grip 동작, 로봇 동작, 움직이는 위치 등을 자세하게 기술해야

\* 울산대학교 생산기계공학과

만 한다. 그래서 로봇 프로그래밍을 간단히 할 수 있는 방법의 개발이 요구되어진다.

사용자에게 편리한 프로그래밍의 방향으로서 본 연구에서는 로봇 응용을 위한 함축적인(Implicit) 프로그래밍, 일반적으로 Task-Level 로봇 프로그래밍으로 알려진 시스템을 CATIA 환경에서 개발하고자 한다. 특히 프로그래밍을 위한 과제 기술에서 단순화를 추구하였다. 이 시스템에서는 로봇이 작업을 어떻게 수행하느냐가 아니라 무엇을 행하는 가로, 즉 공정으로써 "피스톤 링을 피스톤에 삽입" 프로그램이 된다. 이를 위해 공정의 수행에 요구되는 로봇동작을 분석하여 기능 함수로 기술하였다. 기능 함수로 표시된 각 동작들을 해당 로봇언어로 전환시킴으로써 계획된 공정이 실행되도록 하였다. 이것을 통해서 프로그래밍의 단순화뿐만 아니라 생산의 유연성 및 경제성과 자동화에 크게 기여할 것이다.

## 2. Task - Level 프로그래밍과 구성요소

기존 명시적인 프로그래밍에서는 한 기능의 수행에 필요한 정보들은 사용자로부터 정확하게 주어져야 한다. 도달점들은 공간 직교좌표나 로봇 축좌표로 기술된다. 이러한 프로그래밍 방법은 실행될 과제와는 연관성이 없이 프로그래머나 작업자에게 이해하기 어려운 형태, 예를 들면 PTP X -408.0 Y -979.7 Z +537.7 A +104.28 B -0.210 C -17 로 구성된다.

반면에 함축적인 프로그래밍, 즉 Task - Level 프로그

래밍의 목표는 원하는 최종 상태로 로봇의 운동을 기술하는 것이다.<sup>(5-6)</sup> 로봇이나 그의 좌표계가 프로그래밍에 중요한 것이 아니라, 수행할 과제가 핵심적이다. 아주 이상적인 경우에는 아래와 같이 우리 사고대로 자연스러운 말로 표현하여 프로그래밍을 할 수 있다.

조립 "타이어"

위 조립과제의 해결은 몇 가지 부분해, 잡음 "타이어", 끼움 "타이어", 체결 "타이어" 등의 결합으로 이루어진다. 이 부분해도 각 세분화된 로봇동작의 연속으로 구성되어 있다.

이러한 표현의 실현을 위해서 기술된 과제는 로봇의 주변환경과 로봇, 조립물, 조립과정에 대한 기하학적이거나 기술적인 정보에 근거해서 프로그램 시스템에서 로봇 운동 프로그램으로 전환되어야 한다. 이를 위해서 시스템이 갖추어야 할 3가지 기본요소를 Fig. 2에 나타내었다.

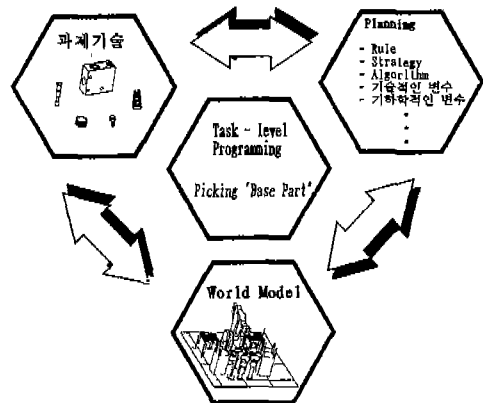


Fig. 2 Task-level programming system

과제 기술은 하나의 작업 단위에 대해서 문장 구조적으로는 자연스러운 언어로 쓰여지도록 한다. 작업 환경을 묘사하는 주변 모델은 프로그래밍 과정에 필요한 모든 기초정보를 제공한다. Planning 모듈은 주변 모델의 정보 위에서 자연 언어로 기술된 과제를 명시적인 세부 지시로 전환한다. 이에 개발된 알고리즘(Algorithm), 법칙(Rule), 전략(Stratgy) 등이 사용된다.

