

금형의 자동 연마 System 개발

Development of Automatic Die Polishing System

연구기관
한국과학기술연구원

과 학 기 술 처

제 출 문

과학기술처 장관 귀하

본 보고서를 "금형의 자동연마 시스템 개발" 사업의 세부과제인 "금형의 연마 작업을 위한 Off-line Programming System 개발" 과 "로봇를 이용한 금형의 연마작업 자동화 기술개발" 연구의 연차보고서로 제출합니다.

1992 년 10 월

주관연구기관 : 한국과학기술연구원

총괄연구책임자 : 강 무진

연구 책임자 : 김 문상 (로봇응용 및
유공압 연구실
선임 연구원)

박종오 (로봇응용 및
유공압 연구실
선임 연구원)

연구 원 : 강 성철 (로봇응용 및
유공압 연구실
연구 원)

김미경 (로봇응용 및
유공압 연구실
연구원)

엄두간 (로봇응용 및
유공압 연구실
연구원)

강효식 (로봇응용 및
유공압 연구실
연구원)

송태오 (로봇응용 및
유공압 연구실
선임 기사)

노덕문 (CAD/CAM
연구실
선임 기사)

**금형의 연마작업을 위한 로봇 Off-line
Programming System 개발**

**Development of Robot Off-line Programming
System for Die Polishing**

여 백

INDEX

1. INTRODUCTION	15
2. DEVELOPMENT OF SYSTEM (PROPS)	19
3. DEVELOPMENT OF COLLISION CHECKER	38
4. INVERSE KIEMATCS ALGORITHM	44
5. PROPS DATA BASE	63
6. CONCLUSION	84
REFERENCES	86
APPENDICES A. DATA BASE ROUTINE	89
B. AutoCAD DXF FILE IMPORT	94

여 백

목 차

제 1 장	서 론	15
제 2 장	SYSTEM (PROPS) 의 개발과 구성	20
제 3 장	COLLISION CHECKER의 개발	38
제 4 장	INVERSE ALGORITHM 의 구성	44
제 5 장	PROPS DATA BASE	63
제 6 장	결 론	84
	참고 문헌	86
	부 록 A. Data base Routine	89
	B. AutoCAD DXF file 의 import	94

여 백

요 약 문

I. 제 목

금형의 연마 작업을 위한 Off-line Programming System의 개발

II. 연구개발의 목적 및 중요성

금형 가공의 마무리 작업 중 사상 (polishing) 작업에 로봇을 이용하여 전문적인 지식이 없는 사람도 쉽게 운용할 수 있도록 자동화 함으로써 공정의 고속화를 꾀하는 방안이 제시되었다.

본 연구는 3차원 자유곡면형상을 대상으로 하는 로봇 작업 프로그램을 CAD 시스템을 이용하여 얻어진 위치 (position) 및 방향 (orientation) 데이터를 통해 열외 (Off-line) 적인 방식으로 구성하는 방법에 관한 내용으로, 금형설계 데이터를 로봇 기술에 이용하여 금형생산 라인에 집적시키는데 그 목표를 두고 있다.

즉 금형, 로봇, 공구를 편리하게 선정하여 실제 가공과 상사성을 갖는 시뮬레이션 (simulation) 을 컴퓨터상에서 손쉽게 수행함으로써 충돌이 배제되고 관절제한에 위배되지 않는 경로 (path) 를 따라 이동하는 로봇의 작업 프로그램을 생성하여 제어기 (controller) 에 전송하는 열외 시스템 (Off-line System) 의 개발을 주 목적으로 한다.

이러한 기능을 갖춘 PROPS의 개발로 인해서 사상 작업에 실제 로봇을 사용할 시의 교시 (teach-in) 작업으로 인한 문제점을 해결하여 경비의 절감, 시간의 단축등을 꾀할 수 있게되었다.

III. 연구개발의 내용 및 범위

지난 3차년도까지 PROPS 를 구성하는 기본기능들 즉, 금형과 각 구성장치의 modeling, 로봇의 기구학 modeling, 금형의 정치기능, tool path generation 기능, 가공 animation 및 collision checker 기능 그리고 작업 program 전송을 위한 통신 모듈 등이 완성되었다. 이번 4차년도 보고서는 3차년도까지 구축된 기능들을 기반으로하여 사용자 interface 를 지향하는 연구가 다음과 같이 진행되었다.

- i) CATIA surface modeller 의 surface data interface 기능
- ii) Cell을 구성하는 각 장치의 modeling 을 위한 Autocad DXF file import 기능
- iii) Cell 구성을 위한 각 장치의 선정 및 layout planning 기능
- iv) 작업 simulation 에 적합한 teaching pendant 의 보완
- v) RDB 의 효율화 및 interface 기능의 강화
- vi) RDB 관리에 적합한 menu 구조의 개선

IV. 연구개발 결과 및 활용에 대한 건의

- (1) 실제 robot system에서의 polishing 작업을 미리 computer상에서 graphic system을 이용한 simulation 기법으로 수행하여 실제 robot를 사용할 시의 teach-in 작업으로 인한 문제점을 해결하여 경비의 절감, 시간의 단축등을 피할 수 있게되었다.
- (2) RDB 를 활용하여 layout, toolpath, simulation 으로 구성되는 'task' 를 하나의 단위로하는 작업이 가능해졌고, 따라서 task를 정의하고 저장,수정하는 일이 가능해졌다.
- (3) collision checker의 개발로 인해서 실제 polishing시 일어날 수 있는 공구의 간섭 문제를 해결할 수 있는기반이 마련되었다.
- (4) 화면상에 teaching pendant를 출현시키고 관절운동의 한계 (joint restriction)를 check 하고 로봇트 작업 경로 혹은 조건을 수정할 수 있게 하였다.
- (5) 본 system을 이용하여 생성된 로봇트 작업 데이터를 postprocessor 를 이용하여 처리한 후 실제 로봇트에 전송함으로써 실질적인 integration이 실현되었다.
- (6) CATIA modeller 로부터 surface data 의 import 기능을 추가하여 외부 modeller 와의 호환을 가능하게 하였다.
- (7) Part modelling 기능의 호환성을 높이기 위해 box 와 cylinder 형상에 대한 Autocad DXF file 의 import 기능을 추가하였다.

로봇트를 이용하여 금형의 사상 작업을 자동화할 경우 제품의 형태가 복잡하고 다양해지면 티이칭 펜던트를 이용한 기존의 교시 방법은 많은 작업 시간을 요하게되고 부정확해진다. 본 시스템의 개발

로 인해 사용자는 CAD system과의 interface를 통해 다양한 금형 형상과 작업 환경에 유연하게 대처할 수 있게 되었으며 총 작업 소요 시간, 관절 제한 조건, 충돌등의 해석이 가능한 시뮬레이션을 통해 실제 가공시 발생할 수 있는 오류를 최대한으로 줄일 수 있게 되었다. 실제 polishing robot system 및 기존 CAD package와의 좀더 유기적인 통합을 위하여서는 가공작업에 대한 RDB의 구축, tool 및 주변장치에 대한 세밀한 simulation 그리고 효율적인 preprocessor 즉 외부 CAD modeler와의 interface 기능, 그리고 현장에서 사용되고 있는 다양한 로봇트에 대한 작업 program generator 기능의 구축이 요구된다.

SUMMARY

As a part of the national project "Application of CIM technology to design and manufacturing of dies and molds", a die-polishing system using industrial robot is being developed. An offline programming system for the automatic generation of the robot working program is essential to perform the die polishing procedure. So, we have developed a polishing robot off-line programming system (PROPS), which is an object oriented programming package.

PROPS gets the surface data from the surface modeler and automatically generates the robot path. In this report the main functions and structures of PROPS are introduced. PROPS consists of several modules for simulation as follows. Those are selection of robot system and die surface, layout planning of the die surface within robot workspace, generation of the robot moving path, displaying of error messages such as the joint restriction of the robot kinematics and collision between tool and die and calculation of the cycle time.

As a result of execution of such modules the robot motion program is completed using the tool path and the motion parameters. And the robot working program is transferred to the robot controller through the robot interface.

PROPS executes animation using general inverse kinematic

package which adopts iteration method and uses a relational database in order to get the whole data and to generate the robot working program.

제 1 장 서 론

최근 금형 가공의 마무리 작업 중 사상 (polishing) 작업에 로봇을 이용하여 전문적인 지식이 없는 사람도 쉽게 운용할 수 있도록 공정을 자동화함으로써 가공의 고속화를 꾀하는 방안이 제시되고있다.

현재 로봇 프로그램의 열외적인 생성방식에 의한 프로그래밍 기술은 페인팅 (painting), 코팅 (coating), 납땜 및 조립, 사상과 같은 분야에서 좋은 결과를 얻고 있다. 이러한 분야들은 대부분 작업의 형태가 매우 복잡하여 로봇 시스템을 활용하여 자동화할 경우 기존의 프로그램방법인 교시 (Teach-in) 방법을 채택하기가 곤란하다. 이는 작업의 형태가 매우 난해하기 때문에 프로그래밍에 많은 시간이 소요될 뿐아니라 정확한 작업 데이터의 형성도 곤란하다. 예를 들어 페인팅, 코팅 그리고 사상 등은 작업 대상이 3차원 곡면이어서 이러한 곡면을 따라 무수한 작업 프로그램을 얻기는 거의 불가능하다고 볼 수 있다. 이러한 자유곡면을 따라서 정확한 구동조건을 결정하는 가능한 방법중의 하나가 기존 설계에 사용된 CAD 데이터를 활용하여 열외적으로 얻어내는 방법이다.

본 연구는 3차원 자유곡면형상을 대상으로 하는 로봇의 사상 작업 프로그램을 CAD 시스템으로부터 얻어진 위치 (position) 및 방향 (orientation) 데이터를 통해 오프라인 (Off-line) 적인 방식으로 생성하는 방법으로, 금형설계 데이터를 로봇 가공에 직접적으로 이용하여 금형생산 공정 중 사상 과정을 자동화하는데 그 목표를 두고 있다.

전체적인 구성은 금형, 로봇, 공구를 사용자가 메뉴 (menu) 를 이용하여 편리하게 선정 후 실제 가공과 유사성을 갖는 시뮬레이션을 수행함으로써, 충돌이 배제되고 관절제한에 위배되지 않는 로봇 작업 경로를 만든 후 이 경로를 추적하는 로봇용 작업 프로그램을 생성하여 로봇 제어기내의 code로 변환한 후 제어기에 전송하는 것으로 요약될 수 있다.

그림 1은 Polishing Robot Off-line Program System (PROPS)의 개략적인 구성을 나타낸다.

보고서는 시스템 (PROPS)의 주 메뉴 (Main menu)로 구성되어 있는 자료 입력, 금형의 정치, 가공경로의 생성, 시뮬레이션, 후처리 (postprocessor) 등의 각 모듈 (module)을 중심으로 서술되어 있으며 이러한 모듈들의 효과적인 수행을 위해서 다음과 같은 기능들이 개발되었다.

- 1) RDB 관리에 적합한 menu 구조의 개선
- 2) CATIA surface modeller 의 surface data interface 기능
- 3) Cell을 구성하는 각 장치의 modeling 을 위한 Autocad DXF file import 기능
- 4) Cell 구성을 위한 각 장치의 선정 및 layout planning 기능
- 5) 생성된 로봇용 작업 경로를 따라 로봇 시스템의 애니메이션이 수행되는 그래픽 시뮬레이션 프로그램 (graphic simulation program)
- 6) 사상 공구와 대상 금형간의 충돌을 사전에 검증하는 routine의 개발
- 7) 작업 simulation 에 적합한 teaching pendant 의 보완
- 8) 방대한 데이터의 효율적인 관리를 위한 릴레이셔널 데이터 베이스 (Relational Database)의 구축 및 interface routine 의 강화
- 9) 각 로봇 제어기로의 interface를 위한 후처리 (Postprocessor) 기능 및 통신 모듈 (communication module)의 개발

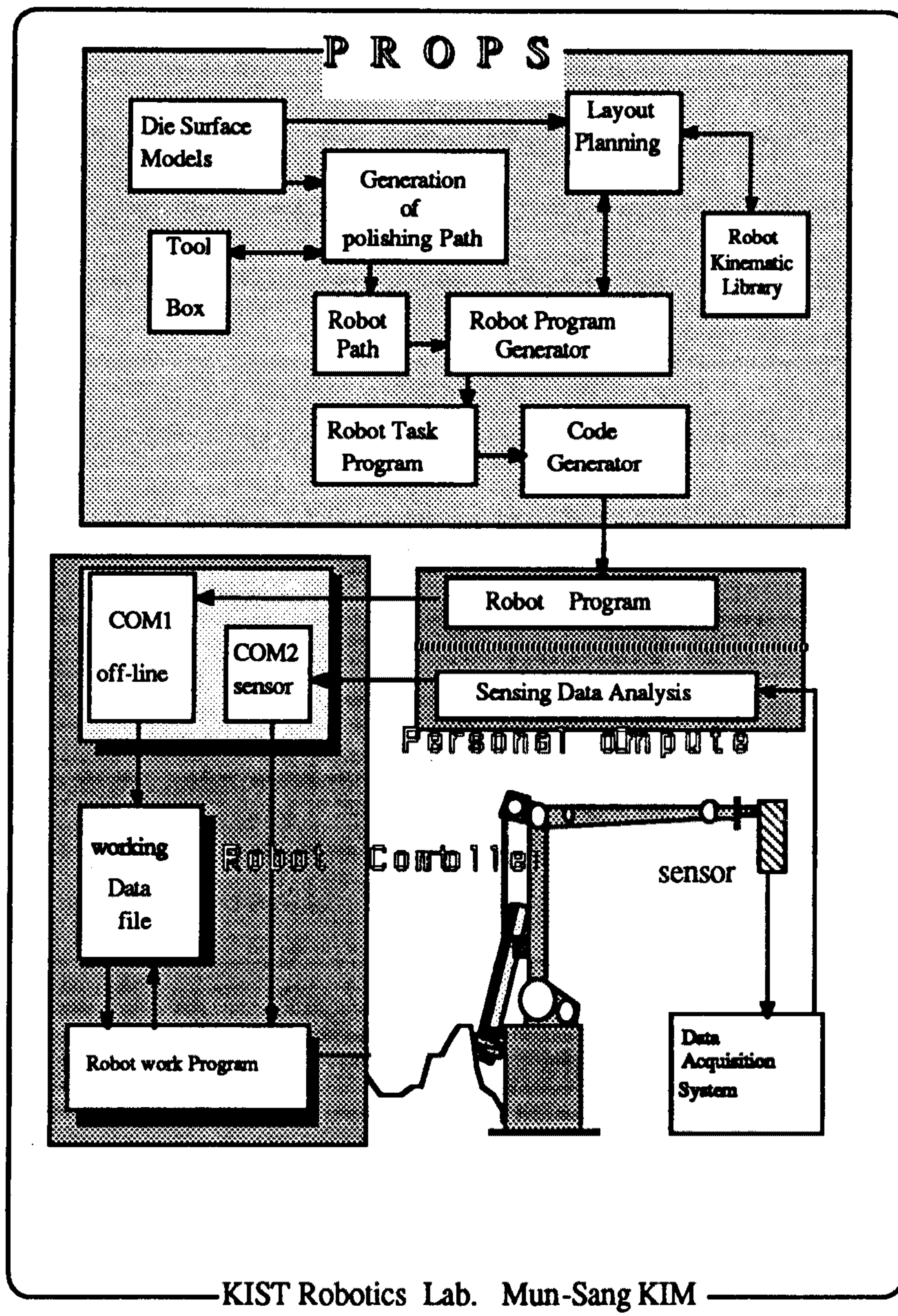


그림 1.1 PROPS와 controller

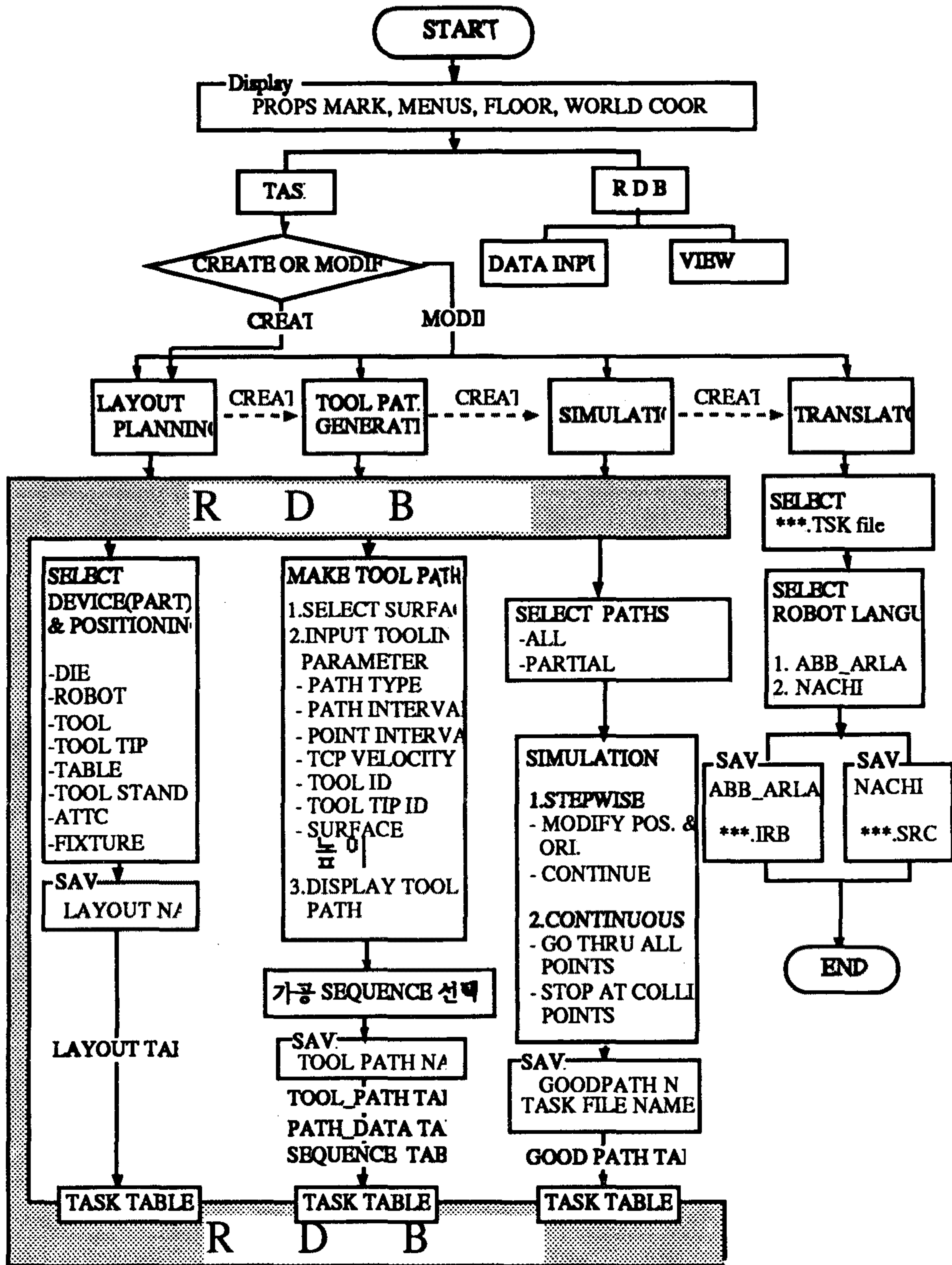


그림 1.2 Flow Chart of PROPS

제 2 장 System(PROPS) 의 개발과 구성

- . Work Station : IRIS 4D series
- . Graphic Package : FIGARO
- . Program Language : FORTRAN
- . Data base : Informix SQL
- . DB query language : ESQL/C

PROPS는 simulation을 수행하기 위하여 main menu를 중심으로 하는 몇 개의 module로 구성되어 있다. 2장의 각 절은 이러한 module에 대한 설명으로 이루어져 있다.

제 1 절 PROPS Main Menu

PROPS main menu 는 TASK, LAYOUT, TOOLPATH, SIMULATION, TRANSLATOR 그리고 RDB 총 6 개로 이루어져 있다.

- . 위의 6 개의 menu 중에서 task, RDB menu 가 최 상위 level 이고, 그 하위에 layout, toolpath, simulation, translator menu 가 존재하게 된다.
- . 각 main menu는 신규 task 인 경우 task, layout, toolpath, simulation 의 순서대로 , 기존의 task 를 수정하는 경우 독립적으로 수행이 가능하다.
- . View transformation menu 는 permanent menu 로서 언제나 수행이 가능하다.

1. TASK MENU

Task 는 PROPS 에서 구현되는 하나의 작업 단위로서 layout, toolpath, simulation 으로 구성된다. PROPS 사용자는 연마작업을 위한 simulation 을 행할 때, 하나의 task를 RDB 에 등록하여야 한다. 이는 차 후에 그 task 가 포함하는 layout, toolpath, simulation 작업들을 생성하거나, 수정하기 위함이다.

Task menu 는 PROPS system 의 최상위 menu 로서, PROPS 작업의 기본 단위인 task 를 생성,수정,저장하는 menu 이다. 그 밖에 이 menu 는 최상위 menu 로서 필요한 '화면 지우기', '작업 종료'의 menu 를 포함한다.

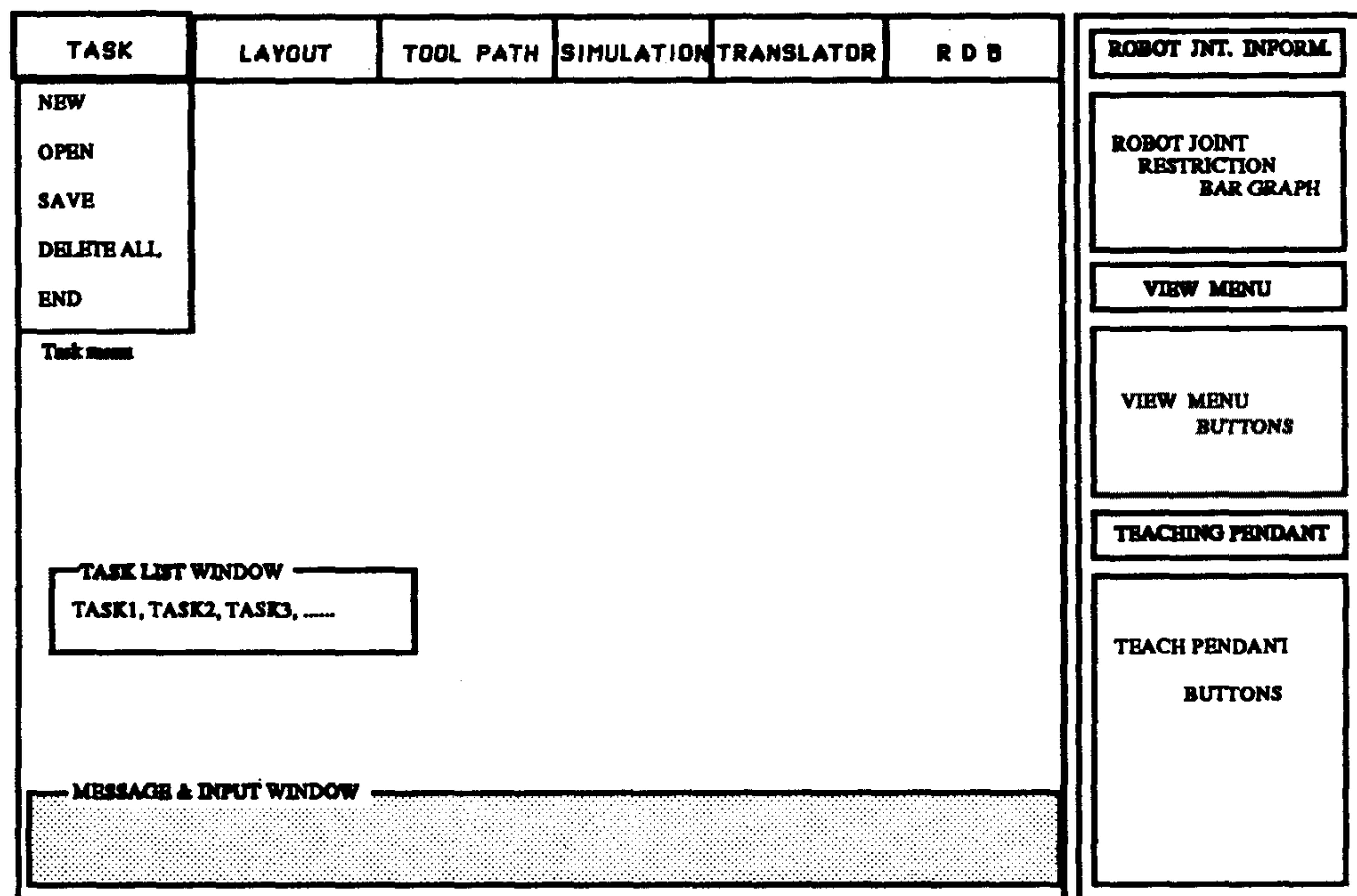


그림 2.1 Task menu

2. Layout menu

Layout menu 는 금형 연마작업을 위하여 필요한 모든 구성요소를 선택하여 정확한 위치에 배치하는 menu 이다.

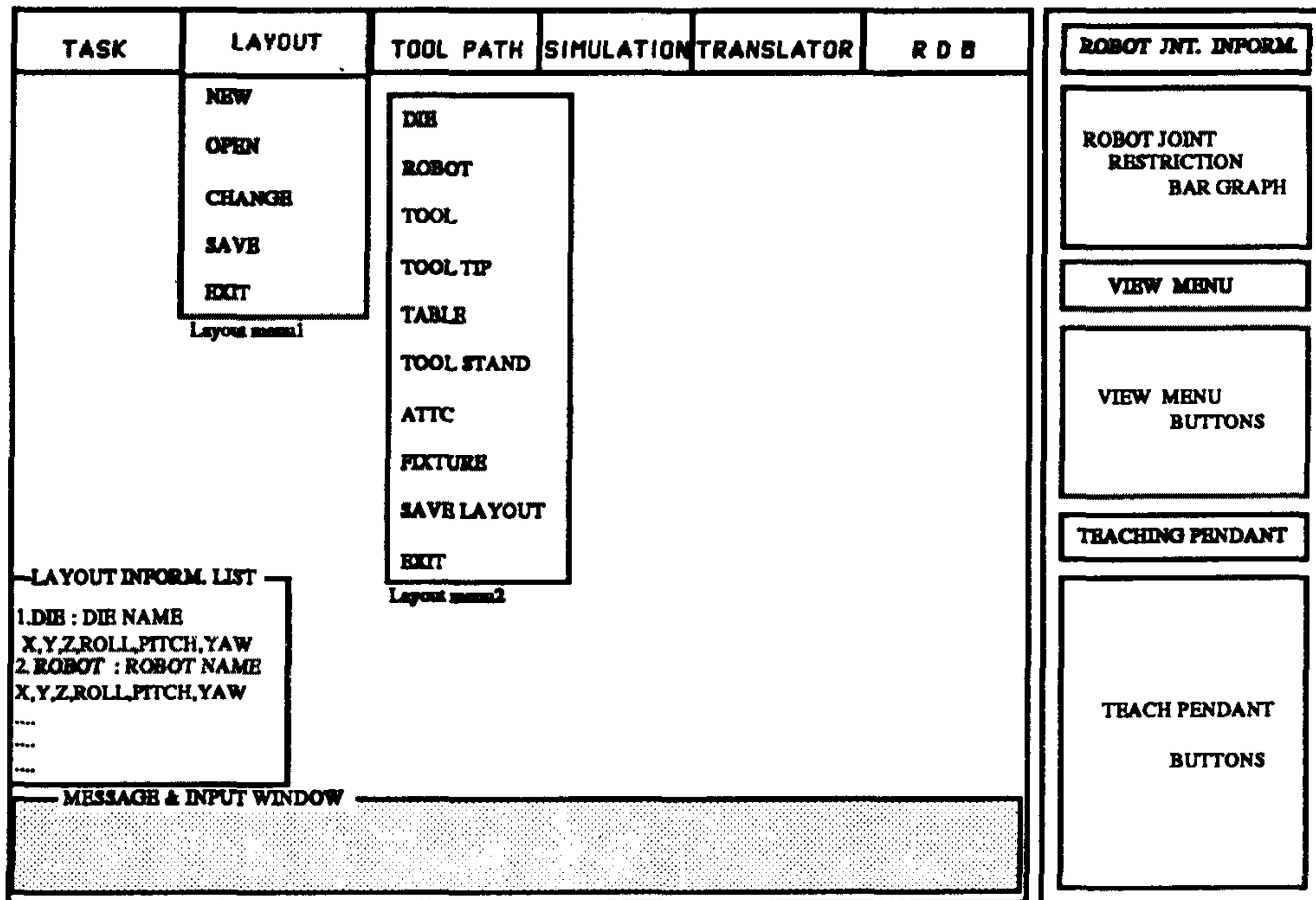


그림 2.2 Layout menu

2.1 Layout menu1

Layout menu1 은 다음과 같은 button 으로 구성되어 있다.

1. NEW : Layout level 에서의 독립적인 작업을 가능하게하기 위해서 새로운 layout 을 생성하고 RDB 에 등록한다.

Layout menu2 가 나타난다.

2. OPEN : RDB 에 이미 저장되어 있는 layout 을 불러들인다.

Layout menu2 가 나타난다.

3. CHANGE : 현재 선택되어 있는 task 가 포함하는 layout 의 내용을 변경한다. Layout menu2 가 나타난다.
4. SAVE : 작업이 완료된 layout 을 RDB에 저장한다.
5. EXIT : Main menu 로 이탈.

2.2 Layout menu2

실제로 작업에 필요한 장치를 선정하여 RDB 에서 가져오고, 기준좌표계에 대하여 정확한 위치에 배치시키는 menu이다. 다음은 layout 을 구성하는 각 part 의 button 이고, 모든 part는 선택한 후에 x,y,z,roll,pitch,yaw 값을 입력하여 정치한다.

1. DIE : RDB의 die table 에서 die geometry data 를 가져와서 화면에 그린다.
2. ROBOT : RDB의 robot table 에서 robot의 geometry data 를 가져와서 화면에 그리고, robot 의 기구학을 정의한다.
3. TOOL : RDB의 tool table 에서 tool geometry data 를 가져와서 화면에 그린다.
4. TOOL TIP : RDB의 tool tip table 에서 tool tip geometry data 를 가져와서 화면에 그린다.
5. TABLE : RDB의 table table 에서 table geometry data 를 가져와서 화면에 그린다.
6. TOOL STAND : RDB의 tool stand table 에서 tool stand geometry data 를 가져와서 화면에 그린다.

7. ATTC (Automatic Tool Tip Changer) : RDB의 attc table 에서 attc geometry data 를 가져와서 화면에 그린다.

8. FIXTURE : RDB의 fixture table 에서 fixture geometry data 를 가져와서 화면에 그린다.

3. Toolpath menu

Layout 에서 선택된 금형의 surface 중에서 가공면을 선정하고, 그 가공면에 가공 조건을 입력하여 path 를 생성하고, 이것이 여러 개인 경우 그들을 조합하고 가공 순서를 지정하여 하나의 독립적인 toolpath 를 생성하는 menu이다.

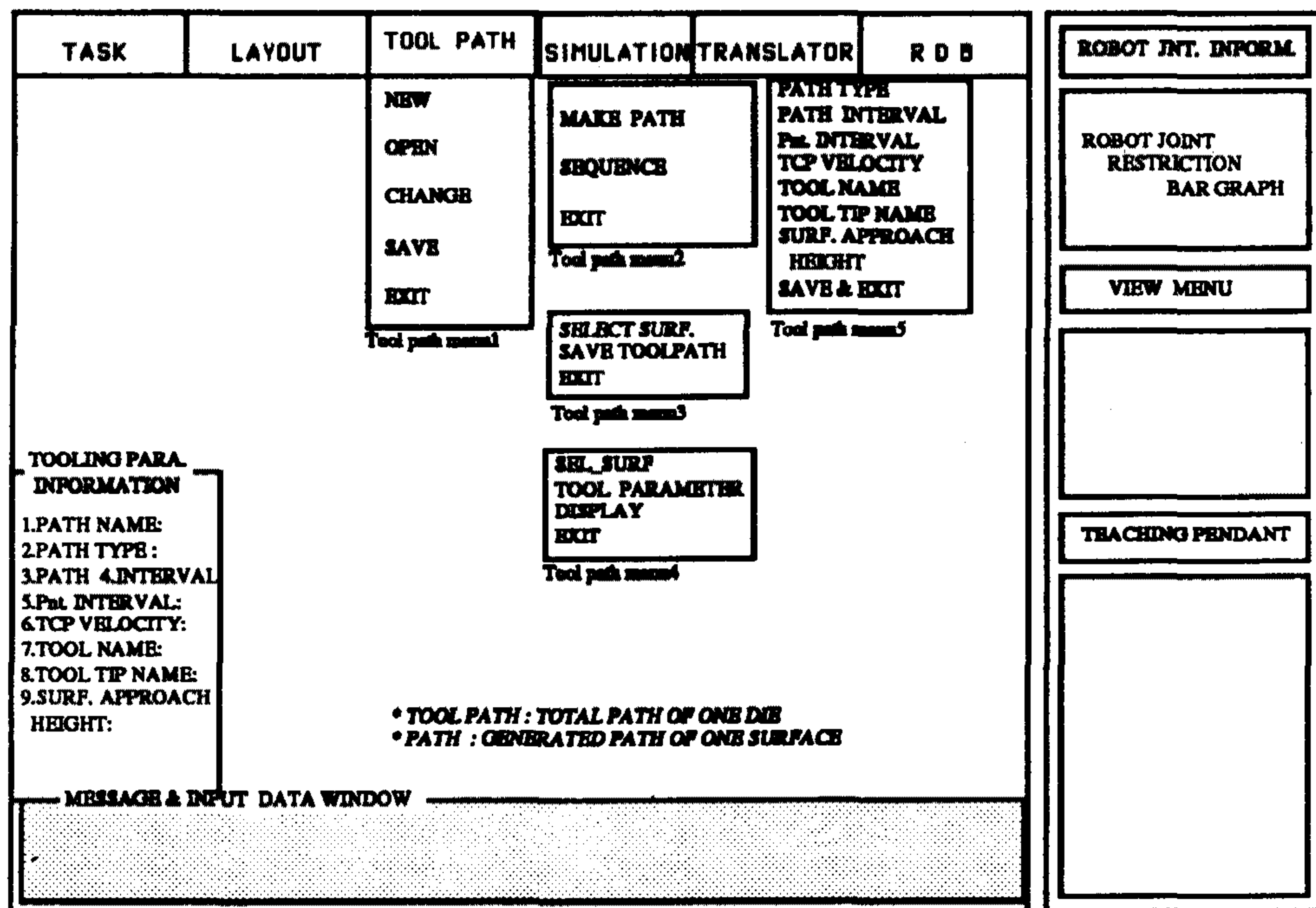


그림 2.3 Tool path menu

3.1 Tool path menu1

Tool path menu1 은 toolpath level 의 독립적인 작업을 가능하게 하기 위해서 layout menu1 과 같은 성격의 기능을 갖는 NEW, OPEN, CHANGE, SAVE, EXIT 5 개의 button 으로 이루어져 있다.

3.2 Tool path menu2

1. MAKE PATH : 화면 상에서 mouse 를 이용하여 가공할 surface 를 선택하고, 필요한 가공 조건을 입력하여 한개의 surface 에 path 를 만드는 menu 이다. toolpath menu4 가 출현한다.
2. SEQUENCE : MAKE PATH menu 에서 생성된 path 들의 가공 순서를 지정한다. toolpath menu3 가 출현한다
3. EXIT : Main menu 로 이탈.

3.3 Tool path menu3

1. SELECT SURF. : Sequence를 지정하고자하는 path 들을 순서대로 택한다.
2. SAVE TOOLPATH : 완성된 toolpath 를 RDB에 저장한다.

3.4 Tool path menu4

1. SEL_SURF : Path 를 만들고자하는 surface 를 mouse pick 으로 선택한다.
2. TOOLING COND. : 가공조건을 입력한다. Toolpath menu 5 가 출현한다.
3. DISPLAY : 생성된 toolpath를 화면에 보인다.

3.5 Tool path menu5

1. PATH TYPE : Path type (Oneway or Return)을 지정한다.
2. PATH INTERVAL : Path 간 간격(mm) 을 입력한다.
3. POINT INTERVAL : Point 간 간격(mm) 을 입력한다.
4. TCP VELOCITY : TCP 속도 (mm/sec) 를 입력한다.
5. TOOL NAME : 가공에 사용되는 tool name 을 지정한다.
6. TOOL TIP NAME : 가공에 사용되는 tool tip name 을 지정한다.
7. SURF. APPROACH HEIGHT : TCP 의 surface 진입 높이를 입력한다.
8. SAVE & EXIT : 지정된 가공 조건을 RDB 에 저장하고 toolpath menu4 로 이탈.

4. Simulation menu

Layout 과 toolpath menu 에서 cell 을 구성하여 가공 animation을 수행하고, 실제 가공 시 발생할 수 있는 문제, 즉 Tool 및 tooltip change 문제, 가공 sequence 문제, tool 과 금형간의 collision 문제, robot joint restriction 문제, 가공 시간 계산 등을 검사하고, teaching pendant 를 이용하여 로봇트 작업경로 data 의 삽입,수정,삭제를 통하여 궁극적으로 goodpath data 를 생성하는 menu 이다.

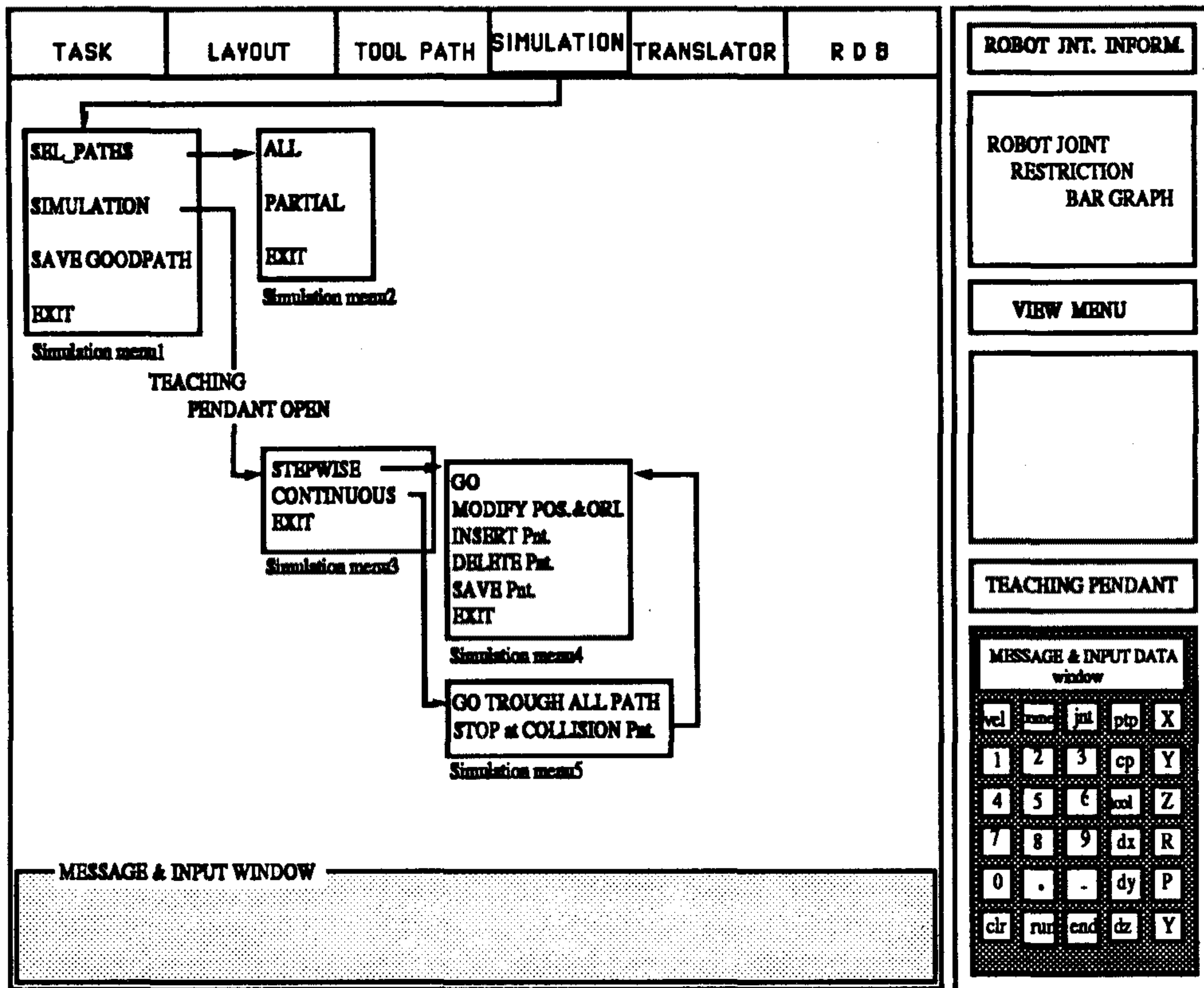


그림 2.4 Simulation menu

4.1 Simulation menu1

1. SEL_PATH : Simulation 을 수행할 path 를 선택한다.
Simulation menu 2 가 출현한다.
2. SIMULATION : 가공 animation 을 수행한다. Simulation menu 3 가 출현한다.
3. SAVE GOODPATH : Simulation 과정에서 생성된 goodpath 를 RDB 에 저장한다.

4.2 Simulation menu2

1. ALL : 지정되어 있는 toolpath 의 모든 path 를 선택한다.

2. PARTIAL : 지정되어 있는 toolpath 의 일부 path 를 선택한다.

4.3 Simulation menu3

1. STEPWISE : 각 point 마다 정지하는 animation 을 수행한다. 정지 후 Simulation menu4 가 출현한다.

2. CONTINUOUS : 연속적인 animation 이 수행된다. Simulation menu5 가 출현한다.

4.4 Simulation menu4

1. GO : 계속 animation 을 수행한다.

2. MODIFY POS. & ORI. : Teaching pendant 를 이용하여 현재의 위치와 방향을 변경한다. Teaching pendant 가 출현한다.

3. INSERT Pnt. : Teaching pendant 를 이용하여 원하는 위치와 방향으로 교시한 후 그 점을 RDB 의 goodpath data table 에 현 위치의 전 혹은 후에 삽입 저장한다. Teaching pendant 가 출현한다.

4. DELETE Pnt. : 현재 point 를 삭제한다.

5. SAVE Pnt. : 변경된 point를 RDB 에 저장한다.

4.5 Simulation menu5

1. GO THROUGH ALL PATH : Collision check 없이 계속적인 animation 을 수행한다.

2. STOP at COLLISION Pnt. : Collision 이 일어날 때까지 animation 을 수행한다.

5. Translator menu

Simulation menu를 수행하면 goodpath point data 가 생성되어 task file 로써여진다. Translator menu는 이를 이용하여 로봇트가 이해할 수 있는 언어로 번역하여 로봇트 program 을 생성한다. 로봇트 언어는 일반화되어 있지 않기 때문에 각 로봇트마다의 특정한 translator 를 개발하여야 한다. PROPS 는 ABB robot 의 ARLA 언어와 NACHI robot 의 SRC file 로 번역하는 translator 를 보유하고있다.

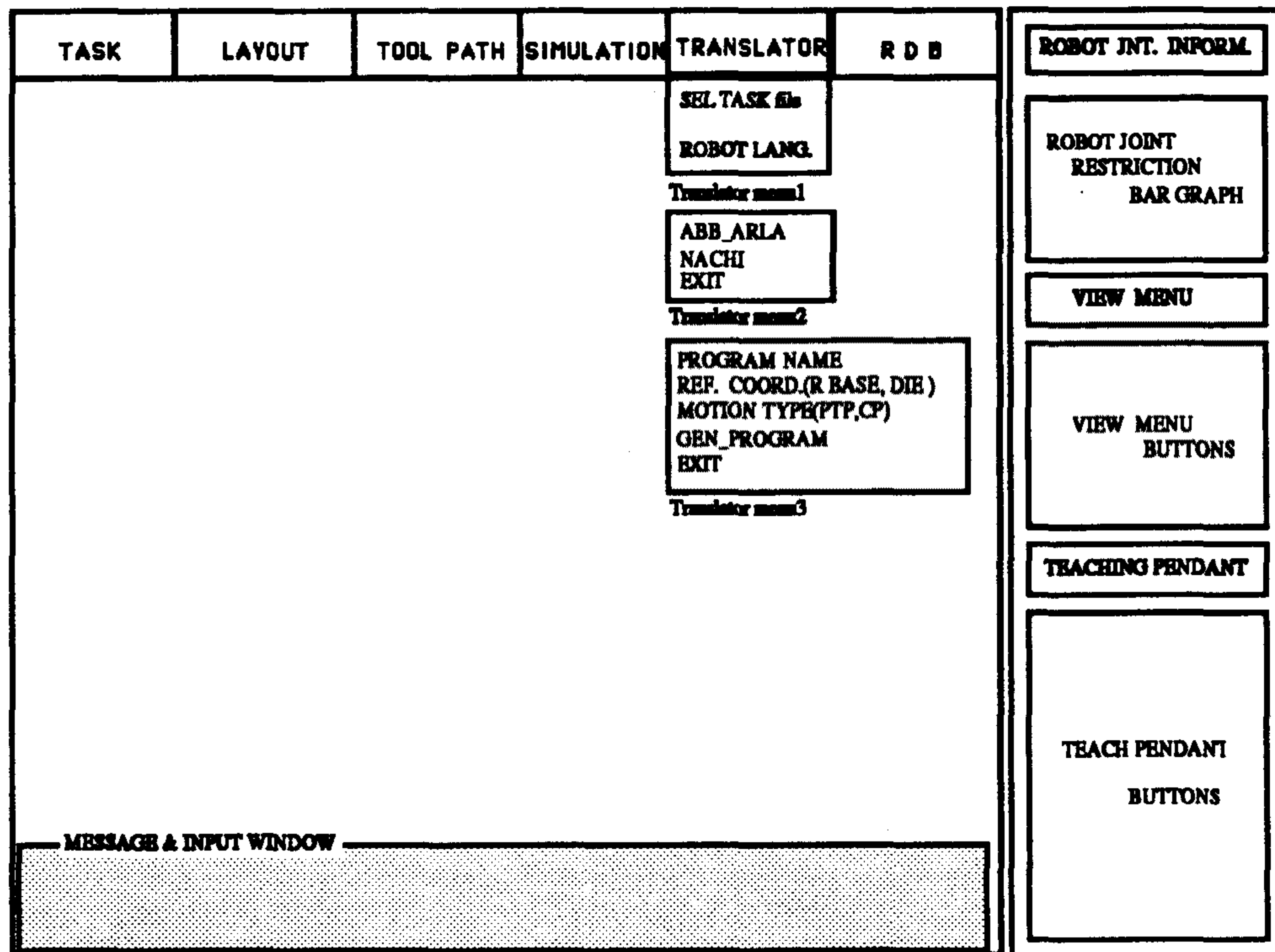


그림 2.5 Translator menu

5.1 Translator menu 1

1. SEL_TASK file : Simulation 이 완료되어 생긴 .task file 을 선택

한다.

2. ROBOT LANG. : Translator 가 보유하고 있는 언어의 해당 robot 이름을 선택한다. Translator menu2 가 출현한다.

5.2 Translator menu2

1. ABB_ARLA : ABB robot 의 ARLA language 를 선택한다.

Translator menu3 가 출현한다.

2. NACHI : NACHI robot language 를 선택한다.

Translator menu3 가 출현한다.

5.3 Translator menu3

1. PROGRAM NAME : Robot program name 을 입력한다.

2. REF. COORD. : 기준 좌표계(robot base or die base)를 지정한다.

3. MOTION TYPE : TCP 의 운동 보간 방식 (ptp or cp) 를 지정한다.

4. GEN_PROGRAM : Robot program 을 생성한다. ABB 인 경우 ***.IRB file, NACHI 의 경우 ***.SRC file 이 생성된다.

6. RDB menu

PROPS 환경에서 CAD data 를 import 하는 기능과, RDB table 에 저장되어 있는 주요 data 를 검사하는 menu 이다. CAD data import 기능에서, die 인 경우 CATIA modeller data format 을 준수하고있으며, 나머지 part 의 경우 AUTOCAD DXF format 을 따르고 있다. 이는 PROPS 가 일반화된 geometry modeling 기능을 갖고있지 않기 때문이다. 따라

서 PROPS 는 외부 CAD modeller 와의 interface module 을 가지고 있다.
 Data view 는 사용자가 PROPS 환경에서 task, layout, toolpath, goodpath, robot 등의 주요 data 를 검사하는 menu 이다.

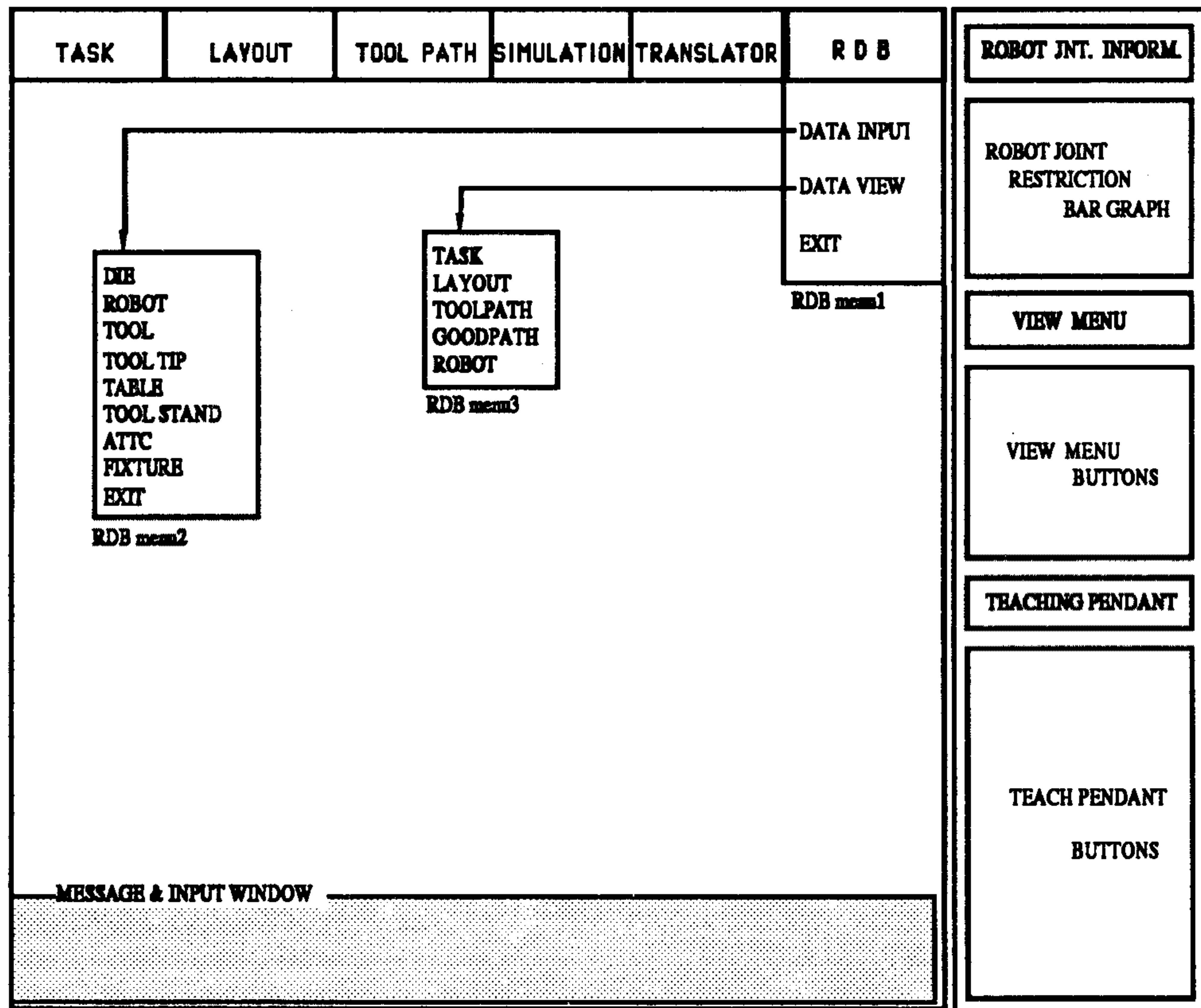


그림 2.6 RDB menu

6.1 RDB menu1

1. DATA INPUT : Die 를 비롯한 7 개의 part geometry data 를 import 하는 menu 이다. RDB menu2 가 출현한다.
2. DATA VIEW : 주요 data table의 내용을 검사하는 menu 이다. RDB menu3 가 출현한다.

6.2 RDB menu2

1. DIE : CATIA format 의 data file 을 import 하여 RDB die table 에 등록, 저장한다.

2. ROBOT : Autocad DXF file 을 import 하여 RDB robot table 에 등록, 저장한다.

TOOL 부터 FIXTURE 까지의 menu 는 robot 와 같은 방법이다.

6.3 RDB menu3

1. TASK : Task가 작성된 일시, task name, 하나의 task 가 가지는 layout name, toolpath name, goodpath name 등을 보인다.

2. LAYOUT : Layout 이 작성된 일시, layout 이 가지는 각 part 의 name, position 을 보인다.

3. TOOLPATH : toolpath 가 작성된 일시, toolpath 가 가지는 각 path name, 가공 조건 등을 보인다.

4. GOODPATH : goodpath 가 작성된 일시, path point data 등을 보인다.

5. ROBOT : 각 robot 의 사양 및 기구학적인 조건, 관절 운동 영역 등을 보인다.

제 2 절 PROPS 와 ROBOT C/R 과의 통신

PROPS 에서 최종적으로 생성된 robot program file 은 각 robot 에 서 지원하는 통신 module 에 의하여 robot controller 에 전달된다. ABB 와 NACHI robot 은 이러한 off-line programming을 위한 program download S/W 를 PC 환경에서 지원한다. 이러한 download S/W 를 이용하여 로보트 제어기에서 쓰이는 code 로 compile 하여 RS232C 포트 로 송신한다. Download S/W 로서 ABB robot 의 경우 OLP3 를 지원하고, Nachi robot 의 경우 PC-CATS 를 지원한다.

그림 2.7 은 ABB robot 의 경우 PROPS 와 ROBOT 의 통신 flow 를 나타낸다.

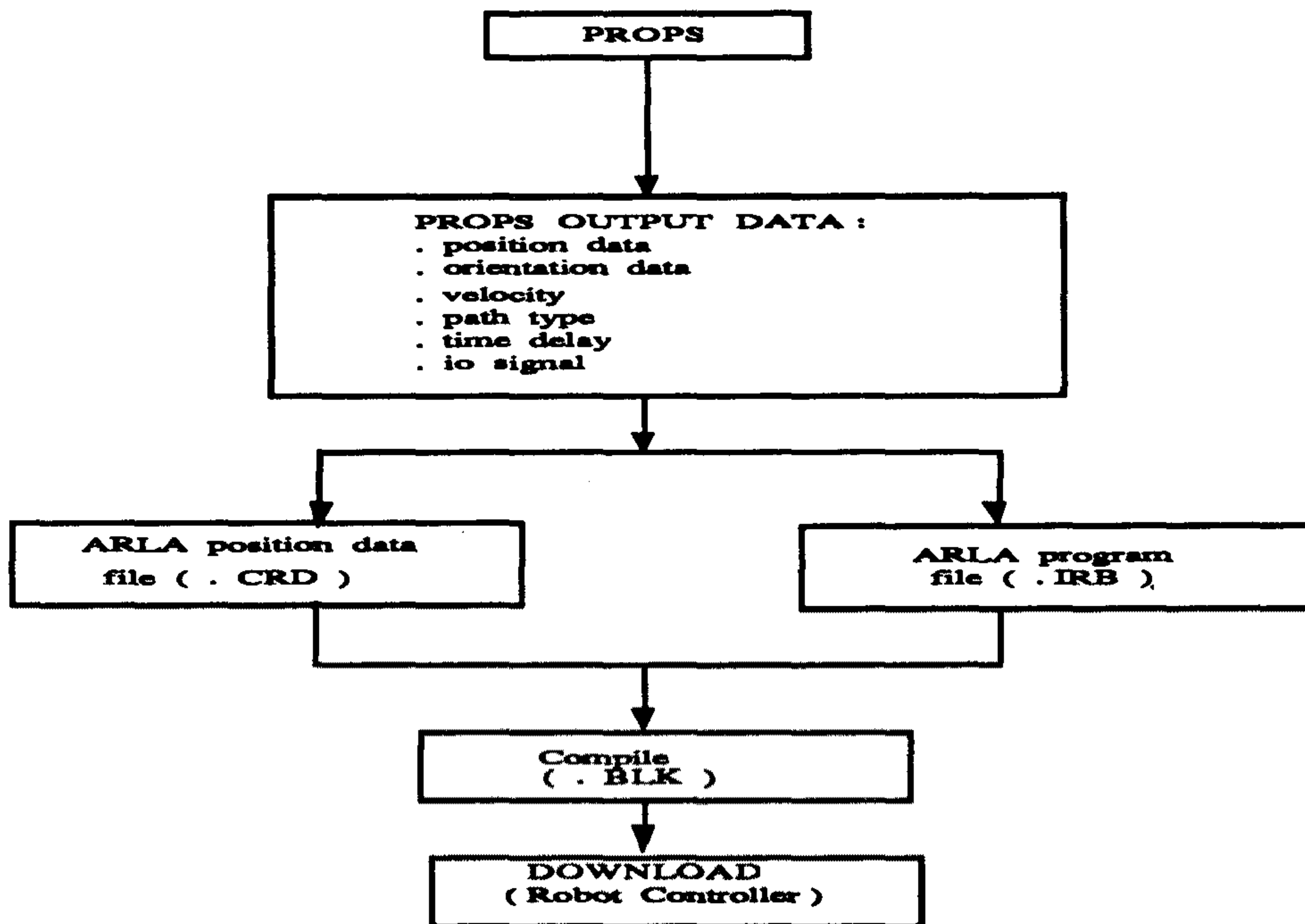


그림 2.7 ABB robot program 의 download

제 3 절 View transformaion menu

View transformaion menu 는 graphic package로서의 시각적 효과, 환경등을 위해서 만든 메뉴로서 program 수행 시 언제든지 선택 가능하도록 되어있다. 이는 User가 원하는 형태의 형상을 위한 menu로서 다음과 같이 구성되어있다.

