GOVP1198901309

'88국책연구개발사업과제
"CIM 기술에 의한
사출금형공장 자동화"
사업의 세부과제

17
110

금형의 연마작업을 위한 로보트 Off-line Programming System 개발

Development of Robot Off-line Programming System for Die Polishing

연구기관
한국과학기술연구원

寄贈
과학기술처
89年 11月 10日

과 학 기 술 처

제 출 문

과학기술처장관 귀하

본 보고서를 "CIM 기술에 의한 사출금형공장 자동화" 사업의 세부과제 "금형의 연마작업을 위한 로보트 off-line programming system의 개발" 과제의 최종보고서로 제출합니다.

1989. 7. 24.

주관연구기관명 : 한국과학기술연구원

총괄연구책임자 : 강 무 진, 김 상 국

연 구 책 임 자 : 김 문 상

연 구 원 : 박 강

여 백

요 약 문

Ⅰ. 제목 :

금형의 연마작업을 위한 로보트 off-line programming System의 개발.

Ⅱ. 연구개발의 목적 및 중요성

금형의 생산에 있어서 마무리 작업인 polishing 작업은 전체 공정시간의 반이상을 차지하는 수작업으로 이루어지고 있는 단순반복공정이다.

이것을 Robot를 이용하여 자동화함으로써 생산경비 및 시간을 절약하게 되어 생산성 향상에 이바지하게 된다. 3차원 곡면을 따라 행하여지는 로보트용 polishing 작업 program을 생성하기 위하여 CAD-system으로부터의 직접적인 연결을 통한 off-line 방식은 로보트 down-time의 극소화 및 사전 Test등을 통한 error 방지등의 효과를 기대할 수 있다.

Ⅲ. 연구개발의 내용 및 범위

가. CAD system과 Off-line programming system과의 접속 및 Robot Kinematic들의 graphic library화

나. Surface의 배치 수정 및 path generation 및 Animation을 통한 가공작업의 simulation.

다. 가공작업 과정을 menu화하여 expert system을 이용한 interactive한 작업화

라. Robot과 PC, Robot과 VAX의 interface

마. Collision Test

Ⅳ. 연구개발 결과 및 활용에 대한 건의

CAD system과 off-line program과의 접속을 하였으며 Robot Kinematic들을 graphic library화하였다. graphic을 이용하여 가공작업을 simulation할 수 있으며 menu를 이용하여 여러작업을 손쉽게 할 수 있게 하였다. 또 Robot과 PC, Robot과 VAX간의 interface를 하였다.

앞으로 이 프로그램을 더 보완 수정하면 금형의 가공공정을 자동화하여 생산성 향상과 경비절감에 많은 기여를 할 것이다.

ABSTRACT

Polishing is a bottle-neck process of die and mold production, because it is time consuming and labor intensive. An attempt for automating the die polishing process using industrial robot is made in KIST CAD/CAM laboratory. As a part of the automation, an offline programming system for the polishing of sculptured surfaces is developed.

The software consists of such functions as robot motion data (RM data) generation, RM data verification, post processing of RM data, optimum layout planning of robot cell, and calibration of robot system.

Once the surfaces to be polished and the robot kinematics are seleted on the graphics terminal, the program determines the optimum relative position between the robot and the surfaces to be polished. To do this, the four standard views of the robot system with the working space of the robot are displayed, and the position of the surface is determined by moving the surface interactively. In order to exactly determine the relationship between the surface coordinate system and the robot coordinate system several reference points on the surface are measured by means of moving the robot kinematics to these points

The RM data both for the iso-parametric path and cartesian

path are generated automatically. To verify the RM data, the program solves the inverse kinematics of the robot system and animates the motion of the robot. The system checks the limits of the robot motion and the collision of the robot body with the surface.