로봇의 응용에서 주변 모델은 전 작업범위내에서 기술적인 데이터의 기술을 포함한다. 로봇 외에도 이 모델로부터 도구, 조립물등에 대한 정적인 정보뿐만 아니라 프로그램 진행동안 로봇과 관계되는 동적인 정보들도 얻을 수 있다. 이와 같이 프로그래밍 시스템에서 중요한 역할

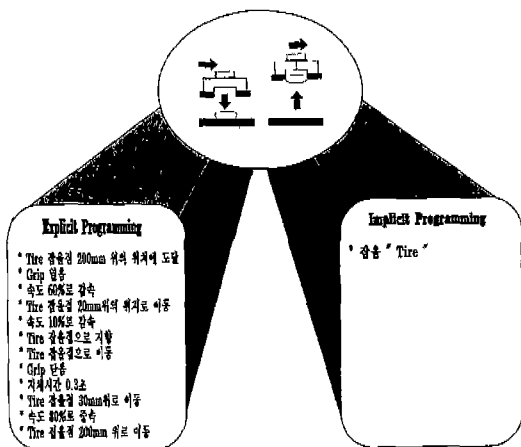


Fig. 1 Difference between conventional programming and task level programming

을 하는 주변 모델을 CATIA System을 이용하여 모델링 하였다. 먼저 CATIA CAD-SYSTEM의 Solid Model<sup>(7)</sup> 법의 도움으로 조립시스템 구성요소들(로봇, 공구, 공급 시스템, 치공구)의 정적 모델이 형성되어진다. 이후 공간 상의 배치 관계와 동적인 연결 상태, 즉 로봇에서 Joint로 연결되는 두 요소의 상대 운동을 정확하게 묘사한 동적인 모델을 통하여 운동특성이 부여된다. 위에서 모델링된 조립 셀 대상물의 모델들을 화면상에서 실제 상황에 알맞게 배치하기 위해서 CATIA는 OPEN메뉴와 Transform기능을 제공한다. 이 기능들의 도움으로 로봇과 주변 대상물들이 실제 생산라인의 환경과 동일하게 모델링되어지게 한다. 새로운 조립 셀을 생성하고자 할 때는 기존의 절차를 수행하면서 Robplace 기능을 이용해서 주변 요소에 도달 가능성을 검토하여 로봇의 위치가 결정되어지도록 한다. 이렇게 하여 조립과제에 가장 적합한 최적의 Layout이 설계되어질 수 있다.

위에 생성된 주변 모델을 가지고 프로그래머가 직·간접으로 얻을 수 있는 중요한 정보를 아래에 나타내었다.

■ 정적인 정보

- 조립셀의 작업공간
- 셀 구성요소, 예를들면 로봇, 공구, 공구대, 치공구, 공급시스템들의 기하학적인 정보
- 셀내에서 구성요소들의 위치와 방향성
- 조립물의 위치와 방향성
- 작업불가능한 범위

■ 동적인 정보

- 공정상의 로봇, 공구 및 조립물의 위치
- 로봇, 주변장치들의 동적인 특성 (Joint 허용범위 안에서의 로봇팔의 운동성 등)
- 조립시 접근속도
- 조립물의 위치와 공정변화 시점
- 운동경로

■ 상관관계 정보

- 접근가능성
- 조립방향성
- 셀내 다른 요소와의 충돌관계

■ 물리적인 정보

- 조립물 표면 정도

- 조립물 무게
- 조립물 성질
  - 유연성
  - 온도
- 잡는 힘

■ 공정관련 정보

- 조립후의 조립물의 형상 및 무게변화
- 조립방법에 따른 물리적인 성질 변화 (용접 및 납접후의 온도변화)
- 접합에 요하는 시간

이 외에도 로봇 프로그램이 생성된 후에 이의 증명을 위해서도 사용된다. 프로그램의 논리성 및 정확한 위치 도달에 대한 평가 등에 사용된다.