LEFRIG -- Left or right translation

UPDOWN -- Up or down translation

ROTAT+ -- X, Y, Z 축을 중심으로 하는 회전 (positive angle)

ROTAT- -- X, Y, Z 축을 중심으로 하는 회전 (negative angle)

ZOOM -- Zoom up or down transformation

EYEUP/DOWN -- 눈의 위치를 화면의 위,아래로 바꾸어 준다

LOOKTY -- Perspective or parallel view

제 4 절 Top view menu

Top view menu 는 animation 중에 사용자가 tool 과 금형 면과의 충돌이 일어나는지의 여부를 직관으로도 알 수 있게하기 위한 menu 이다. 이 menu 는 simulation 의 submenu인 'stepwise' 혹은 'continuous' 에서 사용된다. 그림 2.8 은 animation top view 기능의 한 예이다.

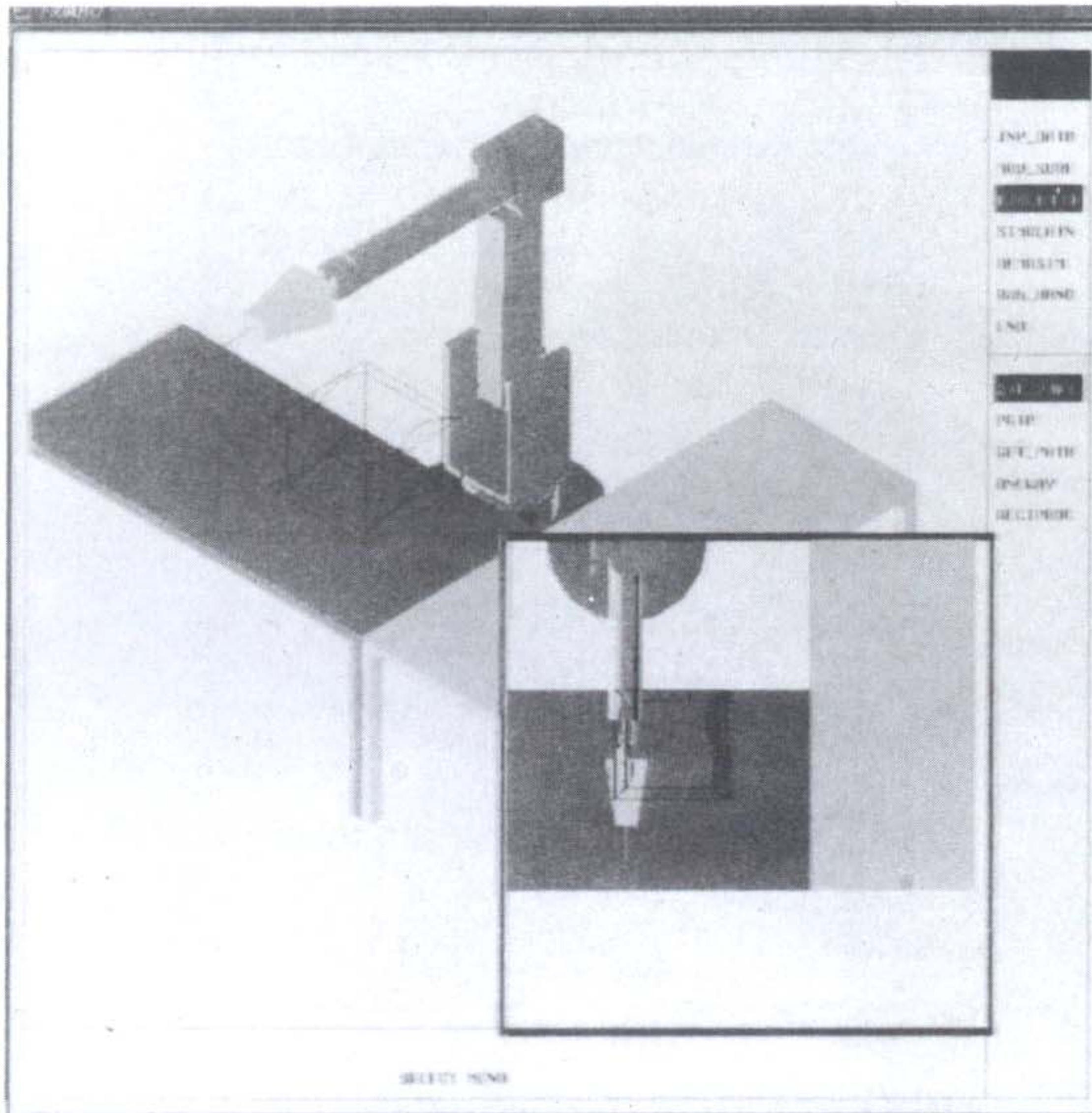


그림 2.8 Animation top view 의 예

제 5 절 Teaching pendant menu

로봇을 이용한 곡면가공에서 곡면 가공 자체에 소요되는 주 작업 경로외에 부작업 경로 즉, 공구의 교환 경로, 초기위치의 선정, 로봇 자세의 변경등에는 Teaching Pendant를 이용하는 교시 방식이 요구되어질 수 있다. 그림 2.9는 PROPS teaching pendant 의 외관을 나타낸다.

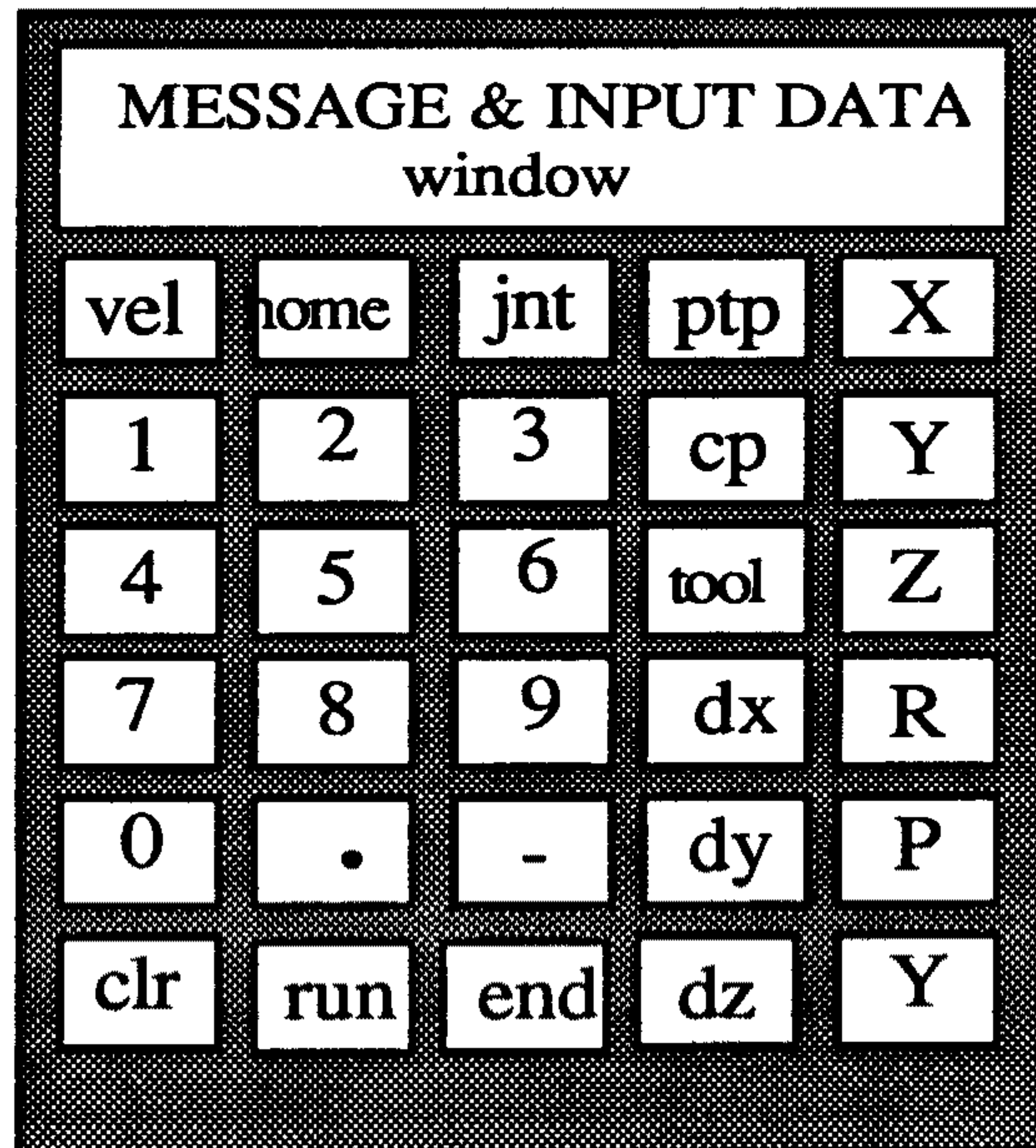


그림 2.9 는 PROPS teaching pendant 의 외관

5.2 Teaching pendant 의 menu

1. VEL : TCP 운동 속도 (mm/sec) 를 입력한다.
2. HOM : 현 위치에서 home position 으로 이동한다.
3. JNT : 각 joint 마다의 독립적인 운동을 수행한다.
4. PTP : 현 위치에서 목적하는 위치로 point to point 보간 방식으로 이동한다.
5. CP : 현 위치에서 목적하는 위치로 continuous path 보간 방식으로 이동한다.
6. TOOL : Tool spin 운동을 지정한다.
7. X,Y,Z,Roll,Pitch,Yaw : Ptp 혹은 cp motion 에서 목적 위치의 위치와 방향을 입력하는데 사용하는 button 이다. 단

Roll,Pitch,Yaw 는 Tool menu 에서도 사용된다.

8. dX, dY, dZ : Tool menu 에서 Roll,Pitch,Yaw button과 함께 사용되며, 각 방향의 미소 운동을 수행하는데 사용된다.

9. Dazit buttons : Ptp 혹은 cp motion 에서 입력값을 button으로 입력할 때 사용된다.

10. RUN : Ptp 혹은 cp motion 을 수행한다.

11. END : Teaching pendant 의 실행을 종료한다.

5.3 Teaching pendant operation

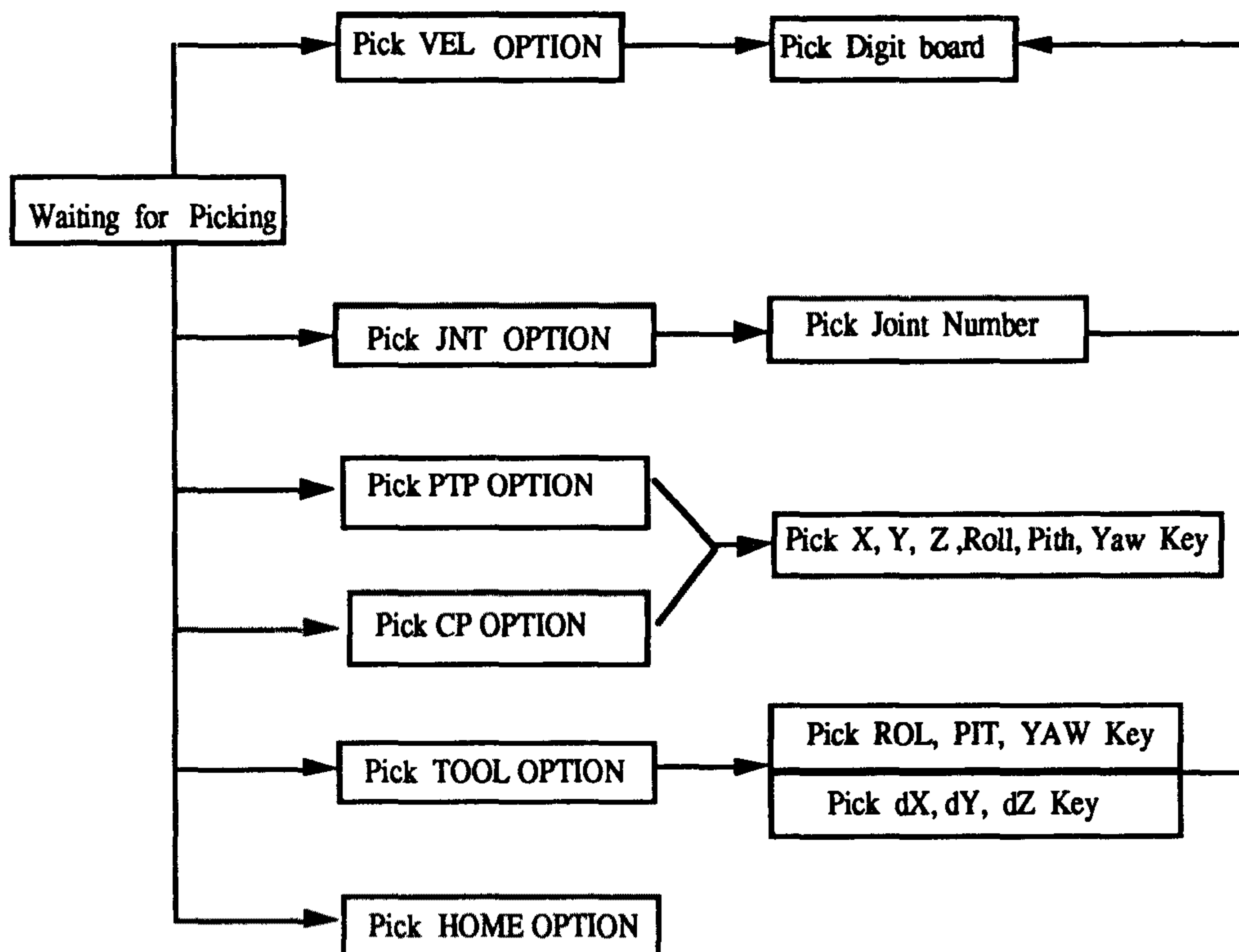


그림 2.10 Teaching pendant operation

제 3 장 COLLISION CHECKER의 개발

본 연구에서 개발한 collision checker를 이용하여 실제 가공시 발생할 수 있는 공구와 금형과의 충돌여부를 미리 simulation 수행시에 check함으로써 경비의 절감과 시간의 단축등을 꾀할 수 있게되었다.

제 1 절 TOOL 형상의 idealization과 계산과정

- tool을 기본 도형인 육면체로 보고 8개의 꼭지점과 cavity를 구성하는 주변 surface와의 간섭을 check한다.
- 각 꼭지점 $Q_i(X_i, Y_i, Z_i)$ 마다 surface에 최단거리로 projection시킨 새로운 좌표값 $P_i(X_i, Y_i, Z_i)$ 와 그 점에서의 법선 벡터 $N_i(N_x, N_y, N_z)$ 를 계산한다.

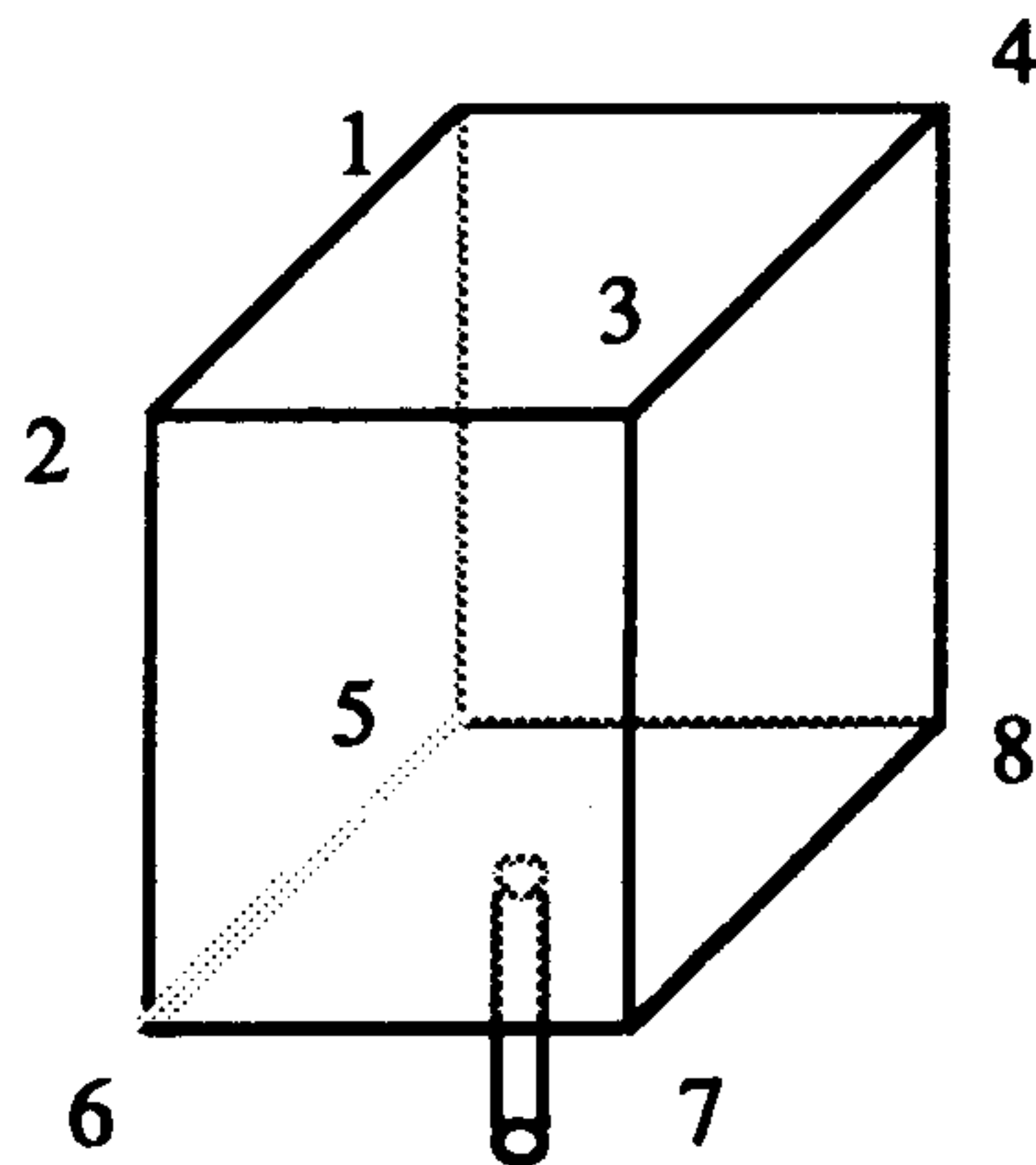


그림 3.1 공구의 형상

- . 여기서 cavity를 구성하는 주변 surface는 법선 벡터의 방향에 일관성(+/-)이 있게 modeling이 되어야 한다.
- . $R_i (= P_i - Q_i)$ 를 계산한 후에 법선 벡터 N_i 와의 dot production을 이용하여 충돌 여부를 계산한다.
- . 위의 과정은 가공 곡면의 각 path point마다 수행된다.
- . 충돌이 일어났을 경우 화면상에 그 message를 도시하고 충돌이 일어난 path point좌표와 normal vector를 Rdb를 이용하여 기억하고 path correction routine의 자료로 삼는다.
- . simulation이 끝난 후 다음 그림 3.2과 같은 collision point를 도시한다. 이러한 collision point들은 Rdb에 저장되어 공구 교환의 자료로 삼는다.

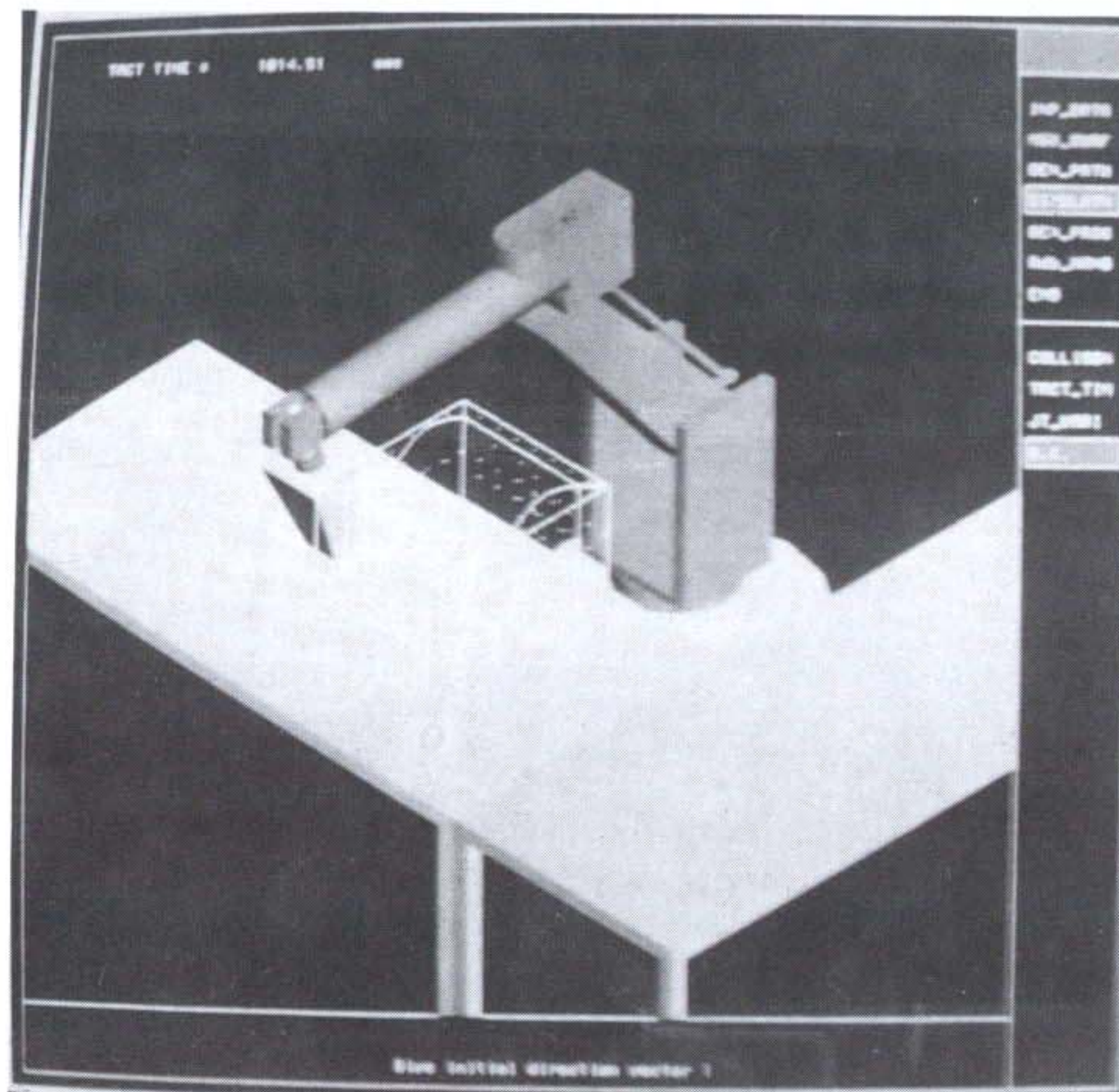


그림 3.2 collision point의 도시

제 2절 기본 Algorithm

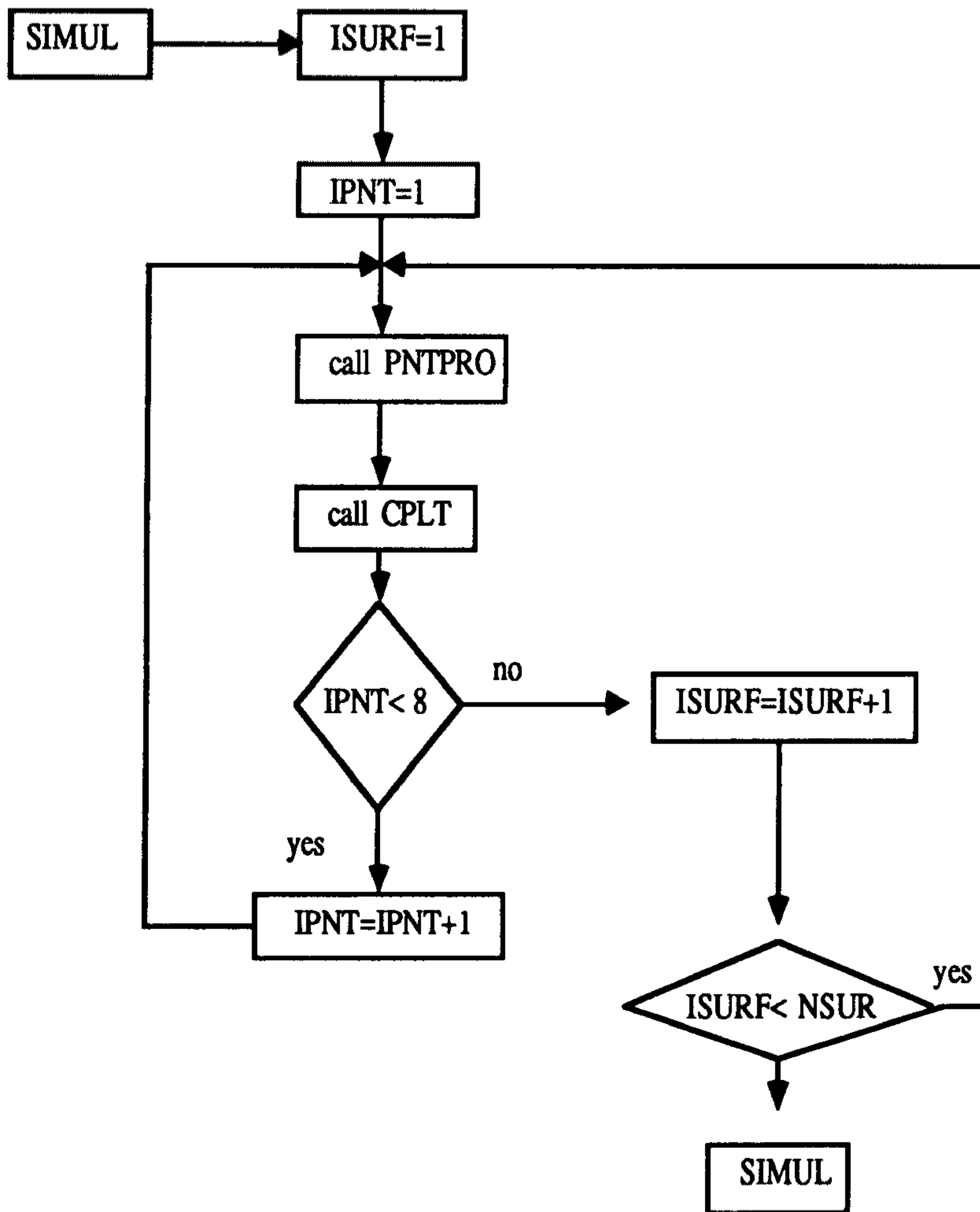


그림 3.3 Collision routine의 구조

제 3 절 충돌이 배제된 경로의 생성

충돌 검증 루틴 (collision check module)을 거치면서 충돌이 일어난 점 (Node value : 1)과 양호한 점 (Node value : 0)에 대한 정보를 갖는 Node data file을 만든 후 충돌이 발생하지 않은 양호한 점들만을 연결하는 새로운 경로를 생성한다.

- 1) Collision check module을 거친 Node array의 구성 : 그림 3.4 참조
- 2) 초기노드의 결정 (node data file에서 최초로 그 값이 영인 점)
- 3) 한 node에서 갈 수 있는 근접 node의 방향 (시계방향과 반시계방향) : 그림 3.5 참조
- 4) 각 node (mother node)마다 8개의 근접 node를 시계방향 혹은 반시계방향으로 check하여 최초로 1이 아닌 node (son node)를 새로운 경로로 설정하고 mother node value를 1로 치환한다. 전체 경로점에 대하여 위의 과정 (3번 과정)을 반복 수행하면 충돌이 배제된 양호한 경로를 생성한다.

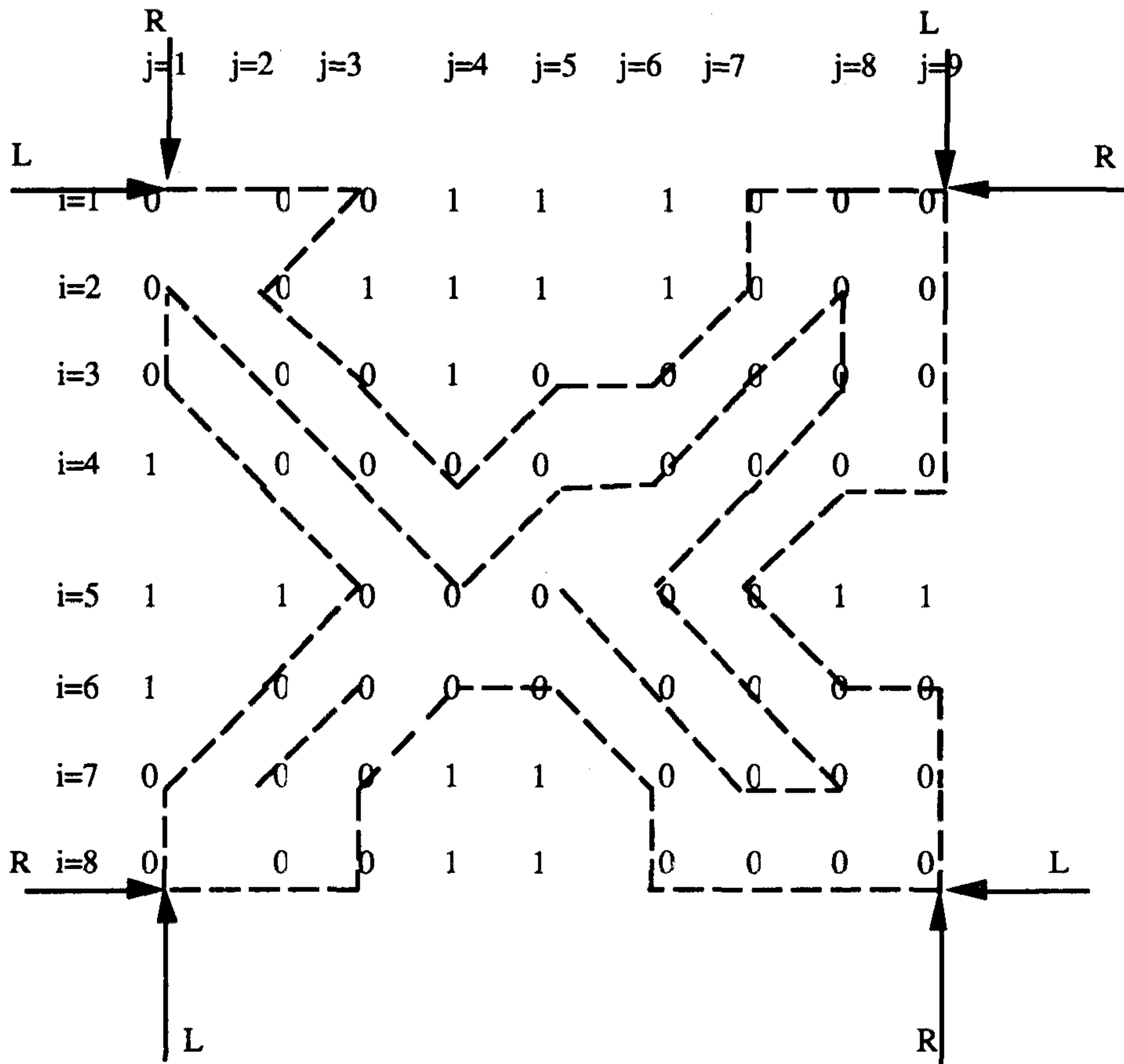


그림 3.4 Node array의 구성

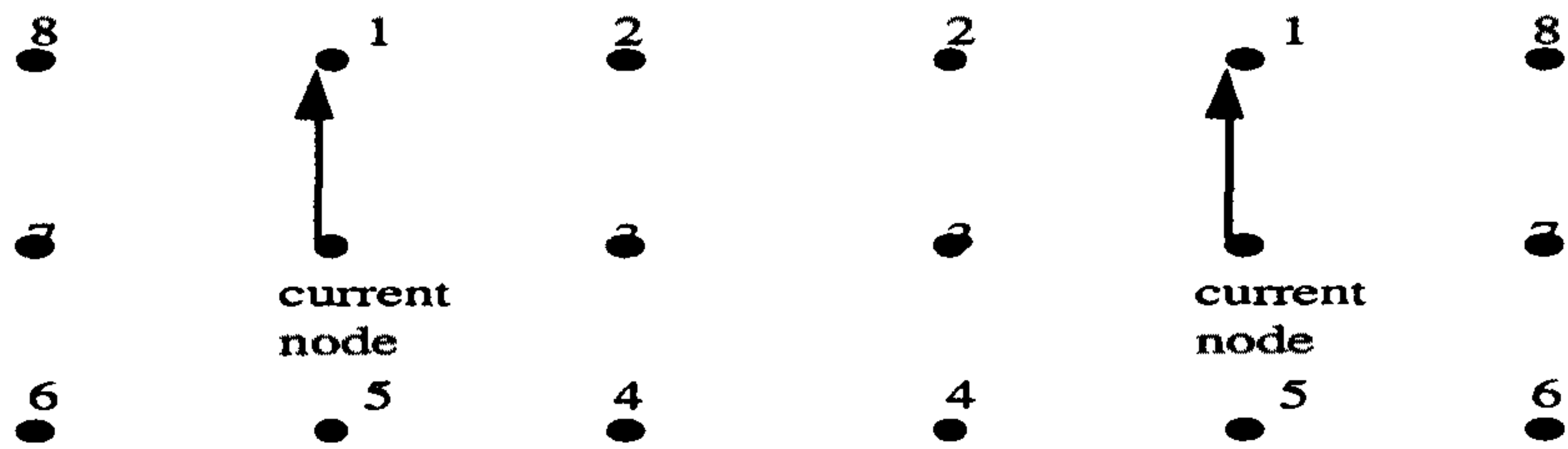


그림 3.5 접근 가능한 근접 node

제 4 절 충돌이 발생한 영역의 처리

Collision check module을 거쳐 생성된 Node array의 정보를 갖는 file에서 Node 값을 반대로 치환하고 위 2절의 routine을 거쳐 작업경로를 생성한 다음 공구 교환경로를 추적하여 공구를 교환하고 다시 한번 simulation을 수행하여 충돌을 검증한다. 이러한 과정을 거쳐서도 충돌이 발생하는 부분에 대해서는 곡면의 법선벡터를 변경하여 해결한다.

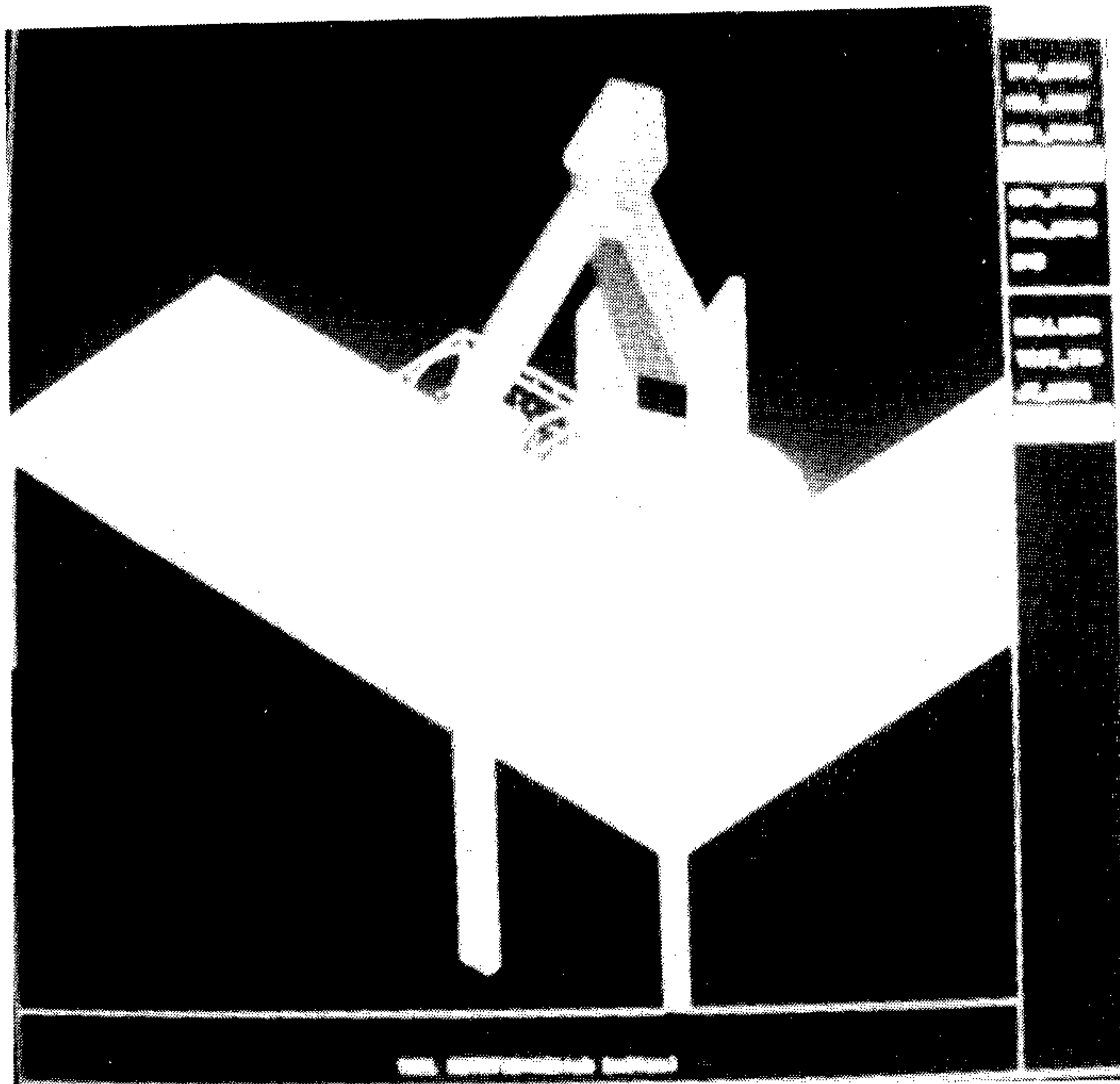


그림 3.6 충돌이 배제된 경로의 생성

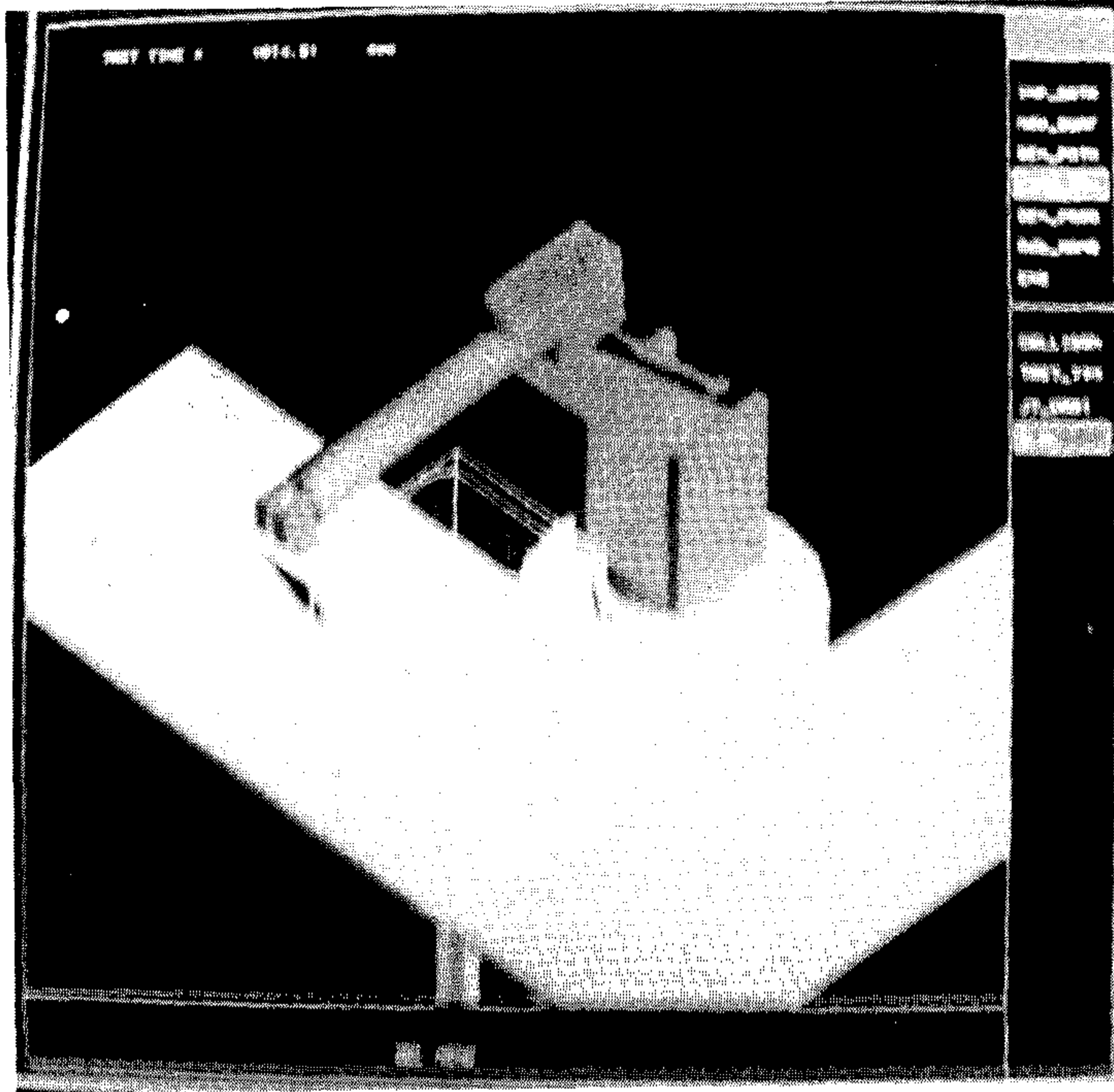


그림 3.7 충돌이 발생한 경로의 생성

제 4 장 Inverse algorithm의 구성

제 1 절 역기구학 (Inverse kinematics) 의 정의

로봇 매니플레이터가 작업을 하는 과정에서 기준 좌표계에 대한 endeffector 의 위치와 방향이 주어진 경우, 그에 해당하는 각 조인트 변

위값을 구해야 하는데 이것이 곧 역 기구학의 문제이다. 따라서 각 조인트 변수의 값이 주어졌을 때 endeffector 의 위치와 방향을 구하는 정 기구학 (forward kinematics) 문제와는 정반대의 문제이다.

이 문제는 실제로 로봇 endeffector 를 지정된 위치로 구동시킬 때 뿐만 아니라, off-line program 상에서 로봇의 작업 위치 (혹은 경로) 에 대한 조인트 변위값을 구하는 문제에서 필요하며 그 밖에도 적용분야가 매우 넓다. 따라서 일반화가 가능하고, 수렴성이 양호한 역 기구학 알고리즘의 개발이 절실히 요구되고 있다.

제 2 절 역 기구학의 접근 방법 (연구 동향)

일반적으로 산업용 로봇으로 많이 쓰이는 6 관절 로봇을 모델로 하여 역 기구학 방정식을 세워보면, 6 개의 조인트 (회전 혹은 병진) 변수를 미지수로 하는 복잡한 비선형 연립방정식으로 귀착된다. 이 비선형 연립방정식을 풀기 위하여 지금까지 여러가지 방법이 연구되었는데, 첫째로 이론적 해석법 (analytical method) 과 둘째로 반복적 해석법 (numerical method) 으로 나뉘어진다.

첫째로 Pieper , Paul 등에 의해 제안된 이론적 해석법은 비선형 연립방정식을 복잡한 산술적인 조작에 의하여 각 조인트 변수값을 구하는 방법이다. 그런데 이 방법은 매니플레이터의 3 개의 회전 조인트 축이 로봇의 손목에서 교차해야하는 전제 조건이 필요하고, 조인트 형

태가 달라짐에 따라 좌표계 변환 행렬 (transformation matrix) 과 자코비안 행렬 (jacobian matrix) 이 달라지므로 일반화된 알고리즘을 개발하기가 불가능한 단점을 가지고있다.

둘째로 반복적 해석법은 주어진 역 기구학 방정식 또는 위치와 방향의 편차함수를 설정하고 반복계산을 통하여 각 조인트의 변위를 계산하는 방법이다. 이 방법은 이론적 해석법과 달리 전제조건이 필요 없으며, 6 관절 로봇의 경우 조인트 형태에 구애받지 않고 최소한 하나의 해를 항상 구할 수 있으므로 조인트 형태에 관계없는 일반적인 알고리즘을 개발할 수 있다. 따라서 본 연구는 일반적인 조인트 형태를 가지는 6 관절 로봇 매니플레이터의 역 기구학 해를 구하기 위한 알고리즘 개발을 목적으로 하고 있으므로 반복적 해석법을 채택하고있다.

역 기구학 문제를 해결하기 위하여 Newton-Raphson 방법을 이용한 반복적 해석기법이 많이 연구되었다. 그런데 이 방법의 기본 원리는 조인트 좌표계와 Cartesian 좌표계간의 미소변화 (differential tranformation) 관계를 정의하는 자코비안 행렬을 구함으로써, Cartesian 좌표계의 변화량으로부터 조인트 좌표계 상의 새로운 값을 구하는 것이다. 위의 이러한 자코비안 행렬은 로봇의 기구학적 자세에 따라 자코비안의 행렬식의 값이 영이되는 특이 (singularity) 문제가 발생할 수 있다. 이를 해결하기 위하여 알고리즘 상에서 특이 상태를 검사하고 이를 보정하는 작업이 필요하다. 이러한 문제들로 인하여, Newton-Raphson 방법은 로봇의 자세에 많은 예외적인 요소를 내포함으로써 해를 구하

는 과정을 복잡하게 한다. 또 하나의 반복적 해석법은 편차 함수 최소화 방법 (deviation function minimizing method)이다. 이 방법은 로봇 end-effector 의 현재 위치, 방향과 목표 위치, 방향 간의 차이를 편차 함수로 설정하여 그 차이를 최소화하기 위한 조인트 변수 값을 구하는 방법이다. 이 방법을 영 기준 위치법을 이용하여 적용하면, 한 개의 미지수 (조인트 변수) 에 대한 비선형 방정식으로 귀착 되므로 자코비안 행렬이 필요하지 않게 된다. 따라서 역 기구학 알고리즘의 일반화가 용이하게 되고, 자코비안으로 인한 특이 문제가 발생하지 않으므로 로봇의 기구학적 자세에 따른 수렴의 불안정성을 피할 수 있다. 따라서 근래에 와서는 이 방법이 많이 연구되고 있으며 본 알고리즘에서도 이 방법을 채택하고 있다. 그리고 본 알고리즘에서는 보다 안정된 수렴성과 계산 시간을 단축시키기 위하여 근접 위치 방법 (near position method) 을 개발하여 적용하였다. 근접 위치 방법이란 이론적 해석법으로 풀기가 불가능한 기구학을 갖는 6 관절 로봇을 반복적 해석법을 사용한다는 것을 전제로하여, 초기 위치를 목표 위치에 가능한 근접하게 잡아서 반복 계산을 수행하는 방법으로서 로봇의 기구학적 자세에 따른 수렴의 불안정성을 방지하고, 계산 시간을 단축하는데 그 목적이 있다.

제 3 절 영 기준 위치 방법

편차 함수 최소화 방법을 이용함에 있어서 편리함을 위하여, 로봇의 형상을 기술하는 방법으로 영 기준 위치 방법을 도입하였다. (그

림 4.1) 그림 4.1은 N 자유도의 링크 기구를 영 기준 위치 방법을 이용하여 표시한 것이다. 그림에서와 같이 좌표계는 고정 좌표계 하나만 있으면 된다. 영 기준 위치 방법을 이용하여 로봇 매니플레이터를 비롯한 링크 기구를 기술할 때, 기구학적 변수 (kinematic parameter) 를 정의하면 다음과 같다.

- Joint vector (u_k) ; k 번째 joint 의 축 방향 단위 벡터
- Body vector (b_{k+1}) ; k 번째 joint 를 시점, k+1 번째 joint 를 종점으로 하는 벡터
- Axial vector (u_a) ; end effector 가 가리키는 방향의 단위 벡터
- Tranverse vector (u_t) ; axial vector 와 직각 방향의 단위 벡터 (단, zero position 일 경우에는 $u_{0,k}$, $b_{0,k+1}$, $u_{0,a}$, $u_{0,t}$)

다음에 k 번째 조인트에서의 회전 행렬 (rotation matrix) [R] 을 정의하면,

$$[R_0] = [I]_{3 \times 3}$$

$$[R_k] = [R_{k-1}] [Rot(q_k, u)] \quad (\text{회전 조인트})$$

$$[R_k] = [R_{k-1}] \quad (\text{병진 조인트})$$

여기서,

$$[\text{Rot}(q_k, u_{0k})] = \begin{bmatrix} (u_x^2 - 1)Vq_k + 1 & u_x u_y Vq_k - u_z S q_k & u_x u_z Vq_k + u_y S q_k \\ u_x u_y Vq_k + u_z S q_k & (u_y^2 - 1)Vq_k + 1 & u_y u_z Vq_k - u_x S q_k \\ u_x u_z Vq_k - u_y S q_k & u_y u_z Vq_k + u_x S q_k & (u_z^2 - 1)Vq_k + 1 \end{bmatrix}$$

$$Vq_k = 1 - \cos(q_k), \quad S q_k = \sin(q_k) \quad u_{0k} = (u_x, u_y, u_z)$$

이 된다.

따라서 최초에 임의의 로봇 형상을 영 위치로 설정하였을 때, joint vector 가 $u_{0,k}$, body vector 가 $b_{0,k}$ 이면, k 번째 joint 에서 q_k 만큼 회전 또는 이동하여 생기는 새로운 joint vector, body vector, axial vector, transeverse vector 는 다음과 같다.

$$u_k = [R_k] u_{0k}$$

$$b_{k+1} = [R_k] b_{0,k+1}, \quad k = 1, \dots, N$$

$$u_n = [R_N] u_{0,n}$$

$$u_t = [R_N] u_{0t}$$

영 기준 위치 방법은 좌표계가 로봇 base 에 고정된 것 하나면 충분하므로 로봇의 초기 위치를 설정하는데 유연성이 있고 매우 간편하다. 영 기준 위치 방법의 예로 HYUNDAI 7603 로봇의 초기 위치를 그림 4.2 와 표 1 에 나타내었다. (그림 4.2 & 표 1)

제 4 절 역기구학식의 형성

4-1. 근접 위치의 설정

역 기구학 알고리즘을 수행하기 전에 목표 위치 (Goal position) 에 가까운 초기 위치를 설정하는 일이 필요하다. 이는 수렴의 안정성을 기할 뿐 아니라 계산 시간을 단축하는 데도 필요하다.

역 기구학을 풀기 위한 로봇이 설정이 되면, 일반적인 6-관절 로봇인 경우 조인트 부분에 shift 가 있으면, 이를 가상적으로 제거하여 shift 가 없는 기구학을 갖는 새로운 로봇으로 모델링한다. 그런데 이 가상 로봇의 작업 공간 (work space) 이 원래 로봇의 작업 공간을 전부 포함하지 않는 경우 조인트 변수 값을 구할 수 없게 되는 문제가 발생하므로 이를 방지하기 위하여 로봇의 위치를 결정하는 1,2,3 번째 링크 길이를 조절하여 주는 알고리즘이 필요하다. 위의 과정이 수행되면, 그림 4.3 와 같이 위치 편차 벡터 $d\{P_H\}$ 를 이용하여 근접 위치의 조인트 변수 값 q_i^* $i = 1, \dots, 6$ 을 구한다. 이 q_i^* 값들은 역 기구학 계산을 위한 초기값의 역할만을 하는 것이지 궁극적으로 구하고자 하는 근 (q_i) 은 아니다. (그림 4.3)

목표 방향과 위치 ($[R_H]^0, \{P_H\}^0$) 와 영 위치에서의 방향과 위치 ($[R_H]^0, \{P_H\}^0$) 가 주어졌을 때, 로봇의 손목 까지의 위치벡터를 $\{P_R\}$ 이라 할때,

$$d\{P_H\} = \{P_H\}^G - \{P_H\}^0 \quad (1)$$

$$\{P_R\}' = \{P_R\}^0 + d\{P_H\}$$

가 된다.

$\{P_R\}'$ 로부터 손목 부분을 제외한 3 관절 로봇의 역 기구학 해를 이용하여 q_1^*, q_2^*, q_3^* 를 구한다. 여기서 이 3 관절 로봇이 3R 타입인 경우 Pieper's solution 을 이용하여 구하고, 3R 타입이 아닌 경우는 shift 가 없는 일반적인 3 관절 로봇의 이론적 해석법에 의한 역 기구학 해를 직접 구하여야 한다. 만일 이론적 해석법으로 해를 구하기가 어려운 3 관절 로봇인 경우에는 뒤에 기술할 반복적 역 기구학 알고리즘을 일반적인 3 관절 로봇에 적용하여 해를 구할 수 있다. 3 관절인 경우의 반복적 계산 방법은 방향은 고려하지 않고, 위치에 대한 역 기구학 문제를 풀면 되므로 6 관절 로봇의 경우 보다는 계산 속도가 훨씬 빠르다. q_1^*, q_2^*, q_3^* 가 구해지면 $[R_H]^G$ 로부터 Euler angle solution 을 이용하여 q_4^*, q_5^*, q_6^* 를 구한다. 두가지 손목 좌표계에 대한 Euler angle solution 은 다음과 같다.

$$[R]_3^6 = ([R]_0^3)^{-1} [R_H]^G \quad (2)$$

$$[R]_3^6 = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} \quad (3)$$

Z-Y-X Euler angle 인 경우,

$$q_4^* = \text{Atan2} (r_{21}, r_{11})$$

$$q_5^* = \text{Atan2} (-r_{31}, (r_{11}^2 + r_{21}^2)^{1/2}) \quad (4)$$

$$q_6^* = \text{Atan2} (r_{32}, r_{33})$$

Z-Y-Z Euler angle 인 경우,

$$q_4^* = \text{Atan2} (r_{23}, r_{13})$$

$$q_5^* = \text{Atan2} ((r_{11}^2 + r_{21}^2)^{1/2}, r_{33}) \quad (5)$$

$$q_6^* = \text{Atan2} (r_{32}, -r_{31})$$

위와 같이 shift 가 없는 6-관절 로봇에 대한 역 기구학 해 q_i^* , $i=1, \dots, 6$ 의 set (일반적으로 4 개의 set) 중에서 조인트 운동범위 (joint range) 와 링크의 모드 (upper arm 혹은 lower arm) 에 대한 정보를 고려하여 적절한 q_i^* 를 초기값으로 정하여 역 기구학 계산을 시작한다. 근접 위치를 구하기 위한 알고리즘은 그림 4.4 와 같다. (그림 4.4)

4-2. 편차함수 최소화 방법

편차함수 최소화 방법을 이용하여 역 기구학 문제를 푸는 과정은 다음과 같다.

입력값은,

◦ 목표 방향 행렬 (Goal orientation matrix)

$[R_H^G]_{3 \times 3}$; 절대 좌표계에 대한 end effector 의 방향을 나타내는
행렬

◦ 목표 위치 벡터 (Goal position vector)

$\{ P_H^G \}$; 절대 좌표계에 대한 end effector 의 위치 벡터
가 된다.

그림 4.5 는 수렴중인 한 위치에서의 역 기구학 반복 계산 과정
을 보여 주고 있다. (그림 4.5)

q_k : k 번째 joint 변수 값

$\{ P_H^C \}$: 현재 위치에서 end effector 의 위치 벡터 (Current
position vector)

$[R_H^C]$: 현재 위치에서 end effector 의 방향 행렬 (Current
orientation matrix)

$\{ P_k^C \}$: 현재 고려 중인 k 번째 joint 의 절대 좌표계에 대한
위치 벡터 (joint position vector)

일 때,

k 번째 joint 에서 dq_k 만큼 회전 혹은 이동하였을 때 새로 생긴
방향 행렬과 위치 벡터를 각각 $[R_H'], \{ P' \}$ 라 하면,

회전 조인트의 경우,

$$[R_H'] = [\text{Rot}(dq_k, u_k)] [R_H^C] \quad (6)$$

$$\{ P_H' \} = \{ P_k^C \} + [\text{Rot}(dq_k, u_k)] (\{ P_H^C \} - \{ P_k^C \}) \quad (7)$$

병진 조인트의 경우,

$$[R_H] = [R_H^C] \quad (8)$$

$$\{ P_H' \} = \{ P_H^C \} + dq_k u_k \quad (9)$$

이 된다.

방향 편차 (orientational deviation) d_R 은

$$d_R = \sum_{i=1}^3 \sum_{j=1}^3 \{ ([R_H^G][R_H^C]^{-1})_{i,j} - ([R_H'] [R_H^C]^{-1})_{i,j} \}^2 \quad (10)$$

위치 편차 (positional deviation) d_P 는

$$d_P = \sum_{j=1}^3 (\{ P_H^G \}_j - \{ P_H' \})^2 \quad (11)$$

R 과 P 를 균형있게 고려하기 위하여 균형치 (balacing factor) C 를 두는데,

$$C = d_P / (d_P + d_R) \quad (12)$$

로 정의한다.