A post-processor program is used to convert the RM data to the robot specific movement commands. The robot specific parameters like the velocity and the mode of the robot motion (for example point to point or continuous path control) and the polishing oriented parameters are decided interactively using menu. In order to guarantee the absolute positioning accuracy of the robot system the robot kinematics should be calibrated. A in KIST CAD/CAM lab internally developed three dimensional position measuring and calibration system is used to identify the kinematical parameters.

The offline programming system is implemented as a module of the surface modeler developed by KIST CAD/CAM laboratory. The surface modeler is a menudriven software which can generate sculptured surfaces in various methods: i.e. ruled surfaces, surfaces of revolution, curve net, tensor product surfaces, and blending surfaces, etc. Special functions such as generating surfaces from measured point data, surface smoothness test surface matching for measuring data evaluation are the outstand-

ing features of the modeler. Micro VAX II computer and Megatek workstation are used to develop and excute the program and an IBM PC is used for the post processing and interface with the robot controller.

여 백

CONTENTS

Chapter 1. Introduction 13

Chapter 2. PROPS system 15

Section 1. PROPS System Configuration 15

1. Introduction 15

2. Micro VAX 18

3. PS/2 19

Section 2. Construction of PROPS System 19

1. Surface Modeler 19

2. Kinematic 20

(1) Denavit-Hartenberg Transformation 21

(2) FANUC 24

(3) HYUNDAI 30

(4) MEASUREMENT 35

3. MENU 39

(1) SEL - ROBO 39

(2) SEL - SURF 39

(3) MOV - SURF 39

(4) GEN - PATH 40

(5) SIMULATN 41

(6) OPIION 41

(7) TEST 42

1) CP 42

2) CPC 43

3) PTP 43

4. Result 44

Section 3. Interface 53

1. VAX 750 to FANUC Robot 53

2. IBM PC to FANUC Robot using RS 232C 56

3. PC to FANUC Robot 59

4. Sensor to Robot 61

Chapter 3. Conclusion 64

부산대학교
도서관 장서

목 차

제 1 장 서 론 ····· 13

제 2 장 본 론 ····· 15

제 1 절 PROPS System Configuration ····· 15

1. 개 요 ····· 15

2. Micro VAX ····· 18

3. PS/2 ····· 19

제 2 절 PROPS System의 구성 ····· 19

1. Surface Modeler ····· 19

2. Kinematic ····· 20

(1) Denavit-Hartenberg Transformation ····· 21

(2) FANUC ····· 24

(3) HYUNDAI ····· 30

(4) MEASUREMENT ····· 35

3. MENU ····· 39

(1) SEL-ROBO ····· 39

(2) SEL-SURF ····· 39

(3) MOV-SURF ····· 39

(4) GEN-PATH ····· 40

(5) SIMULATN ····· 41

(6) OPIION ····· 41

(7) TEST ····· 42

1) CP ····· 42

2) CPC ····· 43

3) PTP ····· 43

4. 결 과 ····· 44

제 3 절 Computer 와 Robot 의 Interface ····· 53

1. VAX 750과 FANUC ROBOT 의 Communication ····· 53

2. RS232C를 이용한 FANUC ROBOT 과 IBM-PC 간의 Interface ····· 56

3. Robot 과 PC와의 Communication ····· 59

4. Sensor 와 Robot 과의 Communication ····· 61

제 3 장 결 론 ····· 64

제 1 장 서 론

생산 시스템에서의 로보트의 이용은 반복성과 유연성으로 가공작업이나 공작물 이동등에서 그 중요성을 더해가고 있다.

특히 시장변화에 따른 제품의 다양화와 제품수명의 단축 등으로 인하여 로보트 system의 유연성(flexibility)등을 더욱 요구하게 되었다.

금형 가공작업에도 있어서 가공작업시의 여러 문제로 인하여 로보트나 전용기계를 이용하려는 시도가 이루어지고 있다. 금형가공 후의 마무리작업인 Grinding, Lapping등이 수작업으로 이루어짐으로 인해서 마무리 작업에 소요되는 시간은 일반적으로 전체공정의 반이상을 차지하며, 이에 투입되는 인원도 매우 큰 부분을 차지한다. 또한 연마작업이 가지는 특성중 반복성이라는 면은 이의 자동화라는 측면을 뒷받침 해주고 있다.