완전한 Task - Level 프로그래밍은 계획만으로 해결될 문제는 아니다. 기대치 못했던 상황, 방해 요소 등으로 프로그램이 국부적인 생산 조건의 변화에 적응할 것을 강요한다. 그래서 센서의 도움으로 외적 환경 변화를 정확하게 파악하여 주변 정보를 생산 과정에 전달하는 지능적인 시스템이 되어야 한다. 그러므로 완전한 Task - Level 프로그래밍 시스템을 실현하여 현장에 적용하기에는 많은 비용과 기술력의 부족으로 당장에는 어려움이 있다. 로봇응용기술의 경제성 때문에 이러한 어려움에도 불구하고 이 방향을 추구하는 약간의 시스템들이 소개되고 있다.<sup>(8-16)</sup> 이들 중 일부 시스템은 조립부품과 주변환경에 대한 CAD-Data로부터 기하학적인 연관성을 고려하여 조립로봇의 경로 계획에 치중한다. 또 다른 시스템에서는 조립계획자가 복잡한 조립과제를 각각의 단위작업으로 분류하고, 기하학적인 형상에 따른 간단한 법칙에 의해 그들의 작업순서를 결정한다. 이에 의존해서 세부계획자, 즉 경로계획자, 공구계획자, 접합계획자가 주변 모델과의 연관성아래 각 작업의 실행에 요구되는 로봇행위와 순서를 결정하는데에 주안을 둔다. 소개된 시스템들은 조립과제의 기술과 법칙기술 등에 많은 시간을 요하고 대개 간단한 조립문제에 응용하였다. 이에 반해 개발된 시스템은 프로그래밍의 계획단계에서 공정의 기술로 로봇프로그램이 자동 생성되도록 하여 프로그래밍의 용이화와 단순화를 도모하였다.

### 3. 공정지향적인 프로그래밍 시스템

#### 3.1 프로그래밍 시스템 구성

로봇 프로그램의 생성은 계획자로 부터 조립순서, 조립에 요구되는 도구들과 그들 상호적인 위치를 나타내는 Layout등에 대한 복잡한 지식을 요구한다. 이들에 대한 정보는 조립과제의 기술과 주변 모델의 모델링과정에서 알려져 있기 때문에 로봇 프로그램을 생성하는 것이 가능하다.

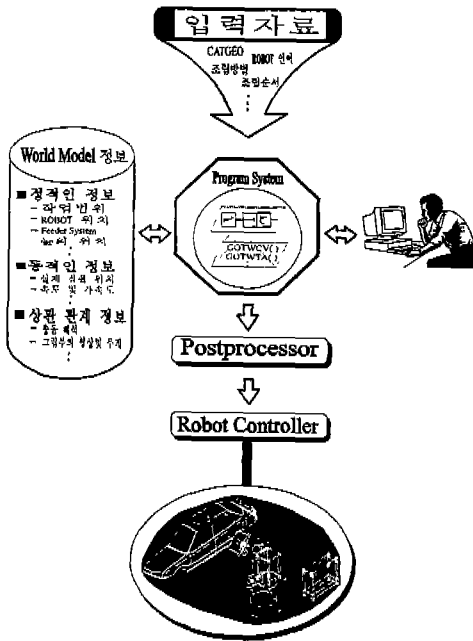


Fig. 3 Structure of the programming system for generation fo robot program

프로그래밍의 초기자료는 조립순서, 조립을 위한 접합 방법과 로봇 프로그램언어이다. 프로그램 시스템은 이들 자료와 생성된 주변 모델 정보의 도움으로 조립과제를 로봇 제어를 위한 언어로 전환한다. 시스템의 작업방법은 먼저 조립을 위한 로봇의 운동은 기능에 따라 분류되어 VDI - 표준 2860<sup>(17)</sup>의 조작 기능(Manipulation function)들로 기술된다. 다음 단계에서 조작 기능들은 그에 상응하는 로봇 프로그램언어로 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 프로그램이 가능하므로 원칙상 기존의 로봇 프로그래밍 시스템들의 Postprocessor는 필요치 않으나, 현장의 다양한 로봇에 응용을 위해 그 기능을 첨가

시키는 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.

공정으로 기술되는 로봇프로그램의 개발을 위해서는 로봇 동작의 분석과 이들을 해당 언어로의 전환은 필수적이다. 다음에 소개된 모델을 통하여 분석과 전환이 체계적으로 이루어지게 하였다.