따라서 편차 함수 (deviation funtion) 는

$$= (1 - C) * d_R + C * d_P \quad (13)$$

가 된다.

편차함수 를 최소화하기 위한 필요충분 조건은

$$\frac{\partial d}{\partial (\Delta q_i)} = 0, \frac{\partial^2 d}{\partial (\Delta q_i)^2} > 0 \quad (14)$$

이 된다. (그림 4.6)

제 5 절 근접 위치 방법과 Kazerounian 방법의 비교

초기 위치를 근접 위치로 구하고 편차함수 최소화 방법을 이용하여 역 기구학 반복 계산을 수행한 결과와, 초기 위치를 영위치로 고정시켜 편차 함수 최소화의 방법을 이용하여 계산한 Kazerounian 의 결과를 수렴 속도의 관점에서 비교하였다. 모델은 현대 7601 로봇 (6R) 과 STANFORD 로봇 (R-R-P-R-R-R) 으로 설정하였고, 80286 AT-Computer 로 계산하였다. (그림 4.7)

제 6 절 잉여 자유도 로봇의 역 기구학문제

본 알고리즘이 6 관절 이상의 로봇에도 적용되어 하나의 해를 구할 수 있음을 보이기 위하여 다음과 같은 잉여 자유도 로봇을 설정하여 역 기구학 해를 구하여 보았다. 초기 위치는 영 기준 위치 방법으로 설정하였고, 많은 해 중에서 초기 위치에 가까운 하나의 해를 구할 수 있다. 따라서 본 알고리즘을 이용해서, 잉여 자유도 로봇의 최적 해를 구하는 문제에도 확장 적용할 수 있게 되었다.

그림 4.8 과 표 2 는 일례로 7R 로봇의 초기 위치와 기구학 계수를 나타내고 있다. 병진 조인트가 포함된 로봇의 경우도 가능하다. (그림 4.8 , 표 2) 그림 4.9 과 그림 4.10 는 7 개 조인트 변수의 수렴곡선을 보여주고 있다 (그림 4.9 , 그림 4.10). 이 해는 그림 4.8 의 초기 위치를 초기 값으로 해서, 이 값에 가장 근접한 하나의 해를 구한 것이다.

Joint	Type	$u_{0,k}$	$b_{0,k+1}$
1	R	(0, 0, 1)	(L_1 , 0, 0)
2	R	(0, 1, 0)	(0, 0, L_2)
3	R	(0, 1, 0)	(0, 0, L_3)
4	R	(1, 0, 0)	(L_4 , 0, 0)
5	R	(0, 1, 0)	(0, L_5 , 0)
6	R	(1, 0, 0)	(L_6 , 0, 0)
		$u_{0,a} = (1, 0, 0)$	$u_{0,t} = (0, 1, 0)$

$$L_1 = 220 \quad L_2 = 600. \quad L_3 = 160. \quad L_4 = 600.$$

$$L_5 = 105. \quad L_6 = 115. \quad (\text{단위 : mm})$$

표 1. 영 기준 위치 방법에 의한 기구학 계수

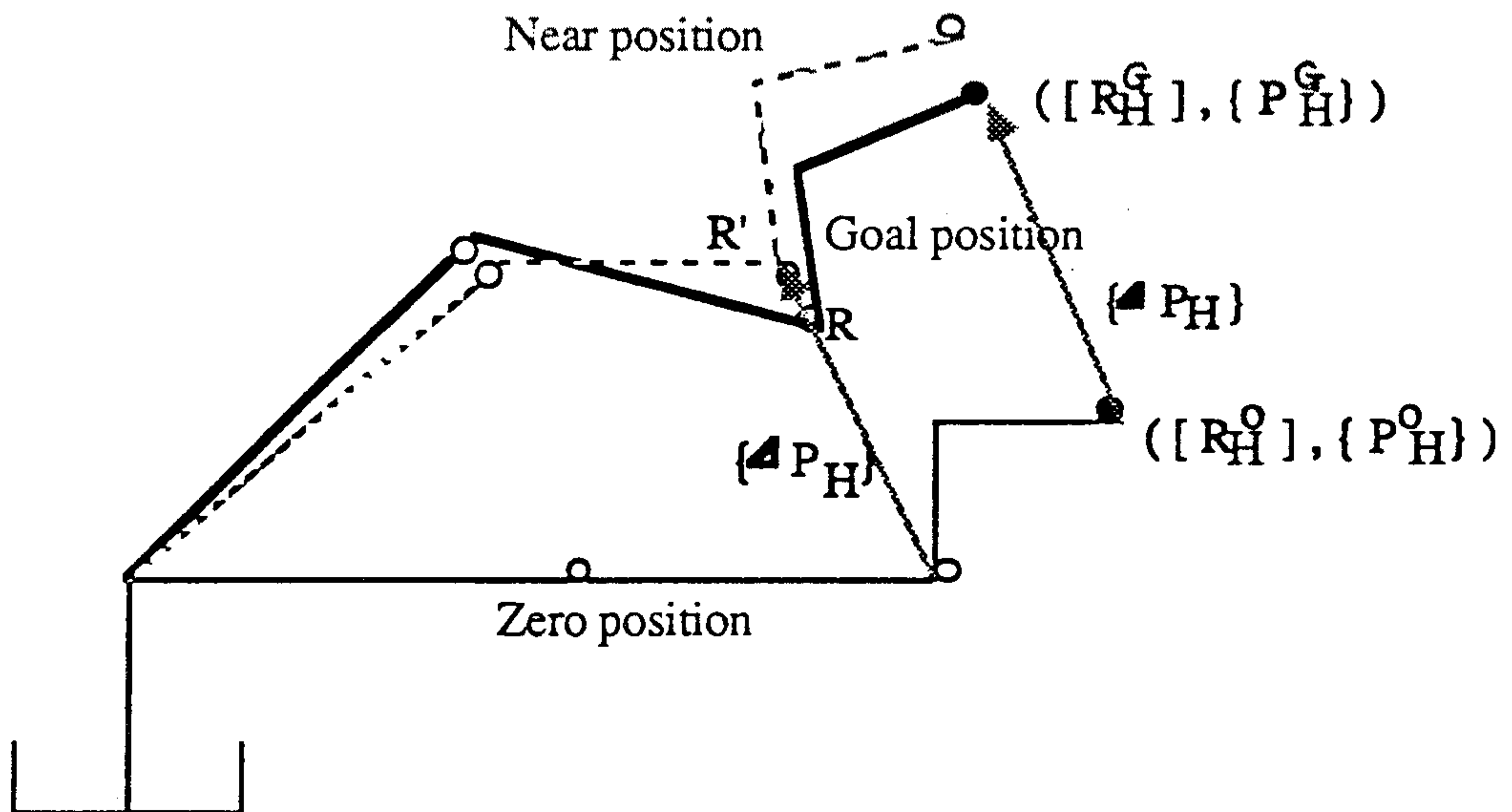


그림 4.3 근접 위치의 결정

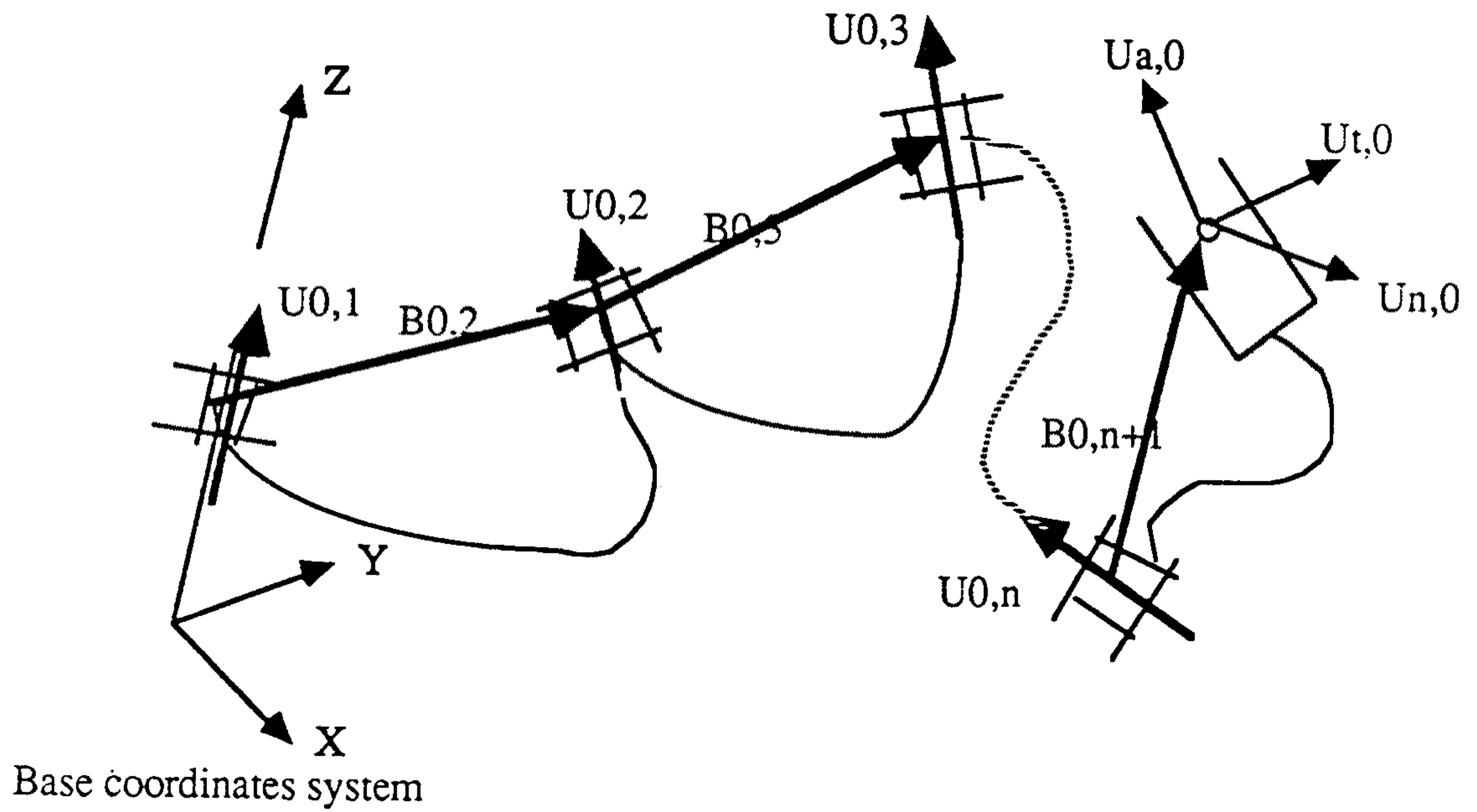


그림 4.1 영 기준 위치 방법

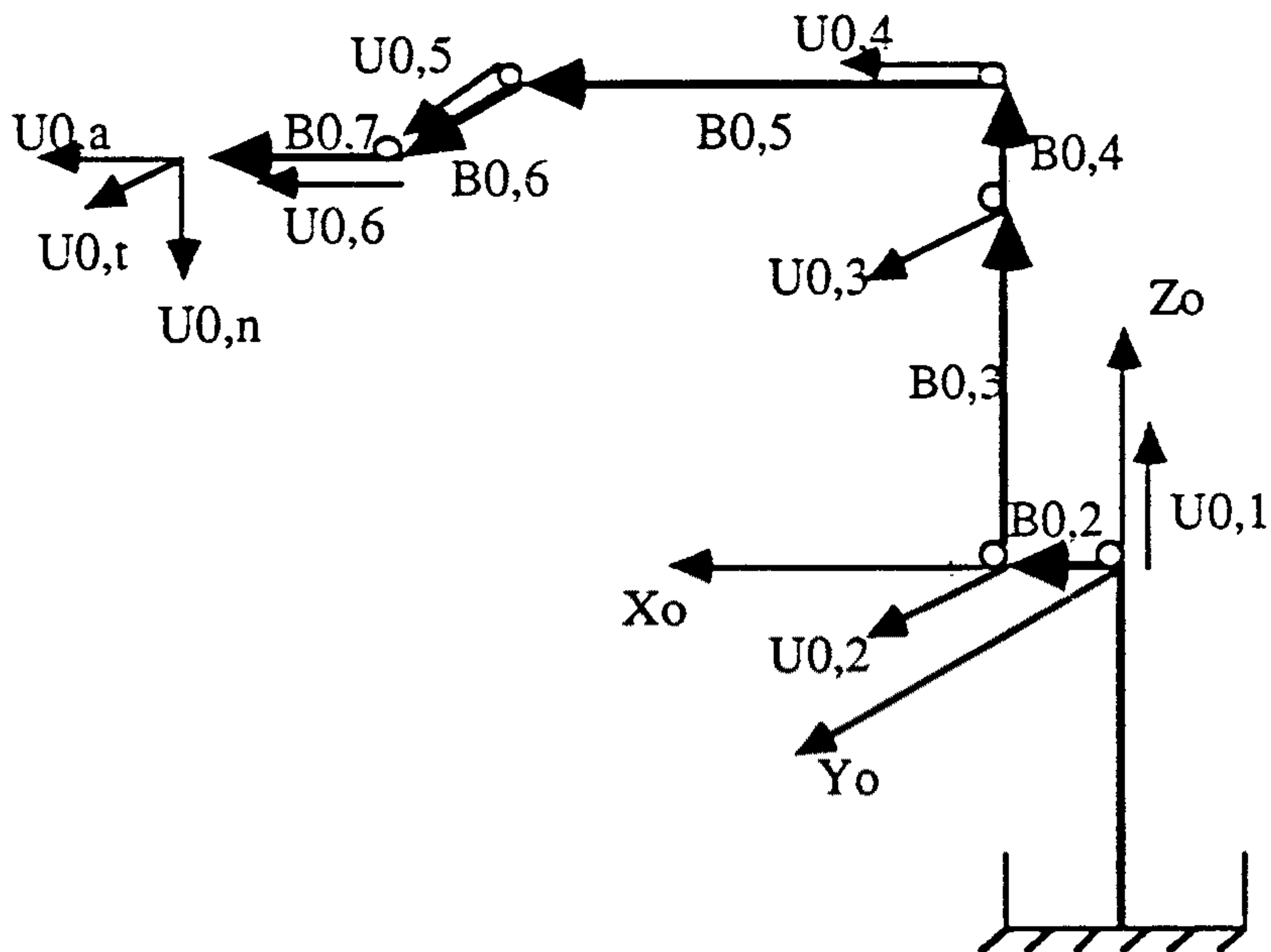


그림 4.2 영 기준 위치 방법에 의한 초기 위치

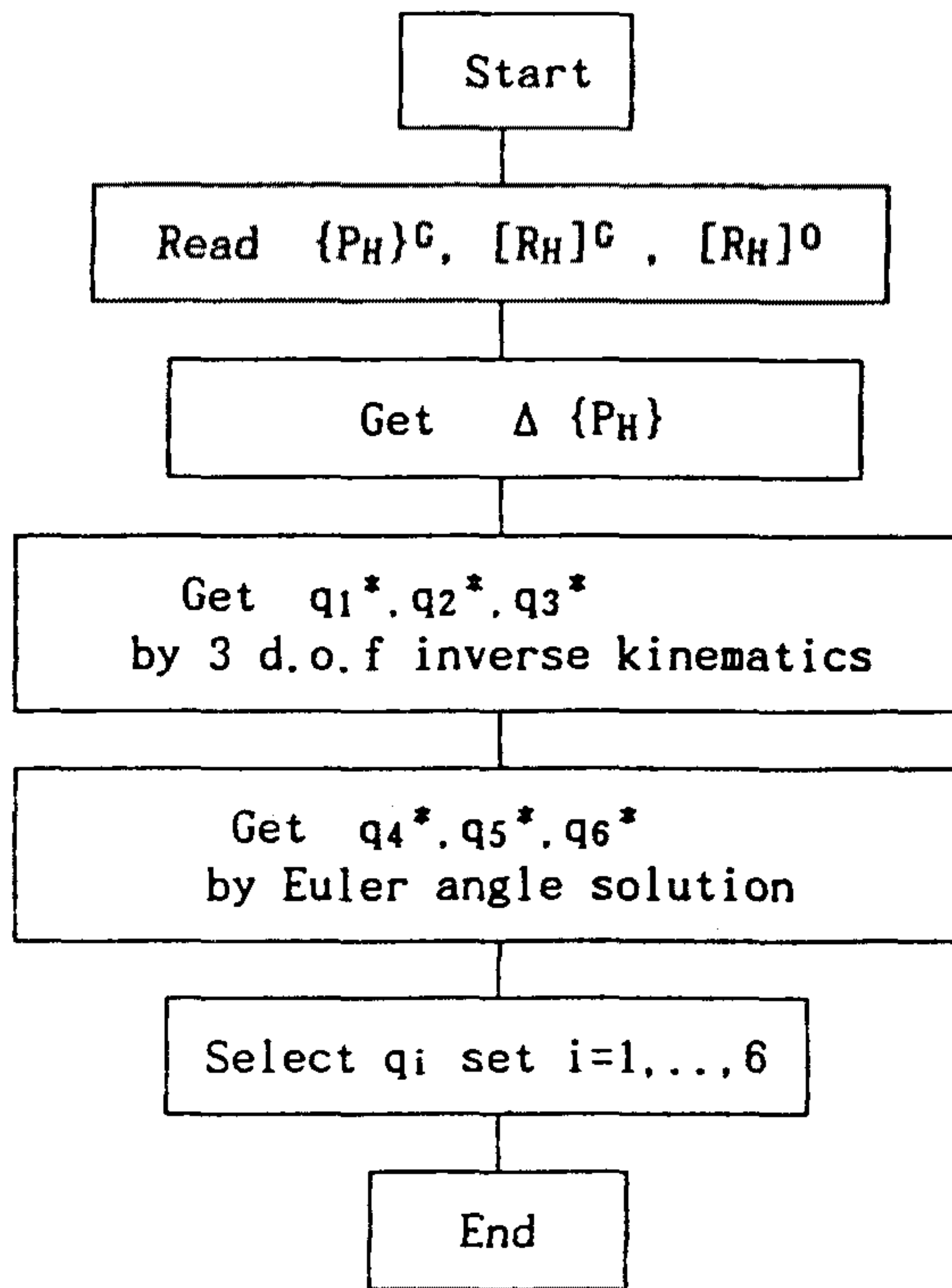


그림 4.4 근접 위치 결정 흐름도

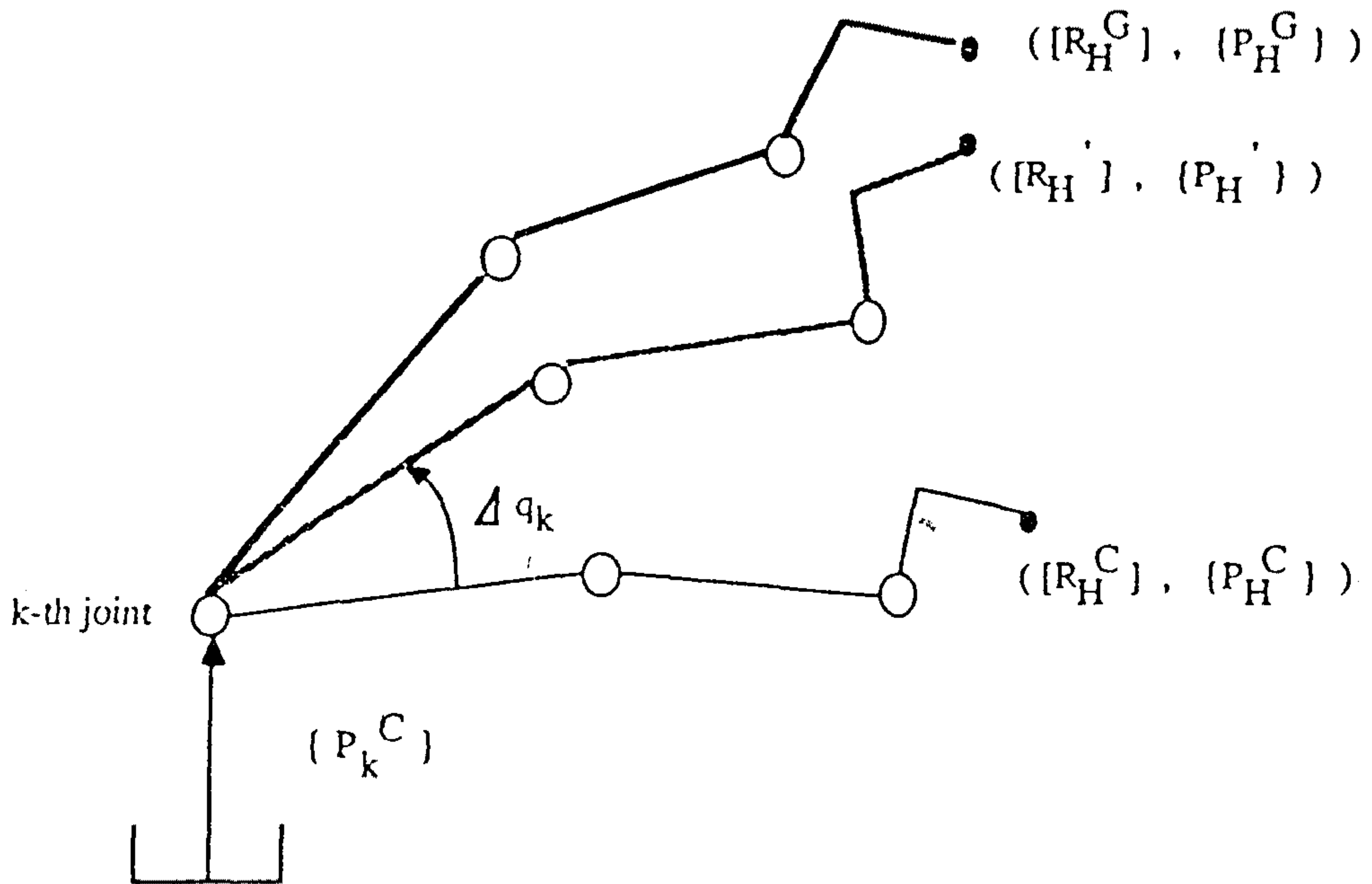


그림 4.5 역 기구학 반복 계산 과정

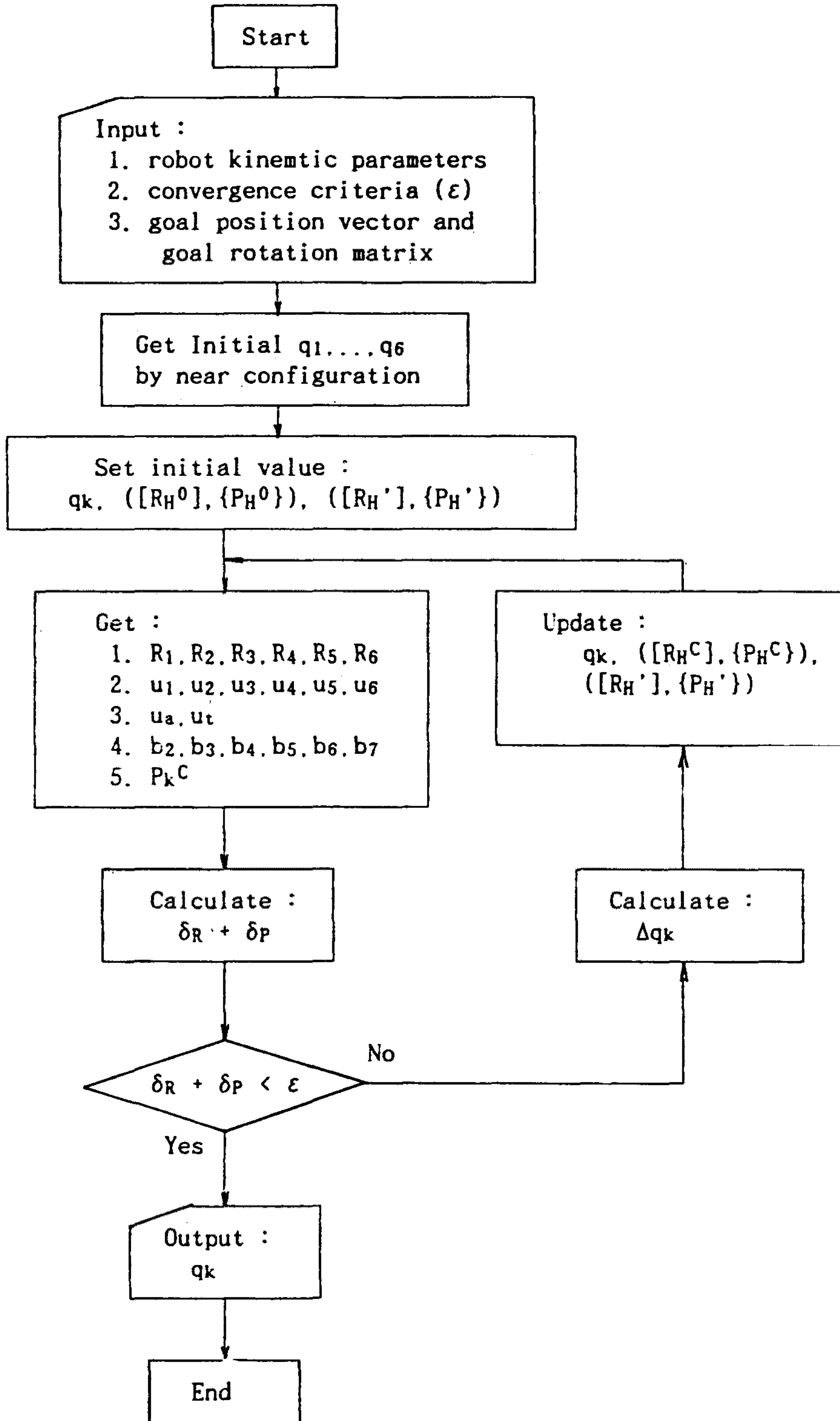


그림 4.6 편차함수 최소화 방법 흐름도

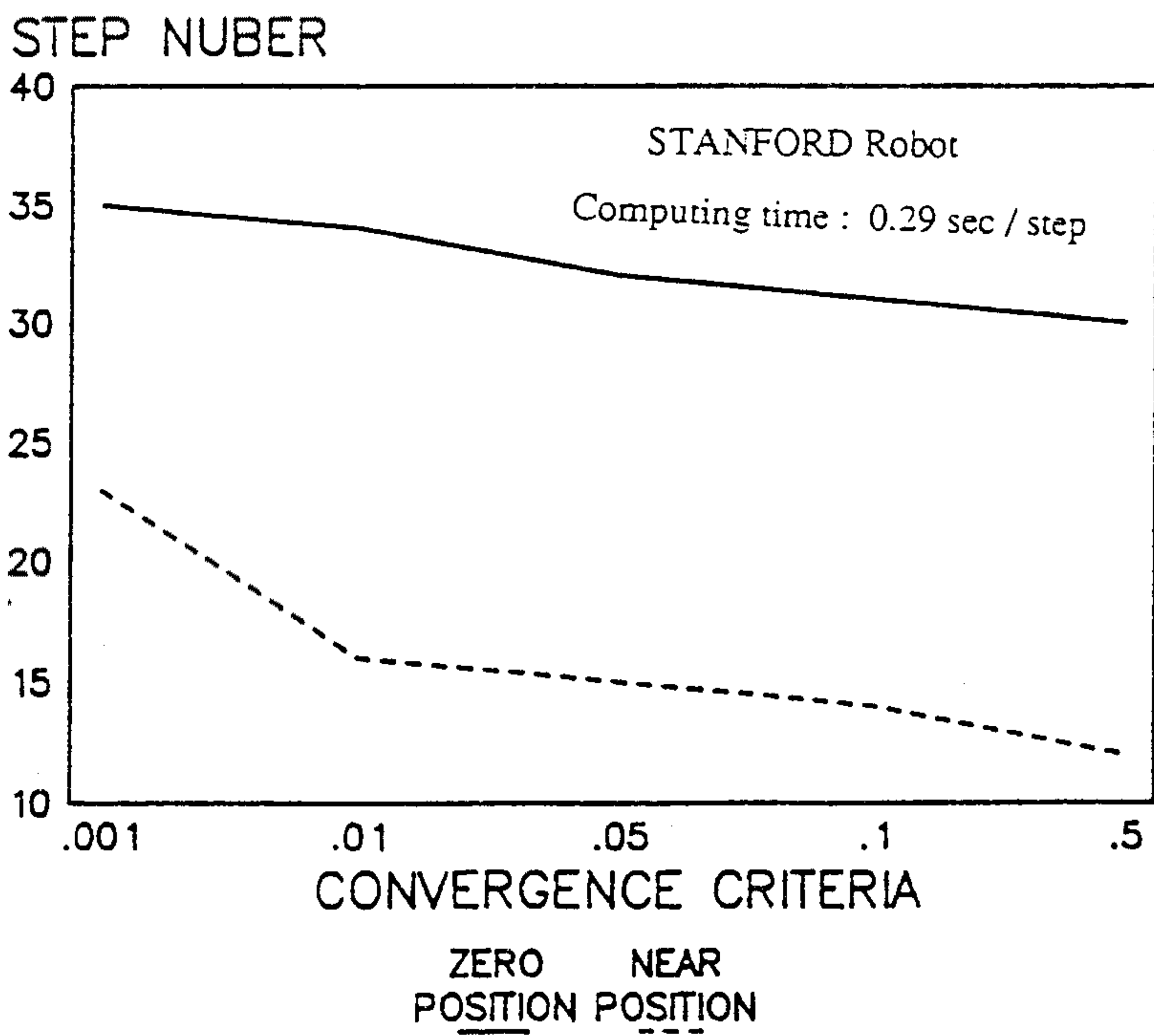
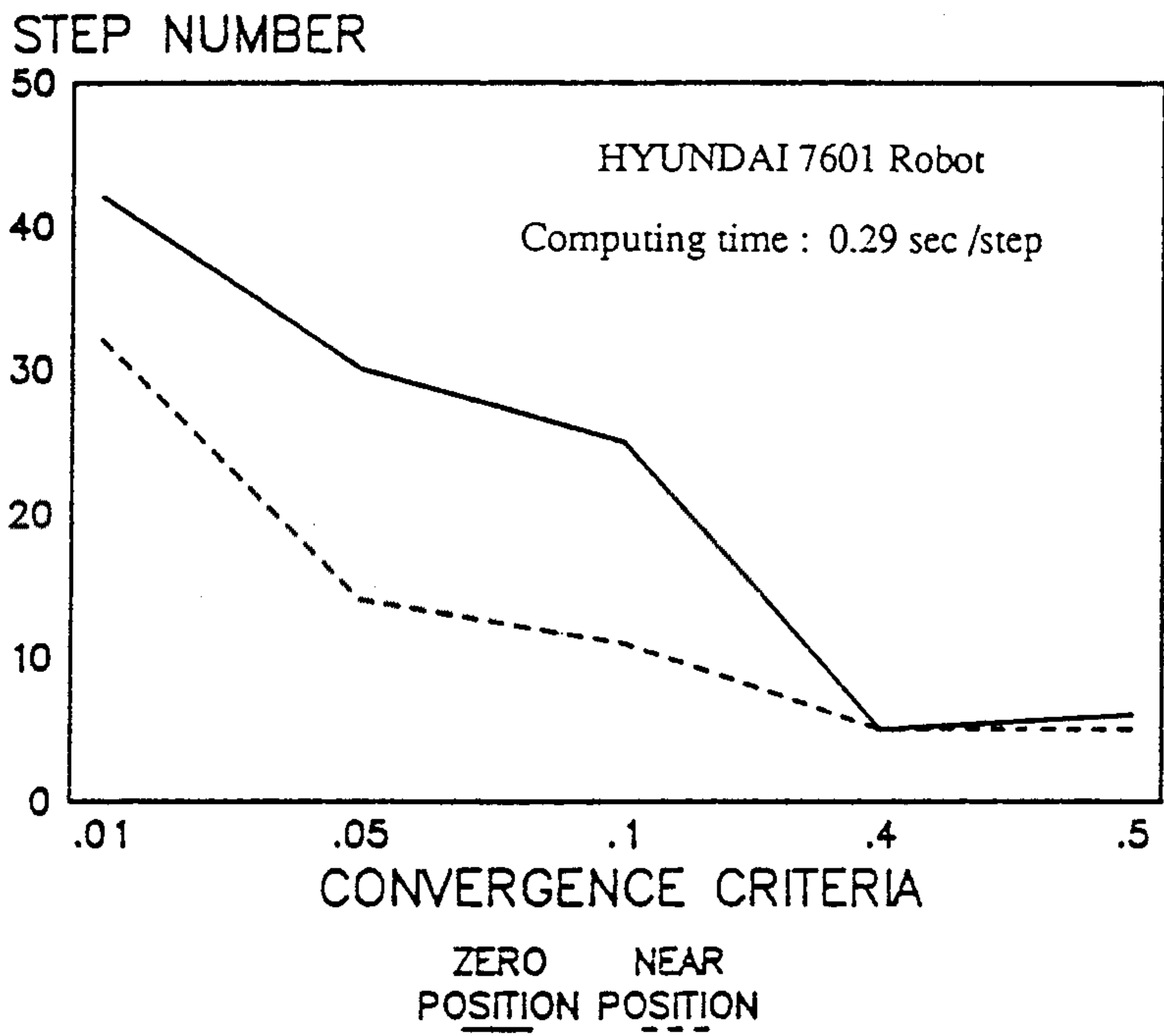


그림 4.7 근접 위치 방법과 Kazerounian 방법의 흐름도

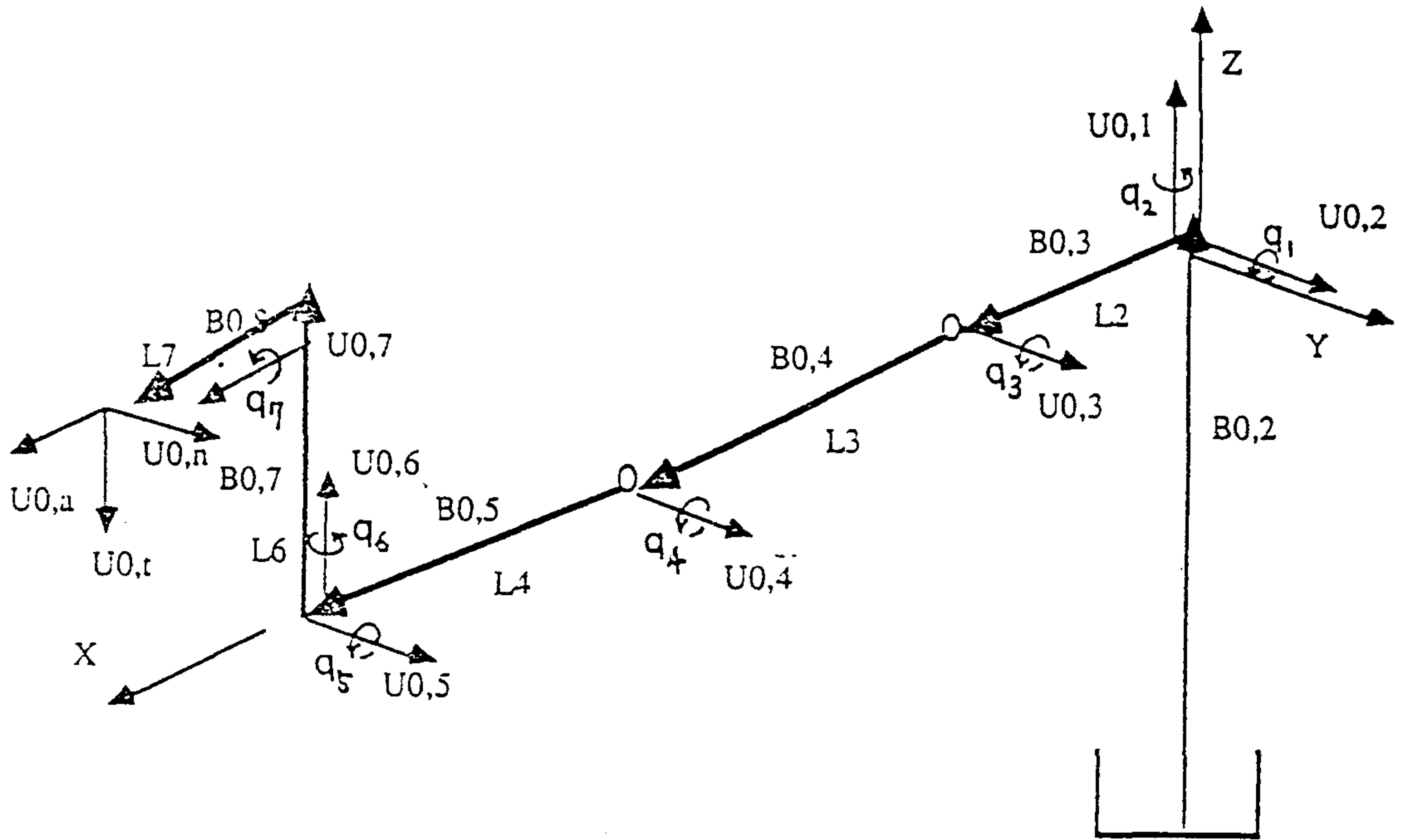


그림 4.8 잉여 자유도 로봇 (7R)의 초기 위치

Joint	Type	$u_{0,k}$	$b_{0,k+1}$
1	R	(0, 0, 1)	(0, 0, 0)
2	R	(0, 1, 0)	(L ₂ , 0, 0)
3	R	(0, 1, 0)	(L ₃ , 0, 0)
4	R	(0, 1, 0)	(L ₄ , 0, 0)
5	R	(0, 1, 0)	(0, 0, 0)
6	R	(0, 0, 1)	(0, 0, L ₆)
7	R	(0, 1, 0)	(0, L ₇ , 0)
		$u_{0,a} = (0, 1, 0)$	$u_{0,t} = (0, 0, -1)$

$$L_2 = 200. \quad L_3 = 300. \quad L_4 = 100.$$

$$L_6 = 100. \quad L_7 = 100.$$

(단위 : mm)

표 2. 잉여 자유도 로봇의 기구학 계수

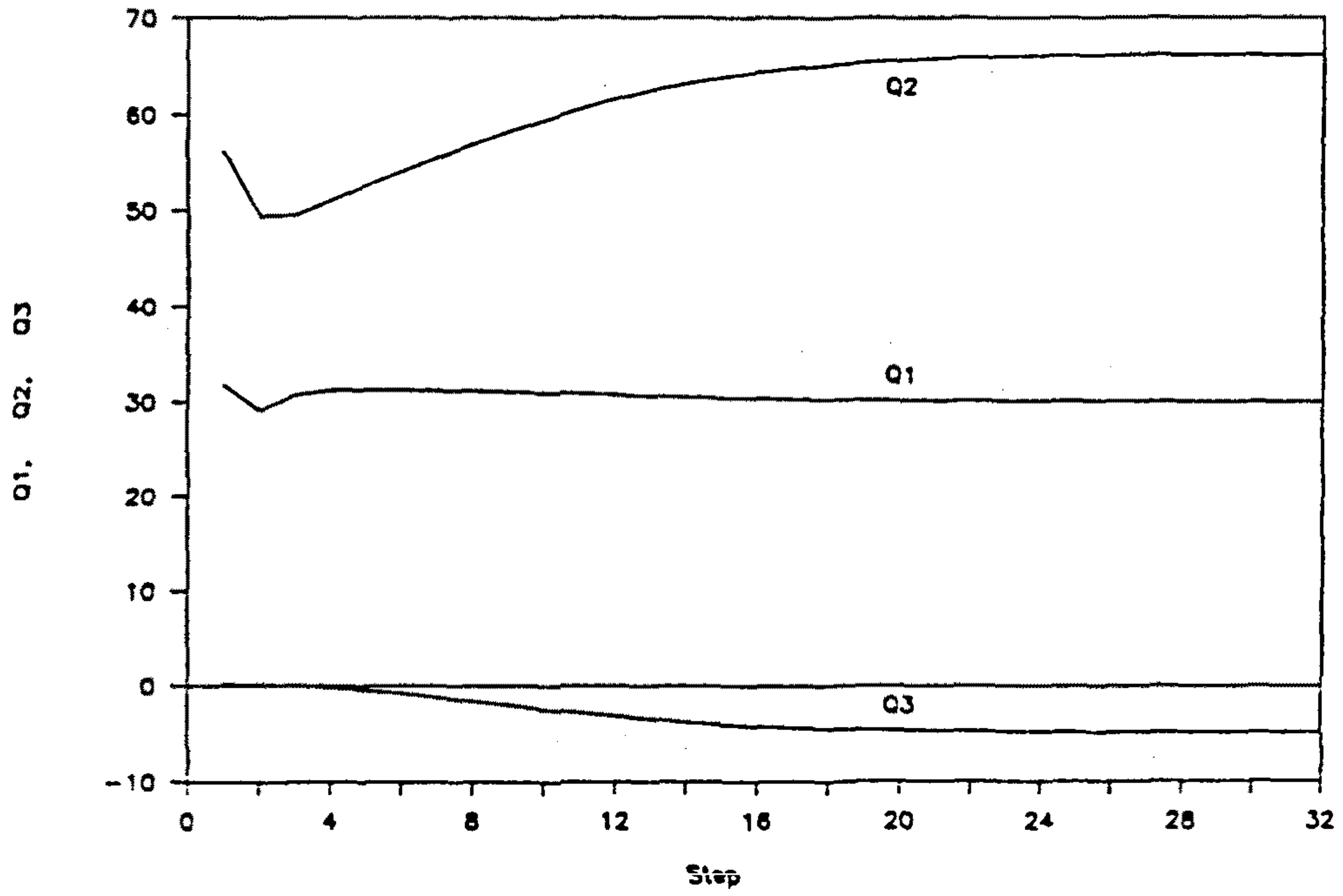


그림 4.9 Q1, Q2, Q3 의 수렴 곡선

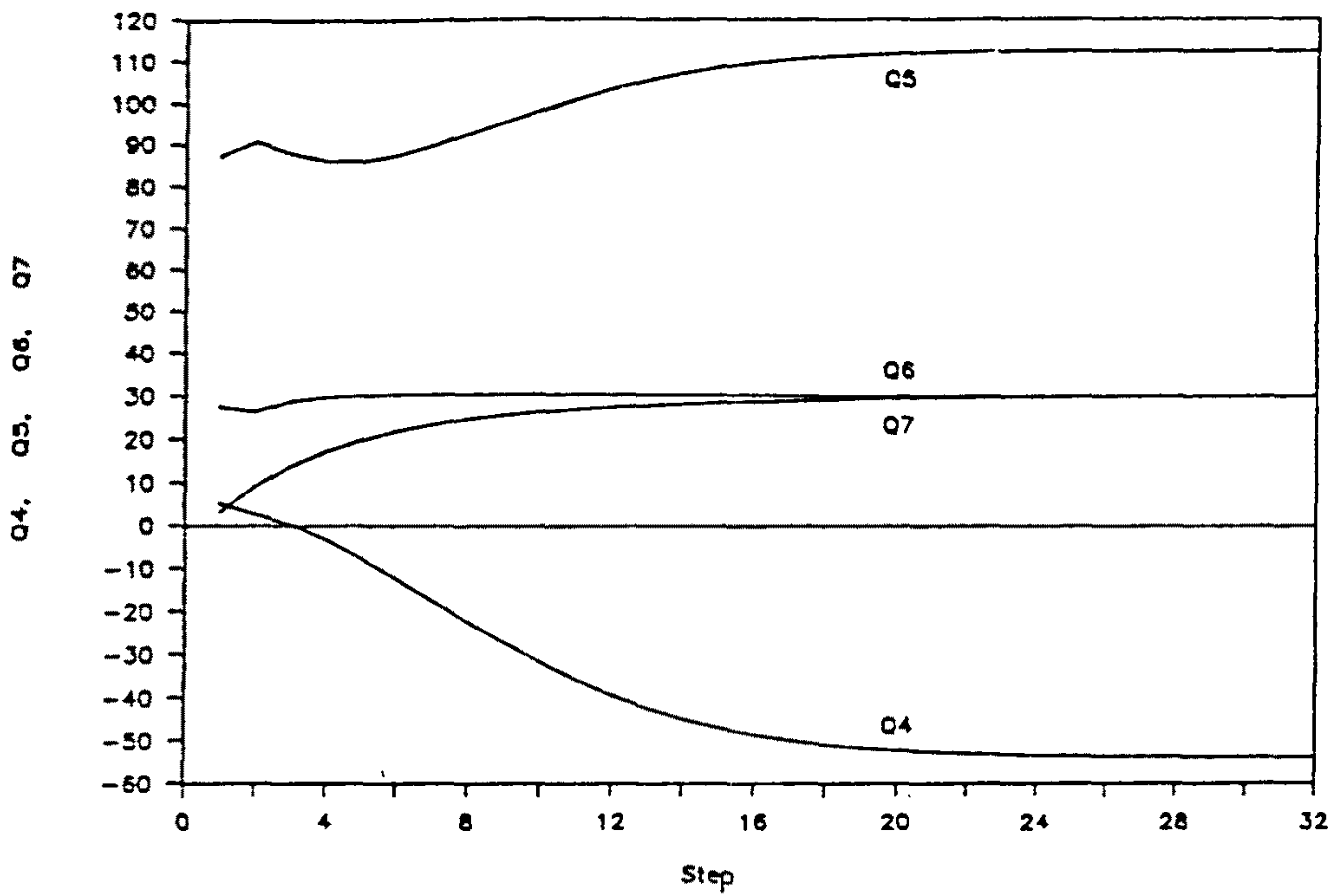


그림 4.10 Q4, Q5, Q6, Q7 의 수렴 곡선

제5 장 PROPS DATA BASE

제 1절 개요

DATABASE는 여러가지 SURFACE 에 대한 정보를 저장하고, 각종 로봇트 나 공구, 주변기기에 대한 정보와 PROPS 가 실행 과정에서 만들어 내는 TOOLPATH point data 나, collision check 한 GOOD PATH point data 등을 정확하게 저장하고, 효율적으로 관리하기 위하여 반드시 필요하다.

PORPS 에서는 DBMS(Data Base Management System) 으로 INFORMIX ESQL/C 를 이용하여 독자적인 DATA BASE 를 구축 하였다. ESQL/C 는 RDB(Relational Data Base)이다. RDB는 data를 저장할 때 computerrized file(textfile, binary file, etc.) 등으로 저장하는 것이 아니라 표 2 와 같이 record 와 field 로 구성 되어있는 table 단위로 data를 받아들인다. Field 는 저장 단위의 이름이고, record는 저장 field 들의 묶음이다. 받아들인 Data는 적당이 분산, 공유하여 정확한 정보를 유지하

	fiel	fiel	fiel	fiel
recor				
recor				
recor				

표 1 DATA TABLE

고, 또한 System에서 제공해주는 최소한의 operation을 이용해서 access 경로를 정의하지 않고도 data를 control 할수있다. 이런 software 를 RDB 라한다.

제 2절 DATA TABLE

DATABASE 는 세가지 부분으로 나누어져있다.

하나는 DIE에 관계된 table 들 이고, 두번째는 ROBOT와 TOOL, 그밖 의 주변장치에 관한 table 들이고, 세번째는 TASK 에 관계된 data 로 LAYOUT 조건, layout 에 따라 생성되는 TOOLPATH point , toolpath 에 서 surface의 순서를 나타내는 SEQUENCE, collision check 를 하는 GOODPATH등과 같이 PROPS 실행 과정에서 생성되는 data를 보관한 다.

PROPS DATABASE 에서 table들 간의 관계는 그림 1 과 같은 구조로 되어있다. Table 과 table 사이의 relation은 ID number 로 연결되어 있다. 이 ID 들은 table에서 필요한 row를 찾는데 사용된다. 경우에 따라서는 두가지의 ID 로 찾기도 한다. 이 ID 들은 data가 table에 insert될때 자동 적으로 unique 하게 부여 한다. ID 는 data를 효율적으로 관리하기 위한 내부적인 장치이다. 그러므로 end user는 ID number가 존재하는지 여부를 알 필요가 없다.

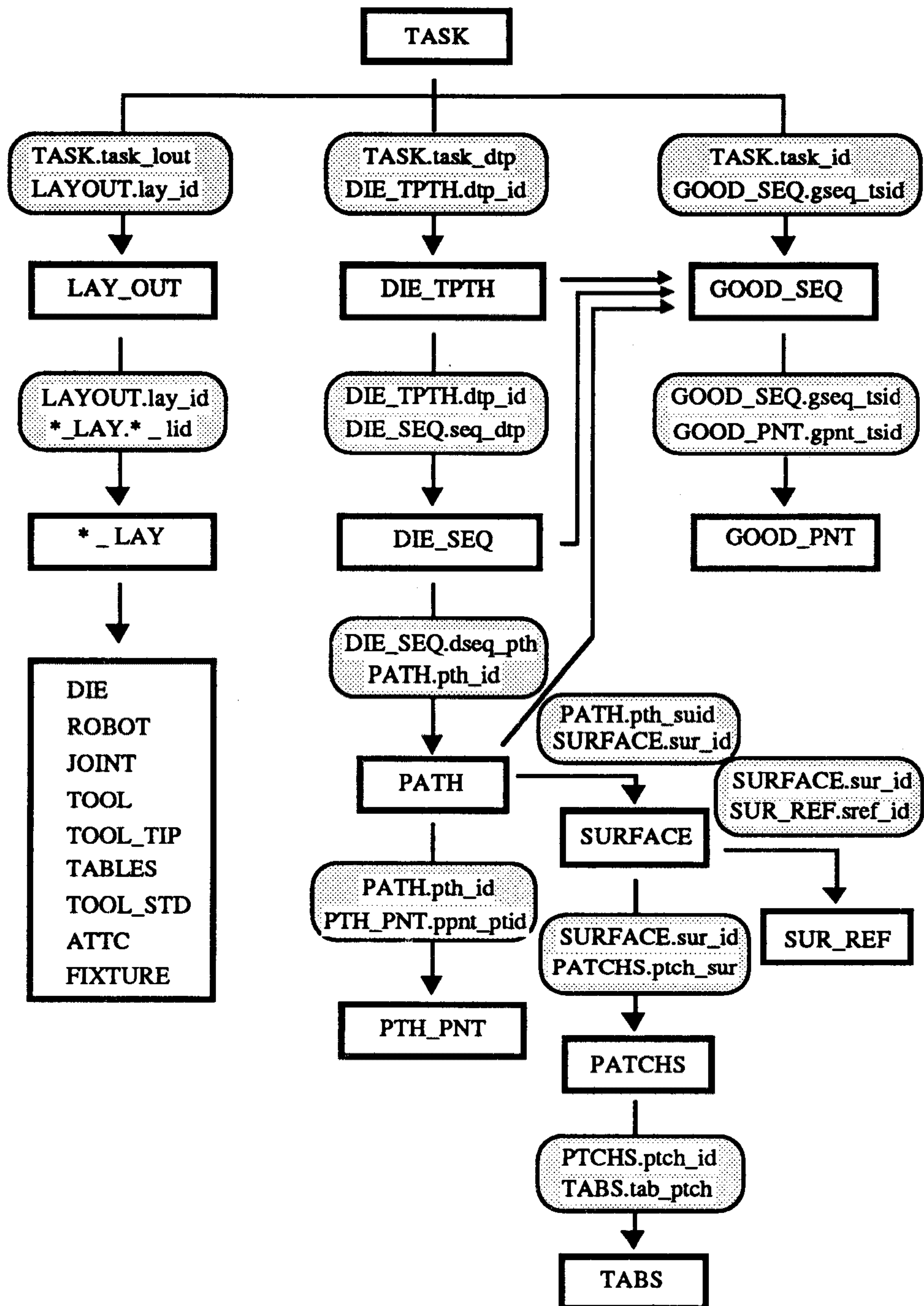


그림 1 Data table relation

1. SURFACE 에 관계된 TABLE

SURFACE 의 point data 와 SURFACE 를 그릴수 있는 정보를 가지고있다. SURFACE 에 대한 DATABASE는 DIE , SURFACE , PATCHS, TABS 로구성되어 있다.

(1) DIE table

DIE 에 관한 정보를 저장한다. DIE의 이름과 부여된 ID, 재질, 거칠기 등 DIE 에대해 참고 할수있는 사항들을 모두 저장한다.

die_id	int	DIE ID
die_name	char(10)	DIE NAME
die_matl	char(10)	DIE material
sur_treat	char(10)	surface treatment
irough	float	initial roughness of die material
trough	float	target roughness of die material

(2) SURFACE table

하나의 DIE는 여러개의 SURFACE로 구성되어 있다. 이 table 은 SURFACE 에 대한 정보를 가지고있다. SURFACE ID 는 DIE에 관계없이 surface 에 따라 unique 하게 부여된다. 그러므로 SURFACE ID 만을

가지고도 어느 DIE에 속하는지 알수있다.

sur_die	int	DIE ID
sur_id	int	SURFACE ID
sur_name	char(8)	SURFACE NAME
ni	smallint	number of patches in U
nj	smallint	number of patches in V
xmin	smallfloat	space occupide by surface
xmax	smallfloat	space occupide by surface
ymin	smallfloat	space occupide by surface
ymax	smallfloat	space occupide by surface
zmin	smallfloat	space occupide by surface
zmax	smallfloat	space occupide by surface

(3) PATCHS table

하나의 surface 당 필요한 patch 의 정보를 가지고있다. PATCHS 의 ID 도 SURFACE의 ID와 마찬가지로 unique 하게 부여된다. 하나의 surface 는 한개 이상의 patch를 가질 수 있다.

ptch_sur	int	surface id
ptch_id	int	patch id
nu	smallint	degree in U + 1
nv	smallint	degree in V + 1
npu	smallint	number of discretizing points in U
npv	smallint	number of discretizing points in V
xminp	smallfloat	space occupied by the patch
xmaxp	smallfloat	space occupied by the patch
yminp	smallfloat	space occupied by the patch
ymaxp	smallfloat	space occupied by the patch
zminp	smallfloat	space occupied by the patch
zmaxp	smallfloat	space occupied by the patch

(4) TABS table

하나의 PATCH 당 필요한 X, Y, Z data를 가지고 있다. 여기서는 PATCH_ID 와 TAB_ID 가 DOUBLE INDEXING된다. 만약 같은 PATCH에 속한다면 여러개의 column은 같은 tab_ptid 를 가지고 있다. 그 갯수는 $nu * nv$ 의 수와 같으며, 그 순서는 tab_id 의해 알수있다.

tab_ptch	int	patch id
tab_id	int	tab 자체 id
tabx	float	X coefficients of the polynomial
taby	float	Y coefficients of the polynomial
tabz	float	Z coefficients of the polynomial

(5) SUR_REF table

surface 의 기하학적인 정보를 제외한 것으로서 surface를 PROPS내부에서 handling하는데 필요한 정보를 가지고 있다. color 나 line type, picking 여부등을 저장한다.

sref_id	int	surface id
sref_col	smallint	color number
sref_vis	smallint	visibility
sref_pick	smallint	detectability
sref_ltype	smallint	line type of element
sref_ptype	smallint	point type of element

2. ROBOT 및 주변장치에 관계된 TABLE

ROBOT와 주변장치에 대한 기본적인 정보와 이것들을 그릴수 있는 FILE NAME 을 가지고 있다.

(1) ROBOT table

ROBOT 에 대한 기본적인 정보와 그릴수있는 FILE NAME 을 가지고 있다. Joint 에 대한 정보는 JOINT table 에 저장한다.

rbt_id	int	ROBOT ID
rbt_name	char(10)	ROBOT NAME
max_l_vel	float	MAX. linear velocity of arm end
inv_kine	char(10)	inverse kinematic method
convent	char(4)	transformation method
rbt_file	char(12)	drawing file of robot
rbt_jnum	int	ROBOT joint number

(2) JOINT table

ROBOT 의 joint에 관계된 정보만을 저장한다. jnt_rbtid 와 jnt_id 두개로 double index 되어있다. 만약 같은 로봇트이면 jnt_rbtid 가 같다.

jnt_rbtid	int	ROBOT ID
jnt_id	int	JUOINT 자체 ID
jt_max	float	MAX. JOINT value
jt_min	float	MIN. JOINT value
jt_max_vel	float	max. joint velocity
theta	smallfloat	the first parameter of D.H
jnt_d	smallfloat	the second parameter of D.H
jnt_a	smallfloat	the third parameter of D.H
alpha	smallfloat	the forth parameter of D.H

(3) TOOL table

TOOL 에 대한 정보를 가지고 있다.

tool_id	int	TOOL NAME
tool_name	char(10)	TOOL NAME
mot_type	char(10)	TOOL TYPE NAME
max_rpm	int	MAX. R.P.M. of the method
max_force	int	Max. applied force of the tool
tool_file	char(12)	drawing file of tool

(4) TOOL_TIP table

TOOL TIP 에 대한 정보를 가지고 있다.

ttip_id	int	TOOL TIP ID
ttip_name	char(8)	tool tip name
abras	char(2)	abrasive of grinding wheel
grain_sz	int	gain size of G.W.
bond	char(2)	bond of G. W.
grade	char	grade of G.W.
structur	char	structer of G.W.
life_tim	float	life time of G.W.
tpth_wd	float	tool path width
trpm	float	recommended RPM of G.W. tip
tvel	float	recommended velocity of G.W. tip
tfd_rate	float	recommended feed of G.W. tip
tforce	float	recommended applied forth of G.W tip
cool	char(6)	냉각제
ttip_file	char(12)	tool tip file name

(5) TABLES table

table에 ID 를 부여하고, TABLE name 과 table 을 그릴수있는 file name 을 가지고 있다.

tbl_id	int	TABLE ID
tbl_name	char(8)	TABLE NAME
tbl_file	char(12)	TABLE 을 그릴수 있는 file name

ATTC (automatic tool tip changer), TSTD(tool stand), FIXTUR(fixture) table 도 TABLES table 과 같은 구조로 구성되어 있다. 이 주변기기들은 surface 와 같이 일정한 format 을 가지기 어렵고, 이것을 그리는 routine을 일반화 한다는 것 또한 어려운 일이다. 그러므로 일정한 정보를 저장할 수 가 없으므로 각각의 주변기기들을 구별해줄 ID 와 해당되는 주변기기를 그릴 수 있는 file name 정도로 최소한의 정보만 저장하도록 하였다.