그러나 금형이라는 대상이 가지는 또 다른 면은 단지 한두개의 수량에 지나지 않다는 면이다. 따라서 마무리 작업을 위하여 로보트가 투입될 시 이의 작업프로그램의 생성이 문제가 된다.

일반적인 Teach-in 방식을 통한 작업 program의 창출은 금형의 마무리 작업을 위해서는 무의미하며 따라서 작업상태의 사전 Test를 겸한 off-line을 이용한 작업 설계 및 programming이 요구되고 있다.

이와 같은 off-line program을 위해서는 다음을 만족해

야 한다.

첫째, 금형가공 process 중 연마작업을 위한 로보트작업 program의 off-line programming software를 개발하여야 한다.

둘째, CAD system과의 intergration을 완성해야 한다.

세째, 로보트의 정확도를 보장해줄 calibration system을 개발하여야 한다.

이를 위하여 자료수집과 함께 로보트와 VAX system과의 communication을 실현하였으며 일반적인 off-line programming system을 개발하였다.

그리고 현재 로보트 system의 Calibration system과 CAD system을 개발하고 있다.

앞으로 전체적인 Off-line programming system을 구현하고 오차보정 system을 구현할 계획으로 있다.

제 2 장　본　　론

제 1 절　PROPS System Configuration

1. 개　요

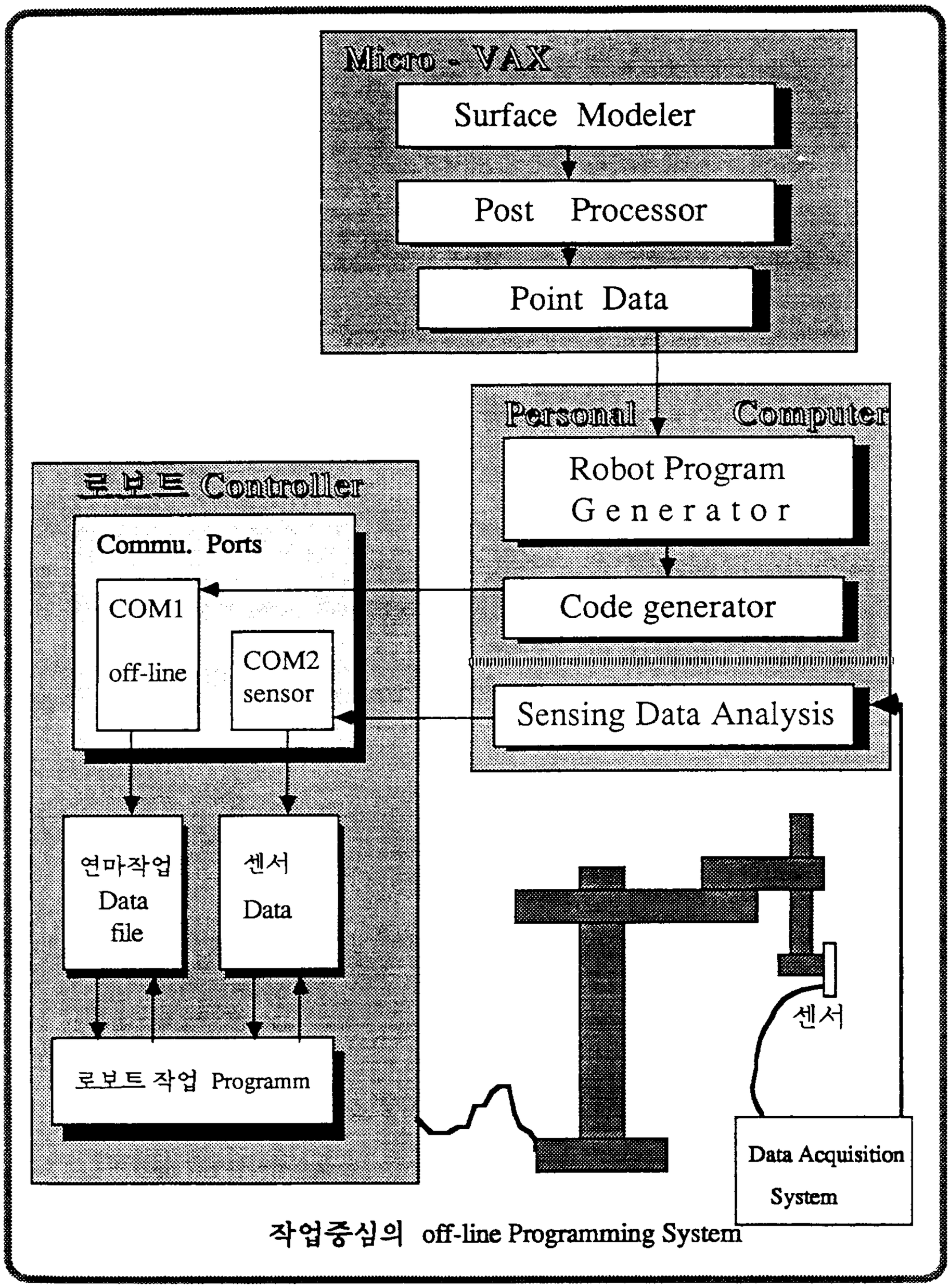
금형의 마무리 작업중 polishing 작업을 로보트를 이용하여 자동화함에 있어, 3 차원 곡면형상에 대한 로보트 작업 program을 기존 CAD system으로부터 position 및 orientation vector를 취득하여 열외(off-line)적인 방식으로 만드는 방식이 제시되었다.