### 3.2 로봇 동작 기술을 위한 모델

로봇 운동의 묘사는 조작 기능(Manipulation function)에 의존한다. 움직임의 각 동작은 정의된 기능을 가지고 기술될 수 있다. 이에 관련된 조작기능들은 Fig. 4에 나타내었다.

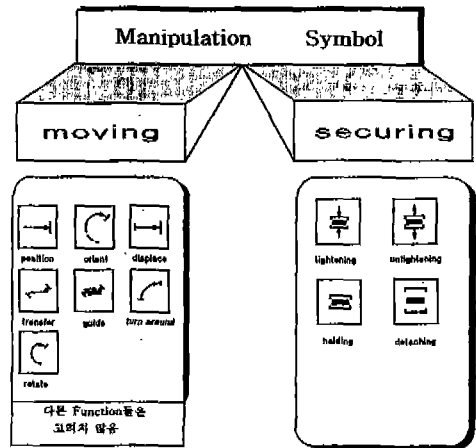


Fig. 4 Symbolic representation of manipulation functions according to VDI-2860

소개된 조작 기능들은 VDI 2860 취급 분야의 일부분인 Moving과 Securing에 속한다. Moving 중에서도 Arrange 와 Feeding은 고려치 않았다. 왜냐하면 이 기능들은 공급(Feeder) 시스템의 기술에 관여되고 로봇 운

등에는 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)을 로봇제어에 관련해서 기능적으로 각각 상세히 기술하면 Fig. 5에 보여진 조작기능들의 순서로 나타난다. 실제 접합이 일어나는 과정을 표시하기 위해 기호(Symbol) Jointing을 첨가했다.

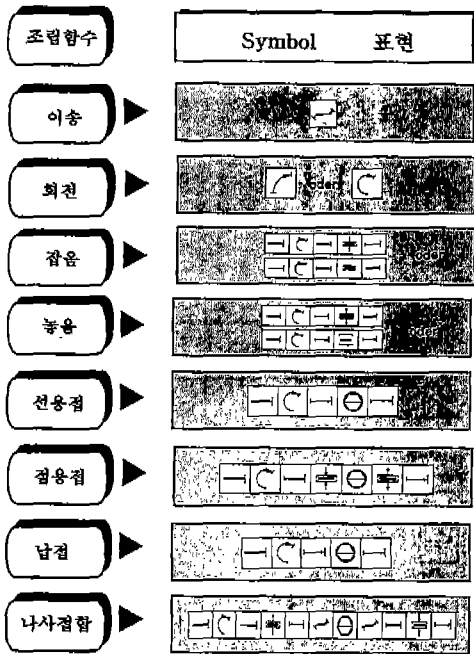


Fig. 5 Symbolic representation of assembly functions

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 구

성된다. Fig. 5의 잡음과 놓음의 아래 부분은 자력 Grip이나 진공 Grip의 사용시를 나타낸다. 로봇의 움직임은 과정이 같기 때문에 접합방법 끼움(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.

이와 같은 조작기능 기호로 조립 과정 및 Handling과정들의 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 시험기능 등을 첨가 함으로써 쉽게 확장가능하다.

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

### 3.3 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. 6에 일 예로써 앞에 소개된 조작 기능들을 수행하기 위한 명령어를 CATGEO로 나타내었다. 그들의 의미와 요구되는 변수들은 CATIA Geometry Interface Reference Manual<sup>(18,19)</sup>에 상세히 기술되어 있다.

Position과 Orient를 가지고 Robot는 정확한 위치에도달해야 하므로 속도를 감속해야 한다. 이를 위해 GOTWSP의 명령어를 사용했다. Transfer에서