3. TASK 에 관계된 TABLE

JOB 의 단위인 TASK당 필요한 DATA를 저장한다. TASK에 대해 independent 한 LAY_OUT, TOOLPATH 의 정보와 PROPS 의 최종 결과인 TEXT 로된 output file 로 출력할수있는 POINT 에 대한 정보와 이를 control 하는 GOOD PATH table 을 가진다.

(1) TASK table

JOB 의 단위인 TASK, LAY-OUT , TOOL-PATH의 이름을 저장한 TABLE.

task_id	int	task ID
task_name	char(8)	TASK ID
task_lout	int	LAYOUT ID
task_dtp	int	DIE 전체의 TOOLPATH ID
task_date	date	TASK가 만들어진 날짜

(2) LAYOUT table

LAYOUT 에 관한 정보를 저장한다. LAYOUT.lay_id 의 숫자는 각 table 의 ***_LAY.*_lid 의 숫자와 일치한다. 이 두 숫자가 같으면 하나의 LAYOUT를 나타내는, LAYOUT.lay_id 에 속한 하위 level 정보이다.

lay_id	int	LAYOUT ID
lay_name	char(8)	LAYOUT NAME
lay_date	date	LAYOUT 이 만들어진 날짜

(3) DIE_LAY table

DIE 의 name 과 그 position data 를 가지고 있다.

die_llid	int	DIE LAYOUT ID
die_lid	int	DIE ID
die_x	float	DIE X position
die_y	float	DIE Y position
die_z	float	DIE Z position
die_rol	float	DIE rol value
die_pit	float	DIE pit value
die_yaw	float	DIE yaw value

RBT_LAY, TOOL_LAY, TTIP_LAY, TBL_LAY, TSTD_LAY, ATTC_LAY, FIXT_LAY table 의 구조역시 DIE_LAY table 의 구조와 같다. 이 table 들은 해당되는 주변기기의 ID와 해당되는 LAYOUT ID , 그리고 그것들의 POSITION DATA 인 x, y, z, rol, pit, yaw 값만 을 저장하고 있다. *_lid 는 같은 LAYOUT 이면 모두 같은 번호가 들어간 다.

(4) TOOL_PTH table

DIE 전체 TOOLPATH 에 대한 정보로 DIE TOOLPATH 에 UNIQUE 한 ID 를 부여하고 DIE TOOLPATH NAME 과 만들어진 날짜를 저장한다. TOOLPATH 의 순서는 DIE_SEQ TABLE 에 저장한다.

tpth_id	int	DIE TOOLPATH ID
tpth_name	char(8)	DIE TOOLPATH NAME
tpth_date	date	TOOLPATH 가 만들어진 날짜

(5) DIE_SEQ table

각각의 SURFACE 에 존재하는 TOOLPATH 의 순서를 나타낸다. TOOLPATH 는 DIE 를 그릴때와 같은 순서로 surface 를 polishing 하는 것이 아니므로 TOOLPATH 할때의 순서를 따로 정해주어야 한다. seq_dtp 와 seq_id 에 의해 double indexing 된다. 만약 같은 TOOLPATH 라면 dseq_tpid 가 같게 된다.

dseq_tpid	int	DIE TOOLPATH ID
dseq_id	int	TOOLPATH 의 순서를 부여하는 ID
dseq_pth	int	해당되는 PATH ID

(6) PATH table

1개의 SURFACE에 속해 있는 TOOLPATH 정보를 가지고 있다. 하나의 surface는 여러개의 path 를 가질수 있다. Pth_id는 unique하므로 pth_id 만을 가지고도 어느 surface인지를 알수있다.

pth_id	int	unique 한 PATH 자체 ID
pth_suid	int	해당되는 SURFACE의 ID
pth_name	char(10)	PATH NAME
pth_type	char(3)	PATH TYPE
pth_intv	float	PATH INTERVAL
pnt_intv	float	POINT INTERVAL
tcp_vel	smallint	현재 point 의 속력
pth_tool	char(10)	가공에 필요한 tool name
pth_ttip	char(10)	가공에 필요한 tool tip name
sur_high	float	surface 진입 높이
pth_date	date	TOOLPATH 가 만들어진 날짜

(7) PATH_PNT table

한개의 SURFACE 에 속한 TOOLPATH 의 point data 를 가지고있다. ppnt_ptid 와 ppnt_id 로 double index 되어있다.

ppnt_ptid	int	해당하는 PATH ID
ppnt_id	int	PATH_PNT 자체 ID
ppnt_x	float	path point의 X 좌표
ppnt_y	float	path point의 Y 좌표
ppnt_z	float	path point의 Z 좌표
ppnt_rol	float	path point의 rol 값
ppnt_pit	float	path point의 pit 값
ppnt_yaw	float	path point의 yaw 값

(8) GOOD_SEQ table

각각의 surface 에 존재하는 toolpath 의 순서를 나타낸다. gseq_strt는 해당되는 surface가 GOOD_PNT table 에서 시작하는 ID를 저장한다. 이것은 surface단위로 polishing 할수있도록 하기 위해서 이다. gseq_done 은 해당되는 surface가 polishing check를 했는가 안 했는가 여부를 저장한다. seq_dtp 와 seq_id 에 의해 double indexing 된다.

gseq_tsid	int	TASK ID
gseq_id	int	GOOD PATH 순서
gseq_pth	int	해당되는 PATH ID
gseq_strt	int	START ID IN GOOD_PNT TABLE
gseq_done	int	check. GOODPATH is made or not

(9) GOOD_PNT table

COLLISION 을 CHECK 한 GOOD-PATH POINT 를 저장한다. gpnt_col 은 collision 이 check되었는 가 여부를 저장해주고, gpnt_del, gpnt_tool, gpnt_ttip, gpnt_mot 등은 각 point 마다 바뀔 여지가 있는 data 들에 대해서 로봇트에 넘겨주기 위해서 한번 더 저장하여 준다.

gpnt_tsid	int	TASK ID
gpnt_id	int	GOOD PATH 자체의 ID
gpnt_x	float	good path X position
gpnt_y	float	good path Y position
gpnt_z	float	good path Z position
gpnt_rol	float	good path rol value
gpnt_pit	float	good path pit value
gpnt_yaw	float	good path yaw value
gpnt_vel	float	POINT VELOCITY
gpnt_col	int	CHECK COLLISION
gpnt_del	float	delay time
gpnt_tool	float	tool ID
gpnt_ttip	float	tool tip ID
gpnt_mot	int	motion type

제 3 절 DATA 의 내부처리

Task, Layout, Toolpath 는 독립적이다. Task는 name 에 따라 하나의 good path(output)을 가진다. Task가 delete되면 해당되는 good path는 없어 지지만 layout이나 toolpath 는 그 자체가 없어지는 것이 아니라 task와의 relation만 없어지는 것이다. 새로운 task 를 만들때는 새로운 layout이나 toolpath 를 만들수도 있지만 기존에 만들어져있는 layout이나 tool path 를 그냥 선택해도 좋다.

제일먼저 TASK name 이 선택되면 TASK table 에서 해당되는 row 를 찾아서 그 row 에 있는 Layout 과 Toolpath 의 ID 를 가지고 온다.

이 layout id 를 가지고 *_LAY table에서 위치 data를 가져오고, 각각의 해당되는 table 에서 그리는 file name을 가져와 layout에서 지정된 모든 것들을 그린다.

Toolpath id 는 DIE_SEQ 에 있는 surface 의 순서에 따라 PATH table 에 있는 point data 를 읽어 들여 toolpath 를 그린다.

Simulation 과정에 들어가게 되면 point data를 double linked list 에 저장 하고 하나씩 읽어 나가면서 collision check를 하고 collision 이 일어나는 point 의 rol, pit, yaw값을 변화시켜 주던가, 아니면 insert 혹은 delete 한다. Insert 나 delete되는 point는 double linked list에 저장되므로 pointer를 움직여 손쉽게 할수있다. 만약 simulation도중에 저장하고자 한다면, 이 linksd list 에 index를 줘서 그대로 GOOD_PNT table 에 저장한다.

PROPS에 새롭게 들어온 경우에 전에 그만 두었던 task를 선택하면 GOOD_PATH table에서 읽어들이는 point값들을 linked list에 저장한뒤 그리면 그전까지 했던 작업상태를 그대로 재현 할수있다.

4.절 DB 실용화를 위한 CATIA format interface

일반적으로 CATIA 에서는 그림 2 와 같은 format 으로 data 를 저장한다. PROPS DATABASE 에서는 CATIA 와의 interface를 위해 SURFACE, PATCHS, TABSFKSMS 라는 세개의 table에 나누어 저장한다. 맨처음 하나의 DIE 에 속해있는 SURFACE 의 NAME을 읽어들이고, 첫번째 block에서 U , V 방향의 patch 수에 해당하는 값을 읽어들이 SURFACE table 에 저장하고, 두번째 block 에서는 U, V 방향의 order 값 과 U, V 방향의 discretizing point number 값을 읽어 들여서 PATCHS table 에 저장한다. 세번째 block 에서 부터 그 다음 SURFACE name 이 나오기 전 까지의 block 은 die 를 drawing 할수 있는 X, Y, Z 값에 해당한다. 이값들은 TABS table 에 저장한다.

SURFACE TYPE = 5
 SURFACE ID = *SUR

Mathematical Block

NI	NJ	ISENS	
XMIN YNIN ZMIN		XMAX YMAX ZMAX	

1st Patch

LACR	NU	NV	NPU
NPV	0	0	0
XMINP YMINP ZMINP		XMAXP YMAXP ZMAXP	
TABX			
TABY			
TABZ			

2nd Patch

Same as above

그림 2 CATIA surface output format

제 5 절 DATA BASE의 개선점

본 4차년도 보고서의 database 는 첫번째, SURFACE data 를 KIST CAD/CAM 실에서 개발된 CASSET 의 database 로부터 독립시키므로서, 그동안 문제가 되어왔던 DATABASE의 독립문제를 완전히 실현하였다.

두번째로 INFORMIX database system 을 사용하여 access를 좀더 효율적이고 간단하게 하였다. 3차년도의 경우 원하는 data를 가져오려면 복잡한 address계산을 한뒤에야 가져올수 있었으나, 본 database는 원하는 변수명만 사용하면 간단하게 가져올수 있기 때문에 효율성이 좀더 높아졌다.

세번째로는 data들간의 share 과 분산이 효율적으로 이루어져서 Memory 활용을 효율적으로 하였다는 것이다.

제 6 장 결 론

본 4차년도 보고서는 PROPS 의 산업체 이양을 위하여, 시스템의 호환성 및 확장성 증대, RDB 를 이용한 가공정보의 효율적 관리, 사용자 interface 강화를 위한 전체 menu 의 재구성 등을 주요 목표로 하는 연구가 진행되었으며 구체적인 사항은 다음과 같다.

1. 실제 robot system에서의 polishing 작업을 미리 computer상에서 graphic system을 이용한 simulation 기법으로 수행하여 실제 robot 를 사용할 시의 teach-in 작업으로 인한 문제점을 해결하여 경비의 절감,시간의 단축등을 피할 수 있게되었다.
2. Layout, toolpath, simulation 으로 구성된 task 를 작업의 단위로 하는 RDB 를 구축하여, simulation program 으로서의 유연성 및 확장성을 증대시켰다.
3. RDB 관리 및 부분작업의 독립화를 고려한 PROPS menu 구조의 변경으로 사용자 interface 를 강화하였다.
4. 가공 simulation 중 teaching pendant 를 이용하여 가공경로의 수정,삽입,삭제를 가능하게 하였다.

5. Collision checker의 개발로 인해서 실제 polishing시 일어날 수 있는 공구의 간섭 문제를 해결하기위한 방법을 제시하였다.

6. 외부의 CAD modeller data 및 geometric data 를 import 하는 기능을 추가함으로써 PROPS 의 modeling 기능을 보완하고 시스템의 호환성을 증대시켰다.

REFERENCES

1. Faux, I. D. and Pratt, M. J., "Computational Geomtry for designing and Manufacturing", Ellis Horwood Ltd., 1981
2. Ding Qiulin and B. J. Davies, "Engineering Geometry for Computer-aided Design and Manufacturing", Ellis Horwood Ltd.,1987
3. Newman, W. M. and Sproull, R. F., "Principles of Interactive Computer Graphics (2nd ed.)", McGraw-Hill, 1979
4. Richard P. PAul, "Robot manipulators", the MIT press,1982
5. M. Kim, "Parameteridentifikation der kinematischen Kenngrößen von Industrie-robotern", Dissertation, Springer Verlag, Berlin,1987
6. M. Kim, H. Chang, H. You, S. Cho, G. Spur, "A new Calibration System", CIRP Berlin, 1990, submitted
7. John J. Craig, Introduction to Robotics, Eddison Wesley, 1955
8. Wesley E. Snyder, Industrial Robots, Prentice Hall,1985
9. K. C. Gupta, K. Kazerounian, "Improved Numerical Solution of Inverse Kinematics of robot", IEEE,1985
10. Kazem Kazerounian, "On the Numerical Inverse Kinematic of Robot Manipulators", ASME,1987

11. K. C. Gupta, "Kinematic Analysis of Manipulator Using the Zero Reference Position Description", The International Journal Of Robotic Research,1986
12. R.P. Paul, C. N. Stevenson,1983, " Kinematics of Robot Wrists" , The International Journal of Robotics Research
13. Jorge Angeles,1985, " On the Numerical Solution of the Inverse Kinematic Problem", The International Journal of Robotics Research
14. Mauro Benati, 1982," The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms", ASME
15. A.A. Goldenberg, D.L. Lawrence,1985, " A Generalized Solution to the Inverse Kinematics of Robotic Manipulators", Journal of dynamic System, Measurement and control
16. Kazem Kazerounian, 1987, " On the Numerical Inverse Kinematics of Robotic Manipulators", ASME
17. K.C. Gupta, K. Kazerounian,1985, " Improved Numerical Solution of Inverse Kinematics of robot", IEEE

여 백

A. DATABASE Routine

1. Create table

(1) cretable create database PROPS
 create all table and create all index

2. Control table

id = 1 insert routine
id = 2 fetch routine
id = 3 update routine
id = 4 delete routine

(1) DIE table

condie(id, die_id, die_name, die_matl, sur_treat, ough, trough)

(2) SURFACE table

consur(id, sur_die, sur_id, sur_name, ni, nj, xmin, xmax, ymin, ymax,
zmin, zmax)

(3) SUR_REF table

consref(id, sref_id, sref_col, sref_vis, sref_pick, sref_ltyp, sref_ptyp)

(4) PATCHS table

`conptch(id, ptch_sur, ptch_id, nu, nv, npu, npv, xminp, xmaxp, yminp, ymaxp, zminp, zmaxp)`

(5) TABS table

`contab(id, tab_ptch, tab_id, tabx, taby, tabz)`

(6) ROBOT table

`conrbot(id, rbt_id, rbt_name, max_l_vel, inv_kine, convent, rbt_file, rbt_jnum)`

(7) JOINT table

`conjnt(id, jnt_rbtid, jnt_id, jt_max, jt_min, jt_max_vel, theta, jnt_d, jnt_a, alpha)`

(8) TOOL table

`contool(id, tool_id, tool_name, mot_type, max_rpm, max_force, tool_file)`

(9) TOOL_TIP table

`conttip(id, ttip_id, ttip_name, abrase, grain_sz, bond, grade, structur, life_tim, tpth_wd, trpm, tvcl, tfd_rate, tforce, cool, ttip_file)`

(10) TABLES table

conttbl(id, tbl_id, tbl_name, tbl_file)

(11) ATTC table

conattc(id, attc_id, attc_name, attc_file)

(12) TSTD table

constd(id, tstd_id, tstd_name, tstd_file)

(13) FIXTUR table

confixt(id, fixt_id, fixt_name, fixt_file)

(14) TASK table

contask(id, task_id, task_name, task_lout, task_dtp, task_date)

(15) LAYOUT table

conlout(id, lout_id, lout_name, lout_date)

(16) DIE_LAY table

conldie(id, die_lid, die_llid, die_x, die_y, die_z, die_rol, die_pit, die_yaw)

(17) RBT_LAY table

conlrbt(id,rbt_lid, rbt_llid, rbt_x, rbt_y, rbt_z, rbt_rol, rbt_pit, rbt_yaw)

(18) TOOL_LAY table

conltol(id,tol_lid, tol_llid, tol_x, tol_y, tol_z, tol_rol, tol_pit, tol_yaw)

(19) TTIP_LAY table

conlttp(id,ttp_lid, ttp_llid, ttp_x, ttp_y, ttp_z, ttp_rol, ttp_pit, ttp_yaw)

(20) TBL_LAY table

conltbl(id,tbl_lid, tbl_llid, tbl_x, tbl_y, tbl_z, tbl_rol, tbl_pit, tbl_yaw)

(21) TSTD_LAY table

conltst(id,tst_lid, tst_llid, tst_x, tst_y, tst_z, tst_rol, tst_pit, tst_yaw)

(22) ATTC_LAY table

conlatc(id,atc_lid, atc_llid, atc_x, atc_y, atc_z, atc_rol, atc_pit, atc_yaw)

(23) FIXT_LAY table

conlfix(id,fix_lid, fix_llid, fix_x, fix_y, fix_z, fix_rol, fix_pit, fix_yaw)

(24) TOOL_PTH table

contpth(id, tpth_id, tpth_name, tpth_date)

(25) DIE_SEQ table

condseq(id, dseq_tpid, dseq_id, dseq_pth)

(26) PATH table

**conpth(id, pth_id, pth_suid, pth_name, pth_type, pth_intv, pnt_intv,
tcp_vel, pth_tool, pth_ttip, sur_high, pth_date)**

(27) PTH_PNT table

**conppnt(id, ppnt_ptid, ppnt_id, ppnt_x, ppnt_y, ppnt_z, ppnt_rol,
ppnt_pit, ppnt_yaw)**

(28) GOOD_SEQ table

condseq(id, dseq_tsid, dseq_id, dseq_pth, dseq_strt, dseq_done)

(29) GOOD_PNT table

**congpnt(id, gpnt_tsid, gpnt_id, gpnt_x, gpnt_y, gpnt_z, gpnt_rol,
gpnt_pit, gpnt_yaw, gpnt_vel, gpntcol, gpnt_del, gpnt_tool, gpnt_ttip,
gpnt_mot)**

B. AutoCAD DXF file 의 import

PROPS layout 을 구성하는 구성장치 및 요소 들의 3 차원 기하학적 형상을 모델링하는 작업이 필요하다. 그러나 PROPS 에서는 일반화된 CAD modeler 기능을 갖고있지 않다. 이를 극복하기 위하여 비교적 단순한 형상의 모델링 기능을 지원하는 AutoCAD 를 이용하여 PROPS 의 part modelling 기능을 보완하는 방법을 고안하였다.

AutoCAD 의 '3D objects' menu 에서 box 혹은 cone (cylinder 포함) 을 모델링하여 AutoCAD 의 data 교환 file 인 DXF 을 생성하고 이를 PROPS 에서 그리기 위한 data 로 변환한다. PROPS drawing routine 은 AutoCAD DXF file 을 직접이용하지 않기때문에 이 변환과정이 반드시 필요하다.

PROPS 의 part geometry 는 금형을 제외하고는 모두 box 와 cone 의 조합체로 이루어져 있기 때문에 box 와 cone 의 import 기능을 부가함으로써 모든 geometry 의 모델링이 가능하게 된다.

B.1 Box

B.1.1 Box 의 modeling

AutoCAD Release10 에서 box 를 모델링하는 방법은 다음의 3 가지로 분류가 가능하다.

- 1) 4개 옆면의 모서리에 해당하는 POLYLINE 1개와 윗면과 아랫면에 해당하는 3DFACE 2개로 정의.

2) Box 의 면 6 개를 모두 3DFACE 로 정의

3) 3DMESH 방법을 이용한 POLYLINE 1 개로 정의

본 보고서에서는 cone 과의 일관성을 고려하여 AutoCAD 사용자가 modeling 하기 간단한 3) 의 방법을 선택하였다.

Box 는 m 방향 mesh 6개, n 방향 mesh 6개의 3Dmesh 로 mapping polyline 으로 modeling 된다. 따라서 DXF file 에 쓰여진 box 의 vertex 수는 18 개가 된다. 이는 data format 은 간단하지만, 불필요한 data 를 많이 갖고있는 단점이 있다.

B.1.2 Box DXF data format

AutoCAD 에서 dxfout 시에 불필요한 data 를 줄이기위해 symbol table data 를 제외한 entity data 만을 저장하는 것이 바람직하다. Dxf file 은 group code 와 following value 로 이루어져, file 상의 두 line 이 하나의 set 가된다. 다음은 box 의 dxf file 의 한 예이다.

0	
SECTION	:도면요소 지정
2	
ENTITIES	: Attribute tag
0	
POLYLINE	:도면 entity 선언
8	
0	: Layer name
66	
1	: Entity follow 표시문자(fixed)
10	

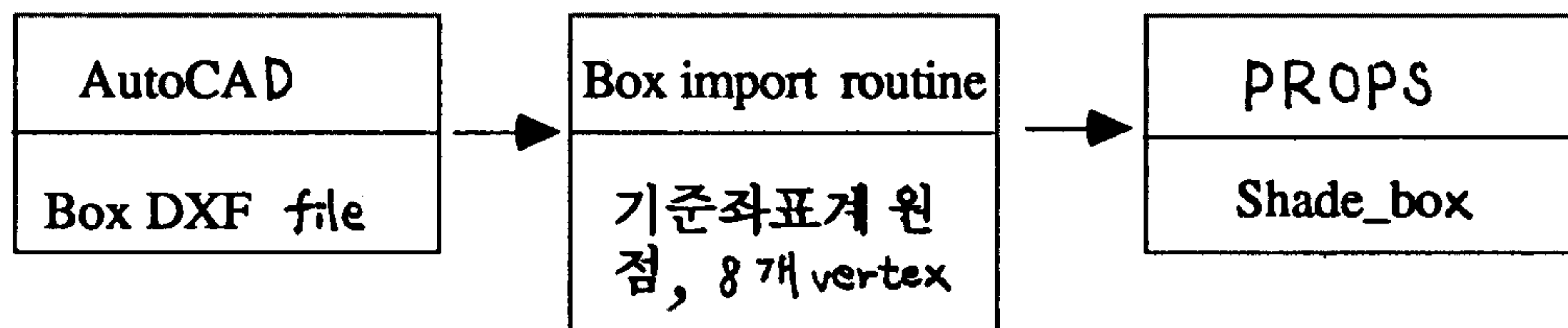
0.0 : 주요 X 좌표
 20
 0.0 : 주요 Y 좌표
 30
 0.0 : 주요 Z 좌표
 70
 16 : 3D mesh 지정 (box 의 ID 로 사용가능)
 71
 6 : m 방향 mesh 갯수
 72
 3 : n 방향 mesh 갯수

0
 VERTEX i (i = 1,16) : Vertex 지정
 8
 0 : Layer name
 10
 X coord. : Vertex 의 x coord.
 20
 Y coord. : Vertex 의 y coord.
 30
 Z coord. : Vertex 의 z coord.
 70
 64 : 3D mesh 의 정점

0
 SEQEND : End of sequence
 8
 0
 0
 ENDSEC : End of section
 0
 EOF : End of file

B.1.3 PROPS 용 box data 로의 변환

PROPS 에서 box 를 그리는routine 은 기준 좌표계에 대한 8 개 vertex 의 x,y,z data 를 필요로한다. 이를 위하여 box 의 dxf file 에서 vertex 의 상관관계를 이용하여 box 기준좌표계의 원점을 구하고, 중복 되지 8 개 vertex data 로 PROPS routine 'SHADE_BOX' 에서 shading 처리 가 된 3D box 를 그린다.



B.2 Cone

B.2.1 Cone 의 modeling

Cone 도 box 와 같은 방법으로 1 개의 polyline 으로 정의한다. 윗면,아랫면의 원을 근사하는 다각형의 vertex 수에 따라 DXF file 의 vertex 수가 결정된다. 단 다각형의 꼭지점 수는 짝수로 한다.

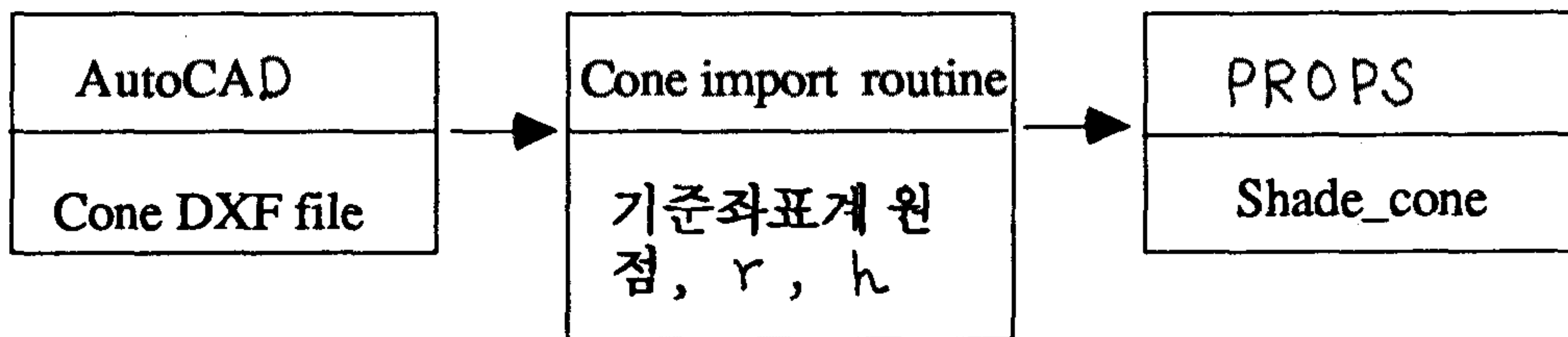
B.2.2 Cone DXF data format

Cone 의 dxf data format 은 앞의 entity 선언부를 제외하고는 box 같다. 밑면의 다각형 꼭지점이 N 개인 경우, dxf data 이 vertex 수는 $2 * N$ 이 된다. Entity 선언부의 data 는 다음과 같다.

0	
SECTION	: 도면요소 지정
2	
ENTITIES	: Attribute tag
0	
POLYLINE	: 도면 entity 선언
8	
0	: Layer name
66	
1	: Entity follow 표시문자(fixed)
10	
0.0	: 주요 X 좌표
20	
0.0	: 주요 Y 좌표
30	
0.0	: 주요 Z 좌표
70	
48	: 3D mesh 지정 (cone 의 ID 로 사용가능)
71	
2	: m 방향 mesh 갯수
72	
16	: n 방향 mesh 갯수 (밑면의 꼭지점 수)

B.1.3 PROPS 용 cone data 로의 변환

PROPS 에서 cone 을 그리는 routine 은 cone 의 윗면 반경, 아랫면 반경 , 높이를 필요로하며, 기준 좌표계의 원점은 AutoCAD 와 PROPS 모두 cone 밑면의 중심이며, 중심축은 Z 축이다. 밑면이 n 각형 인 경우, i 번째 vertex 와 $i+n/2$ 번째 vertex 간의 상관관계를 이용하여 기준좌표계의 원점의 X,Y 좌표 및 반경을 구하고, 밑면과 윗면의 Z 값으로부터 높이와 기준좌표계 원점의 Z 값을 구한다.



여 백

로봇을 이용한 금형의 연마작업 자동화 기술 개발
Development of the Automation Technology
for Die Polishing Using Robot

여 백

요약문

I 제목

로봇트를 이용한 금형의 연마 작업 자동화

II 연구 개발의 목적및 중요성

금형 마무리 공정은 전체 금형 제조 공정의 상당한 시간 비율을 차지하고 있으며 아직 수작업에 대부분 의존하고 있는 실정이다. 그러므로 이 분야의 자동화및 전체 생산공정의 집적화없이 금형 제조공정의 완전한 자동화를 기대할 수 없다. 본 과제는 금형 제작 공정 시간을 단축할 수 있는 정밀 고속 자동 연마 시스템 개발이 목적이다. 또한 금형 연마 로봇트 시스템의 상품화를 위하여 기존 개발된 시스템에 비해 크기가 축소되었으며 몸체의 강인성이 증가되었다.

III 연구 개발의 내용및 범위

기종 개발된 금형 연마로봇트 시스템의 상품화를 위한 Tool system및 system configuration변경과 이에따른 system management S/W의 구성등을 들 수 있다.

IV 연구 개발 결과및 활용에 대한 건의

- 금형 제작용체에서의 기술이전
- 관련부품 개발 중소기업에서의 기술이전

여 백

SUMMARY

The polishing process is counted as a bottleneck for factory automation of Die & Mold and still relies on manual works. So it is very hard to expect the perfect automation of Die & Mold manufacturing line without the automation in this field. In this research works automatic polishing robot system has been developed to reduce the Die & Mold production period. In comparison with already established systems, more compact and more robust body has been considered for the customization of the polishing robot system. For the purpose of the 3 axes simultaneous motor control and more reliability of robot motion, special purpose motor controller was used in our system. System integration S/W has been developed in MS-WINDOWS environment in order to manage the whole system. This developed technology can be applied in actual Die & Mold polishing process in industries concerned.

여 백

CONTENTS

Cap. I Introduction 111

 1. Necessity of R & D 111

 2. Objective and Contents of R & D 113

Cap. II Polishing Surface Analysis 114

 1. Static Analysis 114

 2. Dynamic Analysis 199

Cap. III Polishing pattern 121

 1. Polishing Surface Form 121

 2. Modified Lissajous Pattern 122

Cap. IV System configuration 124

 1. Polishing Tool System 124

 2. System Operation S/W 128

 3. System Lay-out 130

Cap. V Conclusion 131

여 백

목차

제 1 장 서론 111

 제 1 절 연구개발의 필요성 111

 제 2 절 연구개발의 목표 및 내용 113

제 2 장 연마 표면 해석 114

 제 1 절 정적 연마량 해석 114

 제 2 절 동적 연마량 해석 199

제 3 장 Polishing pattern 121

 제 1 절 연마 표면 형상 121

 제 2 절 변형 Lissajous 연마 패턴 122

제 4 장 System configuration 124

 제 1 절 Polishing Tool System 124

 제 2 절 시스템 구동 소프트웨어 128

 제 3 절 시스템 Lay-out 130

제 5 장 결론 131

부록

1. 변형 Lissajous 패턴 공식 및 형상	140
2. 연마 로봇 설계도	142
3. SEL controller 특성 및 사양	154
4. PC-SEL controller, PC-ABB R/C serial communication	157
5. 모터 사양	161
6. SEL controller parameter	162
7. SDK 개발 과정	163
8. 시스템 관리 소프트웨어	164

제 1 장 서론

제 1 절 연구 개발의 필요성

1. 연구개발의 중요성

금형 마무리 공정은 전체 금형 제조 공정의 상당한 시간 비율을 차지하고 있으며 아직 수작업에 대부분 의존하고 있는 실정이다. 그러므로 이 분야의 자동화 및 전체 생산공정의 집적화 없이는 금형 제조공정의 완전한 자동화를 기대할 수 없다. 본 과제는 금형 제작 공정을 단축할 수 있는 자동 연마 로봇에 의한 정밀 고속 자동 연마 시스템 개발이 목적이다.

2. 연구 개발하려는 기술의 세계적 수준의 단계

개념 정립과 생산 활용단계의 혼합 단계에 있다.

3. 지금까지의 연구개발 실적

이 과제의 1.2.3차년도에 연구개발된 실적으로서 다양한 금형 표면 형상에 대한 가공 기술과 공구 집적기술 등에 의한 Robotonic Tool Sytem 및 PC에 의한 통합환경을 개발하였다.

4. 현 기술의 취약점

이 기술의 취약점으로서 다양한 금형 형상에 대응할 수 있는 유연성이다. 기존의 연마시스템에 비해 상당한 유연성을 갖추고 있으나 가공 유연성에 비해 금형 형상은 천차 만별이므로 상대적인 유연성의 부족이 불가피하다.

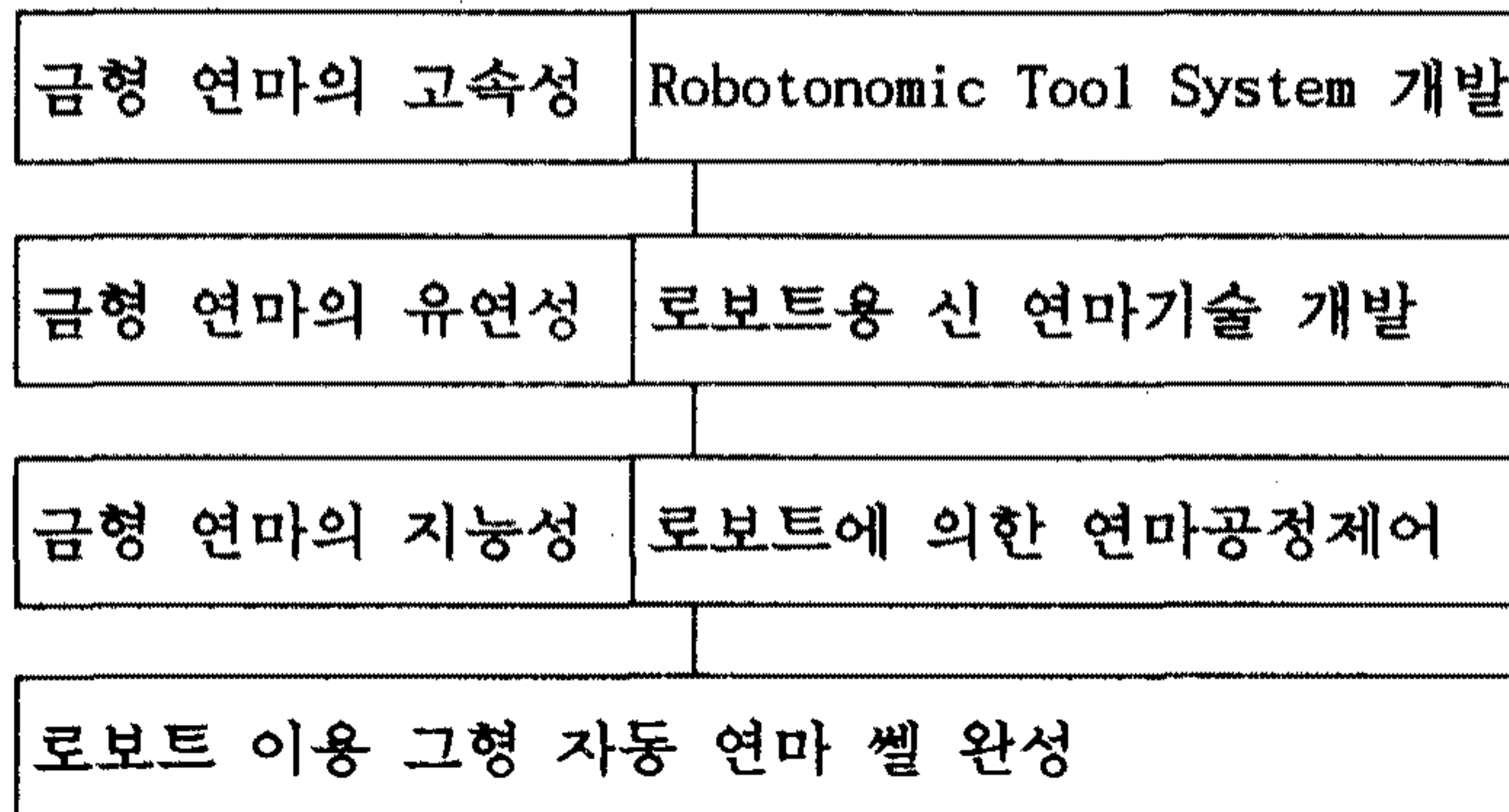
5. 앞으로의 전망

이 기술의 산업계 수요는 필연적이나 해당 산업계의 다른 금형 생산공정의 자동화 수준과 우선 순위에 따라 그 시기의 완급이 있겠다.

6. 기술 도입 가능성 및 경제성

현 세계적인 연구개발기술도 역시 매우 제한된 범위에서 응용가능하므로 큰 효과를 기대할 수 없다. 그리고 위의 이유에 의해 경제성도 취약하다.

7. 중점 추진과제와의 연관성



제 2 절 연구개발 목표 및 내용

1. 연구개발의 최종목표

금형제작 Lead Time을 획기적으로 단축시키기 위한 로봇 이용 정밀 고속자동 연마시스템 개발

2. 연구개발의 성격

기술지식 증대(), 요소기술 개발(○), 제품 또는 공정개발(○), 조사, 분석(), 기타().

3. 연차별 연구 개발 목표 및 내용

구분 연차별	연구 개발 목표	연구 개발 내용 및 범위
1차년도	금형 연마 셀 구성 - 금형 마무리 공정 분석 - Robotic Machinig - Robotonomic Tool	셀 개발 , 요소 제작 모델 정립 모델 정립 모델 정립
2차년도	금형 연마용 자동 공구 시스템 개발	개발
3차년도	로봇 이용 자동 연마 시스템 완성 - 로봇/공구시스템 Interface - 자동 보정 프로그램	개발 제작 개발
4차년도	상품화를 위한 연마 로봇 재설계및 완성 연마로봇 제어방식 변경및 신뢰도 향상 시스템 통합 S/W	개발 개발

제 2 장 연마 표면 해석

제 1 절 정적 연마량 해석

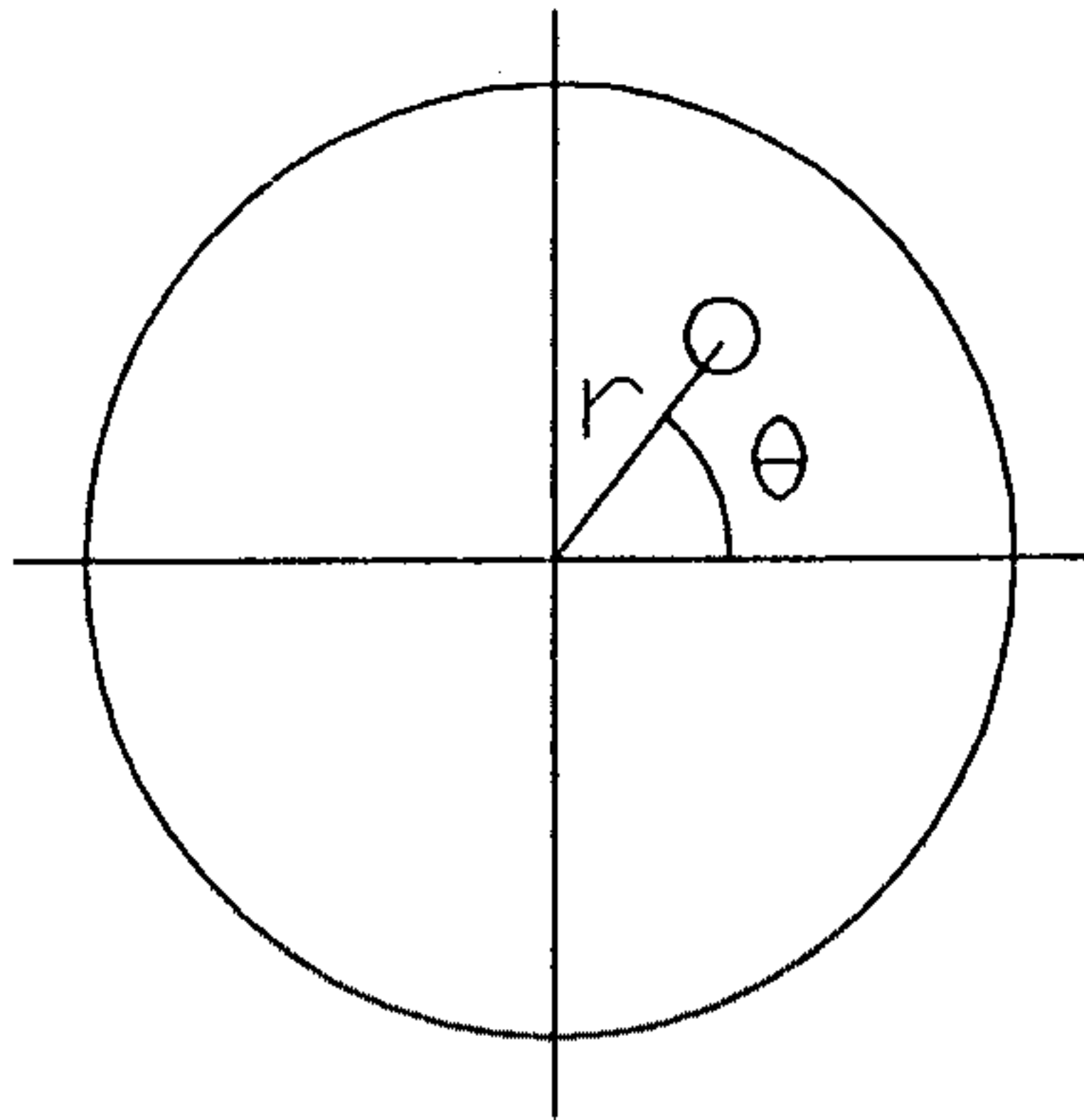
일반적으로 특정한 생산 가공을 거친 금속 표면은 독특한 표면 조직을 가지게 된다. 가공 대상의 공학적 기능을 만족시키기 위해서는 가공 기계의 사양이 적절히 선정되어야한다. 이러한 측면에서 금형 연마용 가공기계는 금형의 공학적 기능을 만족시키기 위하여 적절한 입도와 가공 슷들의 형상, 그리고 가공 기계의 강성등이 요구된다. 이들은 몇가지의 표면 조직을 구성하는 인자에 영향을 주게되며 가공기계와의 연계성을 고려하면 표면조직 구성 인자를 다음과 같이 대별 할 수있다.

- ① 조도 : 표면 형성 도구의 입도에 의한 가공 표면의 불규칙성
- ② 파상도 : 표면 형상 도구나 가공물의 처짐, 진동, 채터, 재료 내부의 스트레인등에 의해 발생하는 가공 파형
- ③ 형상 : 가공물의 일반적 표면 형상

여기서 형상 오차는 금형 연마 가공의 특성상 무시할 수 있으며 가공 경로에 따른 일정속도 유지에의해 형상 오차를 최소화 할 수 있다. 표면 조도 또한 가공 슷들의 입도에 관계되므로 원하는 가공 조도에 해당하는 입도의 선택에 의해 공학적 표면 조건을 만족시킬 수 있다. 일반적으로 파상도는 이상적인 작업 조건에서는 발생하지 않으나 금형 연마의 특성상 최대 관심 인자가 되며 파상도의 감소에 초점이 집중된다할 수 있다. 즉 금형 연마시 연마 슷들과 가공물의 접촉에의해 형성된 표면상에서의 연마량 차별 분포에 의해 표면에 가공 파형이 형성되므로 적절한 연마 슷들 형상의 선택이 요구된다. 일반적으로 자동 금형 연마용 회전 공구들의 틀 팁은 일정한 크기의 지름을 가진 원으로 이루어지므로 주어진 원형내에서의 최적 틀팁 형상 구

성이 요구된다. 본고에서는 파상도 최소화를 위한 인자로 가공 원내에서의 연마량에 대한 표준 편차를 선택하였다. 이에 대한 계산결과를 다음에 나타내었다.

연마량은 일반적으로 다음과 같이 주어진다.



$$S_m = \frac{1}{r_o - r_i} \cdot \int_{r_i}^{r_o} s(r) \cdot dr \quad (1)$$

여기서

S : 연마량

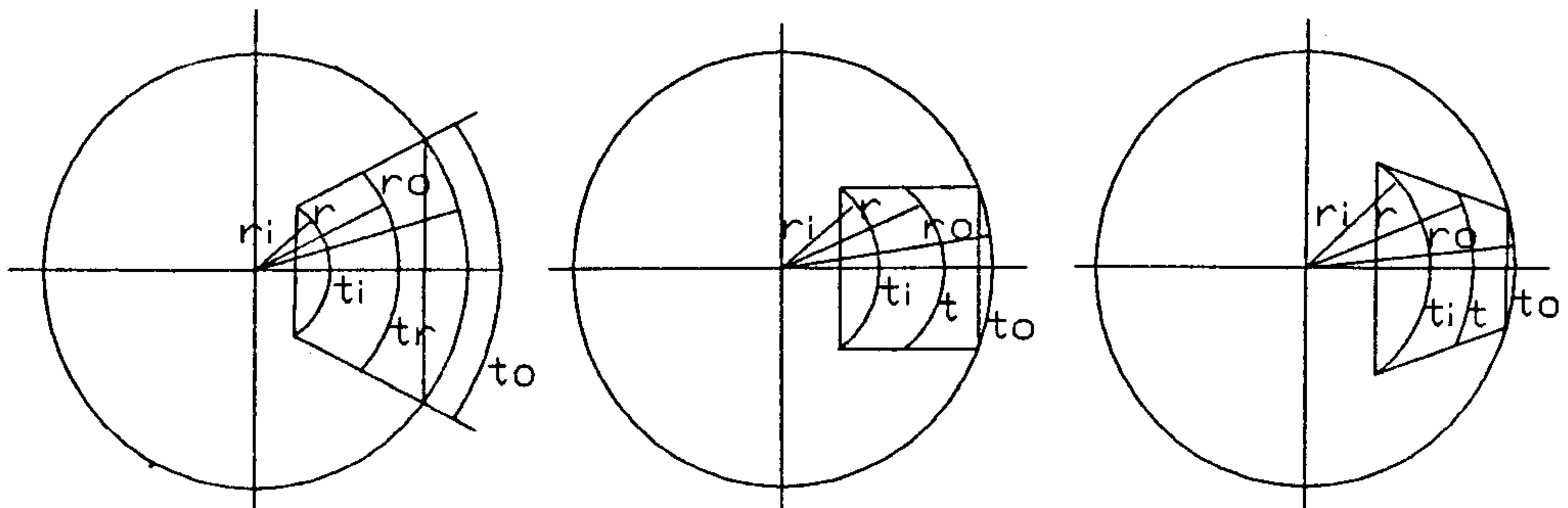
K : 연마비

τ : 1회전당 연마시간

r : 연마 위치 거리

w : 연마 위치에서의 각속도

가공 슷돌이 가공원내 전체를 메우는 형상이 가장 일반적인 경우이겠으나 파상도 감소를 위하여 다음과 같은 3가지의 틀팁 형상을 생각해 볼 수 있다.



위의 세가지 경우에대한 연마량의 적분 시간 τ 는 다음 식에 의해 구해 진다.

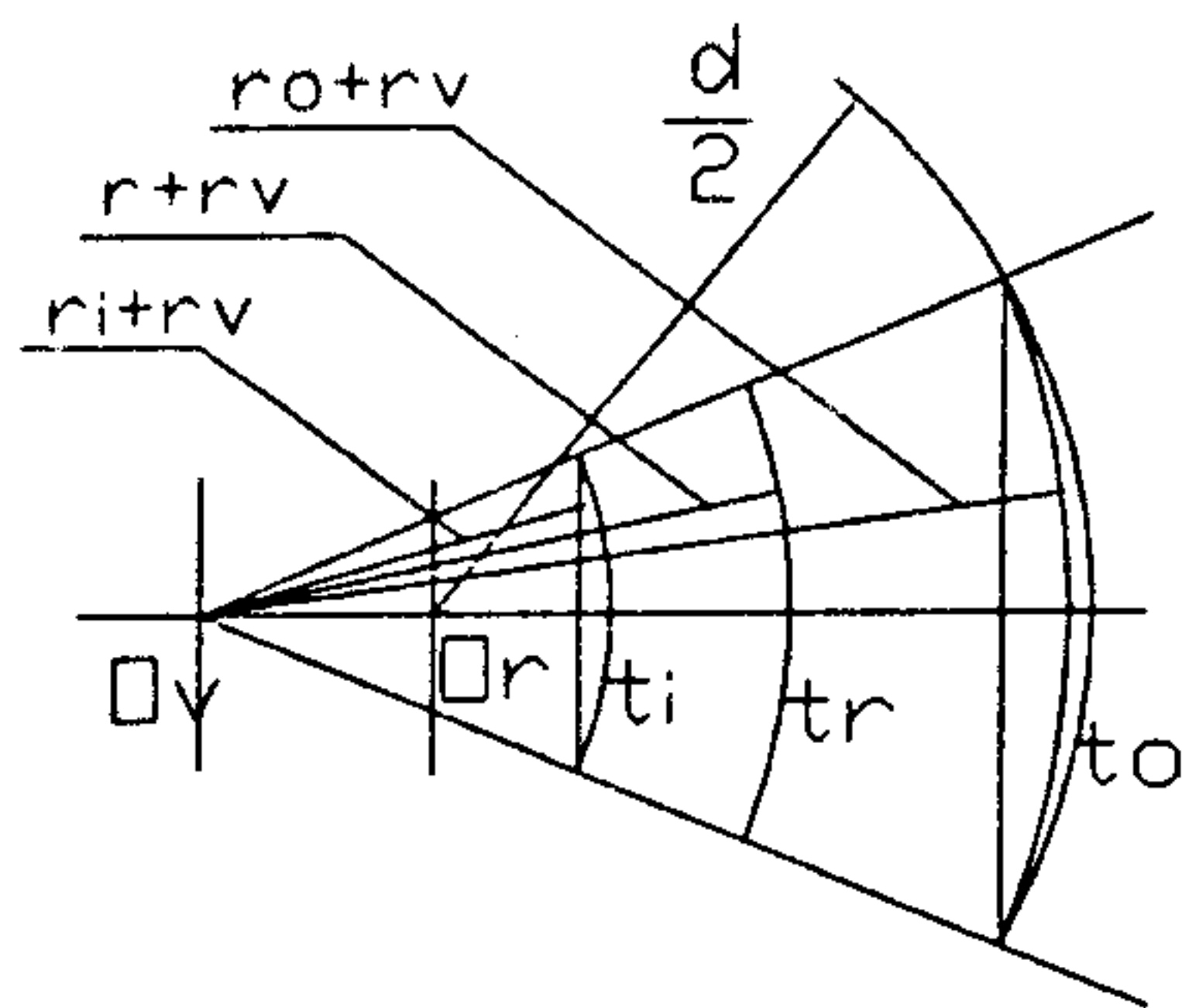
$$\tau = \frac{t_r}{r \cdot w} \quad \text{-----} \quad (2)$$

여기서 위의 3가지 경우에대한 t_r 을 구하면 다음과 같다.

i) $t_o > t_i$ 인 경우

가장 일반적인 형상이며 가공원내 전체를 가공 슷돌이 메우는 경우는 다음과 같은 조건을 만족할 때이다.

$$\frac{t_i}{r_i} = \frac{t_o}{r_o} \quad \text{-----} \quad (3)$$



O_r : 가공원의 원점

O_v : 가상 원점

t_i : $r_i + r_v$ 에서의 틀팁 길이

t_o : $r_o + r_v$ 에서의 틀팁 길이

t : $r + r_v$ 에서의 틀팁 길이

θ : O_r 을 중심으로한 틀팁 각도

위의 그림으로부터 다음의 조건식을 유도할 수 있다.

$$(r_i + r_v) \cdot \theta = t_i \quad \text{-----} \quad (4)$$

$$(r + r_v) \cdot \theta = t_r \quad \text{-----} \quad (5)$$

$$(r_o + r_v) \cdot \theta = t_o \quad \text{-----} \quad (6)$$

식 (4), (6)으로 부터 θ 는 다음과 같이 구해진다.

$$\theta = \frac{t_i}{r_i + r_v} = \frac{t_o}{r_o + r_v} \quad \text{————— (7)}$$

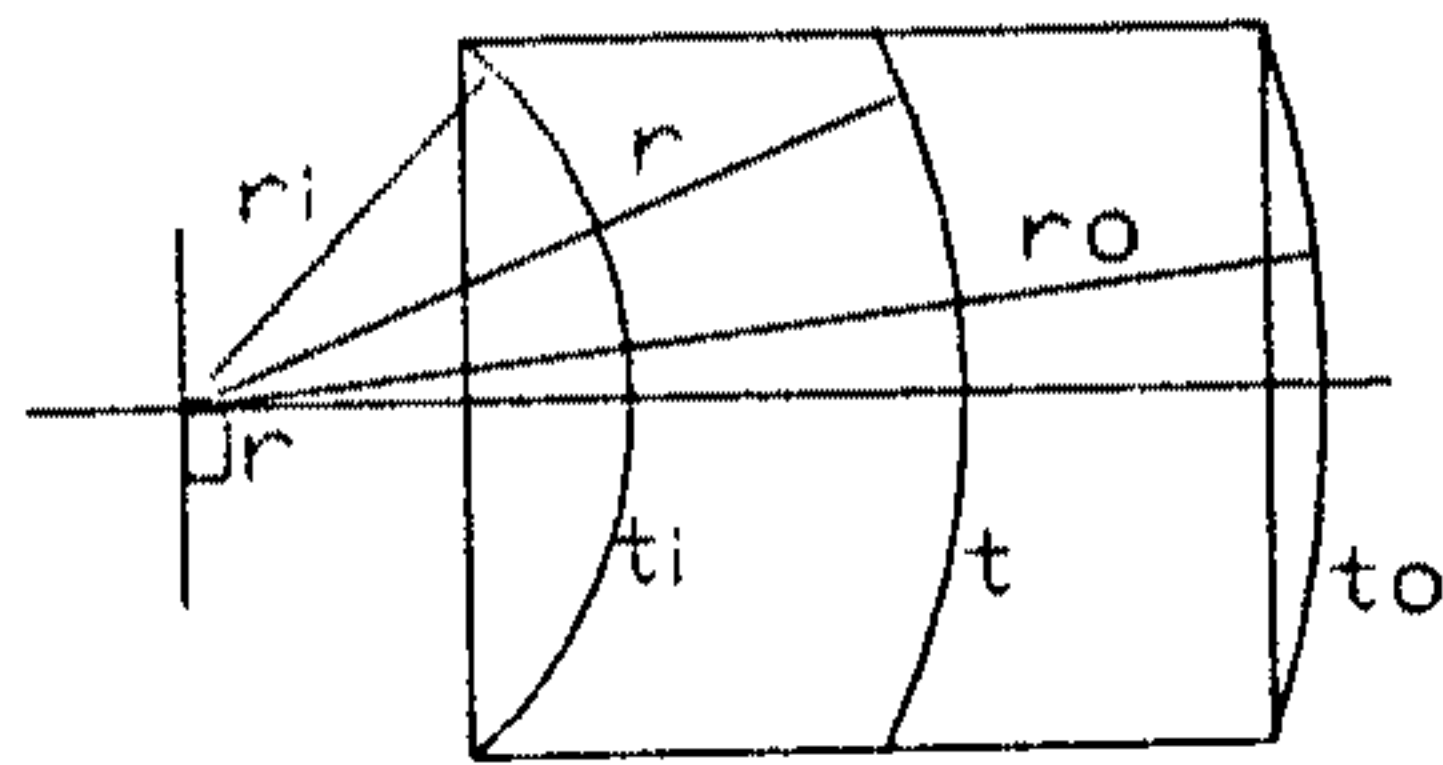
$$r_v = \frac{t_o \cdot r_i - t_i \cdot r_o}{t_i - t_o} \quad \text{————— (8)}$$

$$\theta = \frac{t_i - t_o}{r_i - r_o} \quad \text{————— (9)}$$

식 (8)과 (9)를 식 (5)에 대입하면 t_r 은 다음과 같이 구해진다.

$$t_r = \frac{t_o \cdot r_i - t_i \cdot r_o}{r_i - r_o} + r \cdot \left[\frac{t_i - t_o}{r_i - r_o} \right] \quad \text{————— (10)}$$

ii) $t_o = t_i$ 인 경우



여기서는 아래의 두 조건식을 만족한다.

$$t_r = r \cdot \theta_r \quad \text{————— (11)}$$

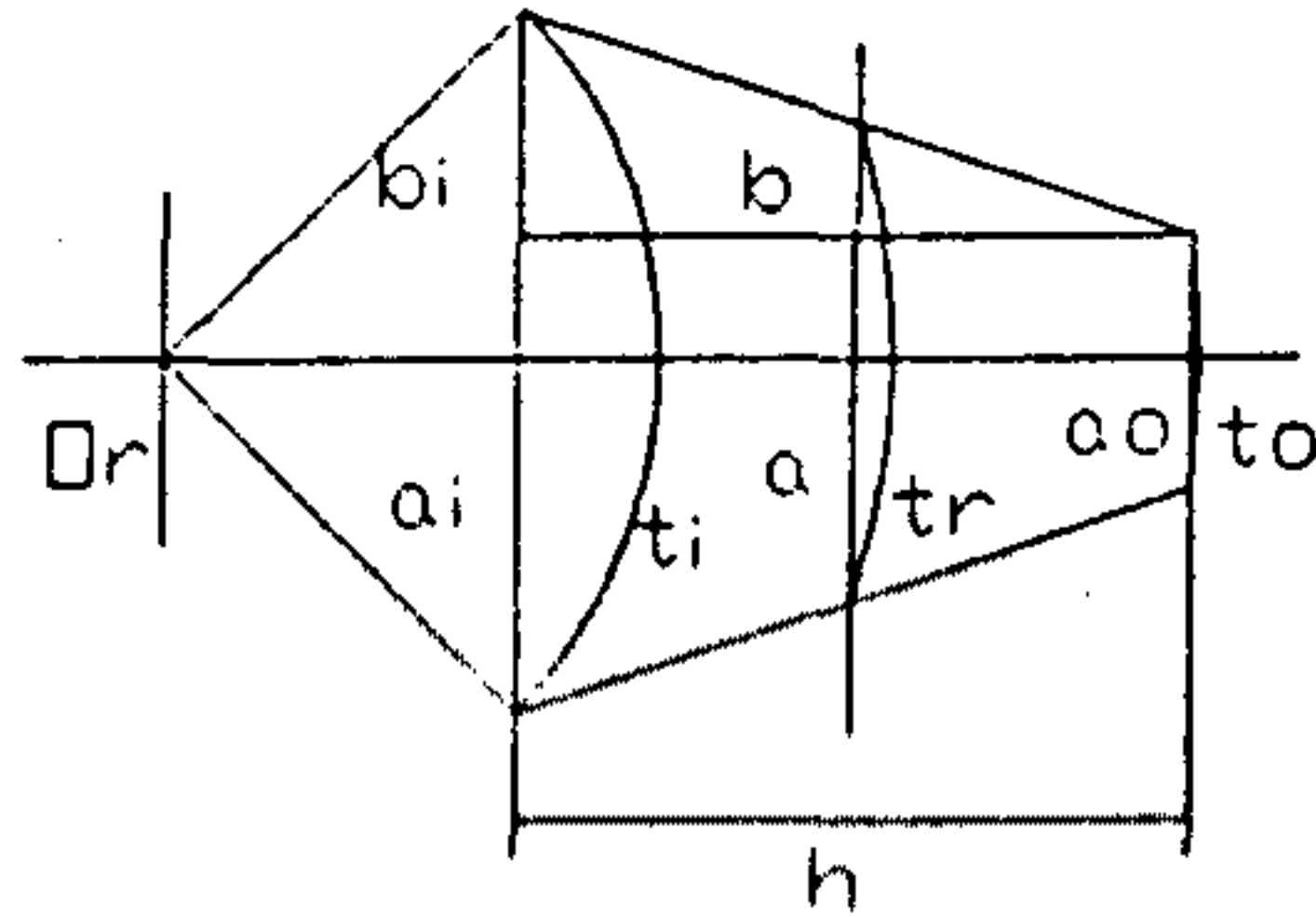
$$\sin \left[\frac{\theta_r}{2} \right] = \frac{t}{2 \cdot r} \quad \text{————— (12)}$$

그러므로 θ_r 과 t_r 은 각각 다음과 같이 구해진다.

$$\theta_r = 2 \cdot a \sin \left\{ \frac{\theta_r}{2} \right\} \quad \text{————— (13)}$$

$$t_r = 2 \cdot r \cdot a \sin \left\{ \frac{t}{2 \cdot r} \right\} \quad \text{————— (14)}$$

iii) $t_o < t_i$ 인 경우



여기서는 아래의 두 조건식을 만족한다.

$$\theta_r = 2 \cdot a \sin \left\{ \frac{a}{2 \cdot r} \right\} \quad \text{————— (15)}$$

$$a = a_o + 2 \cdot \frac{b_i \cdot (r_o - r)}{h} \quad \text{————— (16)}$$

식 (15)와 (16)으로부터 $t(r)$ 은 다음과 같다.

$$t_r = 2 \cdot r \cdot a \sin \left\{ \frac{a}{2 \cdot r} \right\} \quad \text{————— (17)}$$

위에서 구한 세가지 경우의 t_r 로부터 연마량이 계산되며 각 경우에 대한 $r = r_i$ 에서 $r = r_o$ 까지 걸친 연마량의 표준 편차를 구하기 위하여 $s(r)$ 의

평균치를 s_m 이라두면

$$s_m = \frac{1}{r_o - r_i} \cdot \int_{r_i}^{r_o} s(r) \cdot dr \quad \text{————— (18)}$$

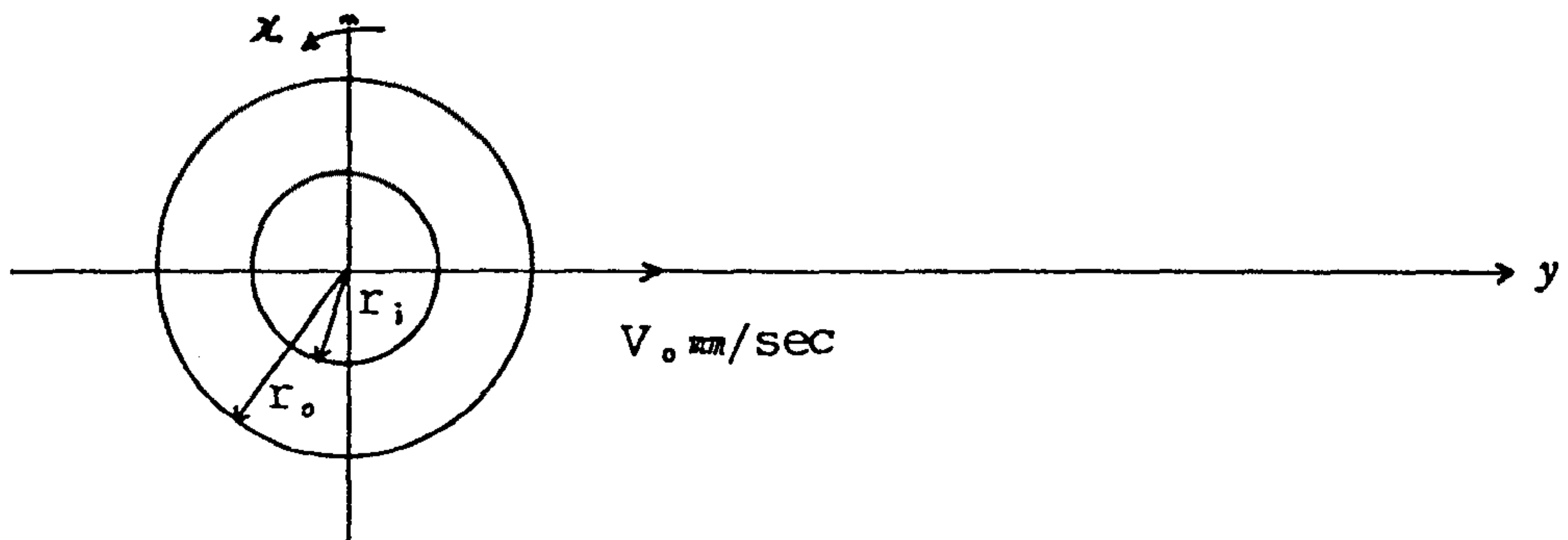
위 식으로 부터 연마량의 표준편차는 다음과 같이 구해진다.

$$\sigma = \sqrt{\frac{1}{r_o - r_i} \cdot \int_{r_i}^{r_o} \left[s(r) - s_m \right]^2 \cdot dr} \quad \text{————— (19)}$$

위식으로부터 t_o/t_i 와 σ 간의 관계를 구한 결과를 그림 1에 나타내었다. 여기서 알 수 있는 바와같이 $t_o/t_i = 1$ 인 경우가 σ 의 최저치를 나타내며 이는 파상도를 최저값으로 유지시켜 줌을 알 수 있다. 본고에서 선택한 mold stone의 형상을 그림 2에 나타내었다.