그림 1-1 과 그림 1-2 는 off-line programming system의 전체구성을 나타내는 그림이다. PROPS 는 그림 1-1 에서 크게 Micro VAX 와 PC와 Robot controller 의 세부분으로 나뉘어진다.

PROPS 는 먼저 Micro VAX 에서 금형의 형상을 CAD system 으로부터 받은 후 금형의 재질과 graphic library 에 내장되어 있는 공구를 interactive 하게 선택한다.

공구의 dimension과 path width가 정해짐으로 가공 path를 generation 하게 된다.

VAX 에서 만들어진 path와 정보들은 personal computer인 PS/2로 넘어온다. PS/2는 내장되어 있는 relational data base와 presentation manager를 이용하여 표면의 roughness, 가압력과 같은 input을 받는다.



KAIST CAD/CAM M. KIM

그림 1-1. 작업중심의 off-line Programming System

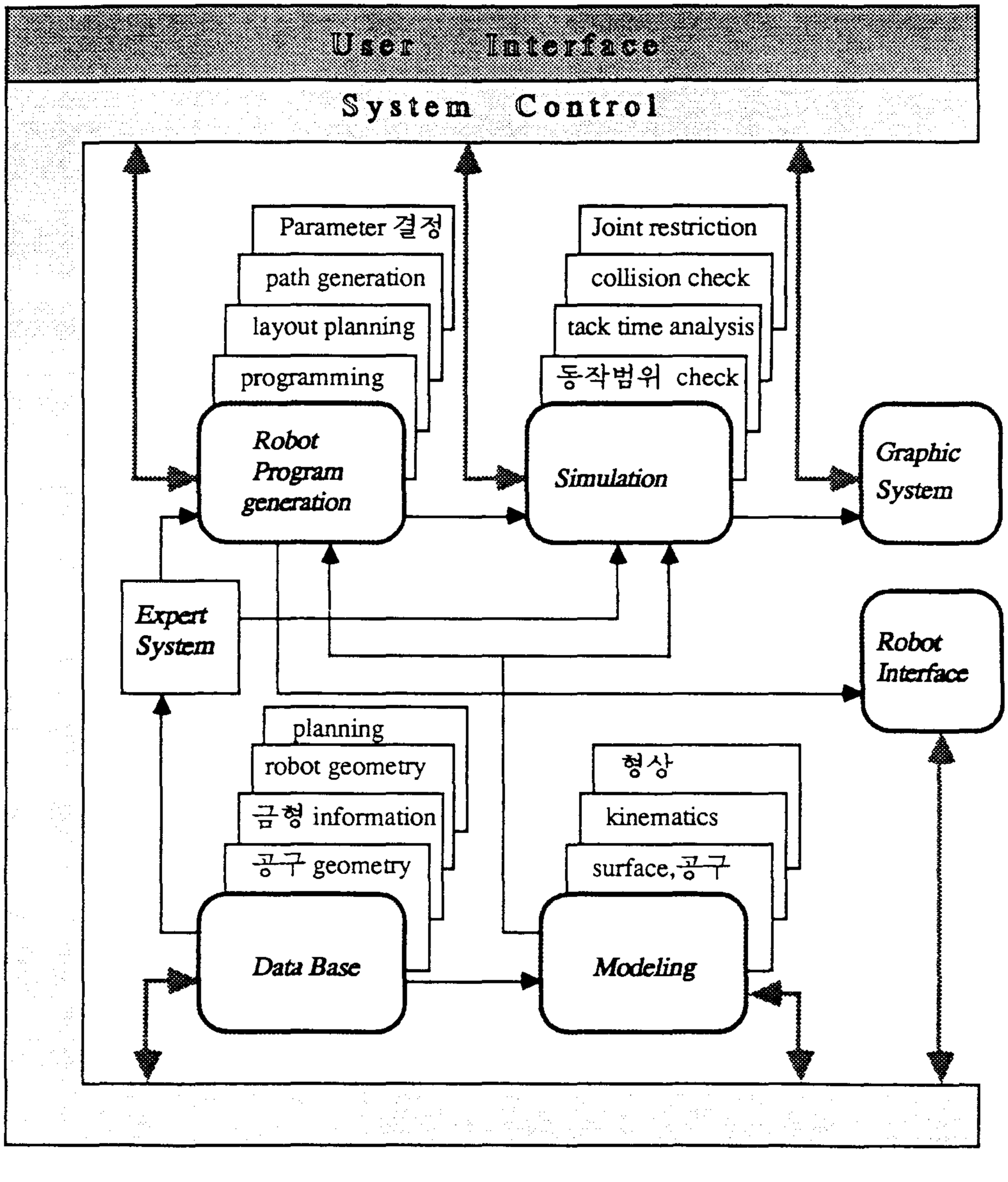


그림 1-2. 주요 system structure

위의 input을 이용하여 인공지능기술(AI)를 이용한 expert system을 사용하여 공구의 r.p.m, 구동속도 등의 robot polishing parameter를 계산해낸다.

이러한 polishing parameter를 이용하여 robot구동 program을 완성하고, Calibration process를 통하여 Kinematical parameter를 정확하게 identify한다. 그리고 Robot를 구동하면서 Sensor의 data를 받아서 제어를 한다.

2. Micro VAX

Micro VAX에서는 먼저 금형의 형상을 CAD system에 의해서 완성한다. 그리고 금형의 재질을 선택한후 graphic library에서 공구를 선택함으로써 공구의 dimension과 그에 따른 path width가 정하여진다. 위의 정보들을 이용하여 path를 generation하게 된다.

path generation 시에는 다음과 같은 사항을 고려해야 한다.

① 하나의 patch는 하나의 공구를 사용한다(가정)

② 공구와 surface의 collision을 check한다.

③ 가공이 끝난 patch와 가공전의 patch를 기억해야 한다.

④ 공구의 교환이 가능해야 한다.

⑤ Optimal한 patch를 찾는 routine이 필요하다.