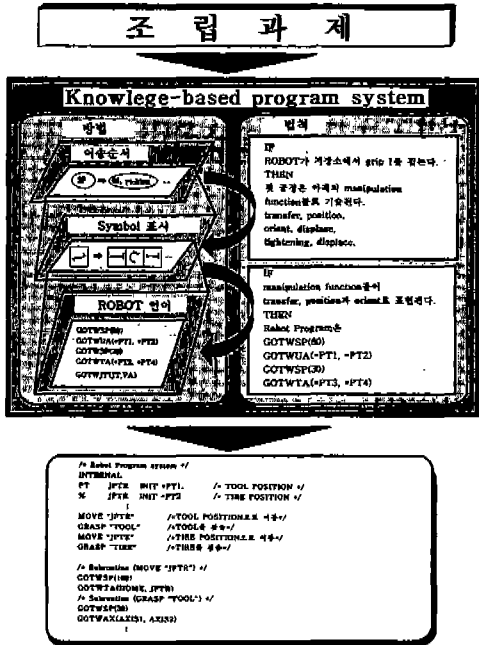


Fig. 8 Working method of the robot program system

- 주 프로그램
- 부 프로그램

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

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

#### 4. 응용사례

로봇 프로그래밍 범위에서 개발된 시스템을 차체, 타이어, 4개의 볼트를 가지고 차체에 타이어를 조립시키는 과제에 적용하여 보았다(Fig. 9).

조립이 기존 시스템에서 수행되느냐 혹은 새로운 시스템의 설계가 요구되느냐의 결정에 따라 조립과제에 최적의 시스템을 CATIA 여러기능의 도움으로 주변 모델화

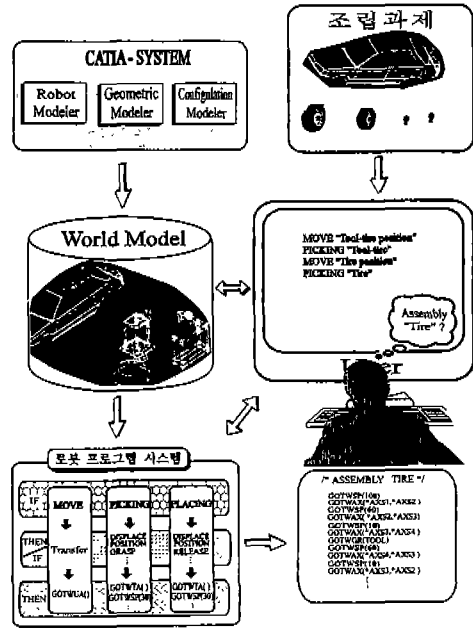


Fig. 9 Automatic generation of robot program

시킨다. 사용자는 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. 타이어 조립순서에 따라 요구되는 로봇의 동작에 의존해서 사용자가 로봇프로그램 생성을 위해 자연어와 유사한 조립함수와 대상물을 기술하면 로봇 프로그램은 앞절에서 보인 것처럼 두단계를 통해서 형성되어진다. 이 과정을 통해서 자연어처럼 기술하여 로봇 프로그램이 자동적으로 생성되도록 한다. 이것을 개발된 Program System 이 맡아서 수행한다.

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. 그러나 실제 조립셀을 설치하여 조립의 수행상태를 파악하지는 않았다. 이 부분은 본 연구를 바탕으로 기업에서 추진중에 있다. 현장조립에 있어

서는 모델상에서 계획된 점과 실제 로봇이 도달해야 될 점에서는 차이가 있다. 그 이유로는 프로그램 시스템이 모델링되어진 이상적인 주변 모델을 근거로 하여 이루어졌기 때문이다. 주변모델의 완전한 시뮬레이션은 모든 물리적인 현상을 다 고려할 수 없으므로 불가능하다. 그러므로 현장적용시에는 이 차이점은 교정을 통해서 보정되어야 한다.

주변모델의 모델링을 위해 이용되었던 CATIA-Robotics 사용의 첫 경험으로는 공구의 상세한 운동을 묘사하는데 어려움이 많았다. 나사체결 공구의 묘사에서 회전운동과 직선운동을 동시에 표현할 수 없었기 때문에 실제로 나사를 잠그는 거리만큼 회전운동을 한다는 가정 하에서 직선운동만으로 나타내었다. 또한 로봇 각 동작에 대해 서로 다른 SPEED로 정의를 하지만 실제로 Simulation상에서는 그 차이를 거의 느끼기 힘들었다. 그리고 종종 먼 거리를 움직일 때 이송중 로봇끝단의 공구방향이 뒤돌아쳐 원하는 위치에 도달하지 못하였다. 이 문제의 해결은 두 작업점의 경로상에 중간점을 지나도록함으로써 이루어졌다.