제 2 절 동적 연마량 해석

정적 연마량 해석으로부터 얻은 결과인 그림 2의 틀 팁 형상을 실제 회전과 함께 이동 시켰을때의 동적 연마량을 구해봄으로써 실제 가공시의 타당성을 확보 할 수 있다. 그림 2에 나타낸 틀 팁 형상을 동적으로 수식화 하기가 어려우므로 아래의 그림과 같은 도넛 형태의 단순한 형상으로 바꾸어 계산하고자 한다. 즉 $t_o > t_i$ 인 경우로 식 (3)을 만족하는 경우이며 V_o 에 비해 w 가 충분히 크다고 가정한다.



여기서 연마량 S 를 연마물 이동 방향에 수직인 X 의 함수로 나타내기 위하여 r 과 τ 를 X 와 시간 t 의 함수로 나타내면 다음과 같다.

$$r = \sqrt{\sqrt{r_0^2 - x^2} - (v_0 t)^2 + x^2} \quad \text{————— (20)}$$

여기서 τ 는 다음의 조건식에 의해 구해진다.

i) $0 < x < r_i$

$$\tau = \frac{\sqrt{r_0^2 - x^2} - \sqrt{r_i^2 - x^2}}{v_0} \quad \text{————— (21)}$$

i) $r_i < x < r_0$

$$\tau = \frac{\sqrt{r_0^2 - x^2}}{v_0} \quad \text{————— (22)}$$

툴 팁 제작상의 간편함을 위하여 $r_i = 10$, $r_0 = 20$ 으로 정하였으며 이때의 연마량을 그림 3에 나타내었다. X 방향의 중첩 연마량이 비교적 고르게 분포함을 알 수 있다. $t_0 = t_i$ 의 툴 팁을 이용한다면 더욱 나은 결과를 얻을 수 있을 것으로 기대된다.

제 3 장 Polishing Pattern

제 1 절 연마 표면 형상

연마 패턴 연구의 결과에의하면 Lissajous Pattern의 비교 우위가 입증되었으나 연마 대상 금형의 형상 다양화에따라 그 직접 적용이 어려운 상황이다. 앞서도 지적했듯이 현 기술의 취약점인 다양한 금형 형상에의 유연성 있는 대응이 시급한 과제이며 특히 자동 금형 연마 시스템의 상품화를 고려해 볼때 이는 필수적이라 하겠다. 이에따라 대표적인 몇가지 금형 형상에 대해 고찰해 보고 그에 대한 변형 Lissajous 연마 패턴의 제시와 함께 그 운용 방법을 기술하고자 한다.

일반적으로 넓은 금형 표면을 연마하기 위해서는 로봇트 끝단 또는 연마 tool 최종단의 경로 프로그램이 필요하며 이는 수작업에의한 teaching이나 금형의 3차원 곡면 모델링에의한 CAD 데이터를 이용하게된다. 이때 연마 툴 또는 로봇트와 연마 대상 금형사이의 충돌 회피를 위해서는 패턴 형상의 크기에따라 연마 표면의 가장자리에 적절한 여유를 두어야 하는데 그형상이 직선이 아닐 경우에는 패턴 외곽 형상의 적절한 변형이 요구된다.

연마 표면의 가장자리 형상은 크게 그림 4와 같이 5가지로 대별 될 수 있다. 즉 예각, 둔각, 직선 그리고 원형, 타원형의 5가지 이며 이러한 경우 정사각형의 패턴 외곽 형상만으로는 부족함을 알 수 있다. 이러한 5가지의 가장자리 형상을 모두 만족 시킬 수 있는 패턴 외곽 형상들을 고려해 보면 다음과 같다.

직선, 둔각 <- 정사각형, 사다리꼴 패턴

예각 (60도 이상) <- 정삼각형 패턴

원형, 타원형 <- 정사각형 패턴

가장자리 형상중 가장 문제시 되는 것은 예각으로 이루어진 부분이며 이에 대한 처리는 정삼각형 패턴에 의해서만 가능하므로 연마 tool이 가장자리로의 접근시 패턴 외곽형상의 갑작스러운 변형에 의한 파상도 분포의 불균질 문제를 막기 위하여 가장자리 연마에 대한 패턴 외곽 형상 운용 방법에 대한 알고리즘을 그림 5에 나타내었다. 이러한 패턴 운용 방법에 의해 예각으로 이루어진 가장자리 부분의 연마가 파상도 분포의 불균질없이 처리될 수 있다.

제 2 절 변형 Lissajous 연마 패턴

위에서 기술한 바와같은 다양한 형상의 패턴을 2차원 평면상에 구현하기 위해서는 Lissajous 패턴 공식의 변형이 필요하며 이를 변형 Lissajous 패턴이라 지칭한다. 변형 Lissajous 패턴 공식을 얻기 위해서 일반 Lissajous 패턴 공식을 고려해 보면 다음과 같다.

$$X(t) = X_OFF + X_AMP \times \sin(X_FF \times t) \quad \text{————— (23)}$$

$$Y(t) = Y_OFF + Y_AMP \times \sin(Y_FF \times t) \quad \text{————— (24)}$$

X_OFF : X 방향으로의 오프셋

Y_OFF : Y 방향으로의 오프셋

X_AMP : X 축 진폭

Y_AMP : Y 축 진폭

X_FF : X 축 frequency factor

Y_FF : Y 축 frequency factor

여기서 Lissajous 패턴의 외곽 형상을 변형 시키기 위해서는 X 또는 Y의 진폭을 t에 대해 변형 시켜야 하며 진폭의 다양한 변형에 의해 정삼각형이나

다양한 형상의 사다리꼴 패턴을 얻을 수 있다. 즉 Y 진폭을 고정시킬 경우 X의 진폭을 다음의 식에 의해 변형시킬 수 있다.

$$X_AMP(T) = XA_AMP \cdot \left[\frac{XA_AMP}{Df} \right] \cdot \text{SIN} \left[\left[\frac{Dd}{X_FF} \right] \cdot X_FF \cdot T \right] \quad (25)$$

위의 진폭 변형 공식에서 Df와 Dd를 적절히 변형하면 정사각형에서부터 사다리꼴, 정삼각형 등의 패턴 외곽 형상을 얻을 수 있다. 여기서 각각의 외곽형상에 대한 Dd의 값을 구하면 다음과 같다.

$$\text{정사각형} : Dd = 0$$

$$\text{사다리꼴} : Dd = X_FF + 1$$

$$\text{정삼각형} : Dd = X_FF - 1$$

각각의 패턴에 대한 완전한 공식과 그에 의한 패턴 형상을 부록 1에 나타내었다.

제 4 장 System configuration

제 1 절 Polishing Tool System

1. 기구학적 설계

자동 금형 연마 시스템 운용을 위해 채택된 ABB 로봇은 손목 끝단 이후의 가반하중이 10Kg으로 제한되어있다. 3차원 곡면상에서의 다양한 패턴 형성을 위하여 3자유도 로봇 시스템이 개발되었으며 가반 하중 제한에 따라 보다 가볍고 보다 높은 동강성을 지닌 구조로 재 설계 하였다. 기본적인 3차원 자유 곡면 운동을 위한 로봇 구조를 그림 6에 나타내었다. 여기서 알 수 있듯이 연마 표면의 곡률은 2축과 3축모타의 조화 운동으로부터 유지할 수 있으며 2차원 패턴 형상은 1축과 3축의 복합 운동에 의해 형성된다. 로봇의 정강성 및 동강성을 높이기 위하여 기존의 타이밍 벨트에 의한 구동방식을 지양하고 Anti-backlash gear를 채택하여 툴 팁 운동의 정확성을 기하고 모타 정격 토크 이하의 외력에 충분히 견딜 수 있는 기구학적 구조를 갖추었다. 1축의 동력은 anti-backlash gear에 의해 모타로부터 직접 전달되며 2축은 볼 스크류에 의해 모타의 회전 운동을 상하 직선운동으로 바꾸게 된다. 연마 로봇의 폭을 줄이기 위하여 3축 모터를 로봇의 길이 방향으로 수직하게 세웠으며 이의 동력은 베벨기어를 통해 직접 3축 링크에 전달된다. 또한 최종 rotary bar를 항상 연마 표면에 수직한 방향으로 유지해야 하므로 3개의 기어 조합에 의한 평행 링크를 구성하였다. Mold stone의 연마 표면에 대한 가압력 조정 및 위치 공차를 흡수 하기 위하여 공압 실린더에 의한 Linear compliance를 부착하였다. 이에 대한 로봇 설계도를 부록 2에 나타내었다. 위와같이 설계된 로봇의 운동 제어를 위하여 구동 모터의 회전 각도와 로봇 자세사이의 관계를 정의하고자 한다.

각 모터 1, 2, 3의 회전각을 $\theta_1, \theta_2, \theta_3$ 라 하고 기구부의 최종 회전각을 $\theta_{M1}, \theta_{M2}, \theta_{M3}$ 라 하면 모터 회전각과 최종 회전각 사이에는 다음과 같은 관계가 성립한다.

$$\theta_1 = 1/6 \theta_{M1}, \theta_2 = \theta_{M2}, d_2 = d_{init} - \theta_{M2}/90, q_3 = 1/12 \theta_{M3} \quad \text{--- (26)}$$

위의 식으로 부터 연마 로봇트의 링크 파라미터를 정리하면 다음과 같다.

Link	변수	α	a	d
1	θ	0	0	0
2	d	0	0	$d_{init} - \theta_2/90$
3	θ	-90	0	0

위의 링크 파라미터로 부터 각 조인트의 변환행렬을 구하면

$$A_1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{init} - \theta_2/90 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & 96.5C\theta_3 \\ 0 & 0 & -1 & 0 \\ S\theta_3 & C\theta_3 & 0 & 96.5S\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

위의 변환 행렬로부터 Mold stone의 object position은 다음과 같이 구해진다.

$$X = 96.5C\theta_3C\theta_1 \quad \text{--- (27)}$$

$$Y = 96.5C\theta_3S\theta_1 \quad \text{--- (28)}$$

$$Z = 96.5S\theta_3 - d_{init} - \theta_2/90 - L_t \quad \text{--- (29)}$$

위의 식으로부터 Mold stone의 object position에 따른 모터 구동 각도 해

가 다음과 같이 구해진다.

	Solution 1	Solution 2
$C\theta_1$	$x/(x^2+y^2)^{0.5}$	$-x/(x^2+y^2)^{0.5}$
$C\theta_3$	$(x^2+y^2)^{0.5}/96.5$	$-(x^2+y^2)^{0.5}/96.5$
θ_2	$90(96.5S\theta_3 + d_{init} - L_t - z)$	

2. 패턴 형상 제어

1) 제어 기기의 구성

역기구학 해에의한 모터의 운동 제어의 정밀도에따라 3차원 곡면상의 연마 패턴 형상이 결정지어지므로 모터의 정밀한 제어가 요구된다. 또한 로봇 운동 제어를 위한 중요한 요소로서 로봇의 반복 정밀도및 외부 부하에 대한 강성 유지가 중요하며 이러한 요구 조건을 만족시키기 위하여 정밀 위치 제어를 위한 전용 모터 제어기(SEL controller)를 이용하였다. 이 제어기의 특성및 사양을 부록 3에 나타내었다. 전용 제어기와 시스템 관리를 위한 PC와의 데이터 통신은 RS-232C serial communication에 의해 이루어지며 제어기의 다양한 기능이 PC에의해 구현될 수 있다. 시스템 관리용 PC는 사용자와의 man-machine interface를 구현해주며 user dependant한 데이터를 받아서 패턴 형상 데이터및 형상 제어 프로그램을 생성하고 이를 SEL controller(이하 SEL)로 전송하게된다. 또한 연마 로봇의 가공면 경로 제어를 위한 ABB 로봇의 운용 프로그램에 대한 구동권을 가짐으로써 전체 금형 연마 시스템을 관장하게된다. 이에 대한 처리 절차를 그림 7에 나타내었다. 시스템 관리용 PC와 SEL 및 ABB 로봇과의 데이터 전송을 위한 시리얼 커뮤니케이션에 대한 기본 자료를 부록 4에 나타내었다.

2) SEL controller의 적용 방법

연마 로봇트의 운동을 위한 모터는 로봇트 크기의 감소와 적절한 정격 토크를 얻기위하여 50:1의 기어 감속비를 갖는 하모닉 드라이브 장착 모터로 선정하였다. 이의 사양을 부록 5에 나타내었다. SEL의 모터 제어 방식을 알아보기 위하여 내부의 파라미터 구성 파악이 요구된다. SEL은 외부 모터의 엔코더로 부터 발생하는 펄스에 자체의 채배기능을 거쳐 모터의 운동을 감지하게되며 내부적으로는 부록 6에서 볼 수 있듯이 모터의 회전이 lead screw에의해 직선운동으로 변환된다고 가정을 하고 내부의 LEAD 파라미터를 이용하여 지시된 UNIT 만큼 모터의 회전각도를 제어하게된다. 즉 LEAD를 1mm로 지정하고 엔코더 PPR을 200, 채배를 4로 두면 1 UNIT(1 mm에 해당)만큼의 이동을 지시했을때 모터축의 360도 회전을 위한 800 pulse 만큼의 엔코더 pulse가 발생하는 위치로 모터축을 돌리게된다. 이러한 SEL의 모터 구동 mechanism을 이용하여 연마 로봇트를 제어하기위한 SEL 파라미터 선정 방법에 대해 간단히 기술하면 다음과 같다.

엔코더 PPR은 200 pulse 이며 SEL multiple은 4 채배이므로 SEL은 외부로 부터의 800 pulse를 1회전으로 간주하게된다. 연마 로봇트는 역 기구학 해에의한 회전 각도가 주어지며 각도 단위를 SEL의 변위 단위로 사용하기위하여 파라미터 중의 엔코더 PPR과 LEAD 값을 적절히 계산해 주어야 한다. 모터의 기어 감속비를 고려한 계산 결과를 나타내면 다음과 같다.

1축, 3축 -> PPS : 194, LEAD : 7

2축 -> PPS : 200, LEAD : 4

위와같은 파라미터 선정에 따라 1축과 3축에 대해서는 역기구학 해에의한 각도를 SEL 변위 단위로 사용 가능하며 2축에 대해서는 다음과 같은 단위

변환 관계식이 필요하게된다.

$$Y = \begin{bmatrix} 4 \\ \hline 360 \end{bmatrix} \times \theta_2 \quad \text{-----} \quad (30)$$

SEL controller는 절대 좌표 방식에의해 위치제어를 구현하므로 반복 운동에 의한 누적오차는 무시할 수 있다. 이에따라 3차원 곡면에서의 정확한 패턴 형성이 가능하며 연마면의 품질을 향상 시킬 수 있다.

제 2 절 시스템 구동 소프트웨어

1. 프로그램 개발 환경

시스템 관리 측면에서의 작업자의 편의를 도모하고 시스템 운용 방법에 대한 절차 교육 시간 단축을 위해서는 사용자 요구의 일관성있는 처리 절차와 명확한 처리 알고리즘이 요구된다. 이러한 시스템 운용의 특성을 만족시키기 위하여 프로그램 개발 환경으로 MS-WINDOWS를 채택하였다. MS-WINDOWS는 human interface가 뛰어나고 message 발생에의한 처리로 프로그램 수행 절차가 명확하다는 특성을 가지고있다. Message 발생에의한 프로그램 수행에의해 Multi-tasking이 가능하며 이는 PC의 자원을 최대한 활용할 수 있도록 해준다. 이러한 message 처리에의한 프로그램 수행 환경을 그림 8에 나타내었다. 여기서 볼 수 있듯이 사용자나 외부 device로부터 발생하는 모든 message는 우선 MS-WINDOWS system queue에 저장되며 이는 다시 현재 실행 중인 application queue로 전송된다. 현재 실행 중인 모든 application은 자체의 Message loop를 가지고있으며 이는 항상 WINDOWS에 할당된 자신의 message queue를 조회하여 현재 실행중인 window function에 message를 전달하게된다. MS-WINDOWS 환경에서의 실행가능한 프로그램을 작성하기위하여

는 프로그램 개발 툴인 SDK(Software Development Kit)를 이용해야하며 이에 의한 프로그램 개발 절차를 부록 7에 나타내었다.

2. S/W 구성

연마 패턴은 다양한 패턴 파라미터의 입력에 의해 그 형상이 결정된다. 그외에 툴 팁의 선정에 의해서도 패턴의 크기나 연마량이 변화하므로 사용자에 의한 패턴 파라미터의 입력이 우선 되어야 한다. 주어진 패턴 파라미터에 의해 연마 로봇을 구동할 프로그램과 위치 데이터가 결정되며 이들은 SEL controller로 전송 됨으로써 자동 연마 프로세스의 연마 로봇 구동을 위한 준비가 끝나게된다. 이에 대한 패턴 생성 절차를 그림 9에 나타내었다. 이와같은 절차에 따라 생성된 패턴을 이용한 연마를 위하여 시스템 관리 PC에 의한 각 기기의 통합이 이루어져야 하며 이러한 통합 기술에 의하여 적절한 연마 프로세스를 구현할 수 있다. 시스템 관리 PC에 의한 연마 프로세스를 그림 10에 나타내었다. 패턴 시작 위치 결정과 프로세스 진행 상황 모니터링을 위하여 ABB 로봇 제어기와 On line 데이터 통신이 필요하며 이는 부록 4에 보인 바와같이 ADLP-10과 ARAP 통신 프로토콜에 의해 이루어진다. 패턴 시작 및 종결 위치 결정은 연마 로봇 구동 위치로부터의 거리 오차로 결정되며 이는 ABB 로봇 운동속도 파라미터에 의하여 결정된다. ABB 로봇 위치 확인은 0.5 초 마다의 타이머 인터럽트에 의하여 이루어지므로 오차 구의 반지름은 다음 식에 의해 결정된다.

$$r \geq \frac{0.5 \cdot V}{2} \quad (31)$$

이외에 시스템 운용에 필요한 추가 기능을 그림 11에 나타내었다. 이러한

모든 기능을 갖춘 자동 연마 시스템 운용 프로그램을 부록 8에 나타내었다. 여기서 알 수 있듯이 초기화면은 시스템 기동등의 주요 기능과 프로세스 상태 모니터링 기능등에 의해 구성 되어있고 나머지 운용 기능등은 Menu형식에 의하여 유저의 요구에 따라 동작되게된다.

제 3 절 시스템 Lay-out

자동 금형 연마용 잉여 자유도계 로봇트와 그에대한 제어기, 그리고 시스템 운용 PC등에의한 전체 시스템 Lay-out을 그림 12에 나타내었다. PC와 ABB 로봇트, PC와 SEL 간의 통신은 RS-232C serial communication에의해 이루어진다. PC의 serial port 1이 ABB 로봇트 제어기와 연결되어있고 serial port 2가 SEL 제어기와 각각 연결 되어있다. 자동 금형 연마를 위한 ABB 로봇트의 경로 데이터는 수작업에의한 경로 teaching이나 Off-line 프로그램에의해 로봇트 제어기에 저장된다. 패턴 형성을위한 SEL 프로그램과 연마 로봇트 위치 데이터는 패턴 파라미터 입력과 함께 전송작업에의해 SEL 제어기의 메모리 카드에 저장되고 이의 구동권을 PC가 가지고 있으므로 ABB로봇트 위치 모니터링에 따라 필요시 연마 로봇트를 구동하게 된다. 즉 PC는 ABB 로봇트와 연마 로봇트와의 동기화를 구현해준다. 연마 로봇트와 SEL 제어기사이에는 모터 엔코더 신호와 동력전달용 케이블 외에 로봇트의 홈 위치 복귀신호와 연마용 회전 공구 틀 동력 라인이 있다. 이러한 신호 라인과 동력 라인의 상호 관계를 그림 13에 나타내었다. 연마 로봇트의 홈 위치를 검출하기위하여 광 센서의 일종인 photo interrupt 를 이용하였고 이 신호를 SEL 제어기에게 전달해줄 아날로그 스위치의 단속신호로 이용하였다.

제 5 장 결론

본 연구 과제에서 개발된 금형 연마 로봇 시스템의 주요 특징은 다음과 같다.

- 1) 금형 제조 공정 시간을 단축 하기위하여 정밀 자동 금형 연마 시스템을 개발하였으며 수작업 연마 공정을 대체함으로써 생산성 향상에 크게 기여할 수 있다.
- 2) 로봇에 적합한, 파상도를 감소시킬 수 있는 무방향성 연마 패턴이 개발되었으며 등속 Lissajous pattern이 그 한 예로서 실험 및 비교 결과 보다 향상된 결과를 도출했다.
- 3) 금형 연마 시스템의 유연성을 향상시키기 위하여 다양한 변형 Lissajous 연마 패턴을 구현하였으며 이에의한 다양한 금형의 연마 처리가 가능하게 되었다.
- 4) 연마 로봇의 패턴 운동 반복 정밀도 향상과 정확한 패턴 형상 구현을 위하여 3축 동시 제어용 전용 제어기를 이용하였으며 연마 로봇의 신뢰도 향상과 상품화에 크게 기여할 것으로 기대된다.
- 4) 전체 금형 연마 로봇 시스템의 정보 생성/ 처리/ 제어/ 저장기능등의 시스템 통합 S/W가 개발되었으며 뛰어난 User Interface기능과 통합 관리 기능을 가지고있다.

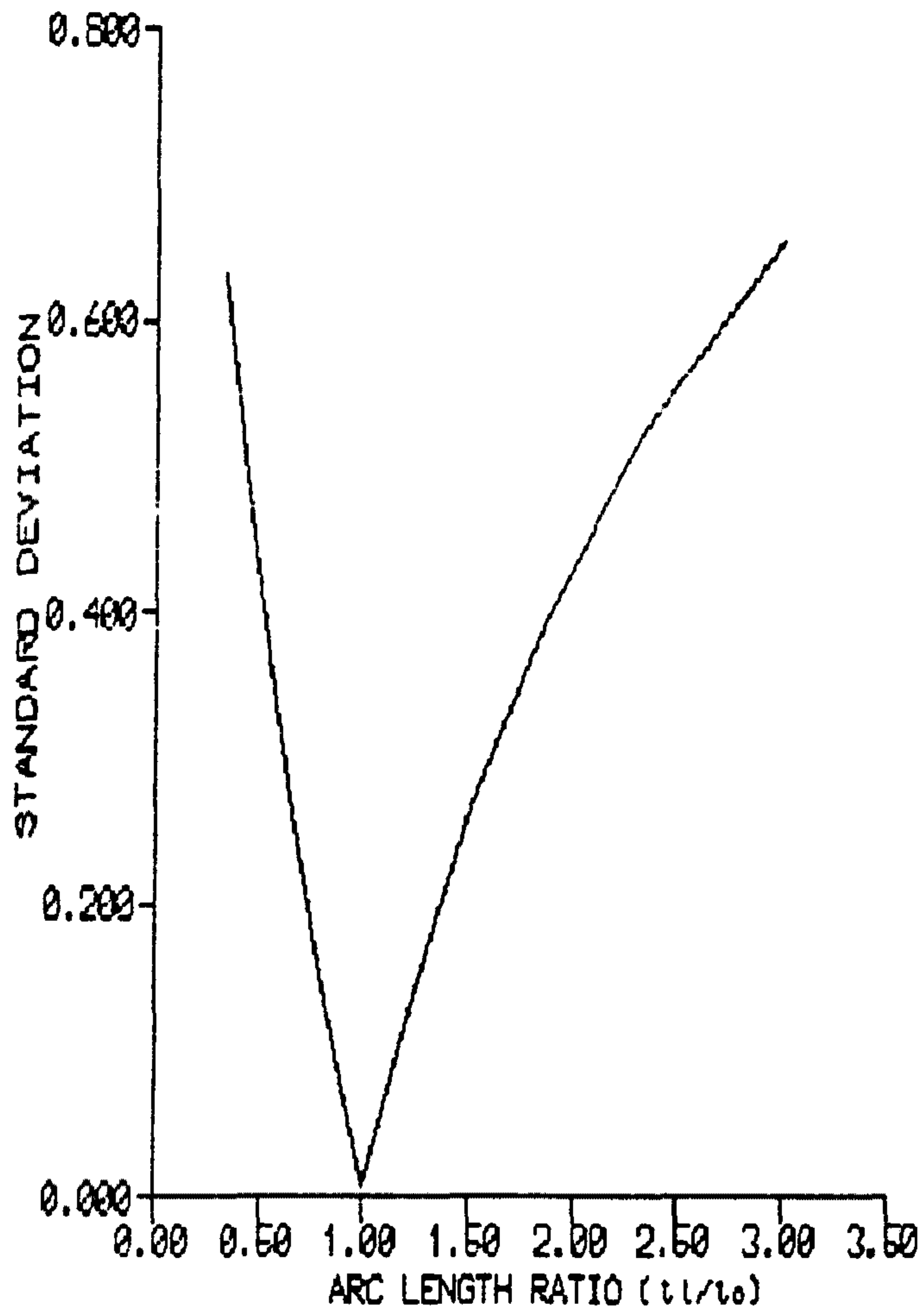


그림 1. 연마량 표준 편차

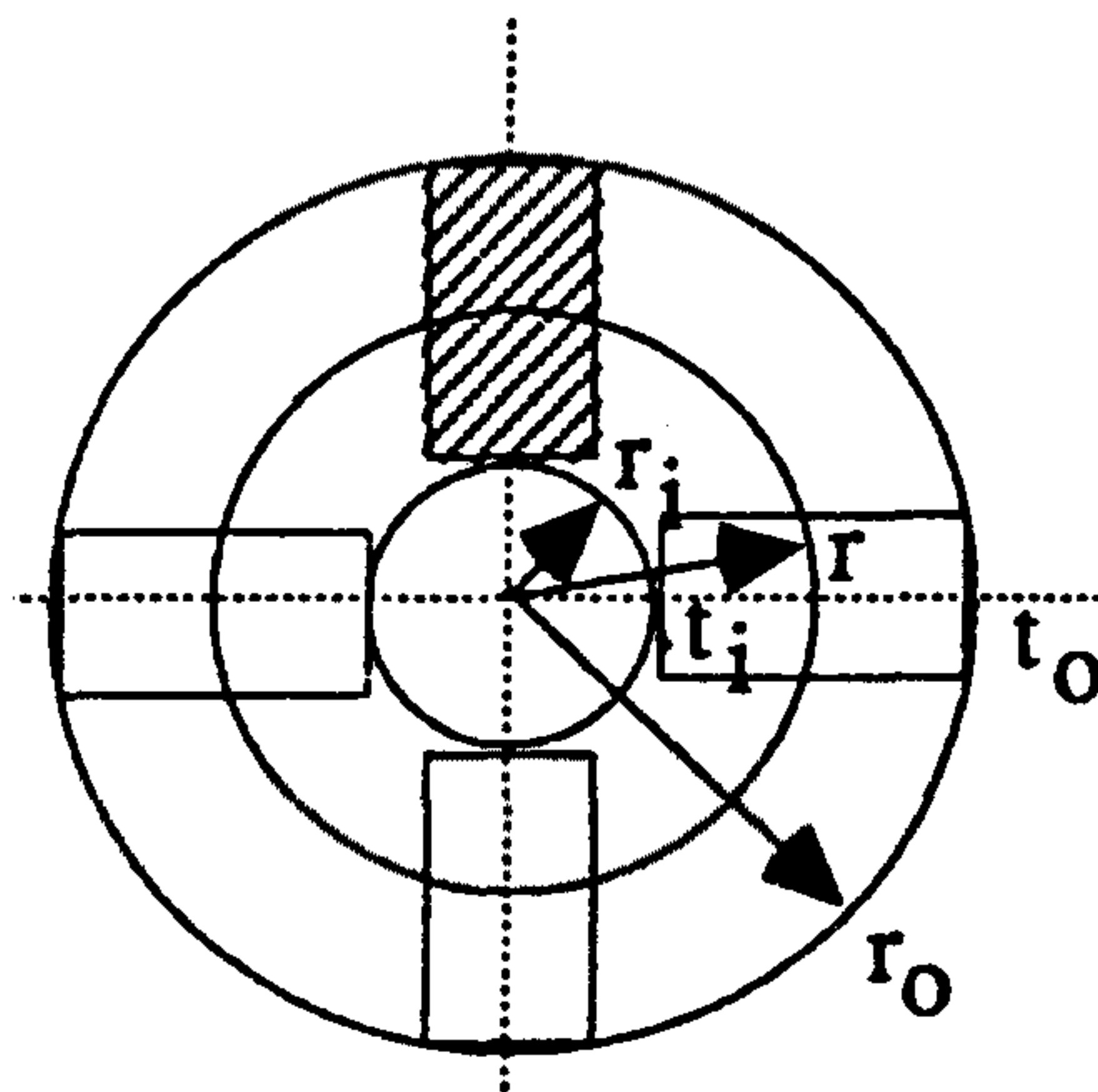


그림 2. 최적 Mold stone 형상

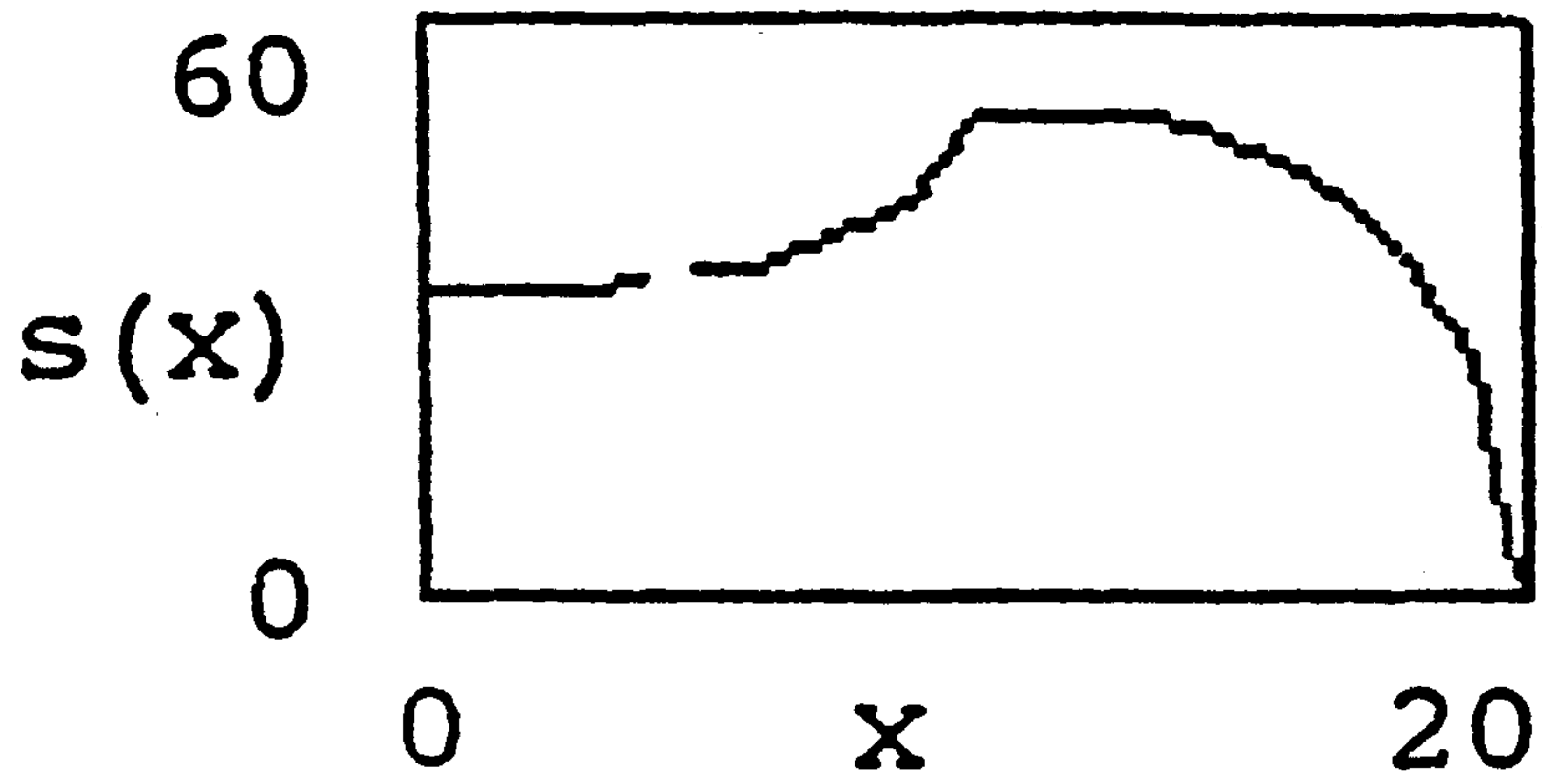


그림 3. 동적 연마량 해석

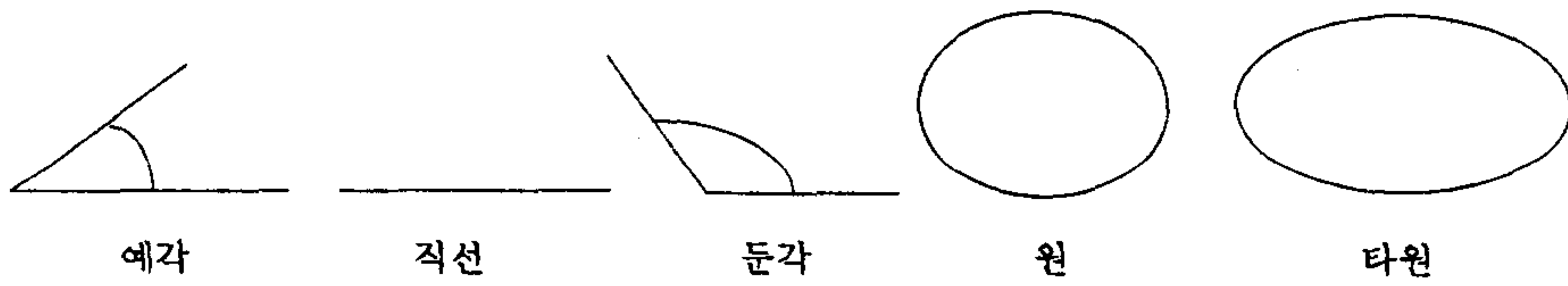


그림 4. 금형의 다양한 가장자리 형상

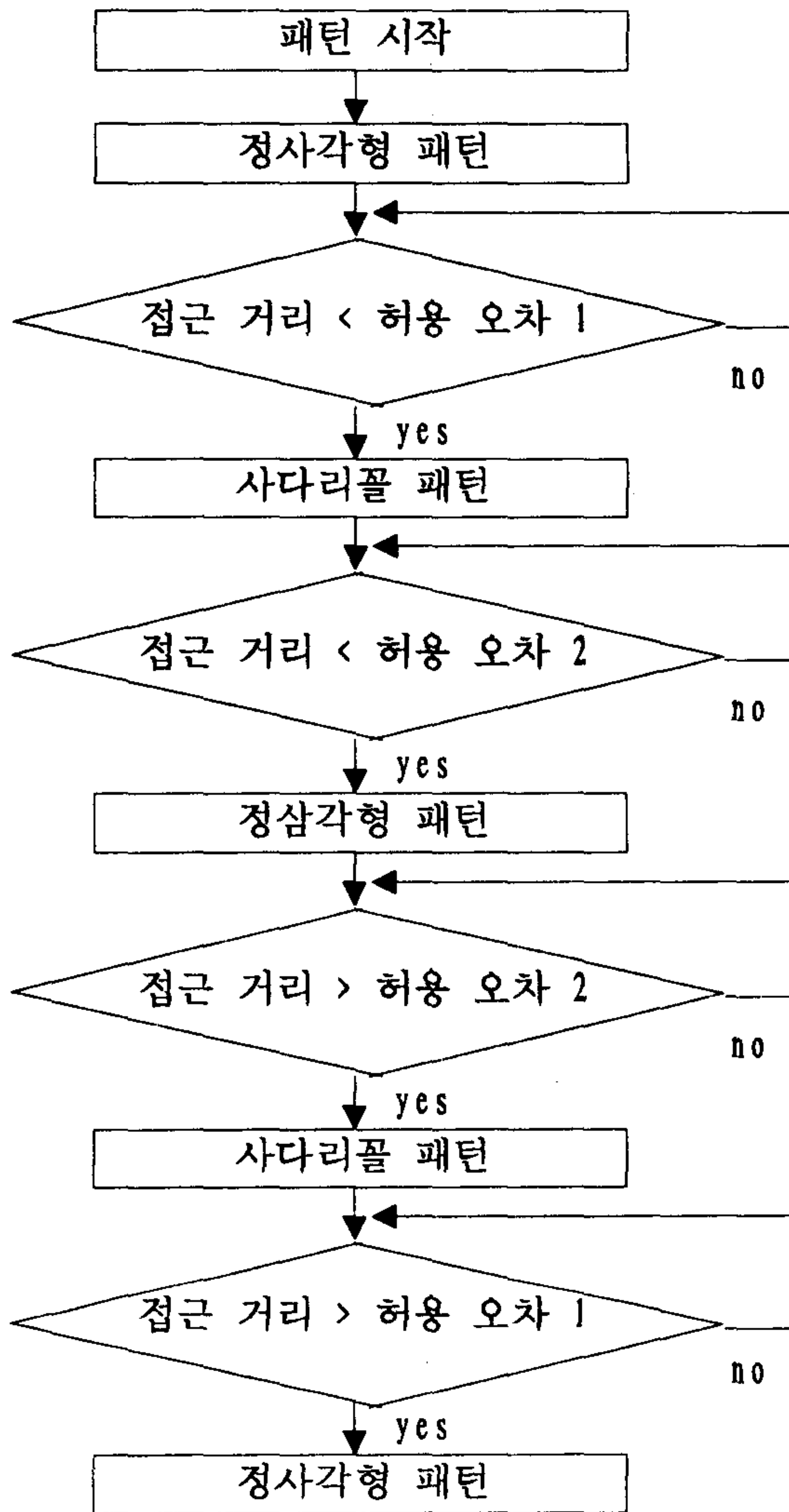
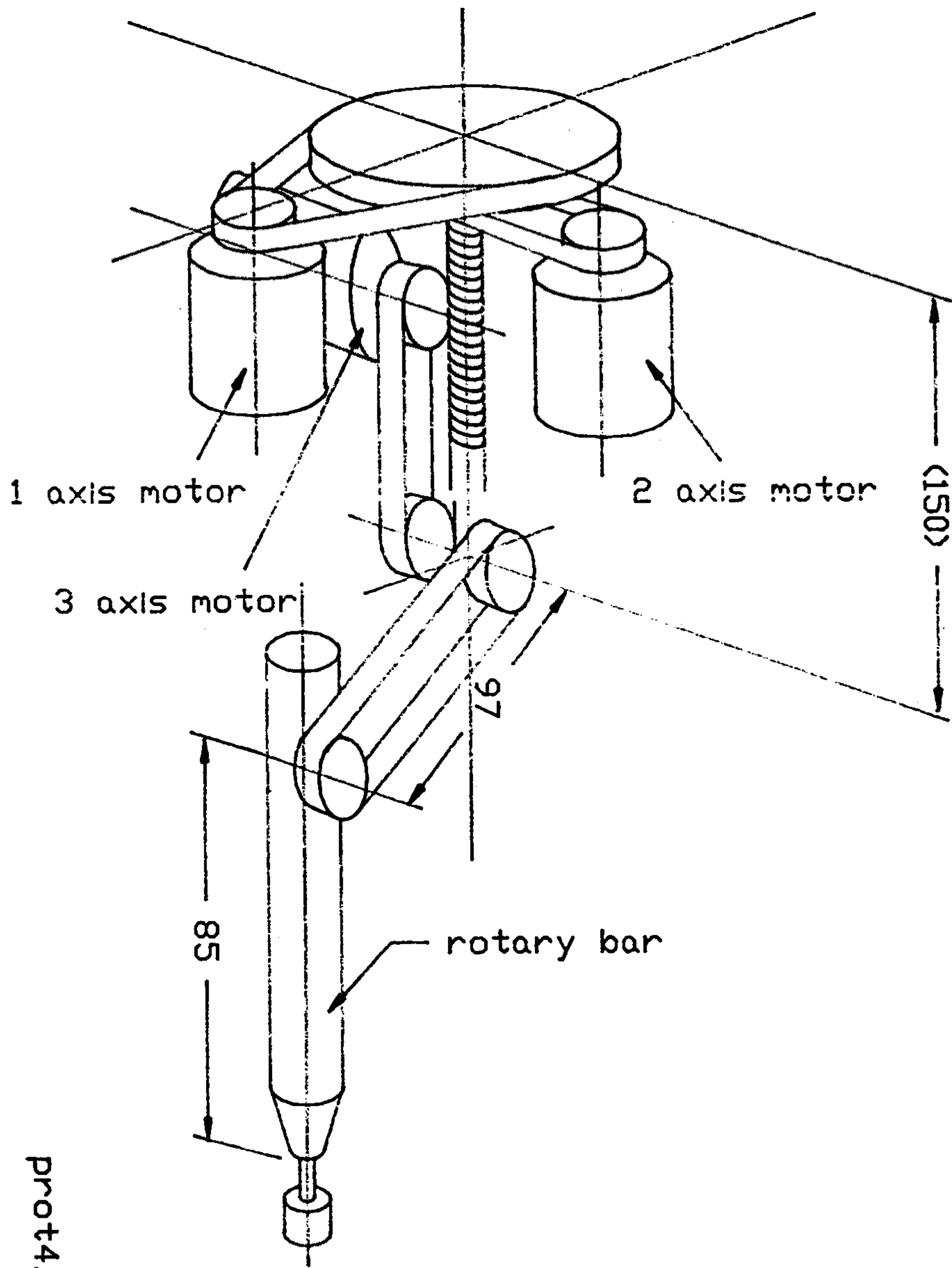


그림 5. 연마 패턴 형상 운용 방법



pro44/5/13/91

그림 6. 연마 로봇트 기본 구조

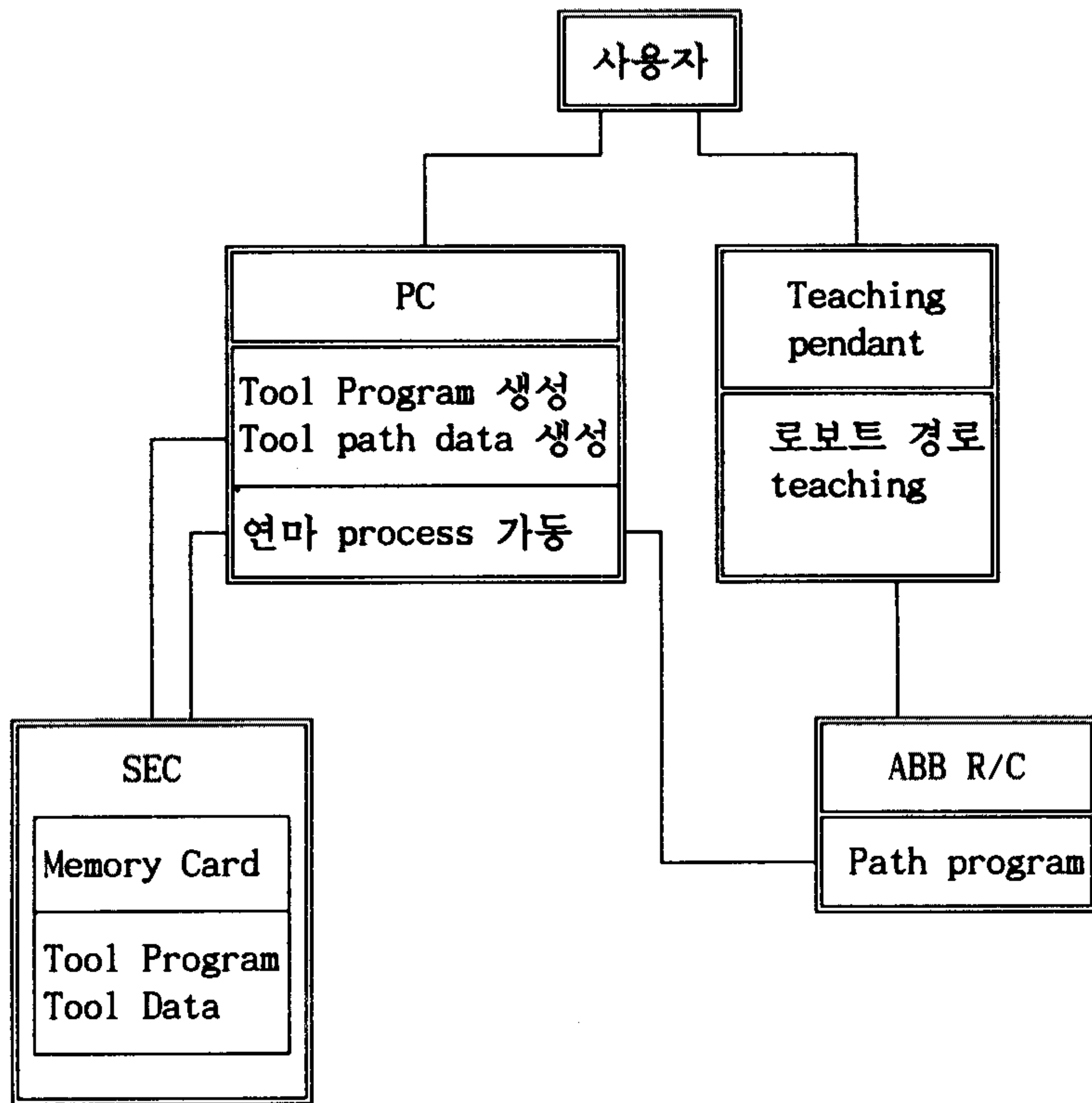


그림 7. 연마 시스템 동작 구조

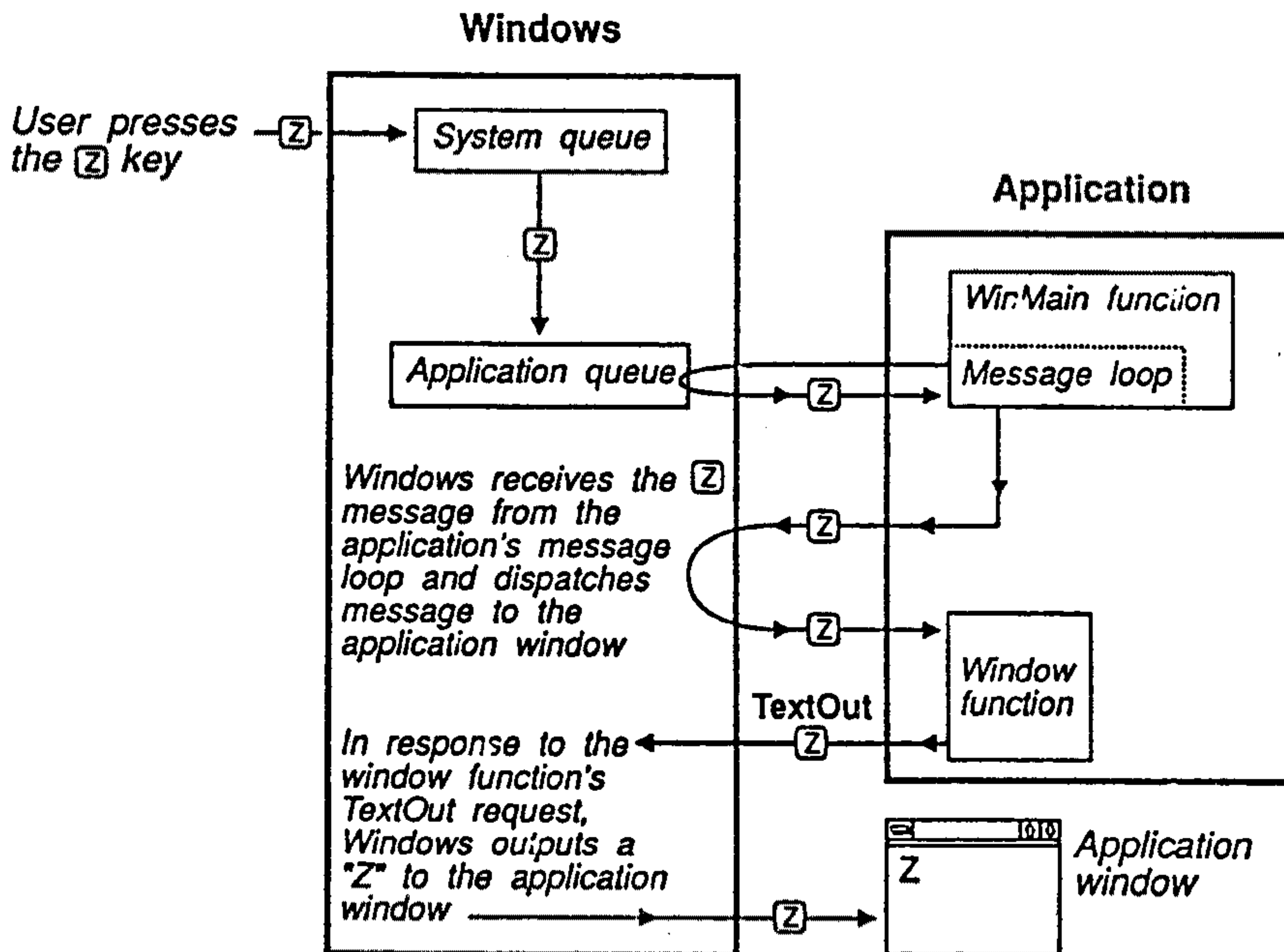


그림 8. MS-WINDOWS 메시지 처리 절차

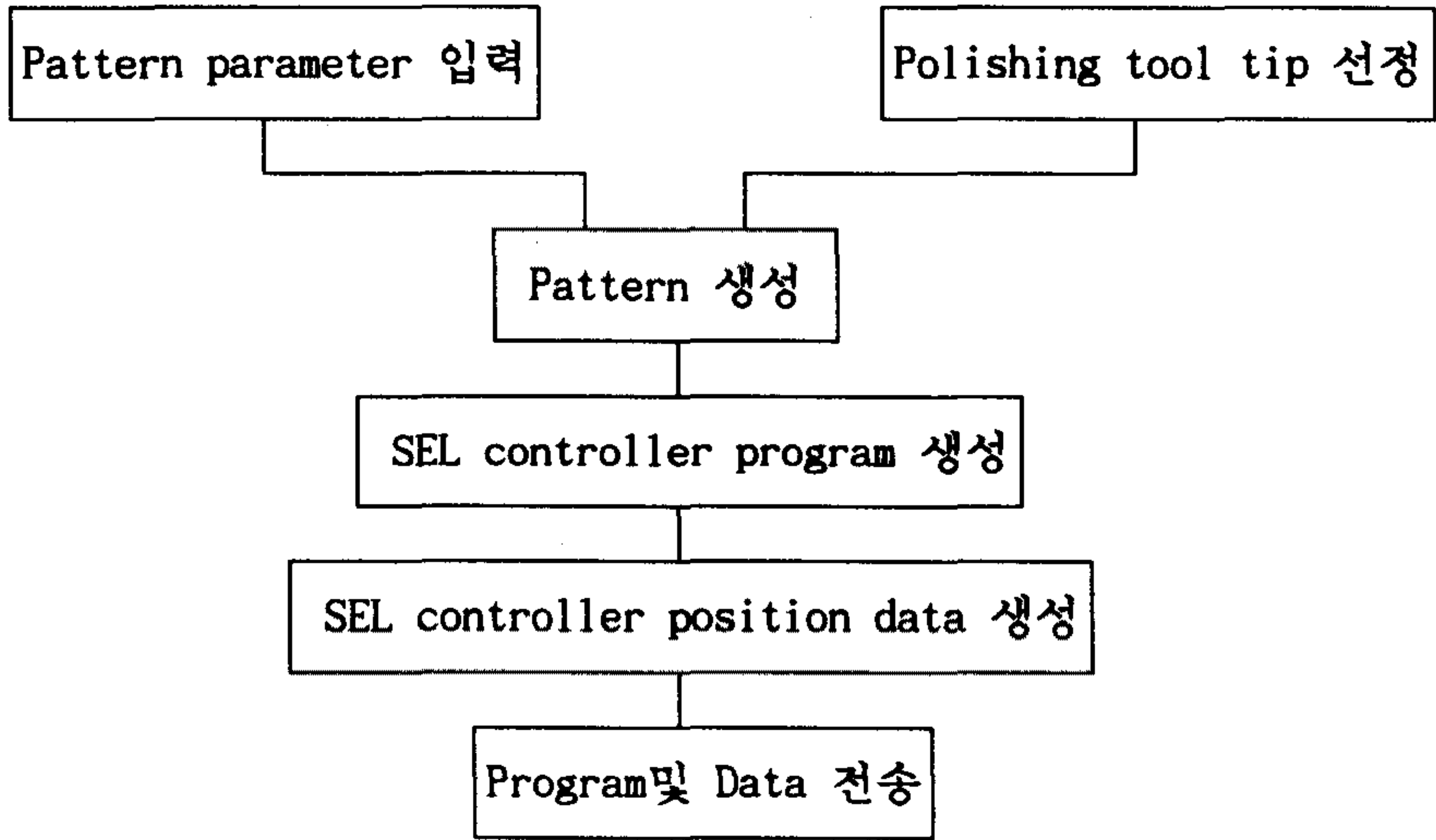


그림 9. 패턴 생성 절차

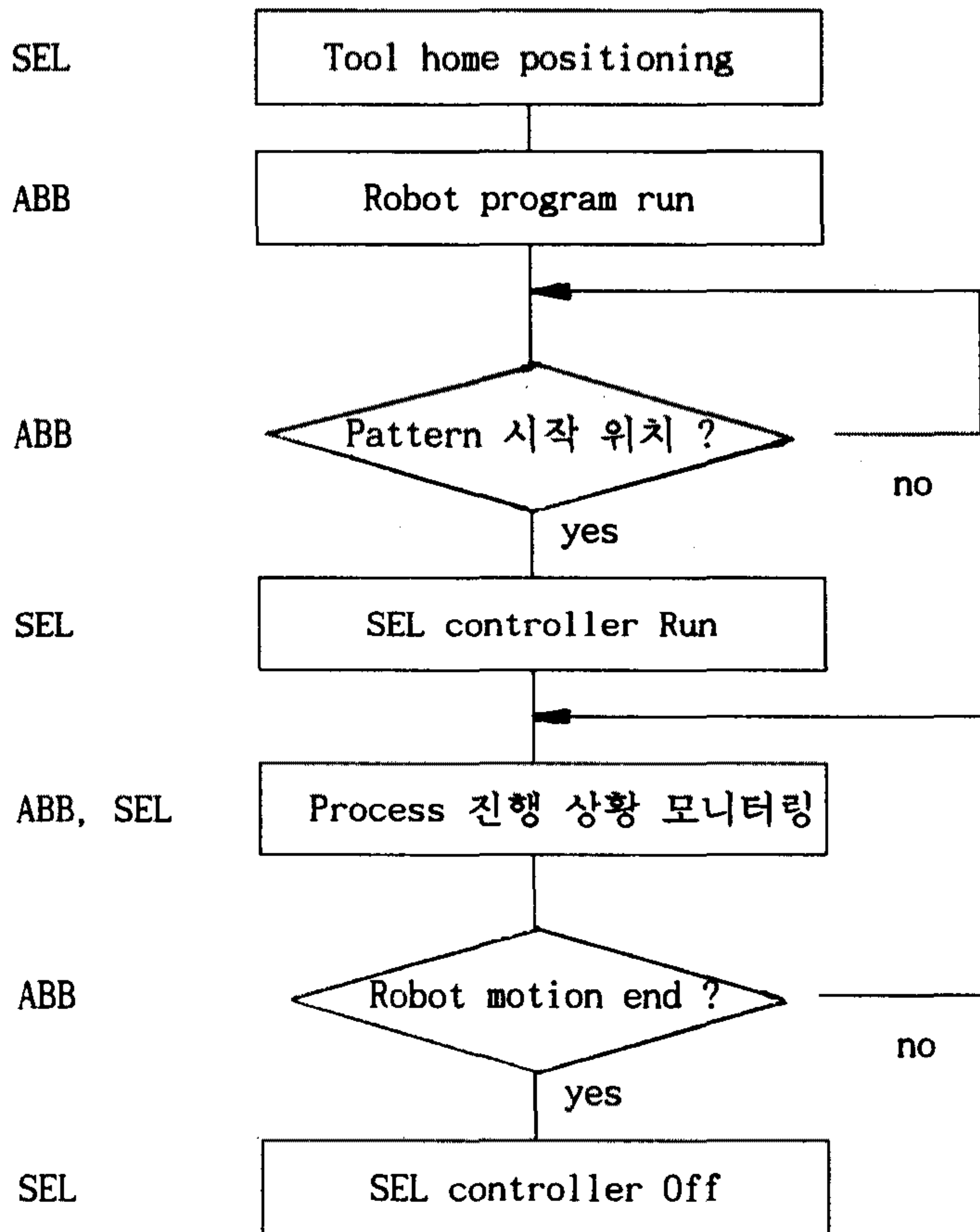


그림 10. 연마 프로세스

Pattern	SEL	Robot	File
Parameter generate	Program data version	Arm power on/off Synchronization Program Run	Load program Save program Load data Save data Exit

그림 11. S/W 메뉴 구성

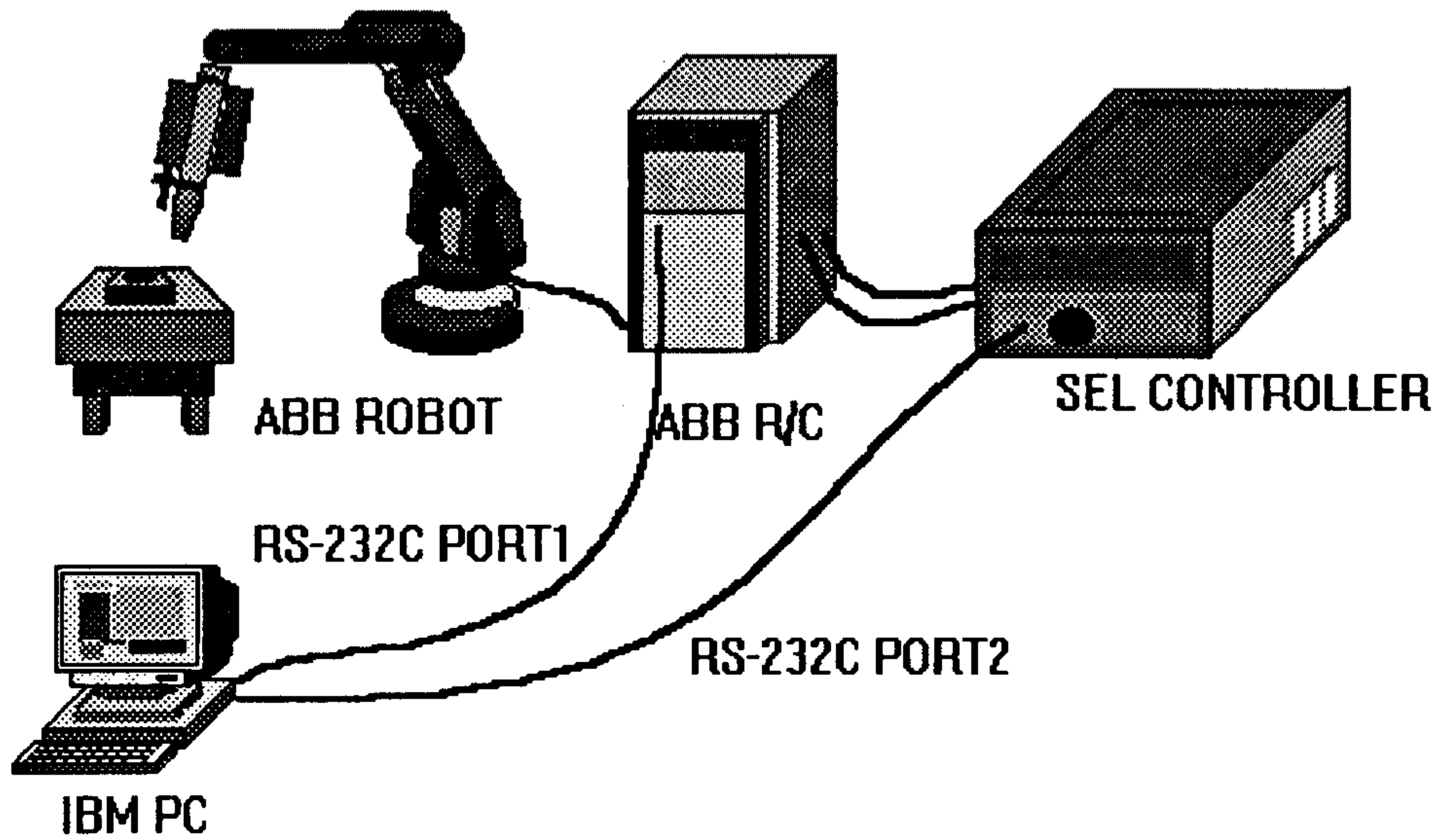


그림 12. System layout

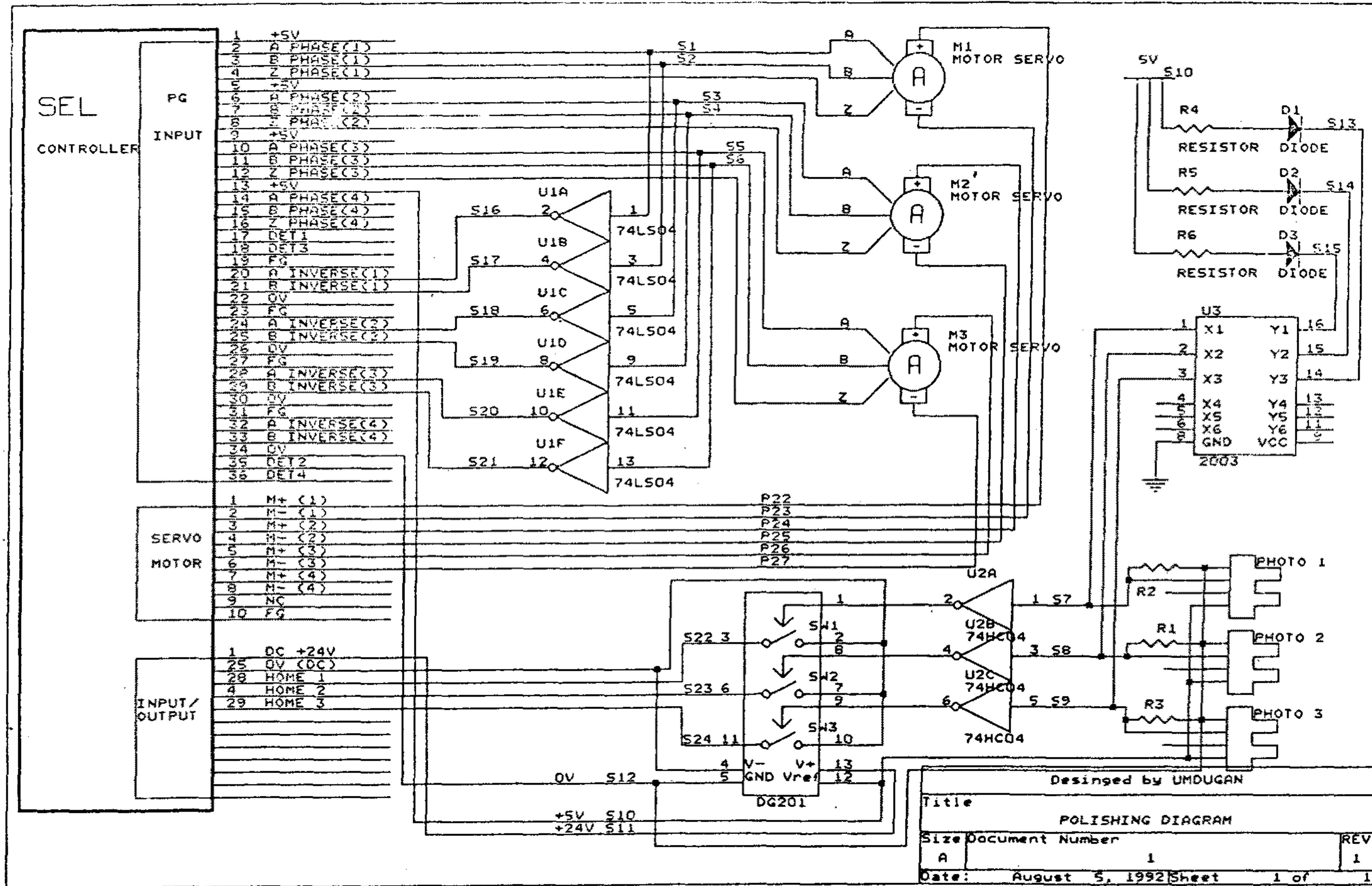
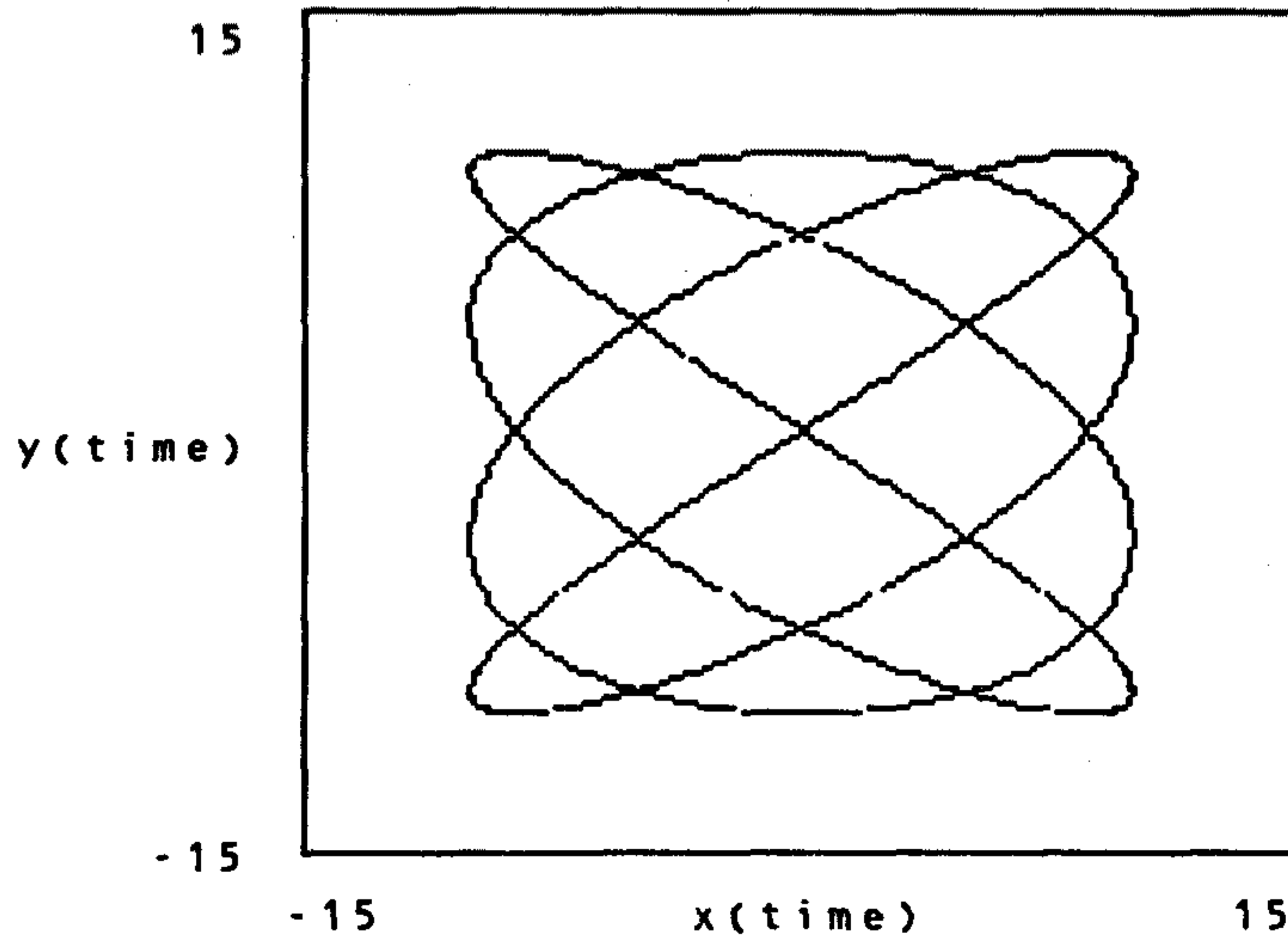


그림 13. H/W 신호 diagram

부록 1. 변형 Lissajous 패턴 공식 및 형상

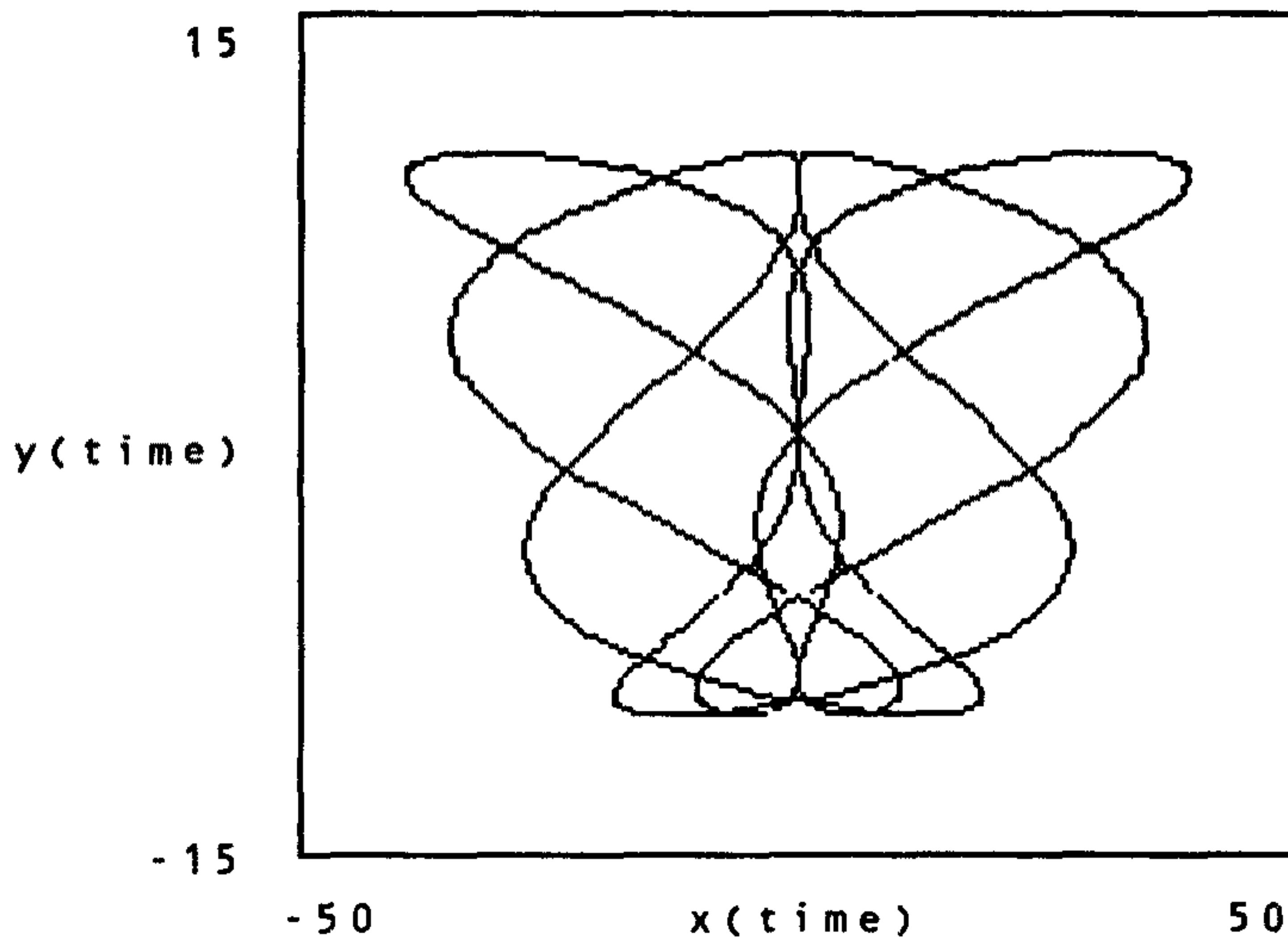
$$x(\text{time}) := x_a \cdot \sin \left[4 \cdot \text{time} \cdot \frac{\pi}{180} \right]$$

$$y(\text{time}) := y_a \cdot \sin \left[3 \cdot \text{time} \cdot \frac{\pi}{180} \right]$$



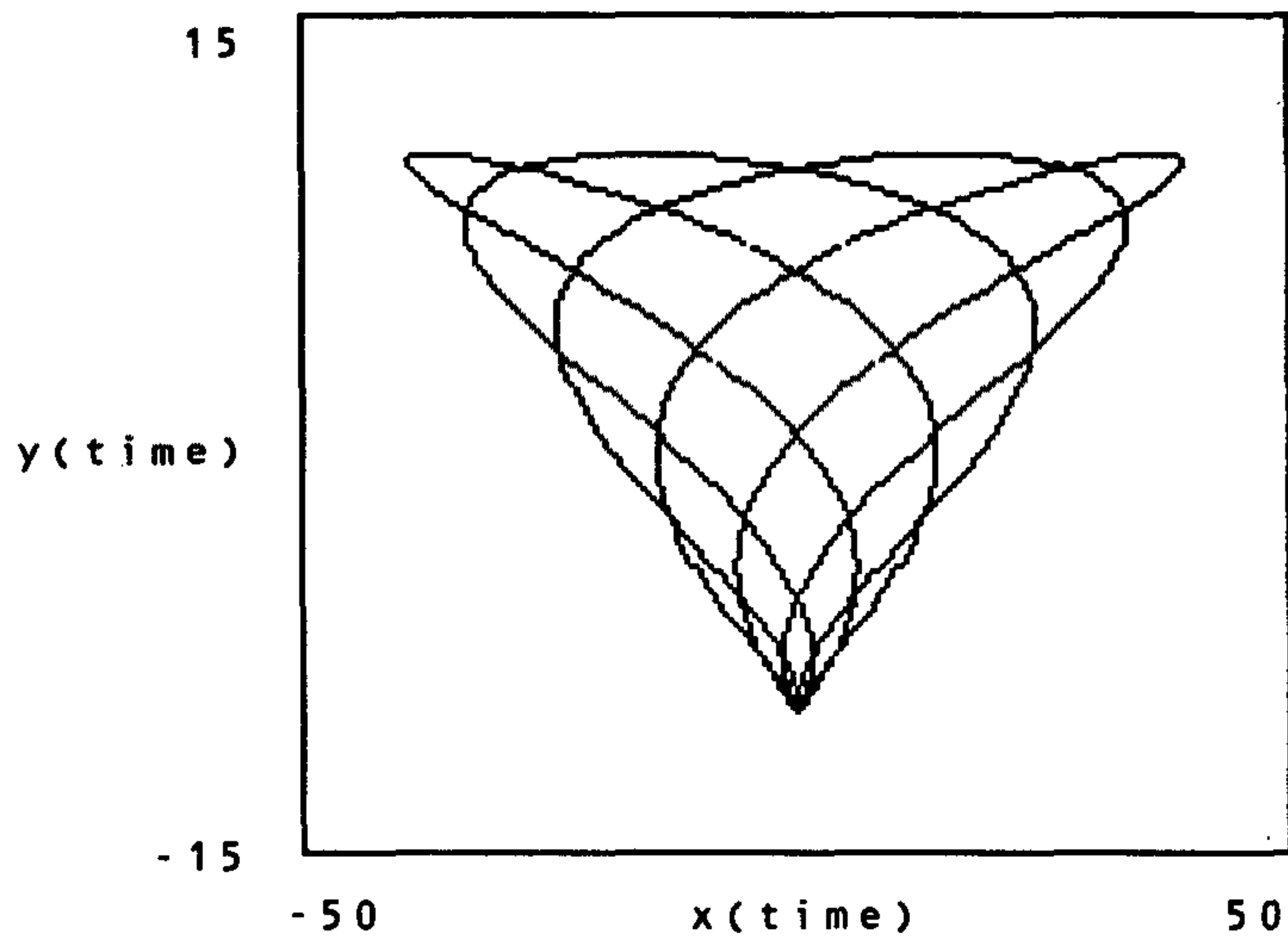
$$x(\text{time}) := x_a \cdot \sin \left[5 \cdot \text{time} \cdot \frac{\pi}{180} \right] \cdot \left[\frac{x_a}{5} \cdot \sin \left[0 \cdot \frac{\pi}{180} + \frac{6}{5} \cdot 5 \cdot \text{time} \cdot \frac{\pi}{180} \right] + \frac{x_a}{5} \right]$$

$$y(\text{time}) := y_a \cdot \sin \left[4 \cdot \text{time} \cdot \frac{\pi}{180} \right]$$

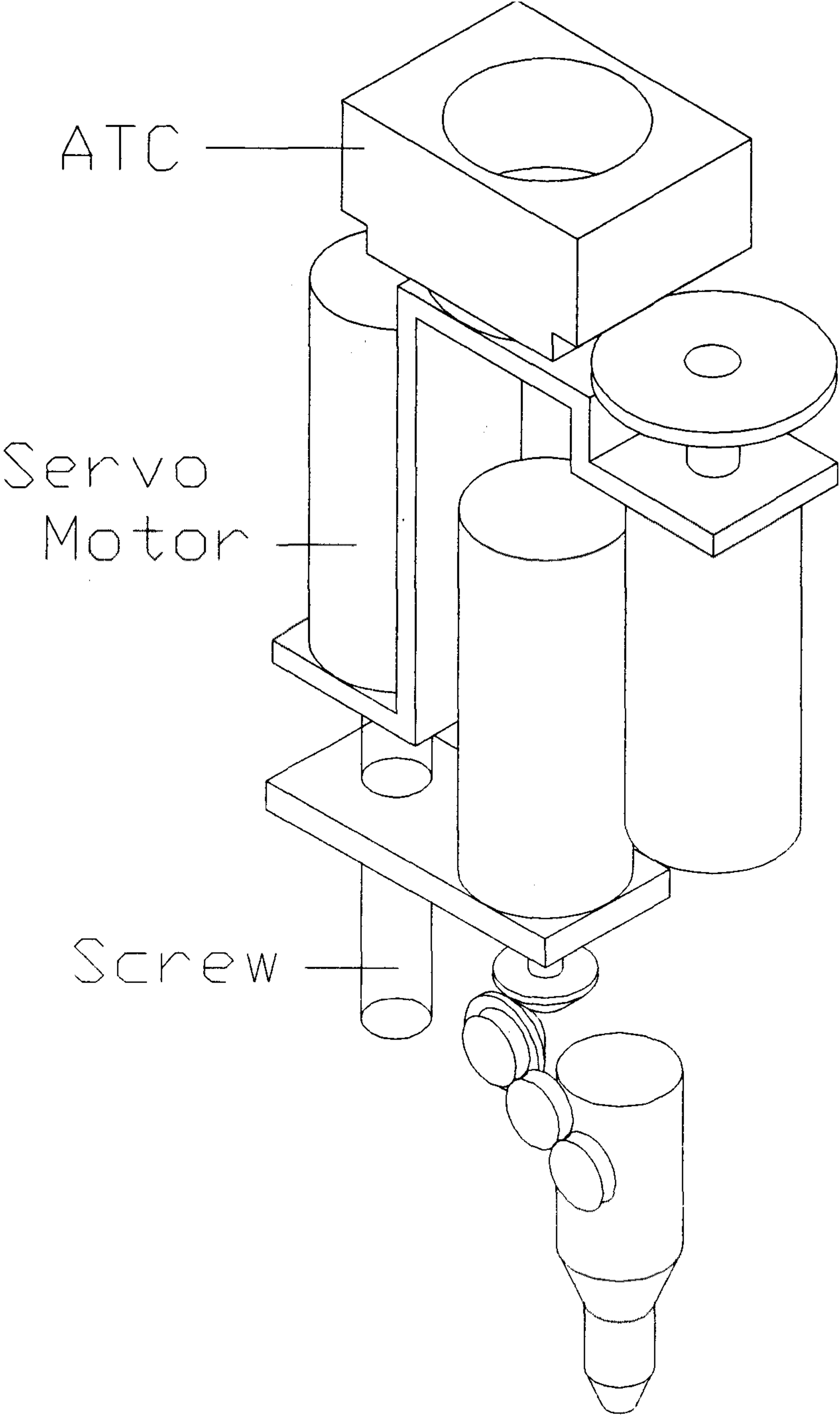


$$x(\text{time}) := x_a \cdot \sin \left[5 \cdot \text{time} \cdot \frac{\pi}{180} \right] \cdot \left[\frac{x_a}{5} \cdot \sin \left[\frac{4}{5} \cdot 5 \cdot \text{time} \cdot \frac{\pi}{180} \right] + \frac{x_a}{5} \right]$$

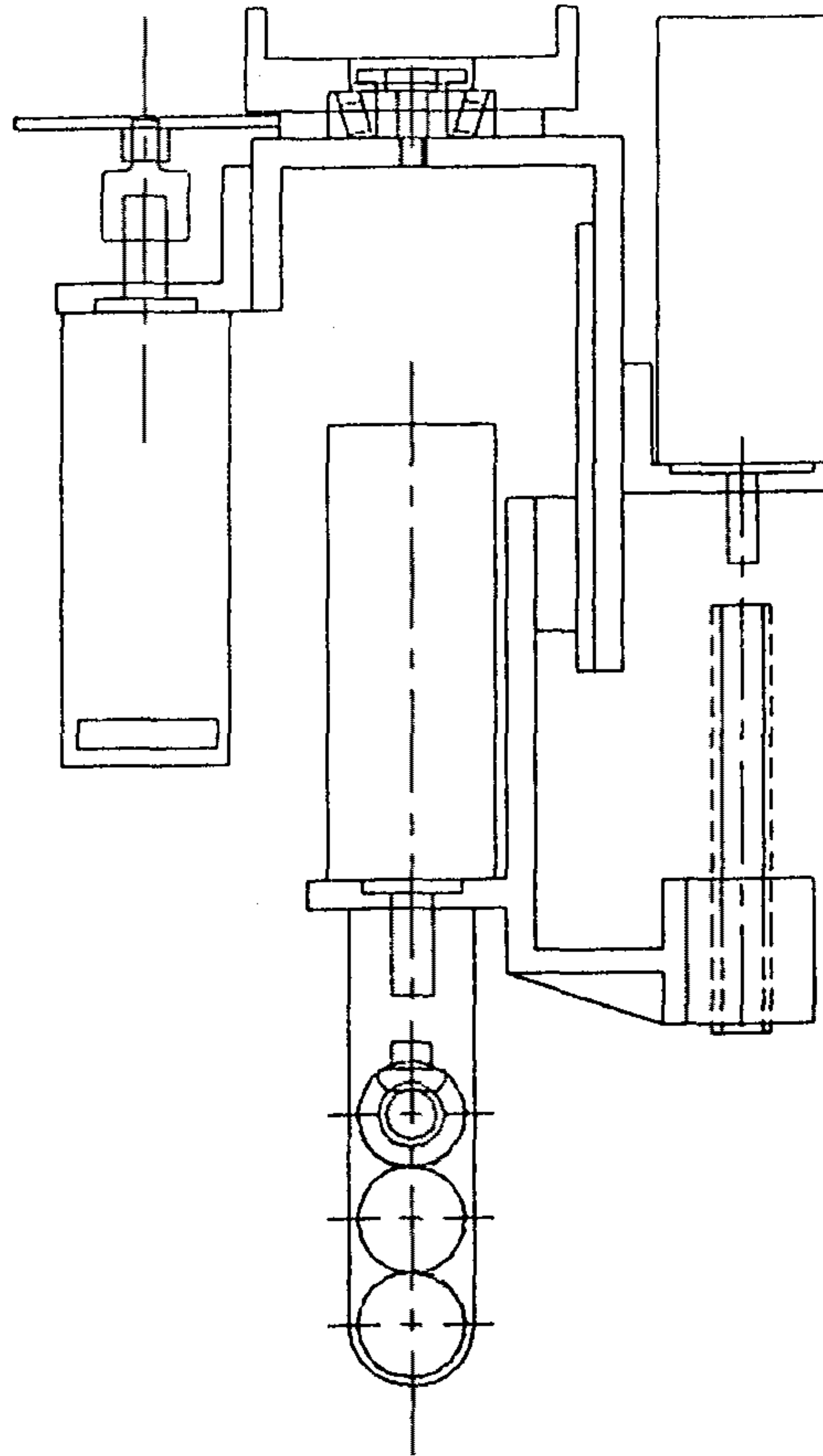
$$y(\text{time}) := y_a \cdot \sin \left[4 \cdot \text{time} \cdot \frac{\pi}{180} \right]$$



부록 2. 연마 로봇 설계도

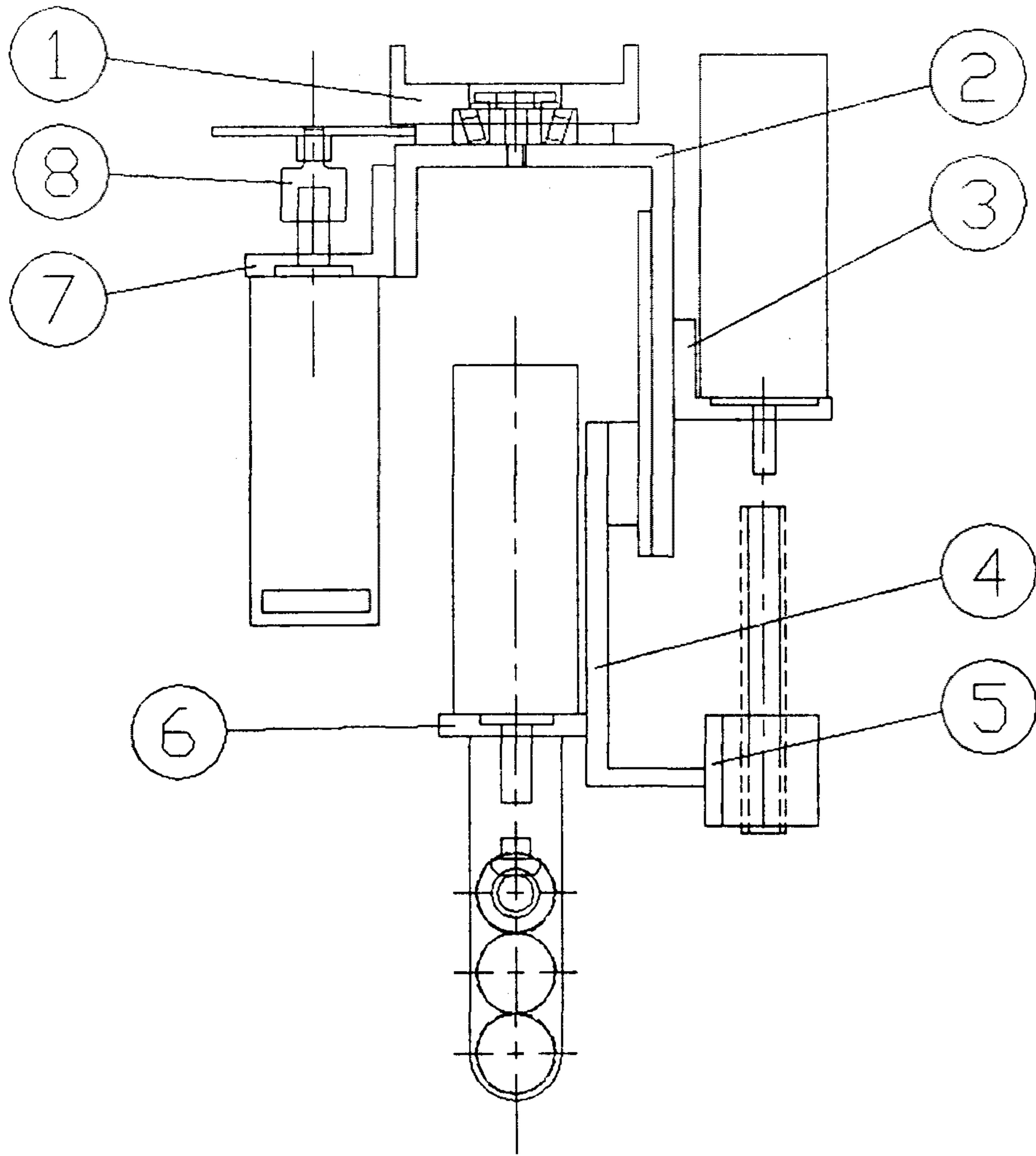


MOTOR R406-011E1X
MOTOR RH-11
MOTOR RH-11



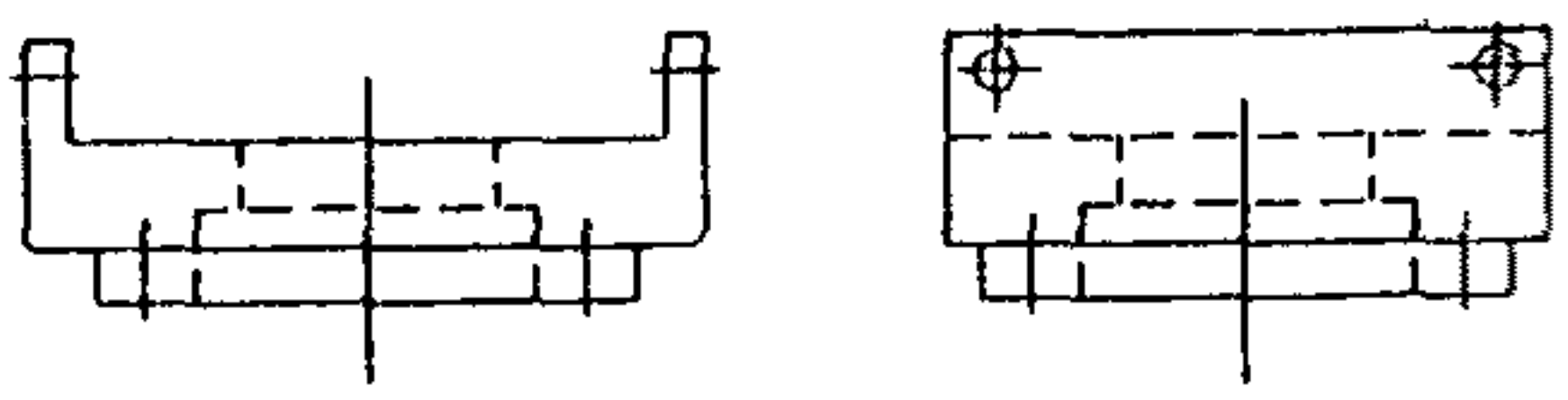
RSR12W
RSR12W RAIL
AP48R-120
F48A3-120
P64A21-64
AP64R-64
M48N-1ABA
M48N-1A
BEARING #30203
BNT1404C
BALL BEARING #1370
BALL SCREW

POLI-1



POLI-2B
LAYER 2 Frezz

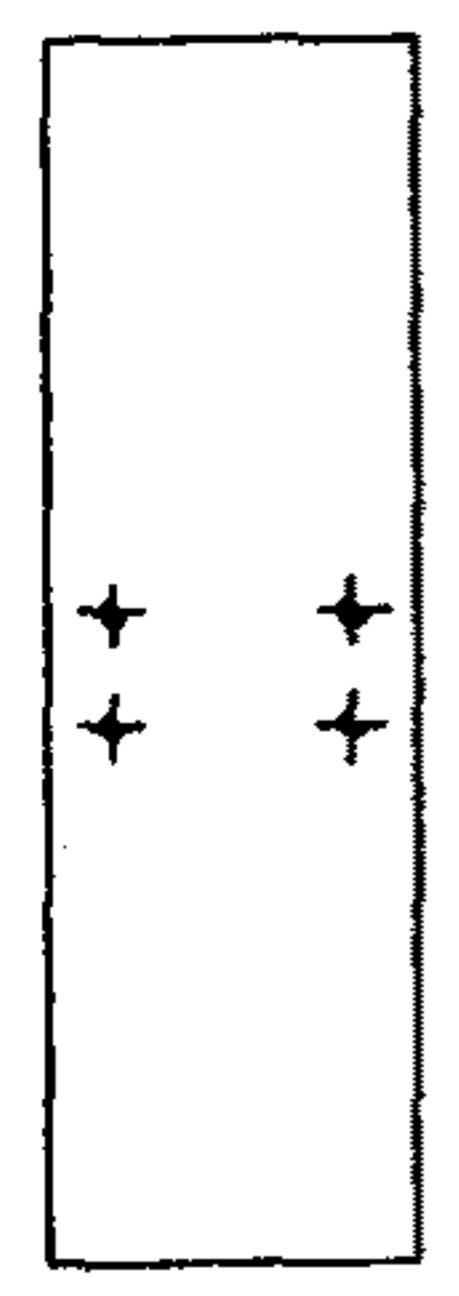
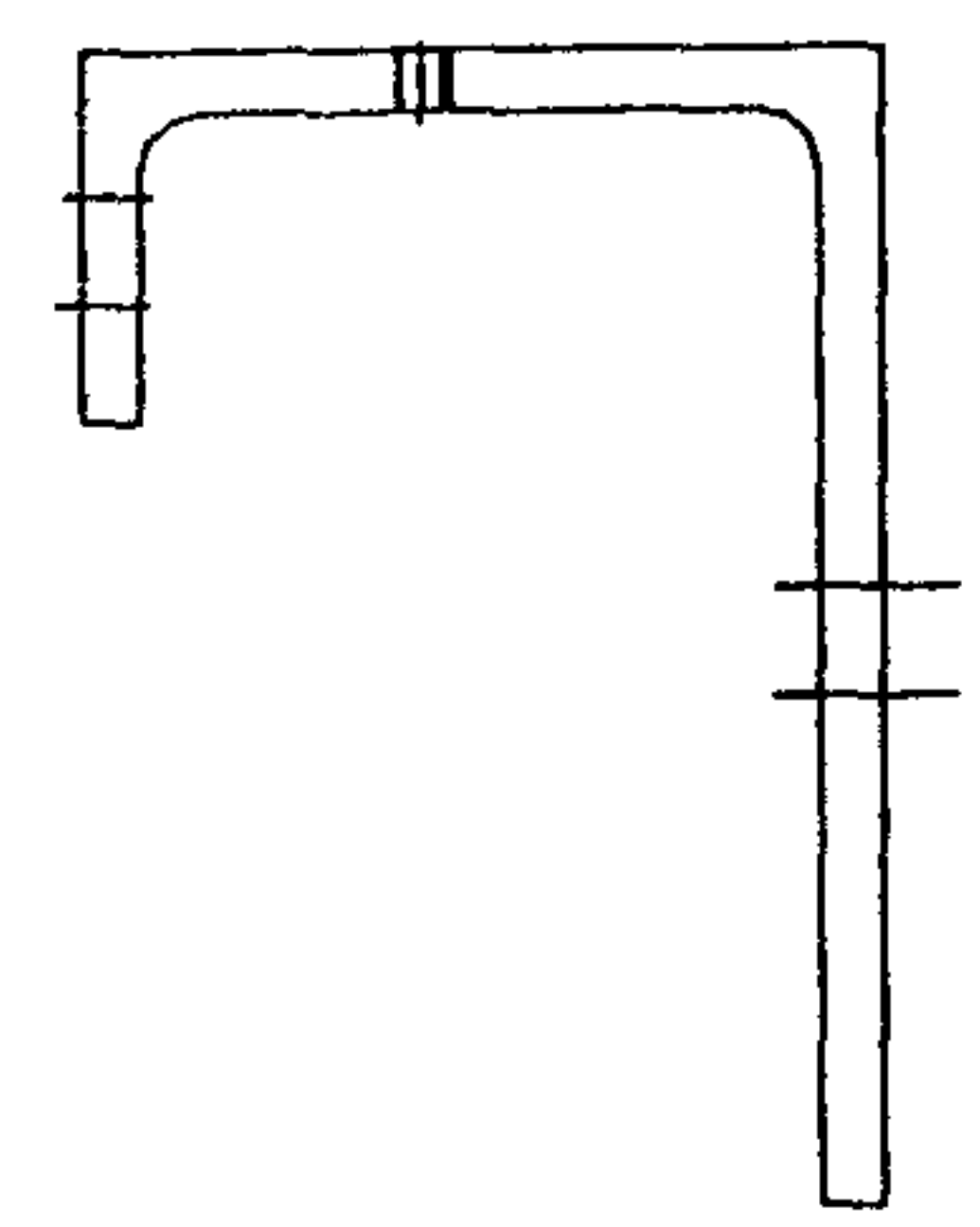
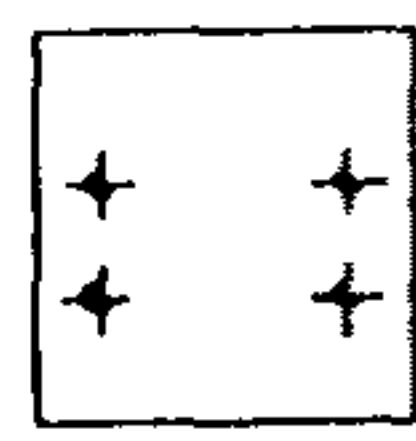
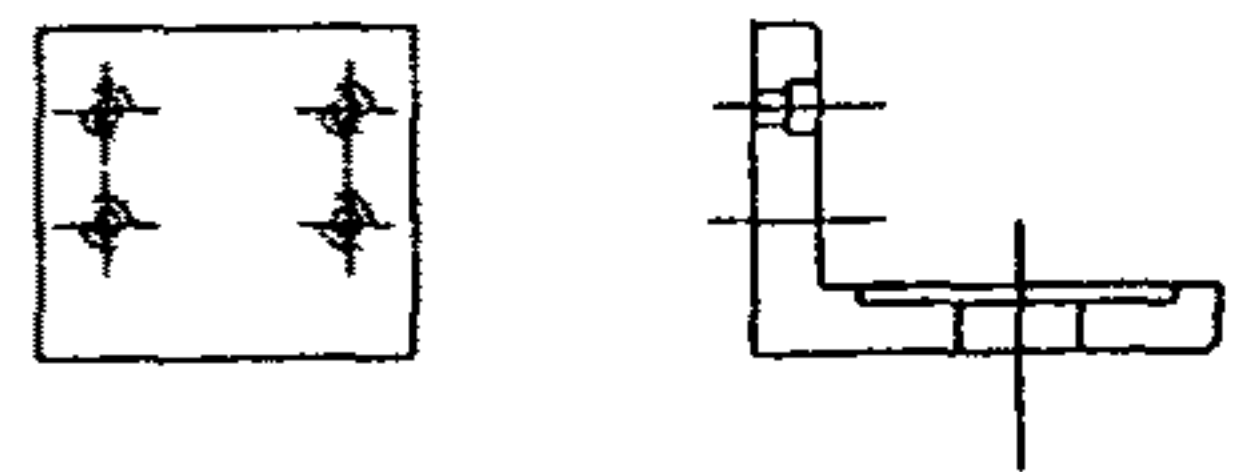
①



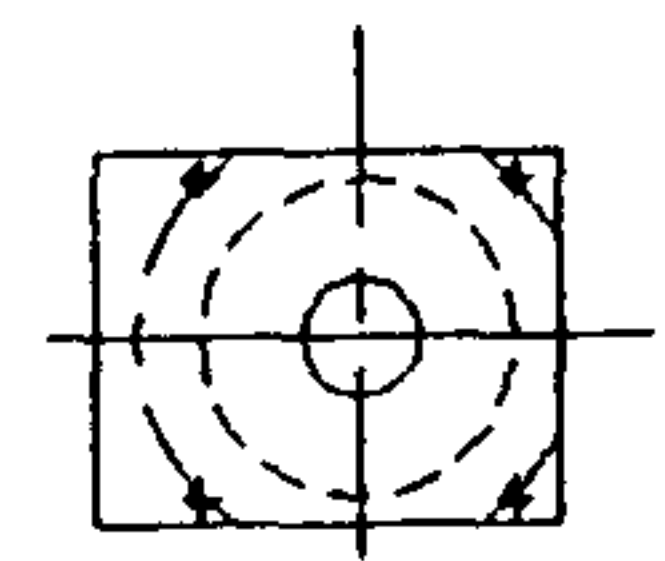
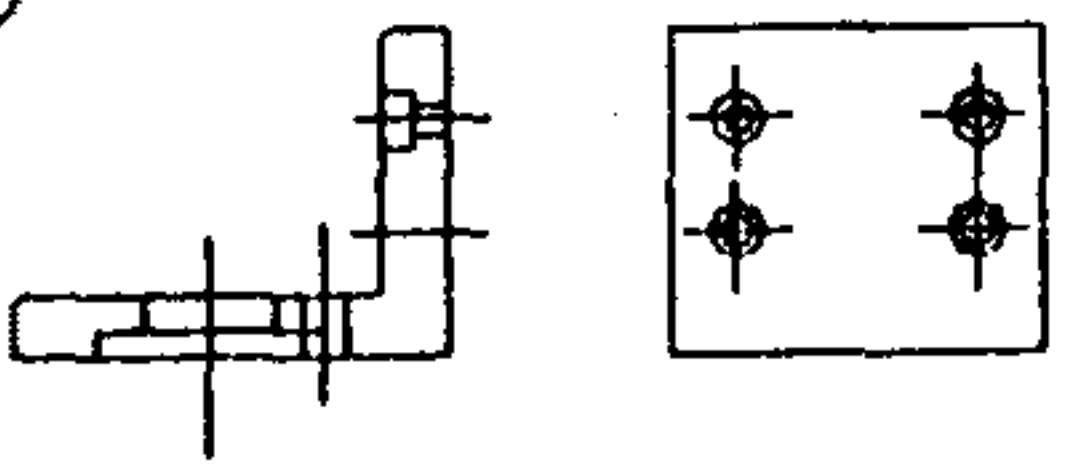
②



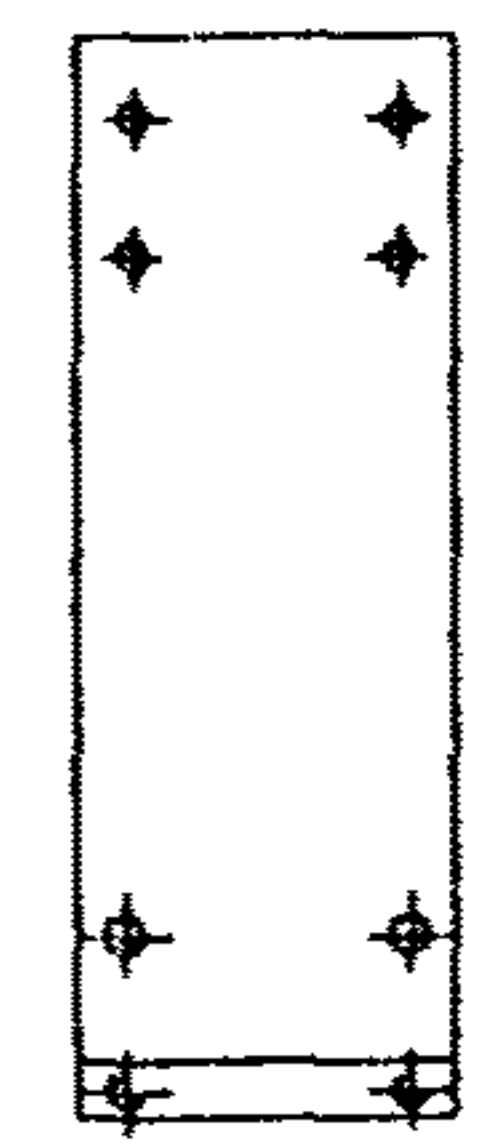
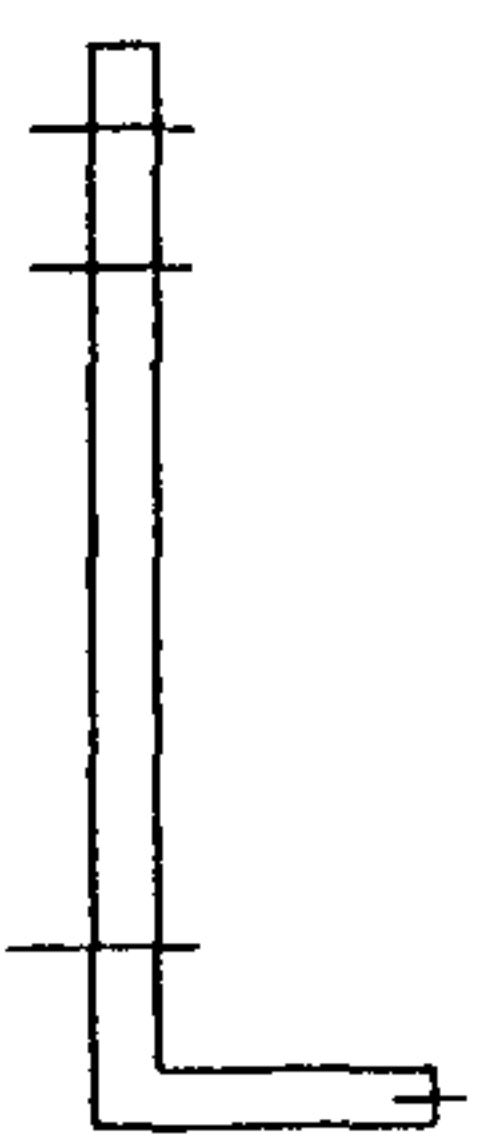
③



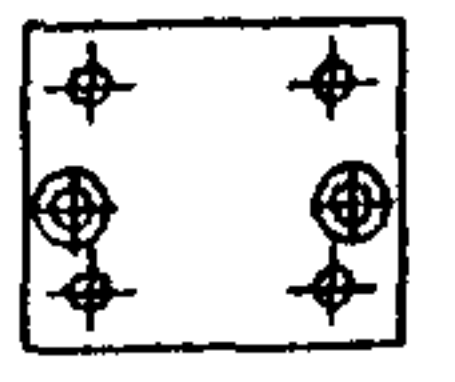
⑦



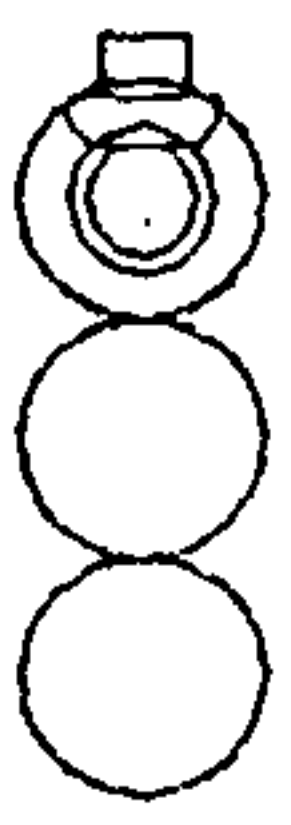
④



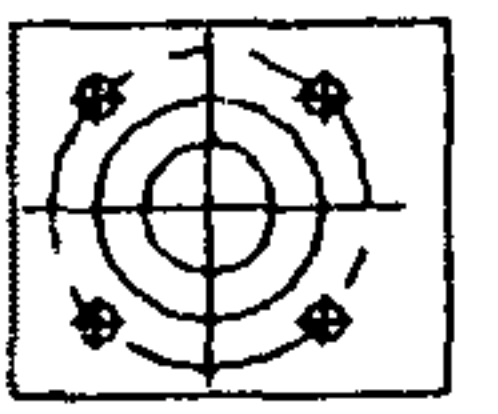
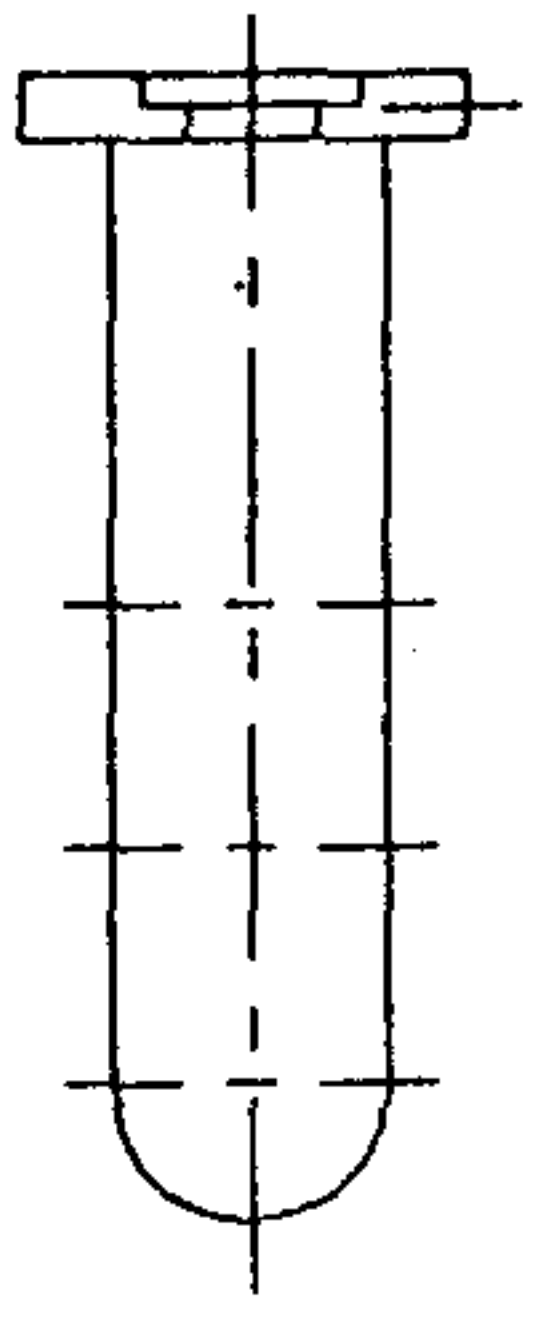
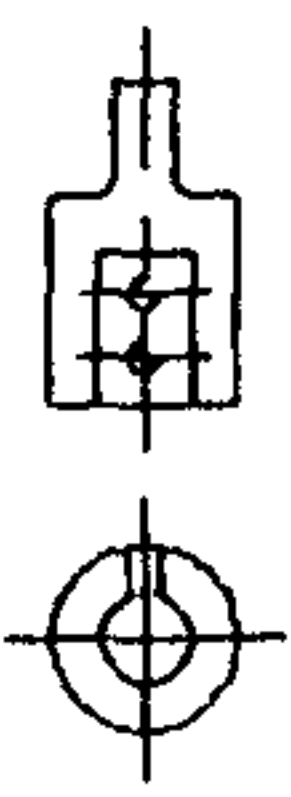
⑤