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

### 5. 결론 및 전망

로봇 프로그래밍은 점점 복잡해지고, 유연성에 대한 요구는 더욱 더 증가된다. 그 유연성은 사용되는 프로그래밍 방법에 크게 의존한다. 이러한 이유와 프로그래밍의 단순화 및 사용자의 편의성은 그래픽 시스템의 지원아래 공정 지향적인 로봇 프로그래밍 시스템의 개발 동기가 되었다.

많은 조립문제를 해석하여 로봇운동의 특성에 따라 조립에 요구되는 공정들을 분류하였다. 이 공정들을 수행하기 위한 로봇 운동의 정확한 분석과 이의 체계화는 본 시스템 개발을 위한 기초를 형성하였다. 개발된 프로그래밍 시스템의 수행능력은 차체에 타이어 조립과제에 적용되어 증명되었다. 이 현장응용력을 통하여 프로그래밍 시스템의 도움으로 로봇 프로그래밍에서 조립과제는 함축적인, 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. Hocken, R. and Morris, G., "An Overview of off - Line Robot Programming Systems", *Annals of the CIRP*, Vol. 35/2/, pp.495-503, 1986.
2. 조소형, 박강, 강성철, 김문상, "금형의 연마작업을 위한 로봇 오프라인 프로그래밍 시스템의 개발", *대한기계학회논문집*, 제 15권 4호, pp.1387-1397, 1991.
3. Constantinescu, V., "Flexible Fertigungszellen und Industrieroboter off - line programmieren", *ZwF*, Nr 4, pp.193-197, 1990.
4. Olschewski, V., "Off-line programmierung flexibel automatisierter Roboter", *VDI - Z*, Nr. 7, pp.60-65, 1990.
5. Nagata, T., "Research Tends in the Off-Line Programming of Robots", *J. Rob. Mech.* Vol. 1, PP. 278-283, 1989.
6. Gerhard, K., "CAD/CAP : Rechnergestuetzte Montagefeinplanung", *Carl Hanser Verlag, Muenchen*, 1990.
7. N. N., "CATIA SOLIDS GEOMETRY Version 3 : Interactive Functions Reference Manual Third edition", *Copyright Dassault Systemes*, December, 1990.
8. Freund, E., Heck, H., Kreft, 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.
9. Bernhardt, R., Dillman, R. and Tierney, K., "Integration of Robots into CIM", *Chapman & Hall*, London, 1992.



10. Rondeau, J. M. and Elmaraghy, H. A., "Roboter programming and task planning", *Manufacturing Review*, Vol. 3, pp.245-250, 1990.
11. Honda, T., Schunichi, K., Funjiie, Y. and Hage, S. "A Flexible Robot Control System with Both Task-Level-Language and Visual Monitoring", *Annals of the CIRP*, Vol. 37/1/, 1988.
12. Weeks, J. K., "Aufgabenorientierte Roboterprogrammierung fuer die flexibel automatisierte Montage", *Industrie-Anzeiger*, Nr. 13, pp. 32-33, 1988.
13. Nnaji, B. O. and Rist, A., "Automatic Robot Programming in Computer Integrated Manufacturing", *Society of Manufacturing Engineers*, MS 89-331, 1989
14. Pan, T.-J. and Luo, R.C., "An AI-based Robot Program Generation System for Assembly Automation", *SPIE*, Vol. 1193, pp. 187-197, 1989.
15. Franzmeier, N., "The Automatic Robot Program Generation System," *Society of Manufacturing Engineers*, MS89-318, 1989.
16. Asada, H. and Izumi, H., "Automatic Program Generation from Teaching Data for the Hybrid Control of Robots", *IEEE Journal of Robotics and Automation*, Vol 2, No. 2, pp. 166-173. 1989.
17. N. N., "VDI-Richtlinie 2860 : Handhabungsfunktionen, Handhabungseinrichtungen Begriffe, Definition, Symbole", *VDI-Verlag*, Duesseldorf, 1990.
18. N. N., "CATIA BASE Version 3 : Geometry Interface Reference Manual Fourth edition", *Copyright Dassault Systemes*, May, 1991.
19. N. N., "CATIA ROBOTICS Version 3 : Geometry Interface Reference Manual Third edition", *Copyright Dassault Systemes*, December, 1990.