⑥

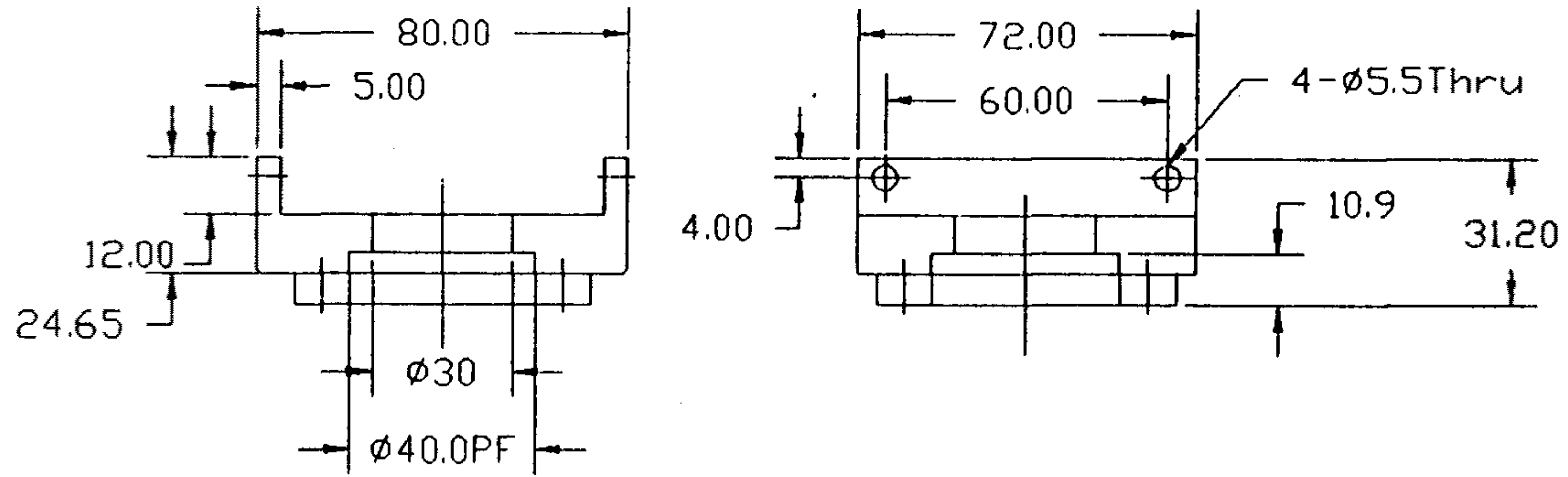


⑧

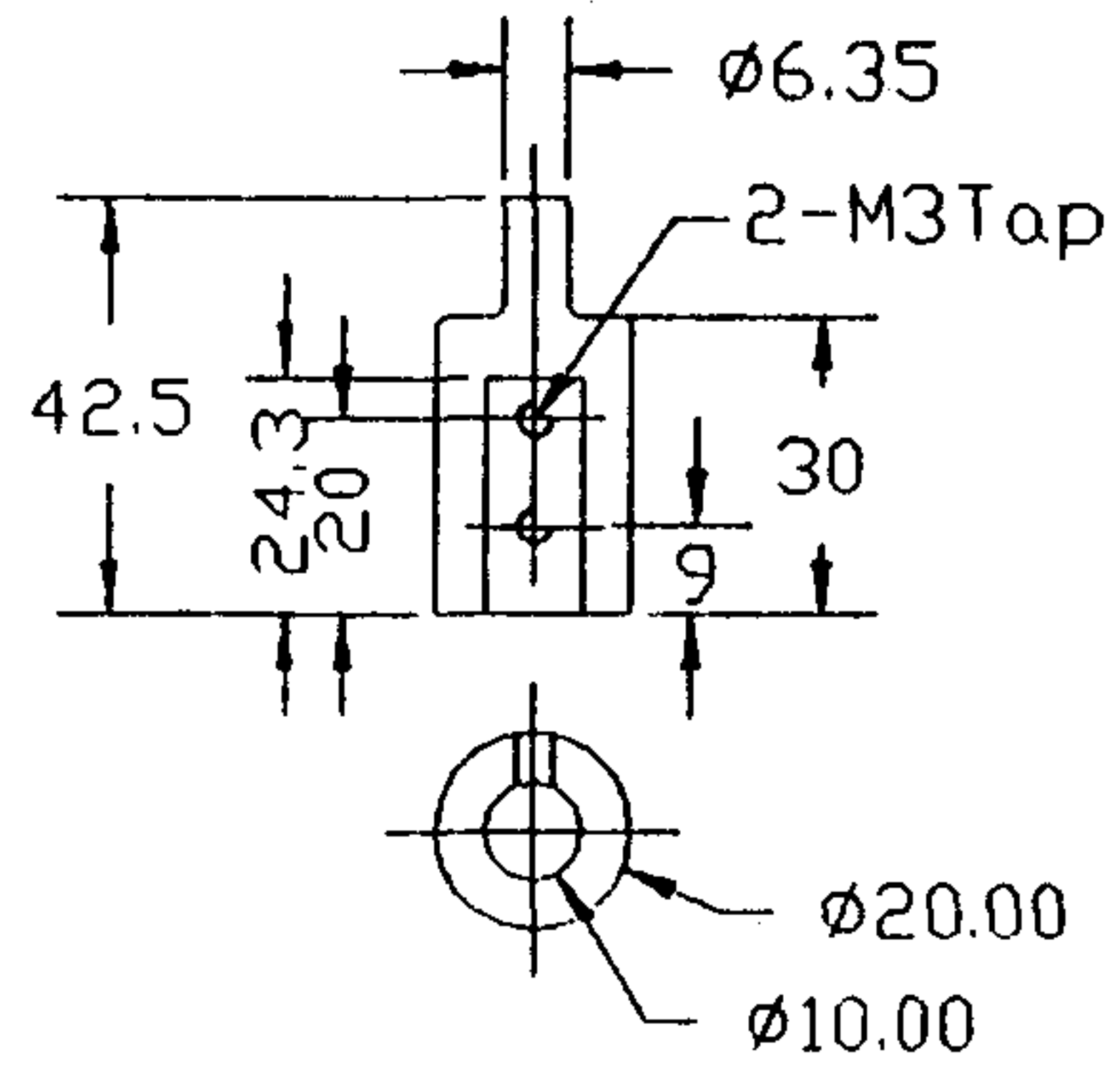


POLI-3

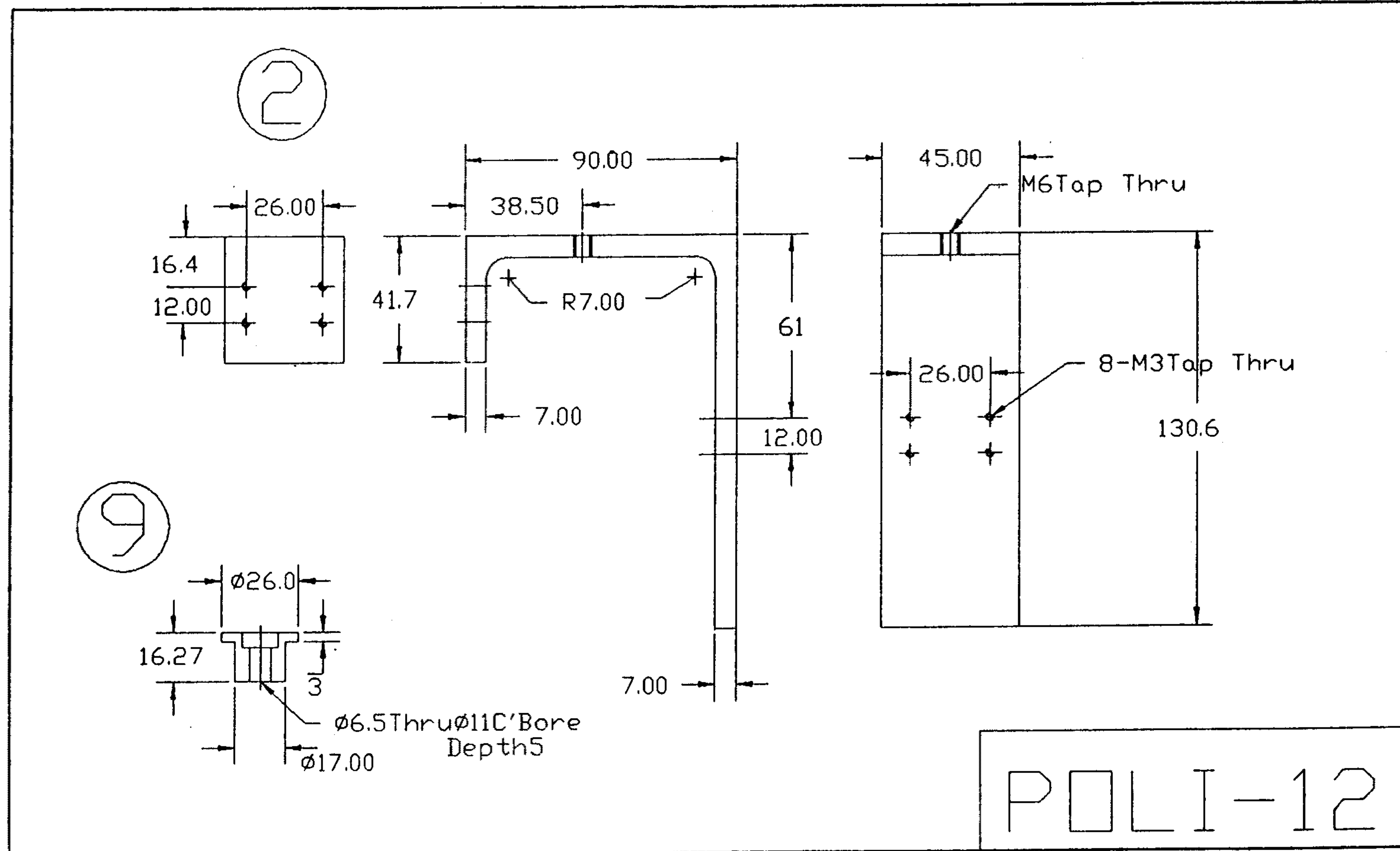
1

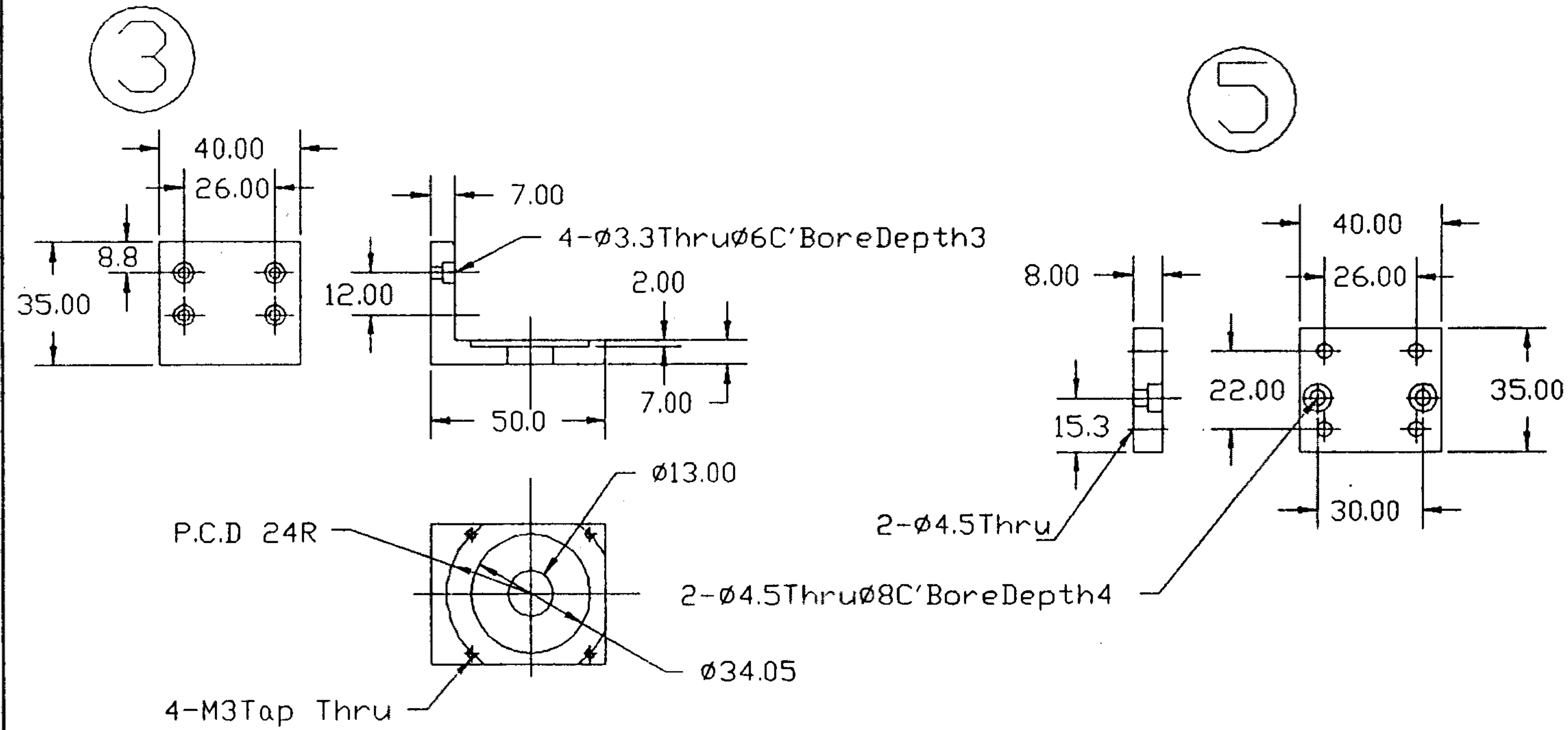


8



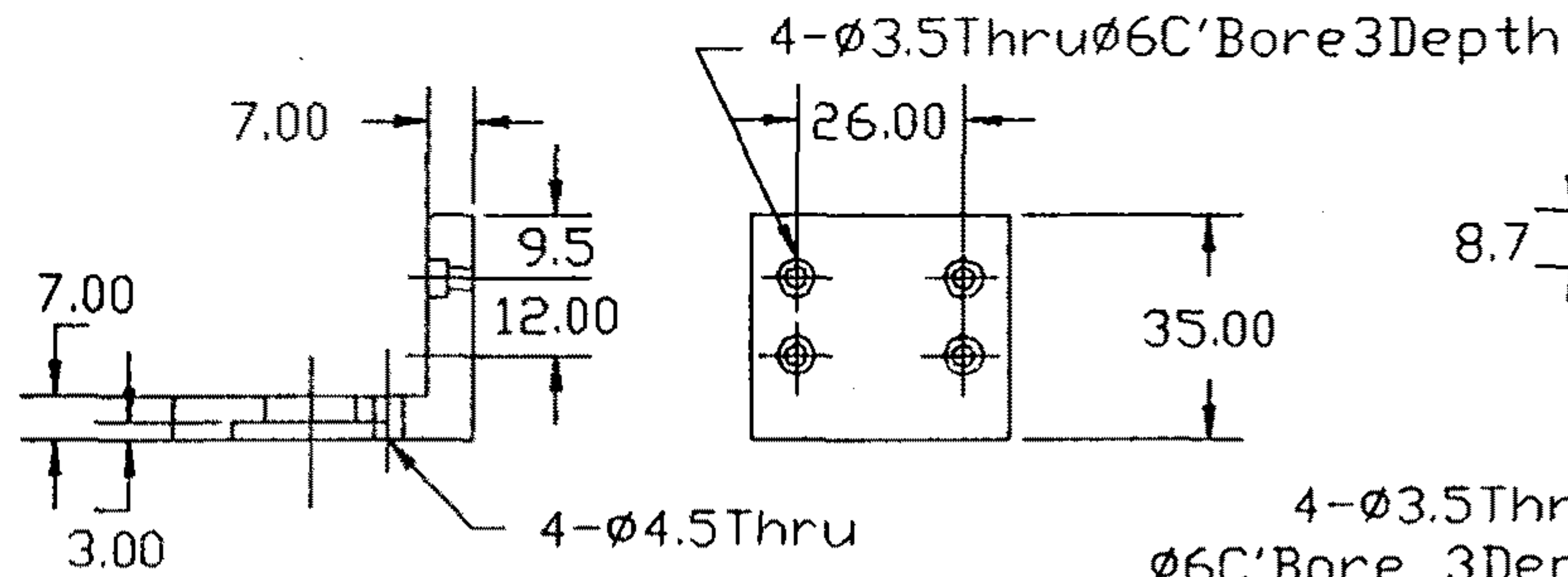
POLI-11



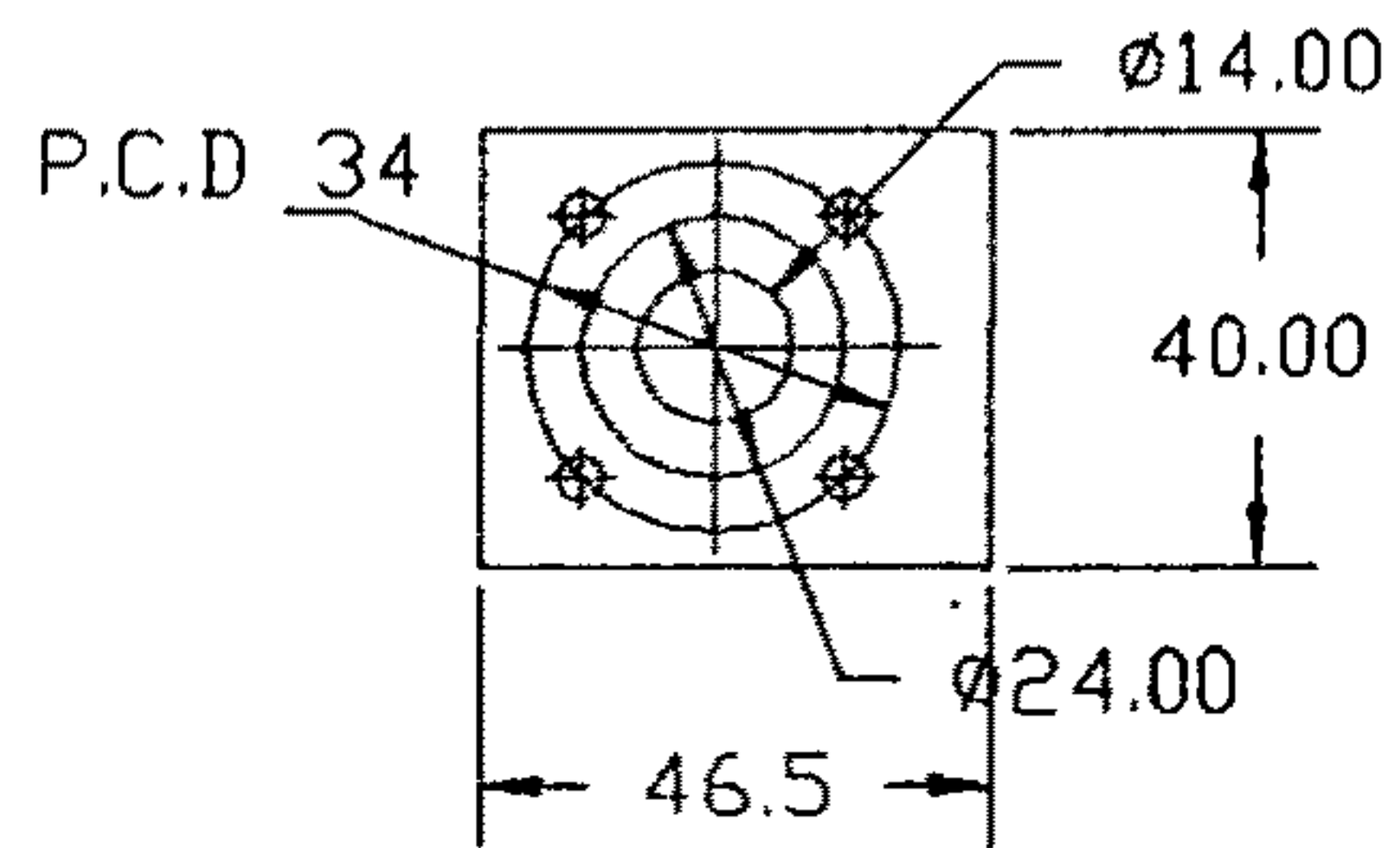
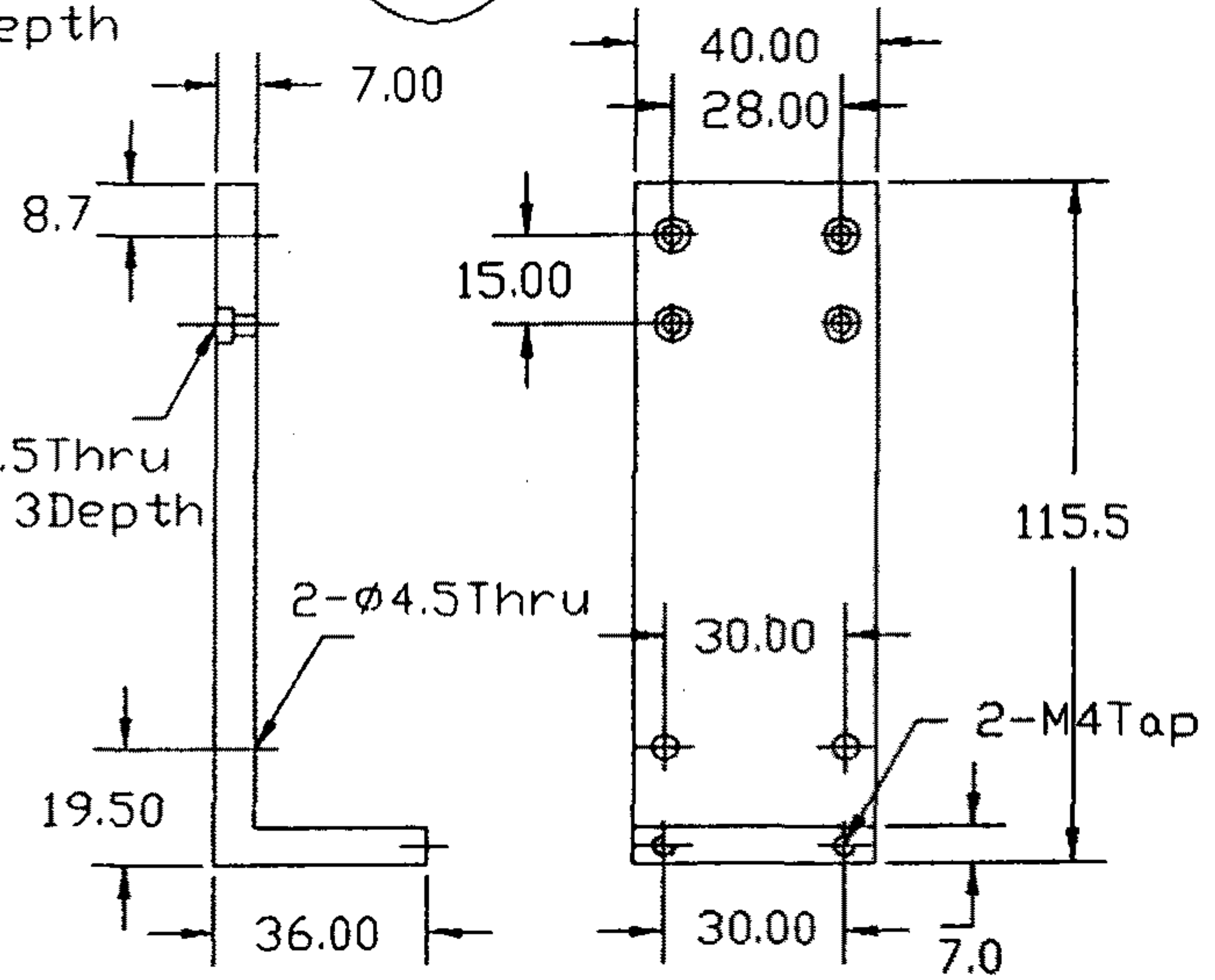


POLI-13

7

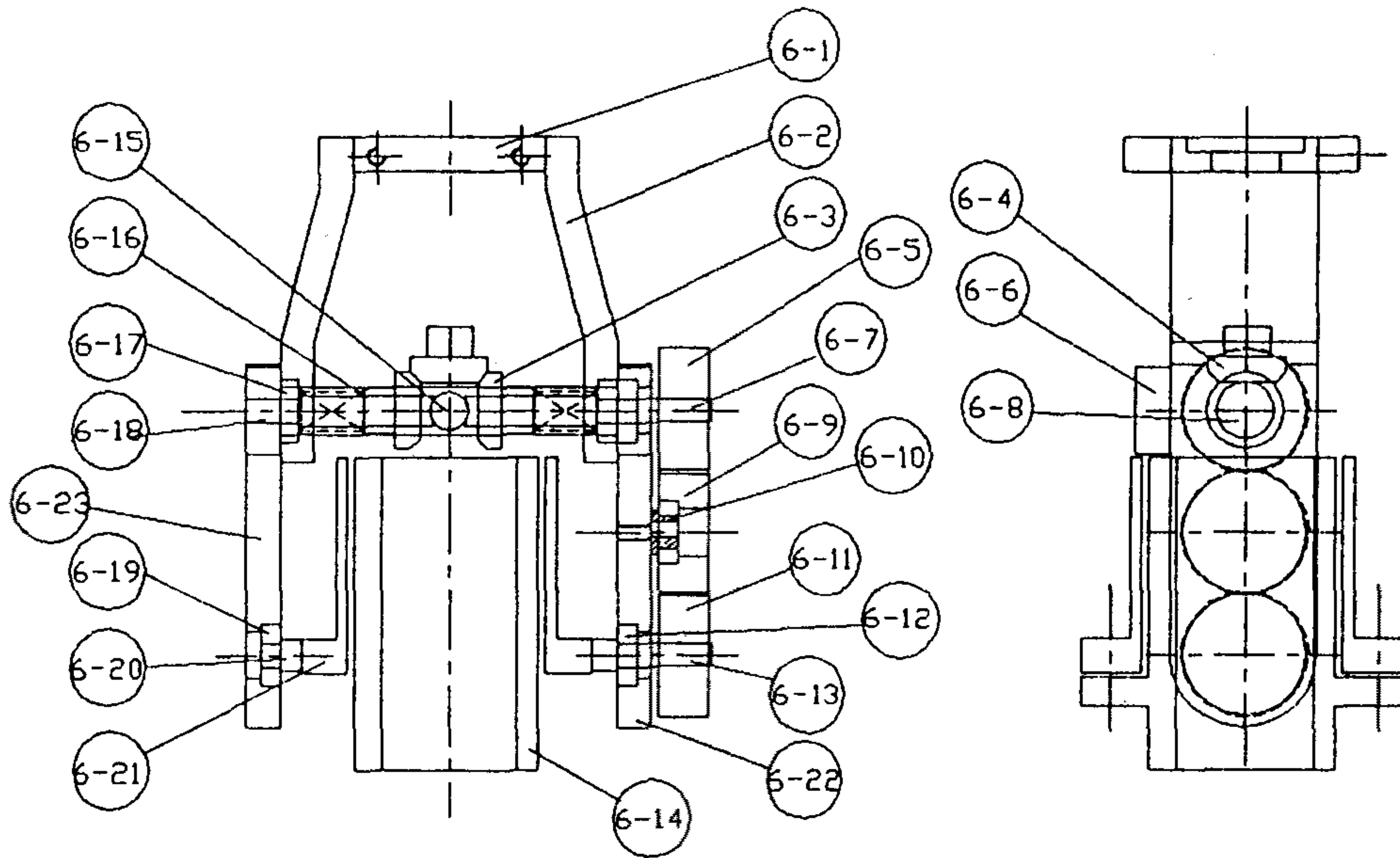


4



POLI-14

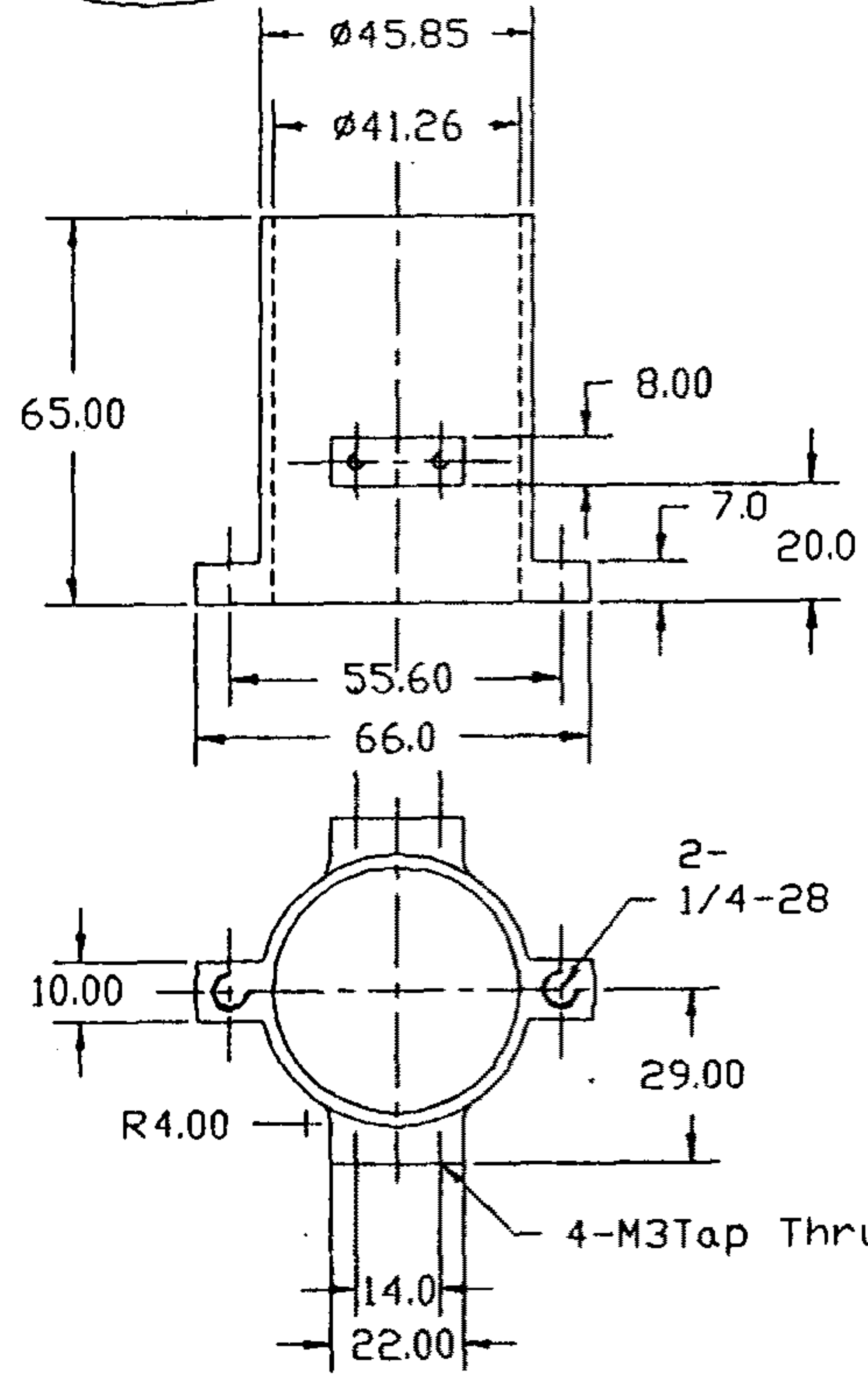
6



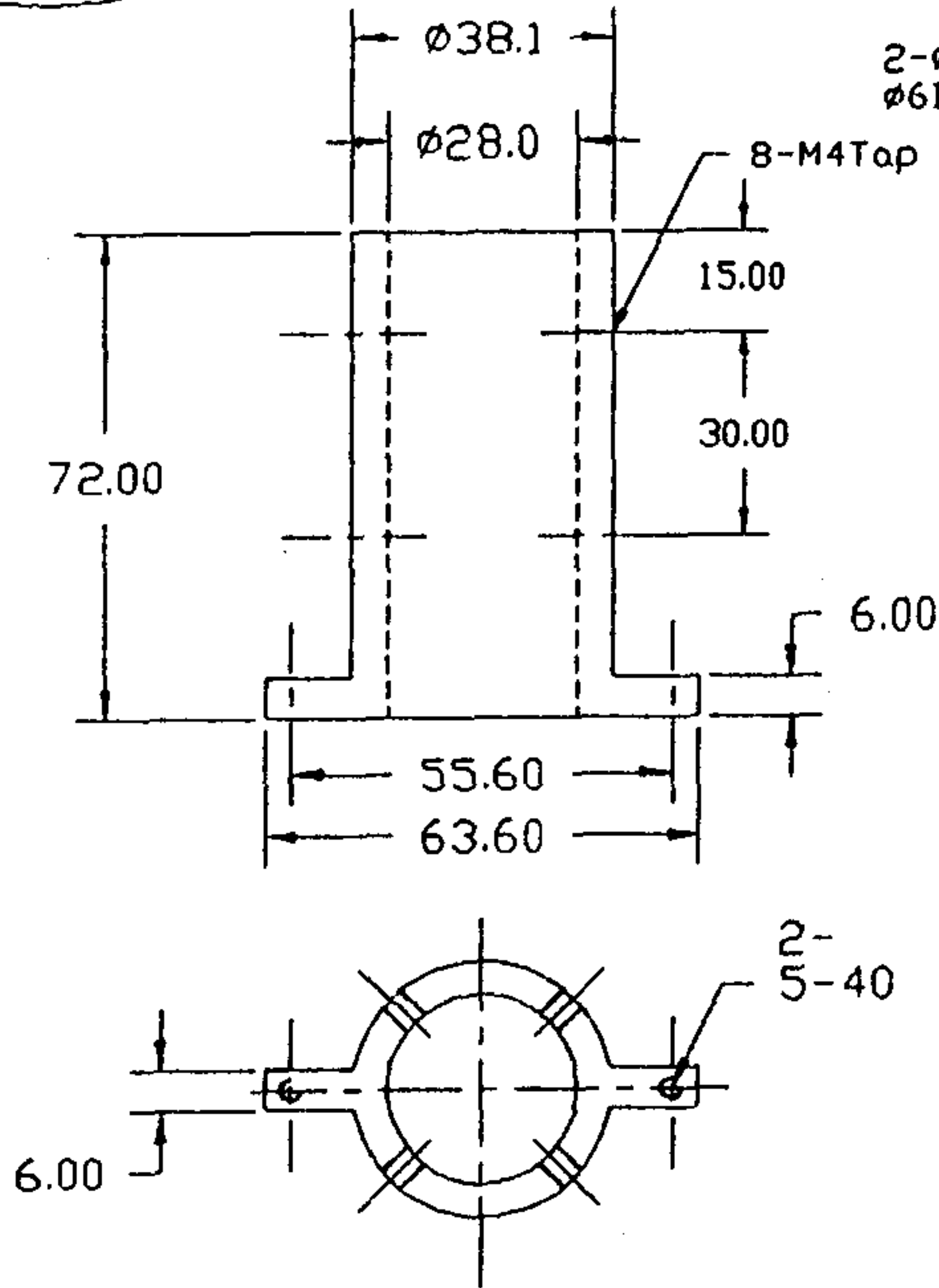
150

POLI-15

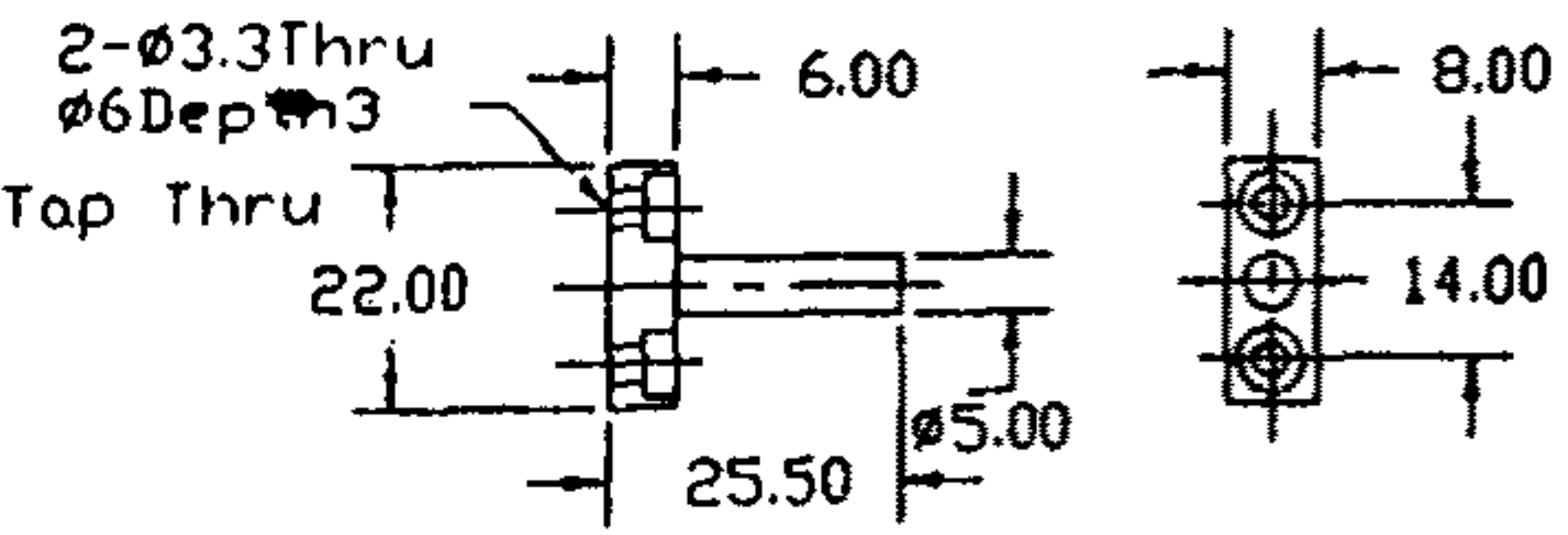
6-21



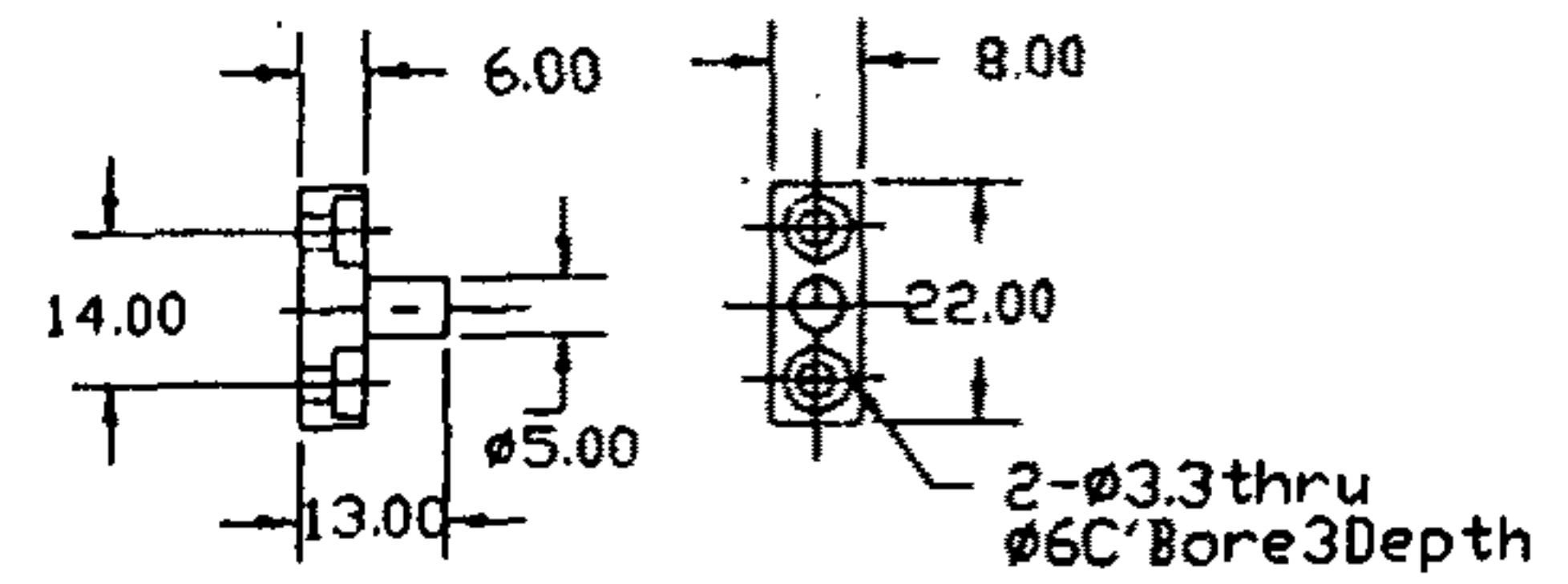
6-14



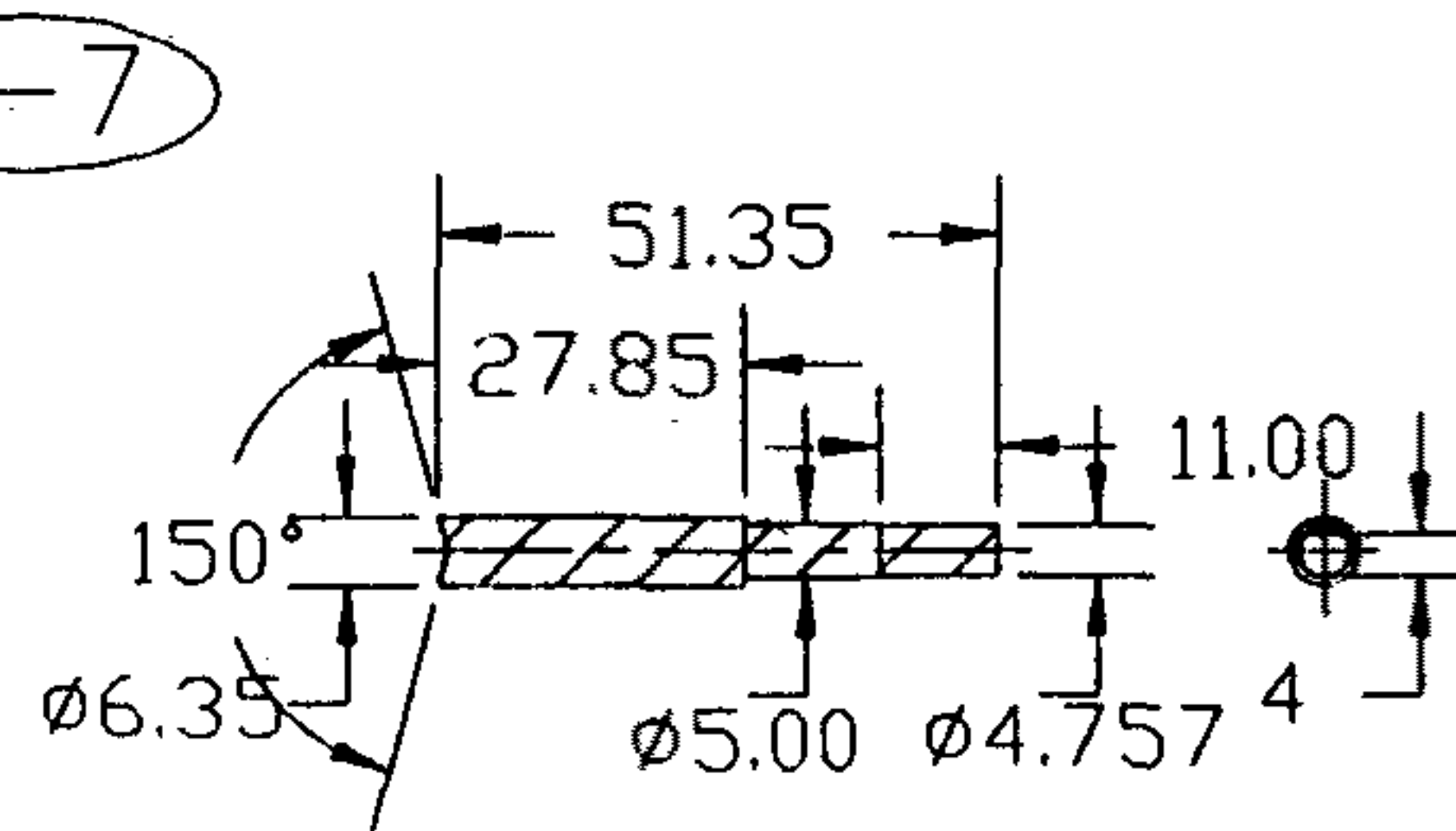
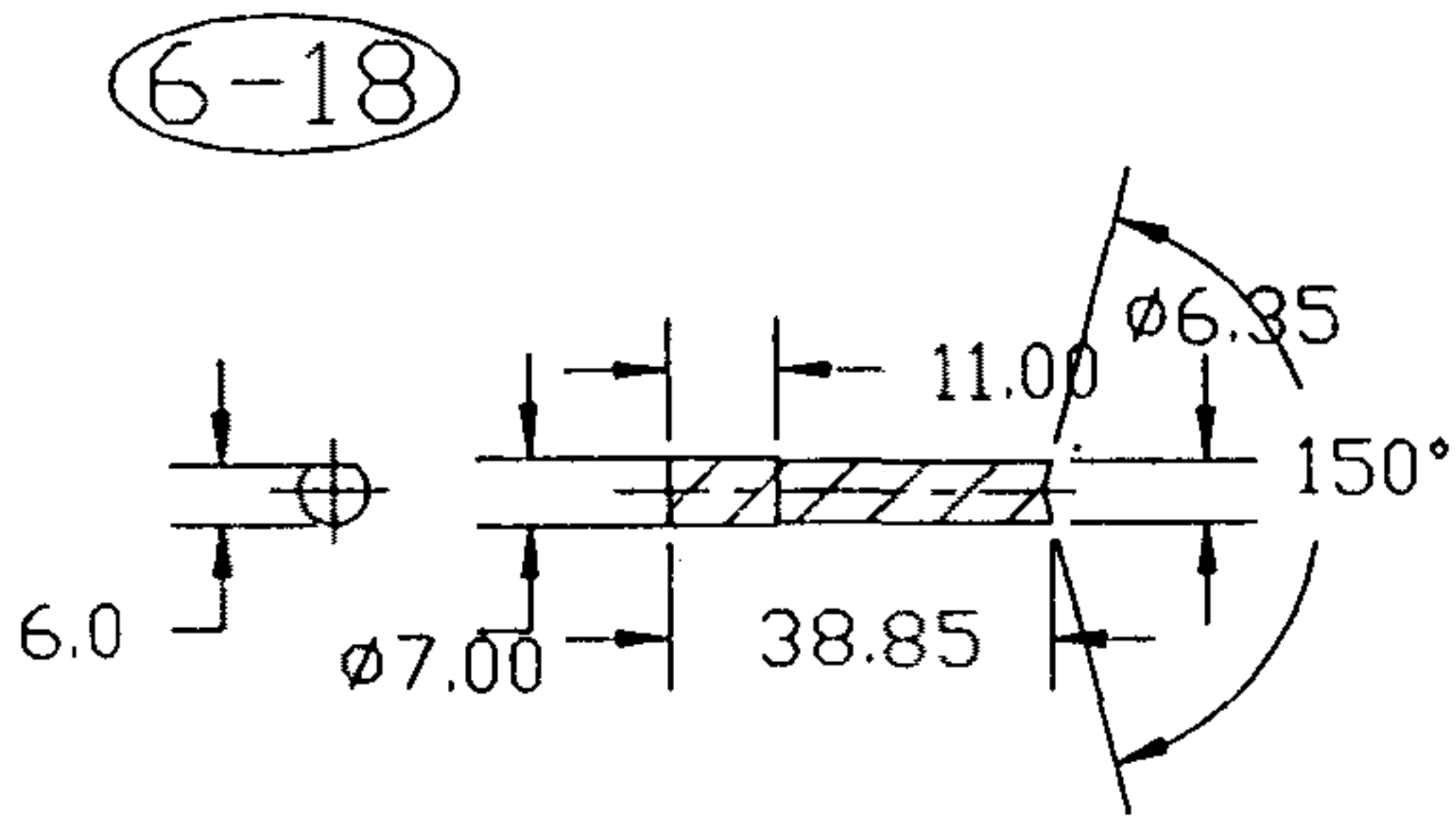
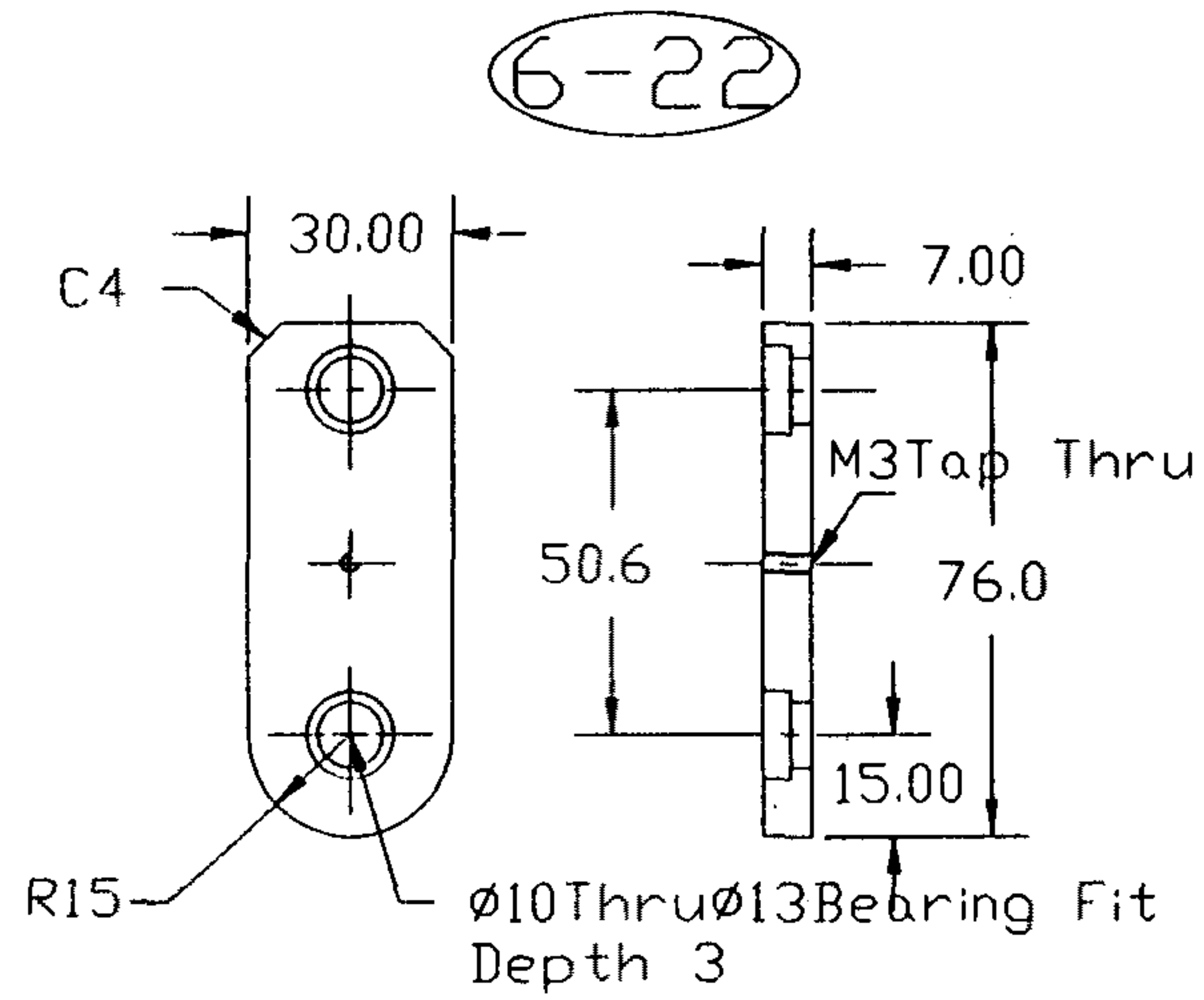
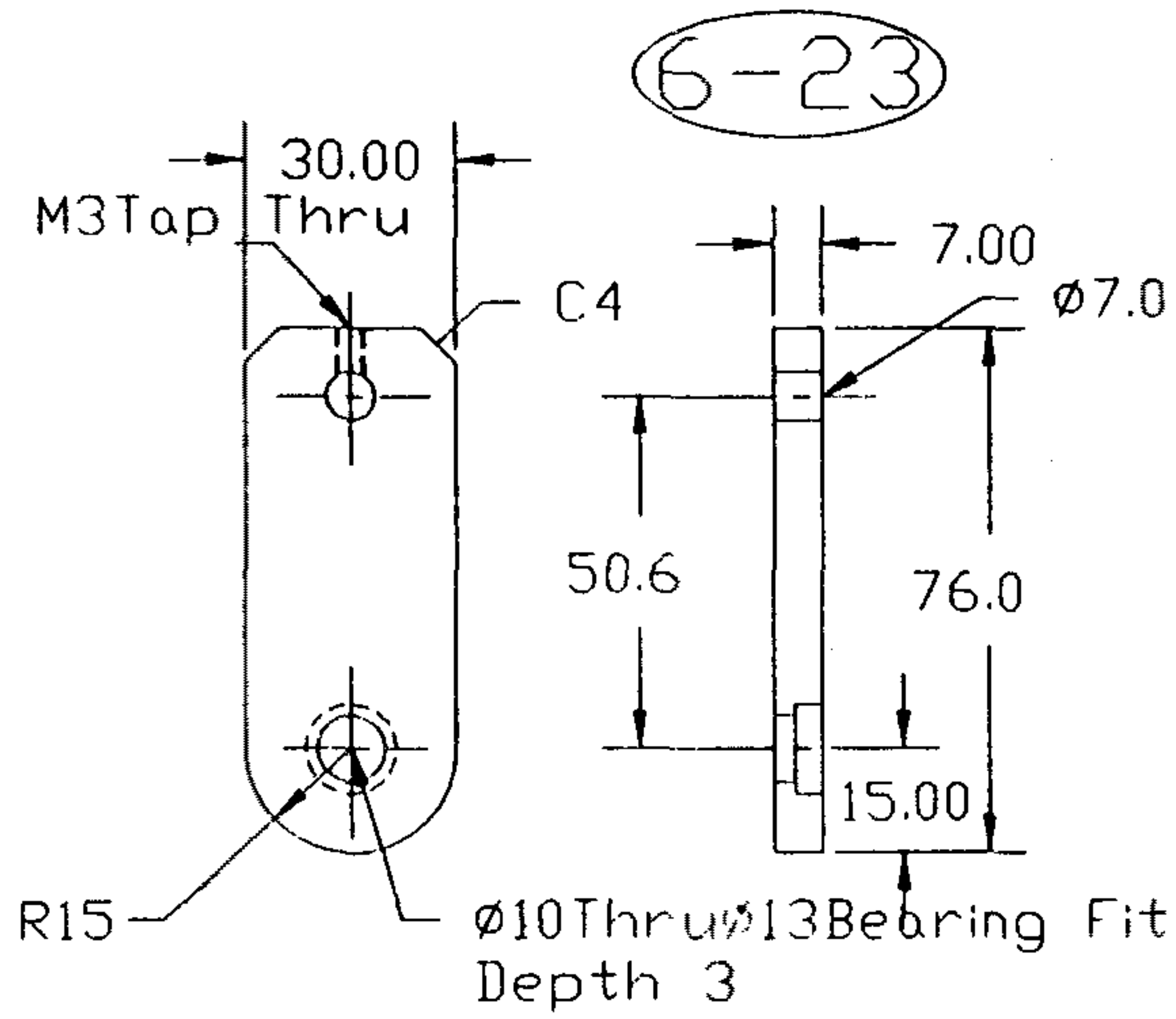
6-13



6-20

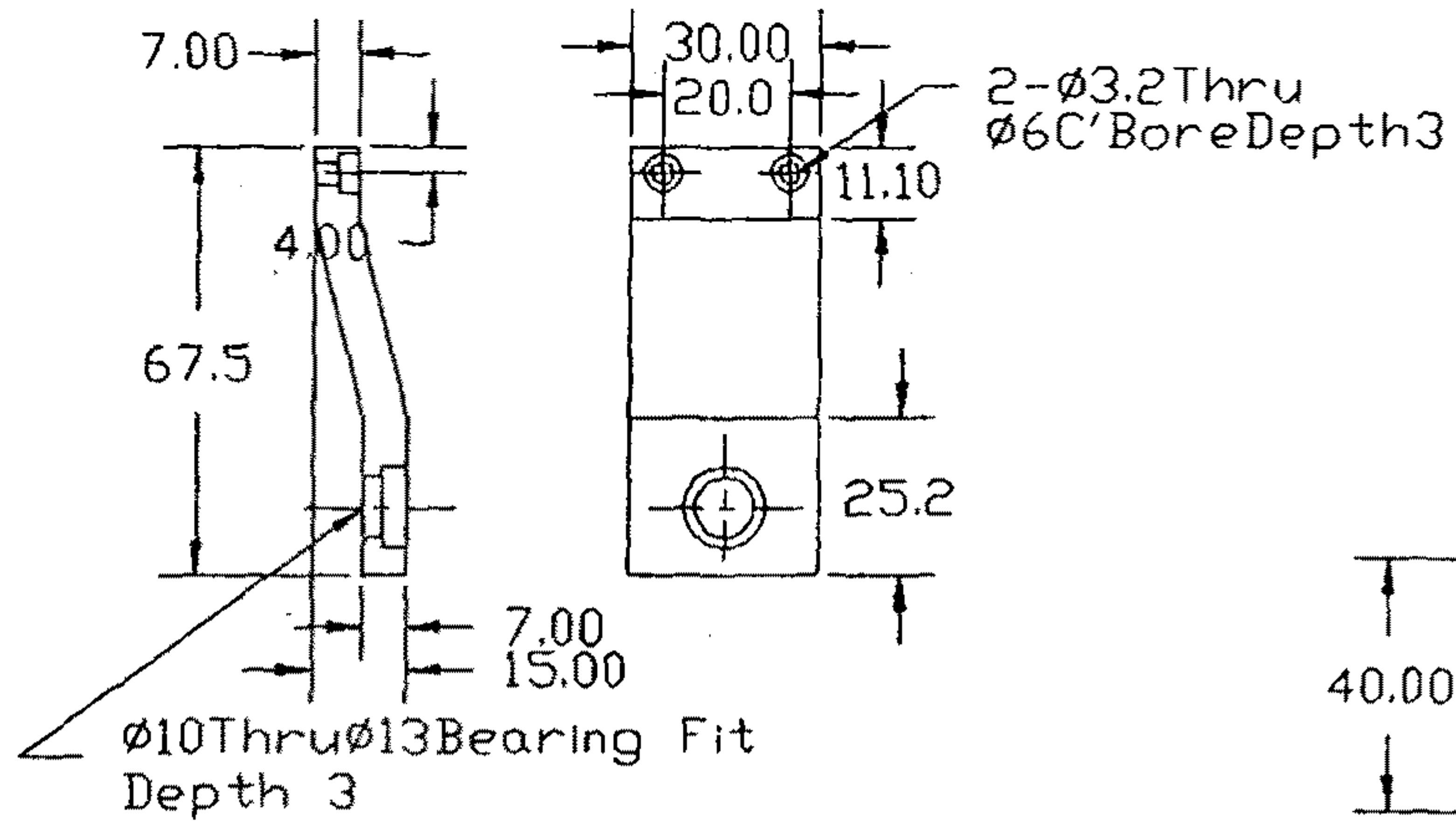


POLI-16

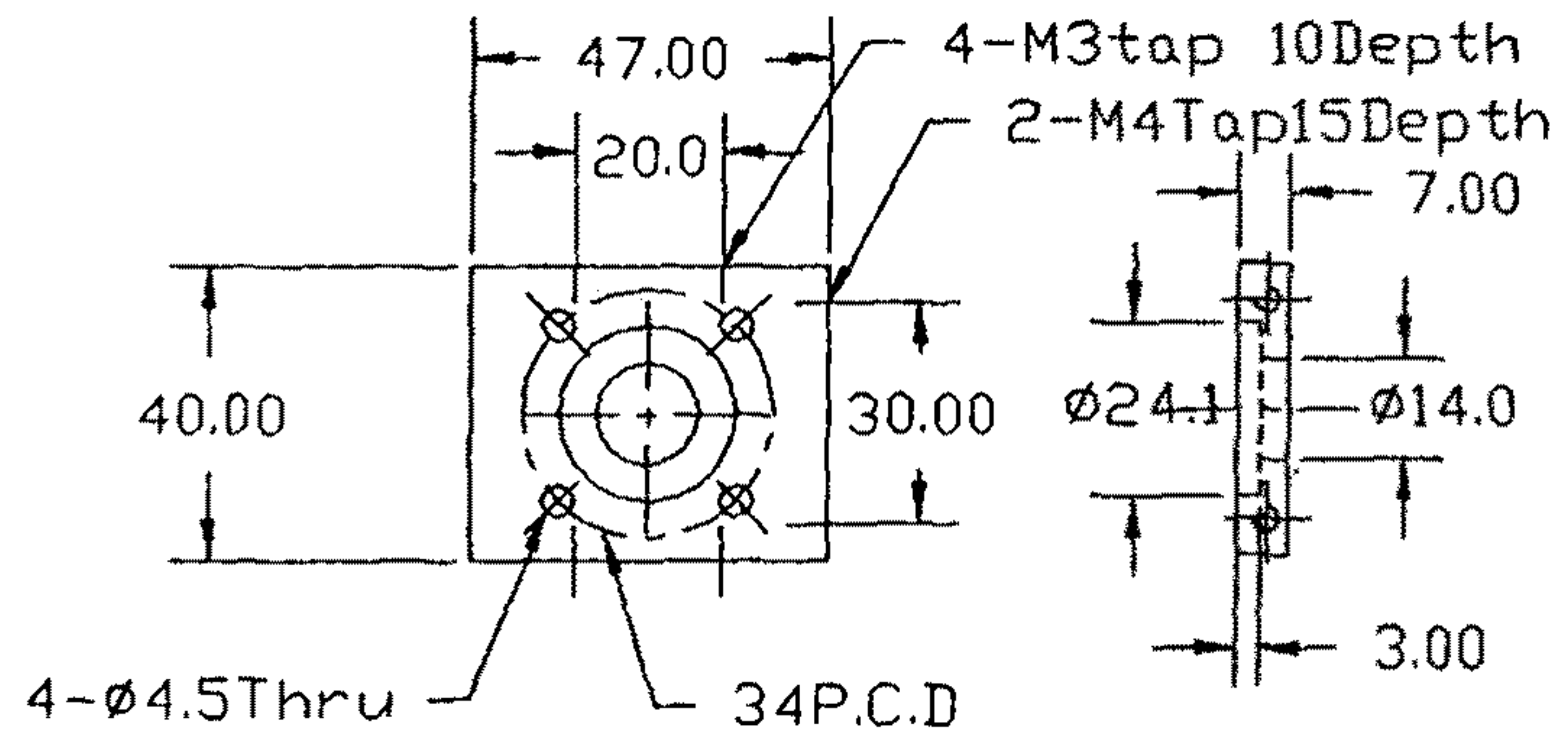


POLI-17

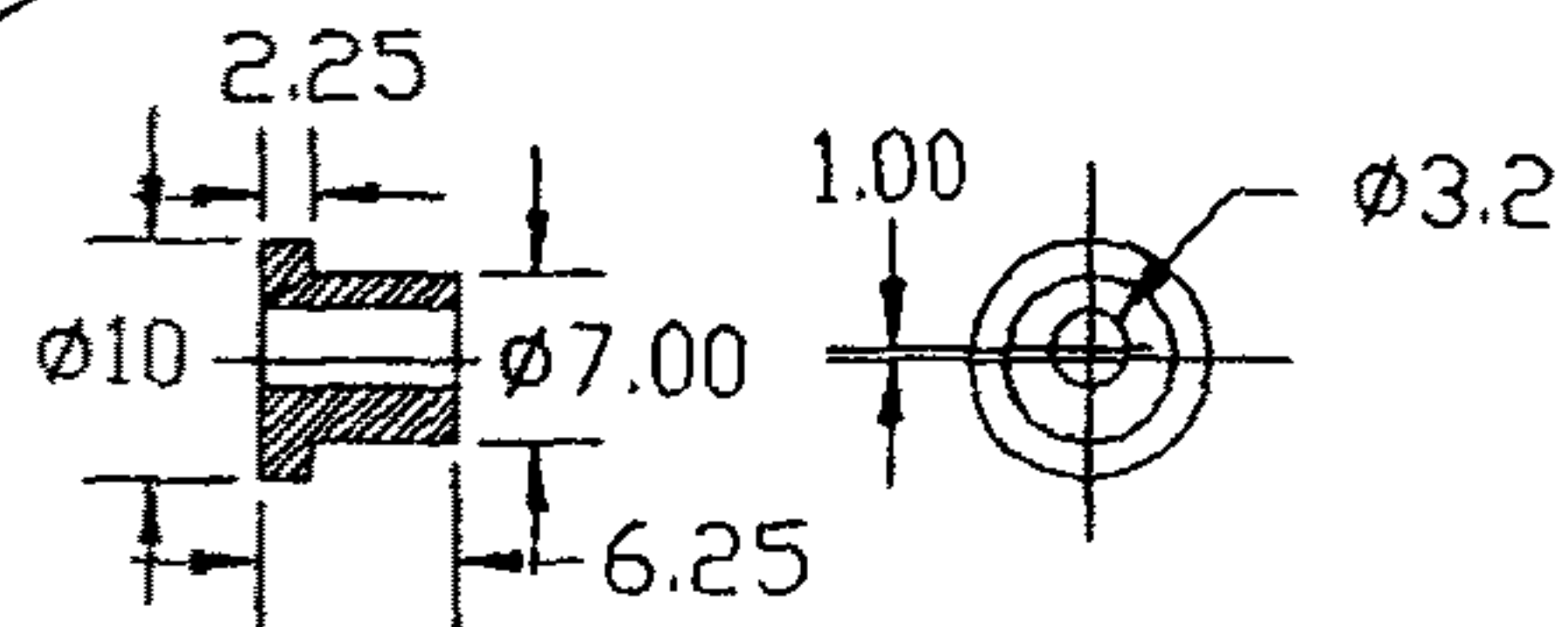
6-2 Material: 2EA



6-1



6-10



POLI-18

부록 3. SEL controller 특성 및 사양

IA MULTI SERVO SEL CONTROLLER 사양

형식	M2 (2 축)	M3 (3 축)	M4 (4 축)
항목	60W x 2	60W x 3	60W x 4
전원 전압	AC100V ± 10%		
전원 주파수	50 / 60 Hz		
전원 용량	약 200W	약 300W	약 400W
사용 온도 및 습도	0-50° C	습도 85% RH 이하	
사용 환경	부식성 가스 없는 것, 먼지 심하지 않은 곳		
절연 저항	500 V 10 MΩ 이상		
노이즈 내구성	노이즈 시뮬레이터에 의한 1000V _{p-p} 1μsec pulse		
중량	약 14kg	약 15kg	약 16kg
보호 기능	전원 전압 체크 PG 접속 체크 과부하 체크	몸체 온도 체크 모터 단선 체크 Soft-limit 체크	
모터 용량	DC 서보 모터 20W - 200W		
기억 용량	약 1500 스텝		
기억 장치	메모리 카드 EEPROM 32KBYTE		
입출력	전용 입력 7점 전용 출력 2점 비용 입력 23점 범용 출력 10점 확장 입출력 (입력 32점 출력 32점) DC 24V 0.5A		
데이터 입력 방식	Teaching BOX 또는 RS232C 통신		
통신 기능	EIA RS232C 비동기 전송		

SERVO 계 사양

항목	사양
제어 방식	로타리 엔코더 A.B.Z상
위치 결정 정도	±2 PULSE
속도 결정	1 mm/sec - 1000m/sec
가감속 설정	0.01G - 1G

Teaching box 사양

항목	사양
사용 온도및 습도	0-40° C 습도 85% RH 이하
사용 환경	부식성 가스 없는것, 먼지 심하지 않으곳
중량	500 g
케이블 길이	2 m
표시	20 x 4 LCD 표시

외부 입출력 사양

입력부

항목	사양
입력 점수	전용 입력 - 7점 범용 입력 - 23점 확장 입력 - 32점(option)
입력 전압	DC24V ± 20%
입력 전류	7 mA/DC24V
ON/OFF 전압	ON-최소 DC16.0V OFF-최대DC5.0V
ON/OFF 응답 시간	ON-MAX20msec OFF-MAX20msec
절연 방식	Photo-coupler

출력부

항목	사양
출력 점수	전용 출력 - 2점 범용 출력 - 10점 확장 출력 - 32점(option)
정격부하전압	DC24V
최대부하전류	40 mA/1점
권장부하전류	20 mA MAX
누출 전류	0.1 mA MAX
잔류 전압	최대 3.1V/40mA
절연 방식	Photo-coupler
출력 단락보호	Pulse 저항 47Ω

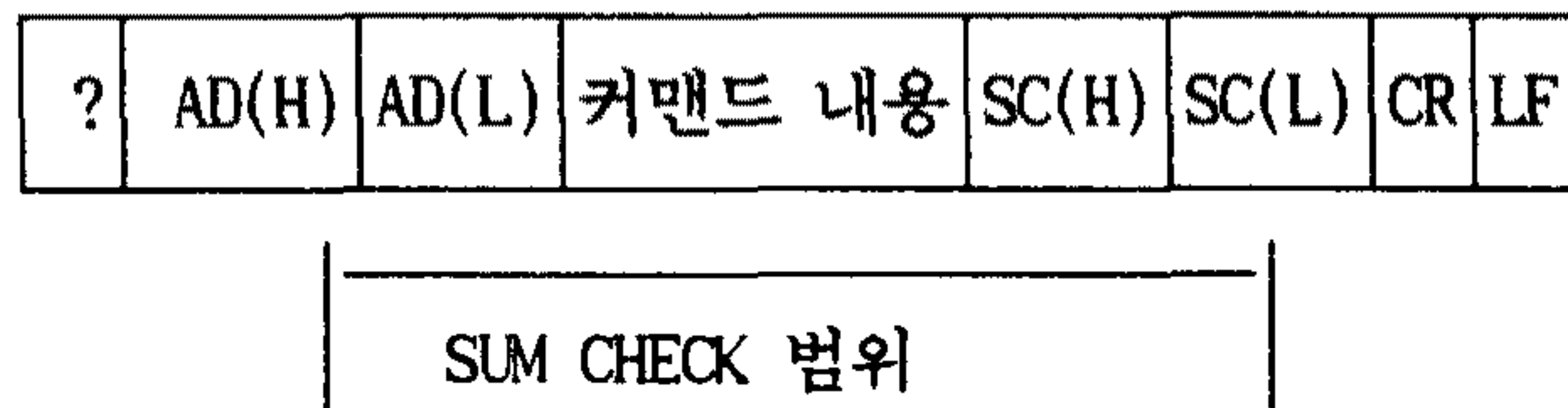
부록 4. PC-SEL controller communication

PC - SEL Serial communication

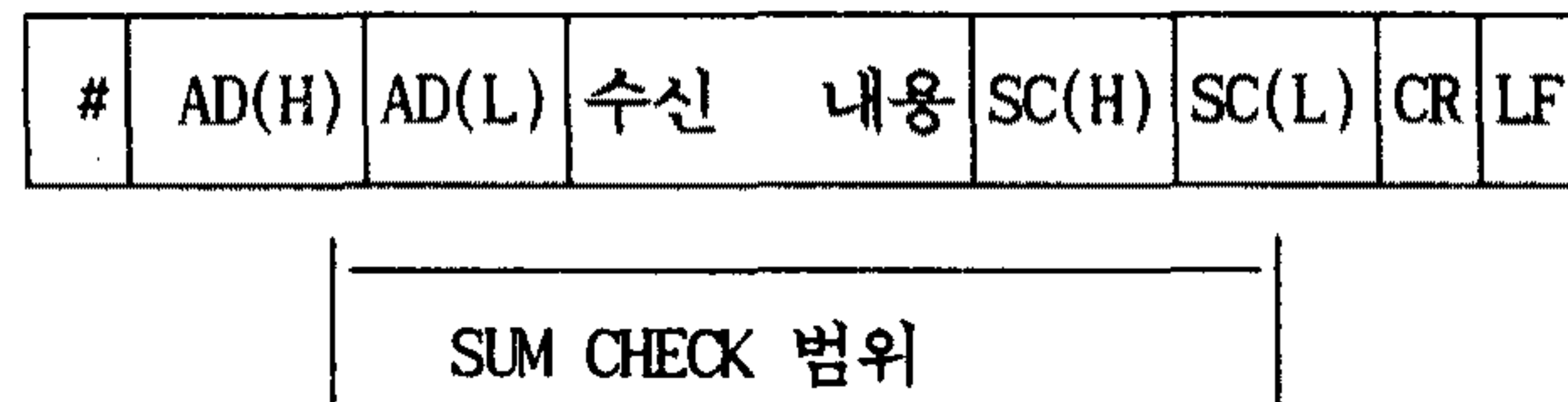
전송 파라미터

Character size : 8 bit
 Parity : none
 Stopbit : 1
 Baud rte : 9600 b/s

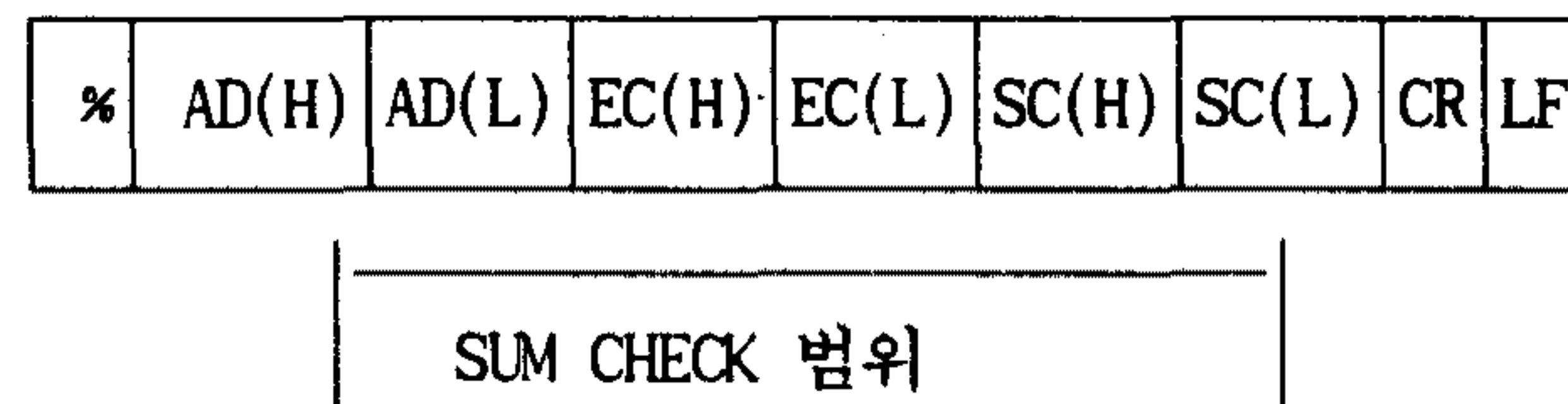
전송 커맨드 규약



수신 커맨드 규약



에러 발생시



부록 4. PC-ABB R/C serial communication

DEFINITIONS

- 2.1
Character A character consists of 8 bits + parity and is always binary coded.
- 2.2
Telegram The series of characters transmitted between robot and superior computer (SC), using the line procedure ADLP 10, which is preceded and succeeded by the control characters DLE/STX and DLE/ETX respectively.
- 2.3
Messages A quantity of information which is logically related and constitutes a unit.
- A message can consist of one or more telegrams.
- Only one message can be included in a telegram.
- The maximum length of a telegram is 128 bytes.
- 2.4
Timeout Timeout is used when the IRB is waiting for a response from SC (5 sec).

COMPUTER LINK HARDWARE

3.1 General

The SC interface must be in agreement with V24,V28/RS232. For distances more than 15 meters a modem is used. The communication speed is 9600 b/s. The character size is 8 bits + even parity and one stopbit.

3.2 Technical description for S2

The computer link for S2 consists of following units:

- DSCA 114 Asynchronous communication board,
5751 0001-AA.
- DSTK 152 Wiring, 2639 0603-K.
- DSTC 120 Connection terminal for DSCA 114,
5752 0001-A.

and as supplements:

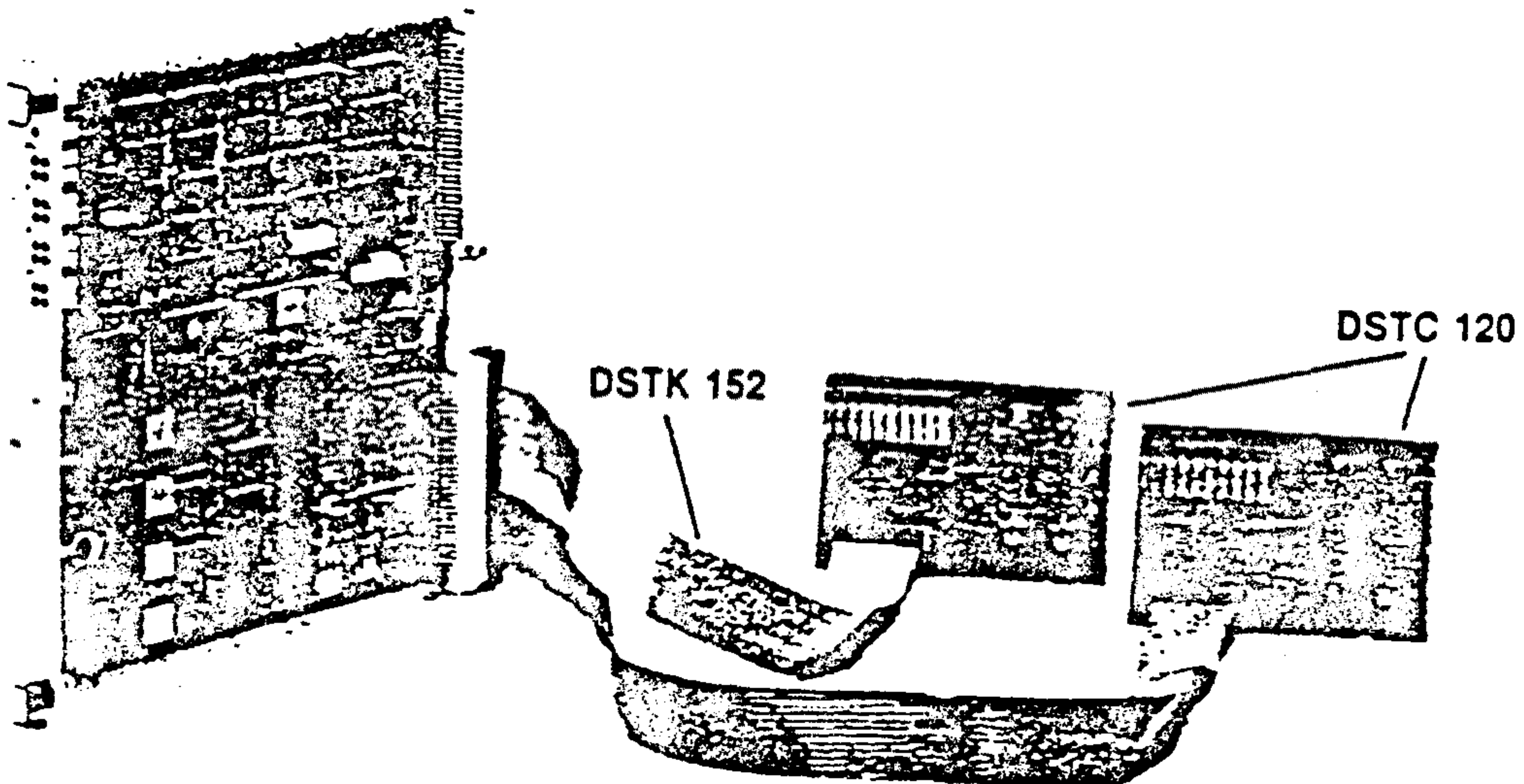
- _ New control program
- (Terminal unit 2166 0559-H)

The circuit board DSCA 114 is to be located in the upper rack at place D14.161. The connection terminal is to be located at the corresponding place in the rear wall of the terminal unit. Wiring DSTK 152 is installed between these. The user connects his cable through an opening in the side of the cubicle and up to the connection terminal, contact X92, as shown on circuit diagram 6704 100-B* A.

NOTE! In order to avoid electrical disturbances the computer link cable should be routed as far as possible from the process wiring.

Communication board DSCA 114

The unit is connected to the parallel bus system and can be used to communicate with units which have asynchronous serial interfaces. Channel 0 is intended for printout, channel 1 for computer link, channel 2 and 3 are not used.



Number of channels	4
Signal interface, outputs	CCITT V 24 and short range modem
Transmission speed	300-9600 bit/s
Word length	5, 6, 7 or 8 bits
Parity	odd, even or no parity
Stop bits	1, 1.5 or 2
Block buffer	128 + 96 bytes/channel
Supply requirement, typ +5V	1.8 A
Isolation test voltage, 50 Hz, 1 min	2.5 kV
Physical data, plug-in board intended for installation in a rack,	1 board place (4 TE)
width	400 g
weight	via DSTC 120
Connection	with ribbon cable
	DSTK 152

부록 5. 모터 사양

TYPE RH-11-6001

- Power : 60 W
- RPM : 100 RPM
- Voltage : 24 V
- Current : 1.3 A
- Torque : 22 Kgf · cm
- Encoder : 200 P/R
A, B, Z 상
- 사용축 : 1, 3 축

(SANYO DENKI CO., LTD.)

TYPE R406-011E

- Power : 60 W
- RPM : 3000 RPM
- Voltage : 24 V
- Current : 4.1 A
- Torque : 22 Kgf · cm
- Encoder : 200 P/R
A, B, Z 상
- 사용축 : 2 축

(SANYO DENKI CO., LTD.)

부록 6. SEL controller parameter

Axis Name	X	Y	Z
Maximum RPM	3000	3000	3000
Encoder (pulse/rev)	194	200	194
Load Screw (mm/rev)	7	4	7
Encoder Multiply	4	4	4
Soft limit , CW(.01 mm)	60000	60000	60000
Brake(1=ON) (mm/s)	0	0	0
Home Direction (0 = CCW)	1	1	1
Home Type (1 = LS)	1	1	1
Axis Home Sequence (1-4)	1	1	1
Home Return Vel (mm/s)	5	5	5
Z Pulse Search Vel (mm/s)	10	10	10
Home Vel 2(not used)	10	10	10
Home Deviation (pulse)	2000	2000	2000
Home Return Offset	0	0	0
Home Current Limit (A)	7	7	7
Position End Band (pulse)	20	20	20
Soft Limit Offset	100	100	100
Position Gain	60	60	60
Speed Gain	80	80	80
Feed/Forward Gain	0	0	0
Integral Gain	30	30	30
Total Gain	15	15	15
Integral Current Limit	50	50	50
Error Range (pulse)	4000	4000	4000
A/D Temp	32768	32768	32768
A/D Motor Overload	23808	23808	23808
Power volt	8704	8704	8704

부록 7. SDK 개발 과정

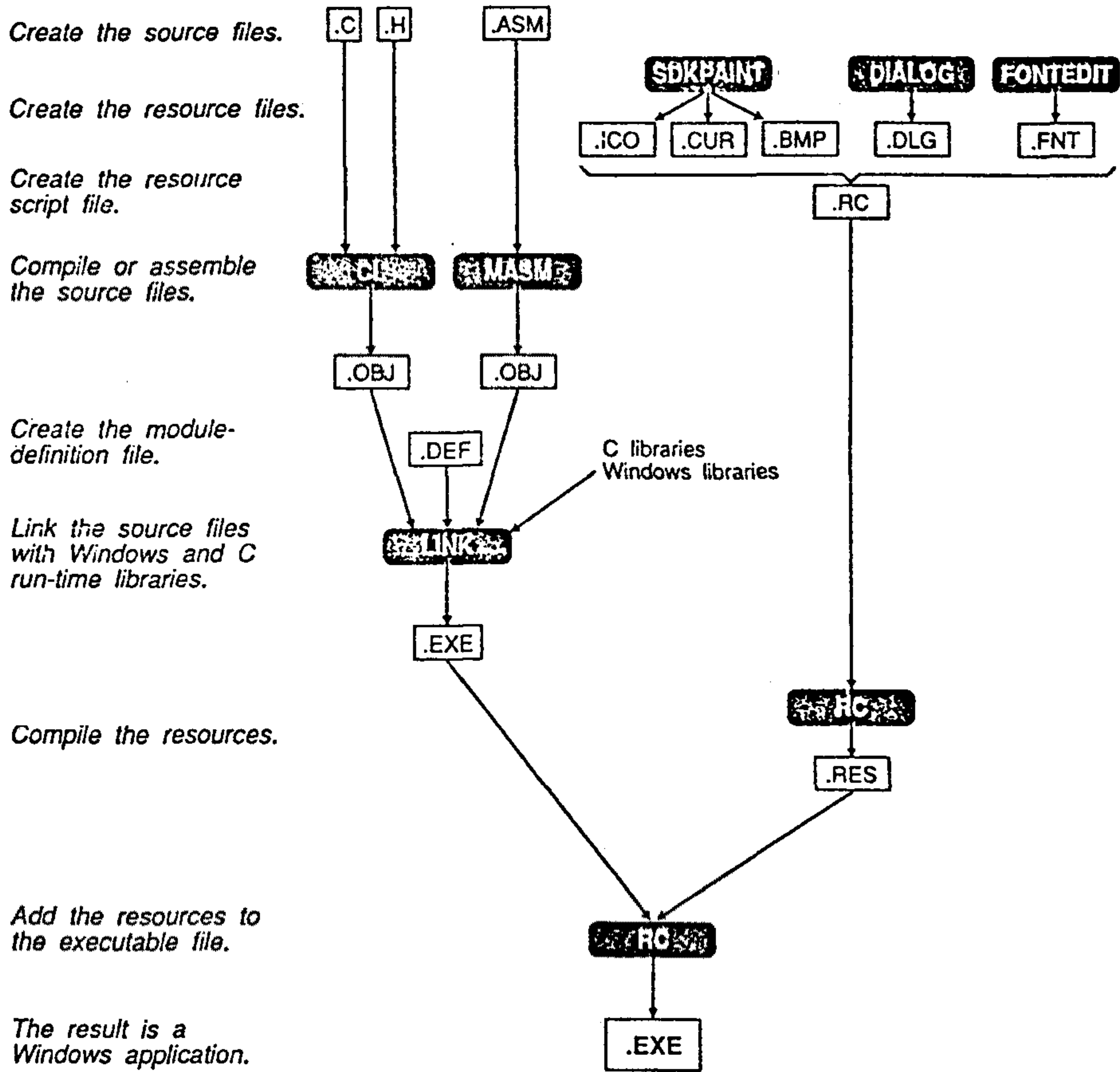
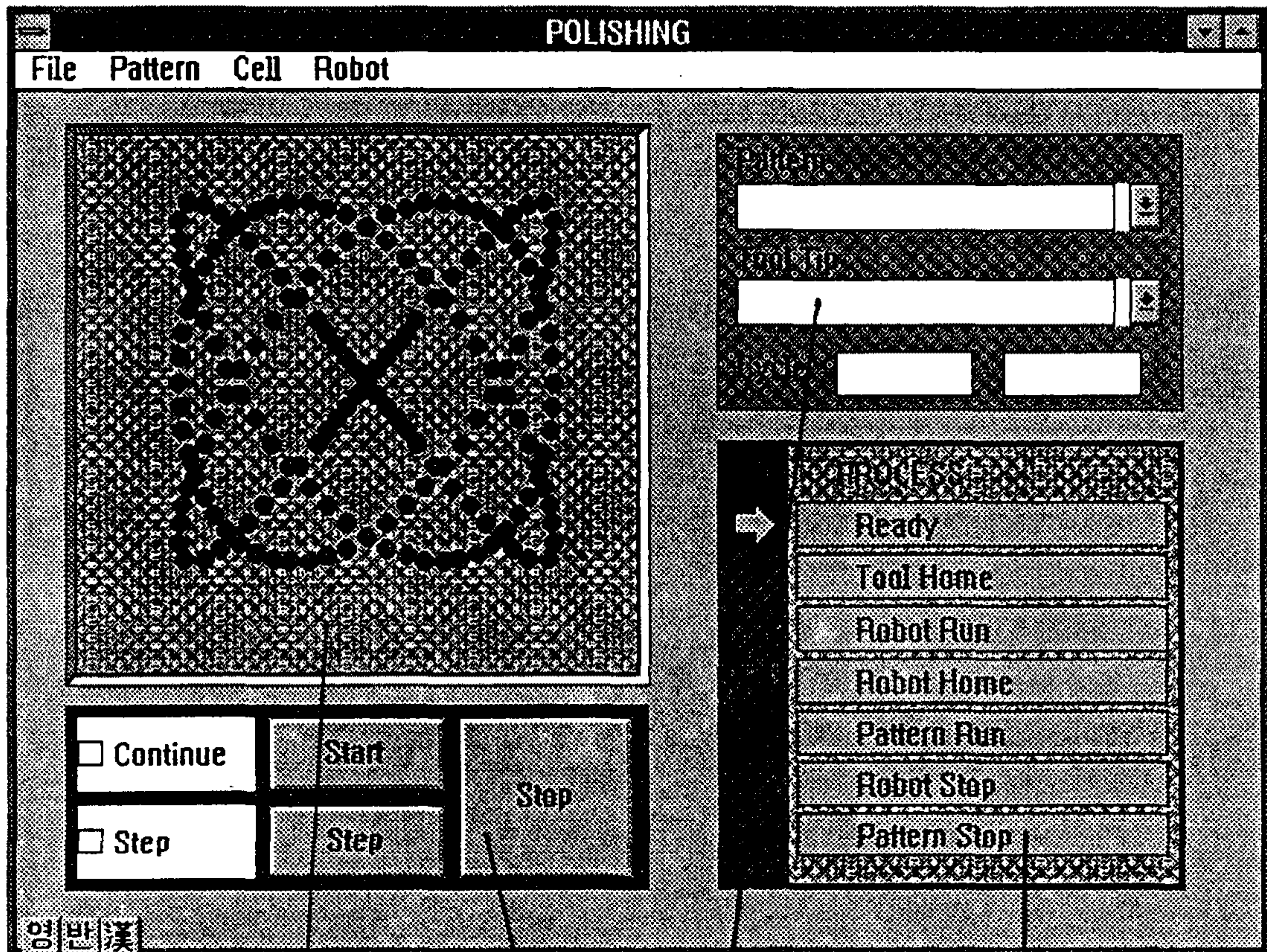


Figure 1.5 Building a Windows Application L02_05

부록 8. 시스템 관리 소프트웨어



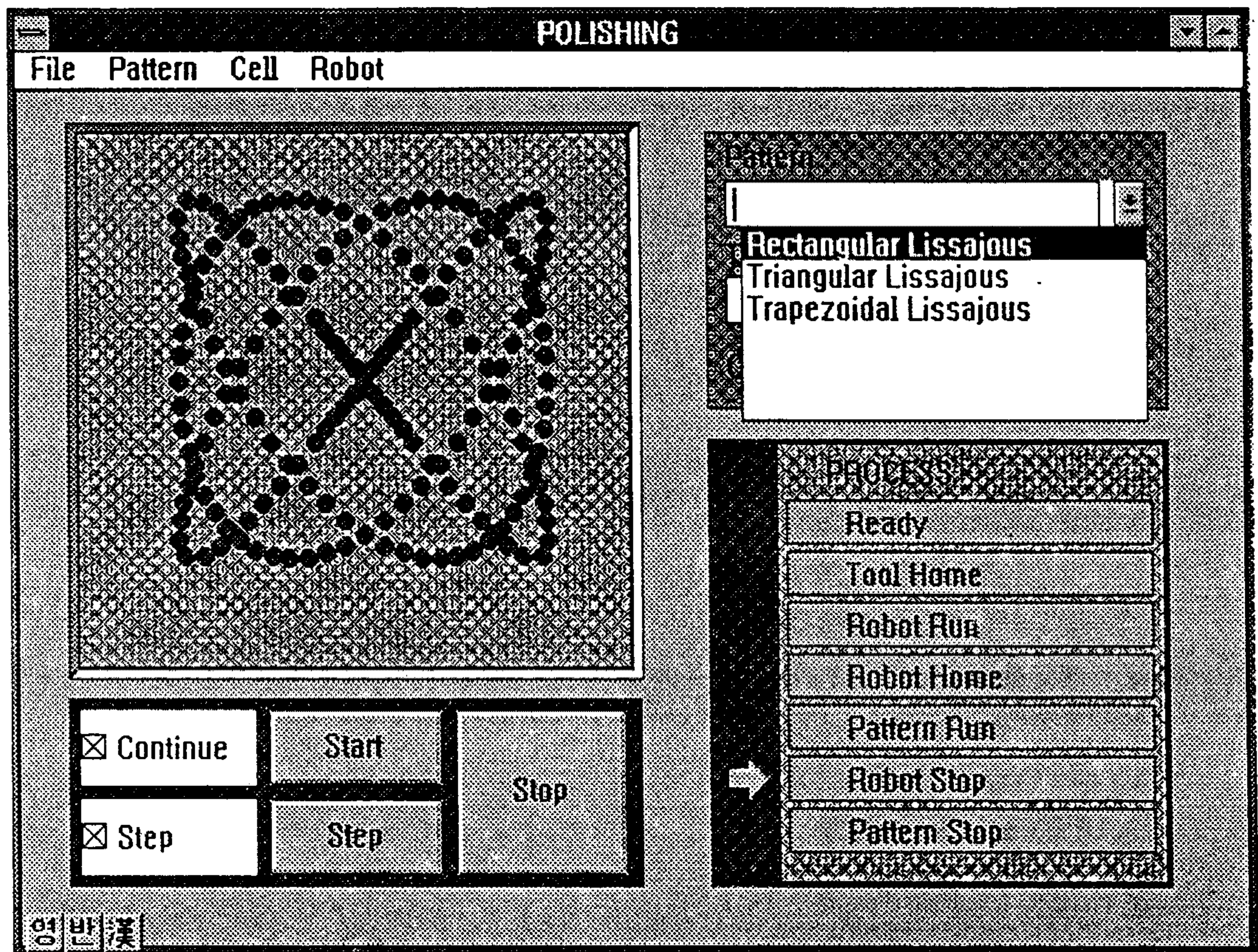
패턴 및 ABB robot
모니터링 윈도우

작업 제어판

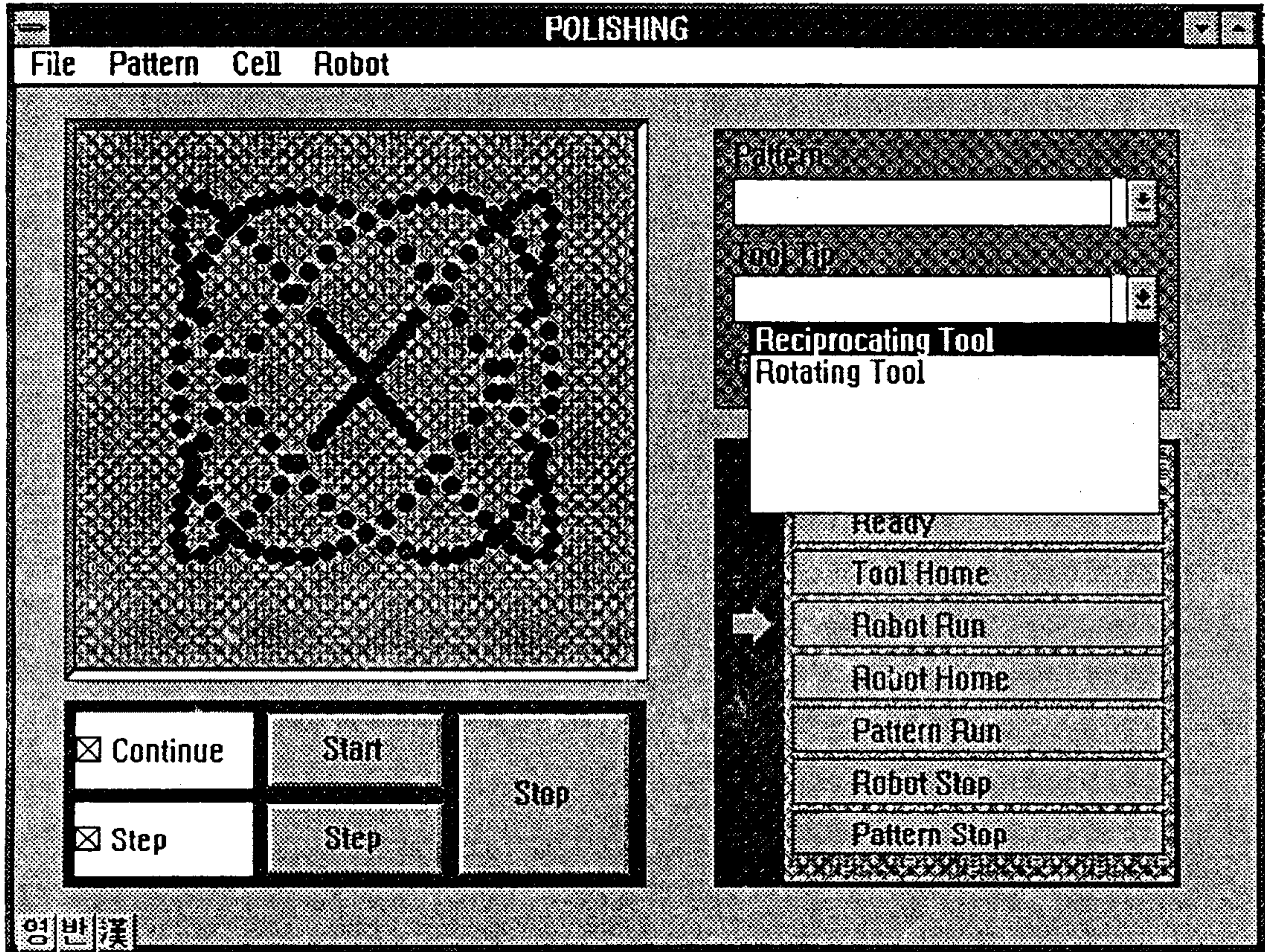
패턴 및 툴팁 선택

Process
모니터링

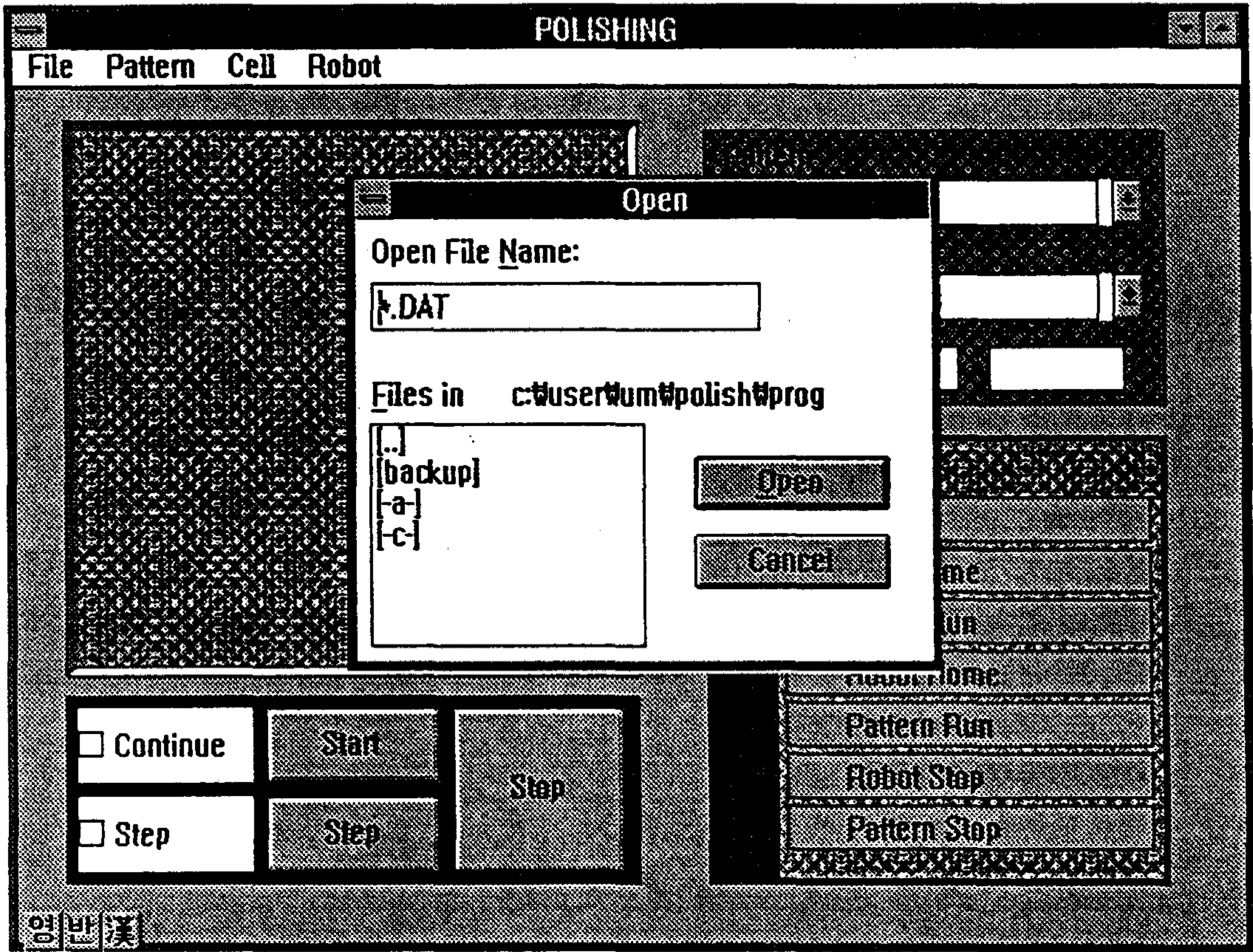
초기 화면



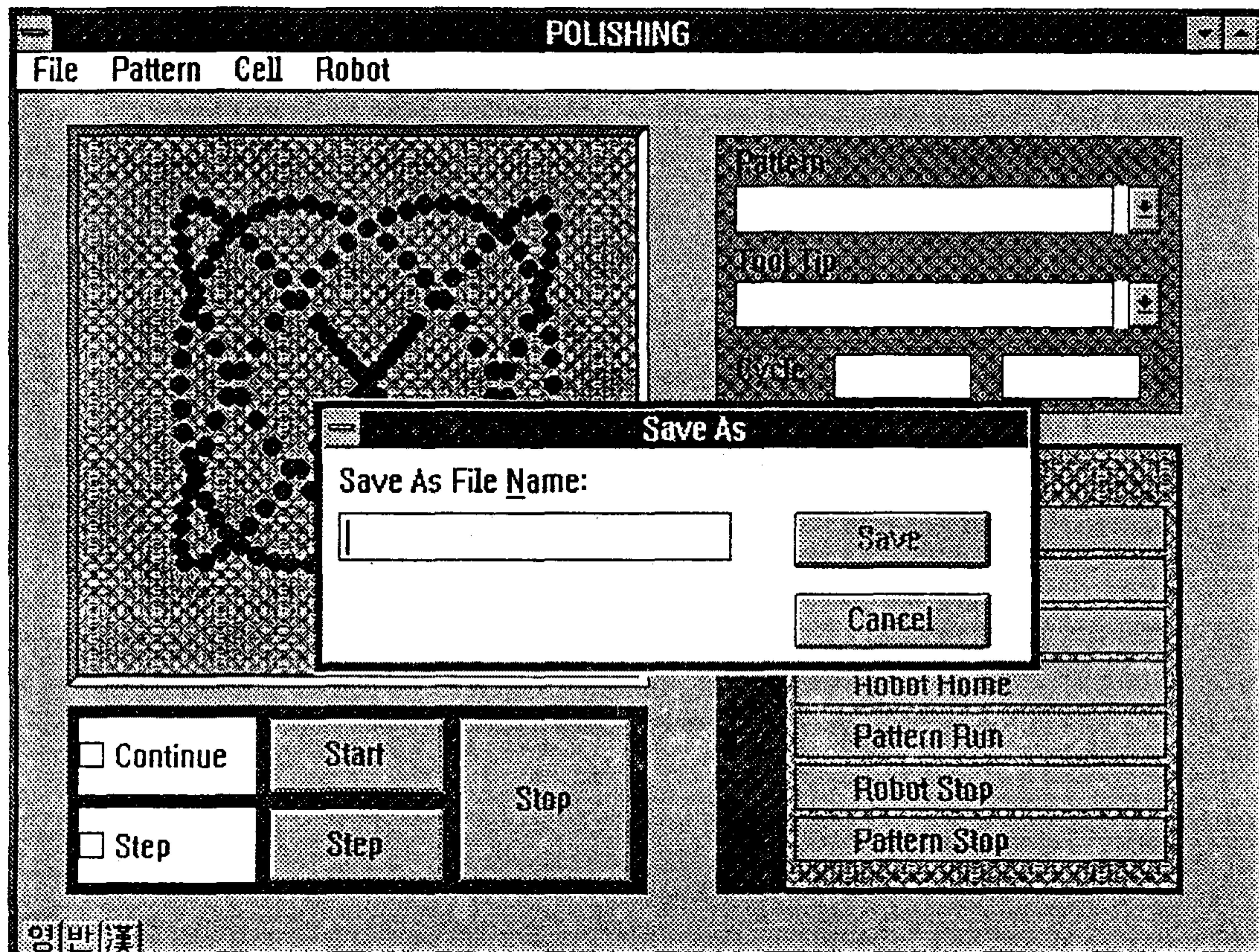
연마 패턴 선택



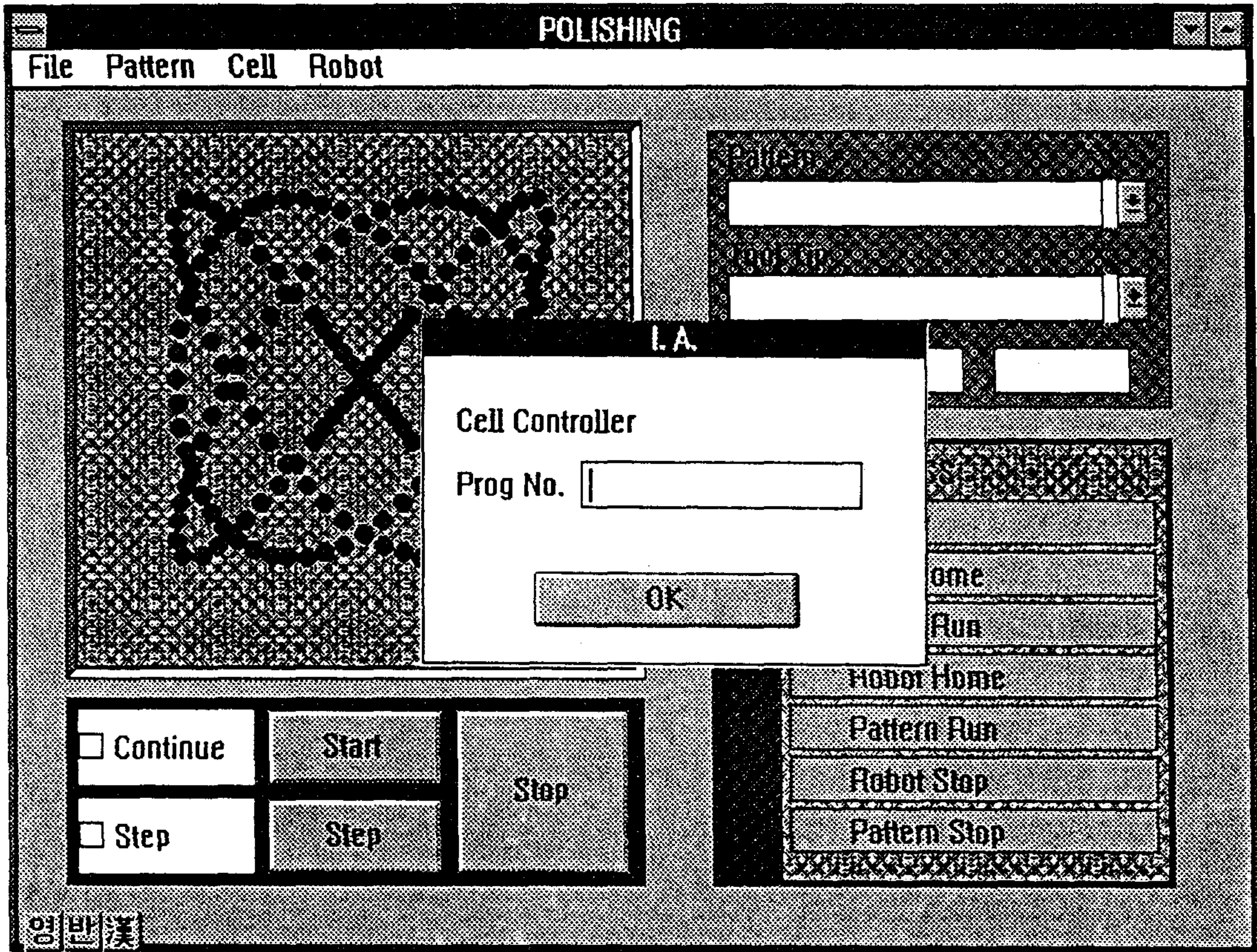
Tool Tip 선택



패턴 데이터 및 연마 로봇 프로그램 열기



패턴 데이터 및 연마 로봇 프로그램 저장



연마 로봇 프로그램 구동

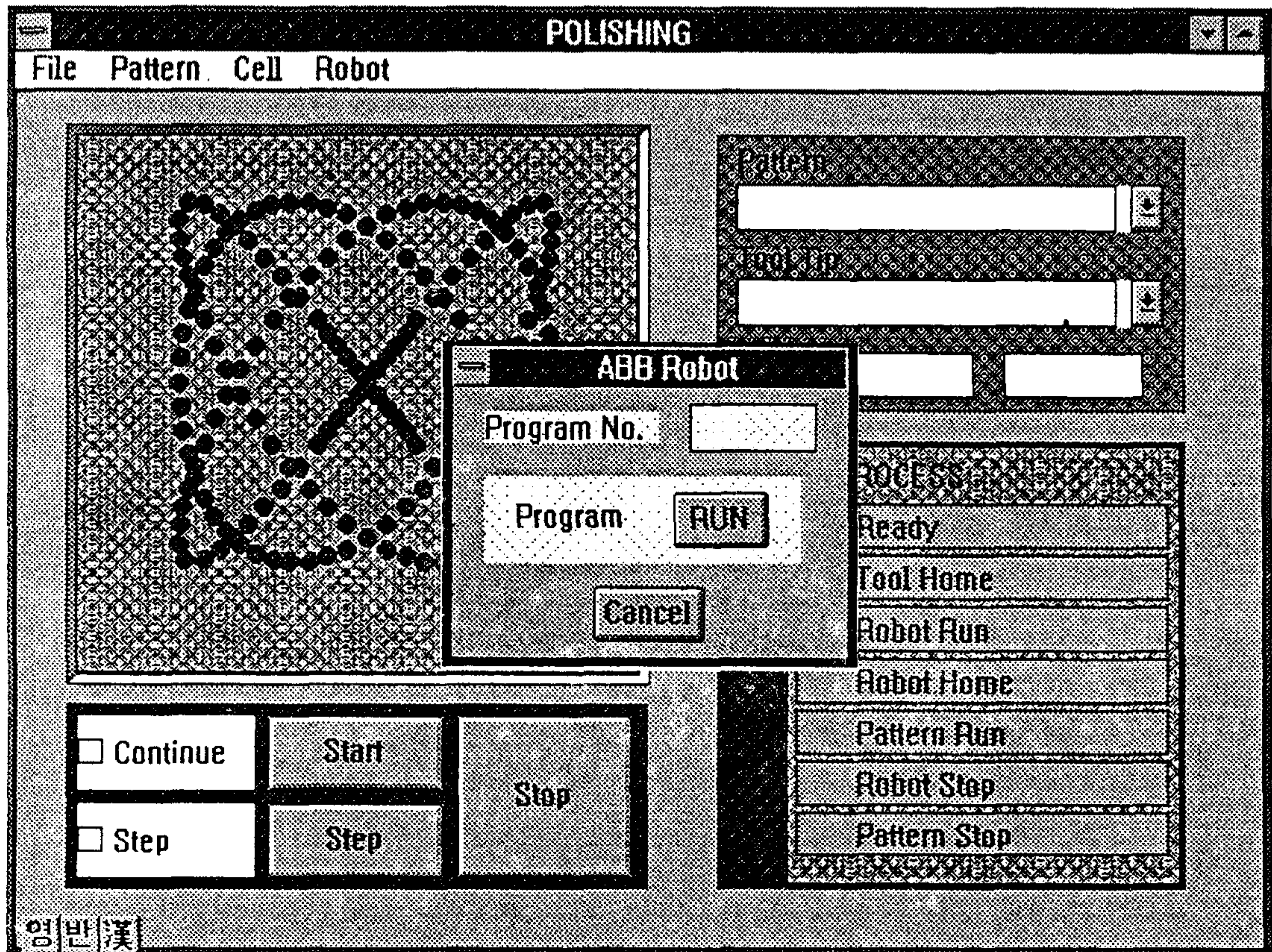


ABB robot 프로그램 구동