3. PS/2

PS/2 에는 Micro VAX 에서 생성된 가공 path, 공구의 종류등의 input 이 들어오고, 그외 가공할 표면의 조도 (roughness)등의 user input은 직접 Key-in 한다.

그 외의 parameter들은 default parameter로 입력된다. 위와 같은 입력을 받은후 인공지능 기술(AI)를 이용한 expert system을 사용하여 가압력이나 공구의 r.p.m, 구동속도 등을 계산해 내고, 이러한 polishing parameter를 이용하여 robot 구동 program을 완성함으로써 PROPS는 끝나게 된다.

제 2 절 PROPS System의 구성

1. Surface modeler

본 연구에는 한국과학기술연구원(KIST)의 CAD/CAM 실에서 개발한 surface modeler를 사용하여 surface를 design 하였다. 이 modeler는 menu를 사용하여 여러가지 방법으로 원하는 surface를 generation 할 수 있으며, graphic driver로는 WAND를 사용하였고 graphic device로는 Megatek 7255을 사용하였다.

WAND는 Whizzard graphic system의 Megatek hardware를 지원하기 위하여 만들어진 software package 이다.

WAND는 SIGGRAPHCORE guidline들을 따라서 쓰여진 FORTRAN library 로 구성되어 있다.

WAND는 다음 네가지 level로 되어있다.

① System level은 device driver와 함께 여러가지 종류의 host computer operating system에 WAND를 적용하는 routine들을 포함한다.

② Workstation level은 모든 Wizzard graphic system의 기능을 지원해주는 routine들을 포함한다.

③ User level은 3D clipping과 perspective뿐만 아니라 modeling과 image transformation까지 포함하는, world coordinate space에서의 완전한 viewing operation들을 제공하고 있다.

④ Extended workstation은 display list memory를 제한없이 사용할 수 있는 것을 포함하는 Hardware를 직접 조작하는 routine들을 사용자에게 제공한다.

본 연구에서는 주로 workstation level을 사용하여 program 하였다.

2. Robot Kinematic

PROPS에서는 여러개의 Robot를 library로 가지고 있게 해서 robot를 선택하여 simulation이 가능하게 하였다. 현재로는 FANUC robot와 비접촉 3차원 측정기가 적용되어 있고 NACHI robot는 개발중에 있다.

각 Kinematic는 Denavit-Hartenberg의 방식에 따라서 transform 되어지며 inverse transformation은 각 Kine-

matic 별로 고유의 것이 적용되어진다.

(1) Denavit-Hartenberg Transformation

인접하는 link사이의 translation과 rotation 관계를 기술하기 위하여 Denavit Hartenberg는 4개의 변수로서 관절이 있는 연속물의 link에 좌표계를 체계적으로 설정하는 matrix 방법을 제안하였다.

DH의 표현 방식은 한 joint에서의 그 전 link의 좌표계에 대한 각 link의 좌표계를 나타내는 4×4 homogeneous transformation matrix를 만들어 낸다.

그래서 연속적인 transformation을 통해서, hand coordinate로 표현되는 end-effect는 dynamic system의 inertial frame인 base coordinate로 표현되고, transform 되어진다.

각 link의 coordinate frame은 다음 세가지 법칙에 의해서 설정되어진다.

i) Z_{i-1} 축은 i번 joint의 운동축을 따라서 놓인다.

ii) X_i 축은 Z_{i-1} 축에 수직하면서 멀어지는 방향으로 놓인다.

iii) Y_i 축은 오른손 좌표계를 만족하는 방향이다.

DH의 표현 방식에는 다음 네가지의 변수가 정의된다.

i) θ_i : Z_{i-1} 축을 중심으로 X_{i-1} 에서 X_i 로의 오른손 법칙에 따른 각

ii) d_i : (i-1)번째 좌표계의 원점에서 X_i 와 Z_{i-1} 가 만나는 점까지의 거리

iii) a_i : 회전된 X_{n-1} 축 즉 X_n 을 따라서의 translation 된 거리

iv) α_i : X_n 축을 중심으로한 회전각

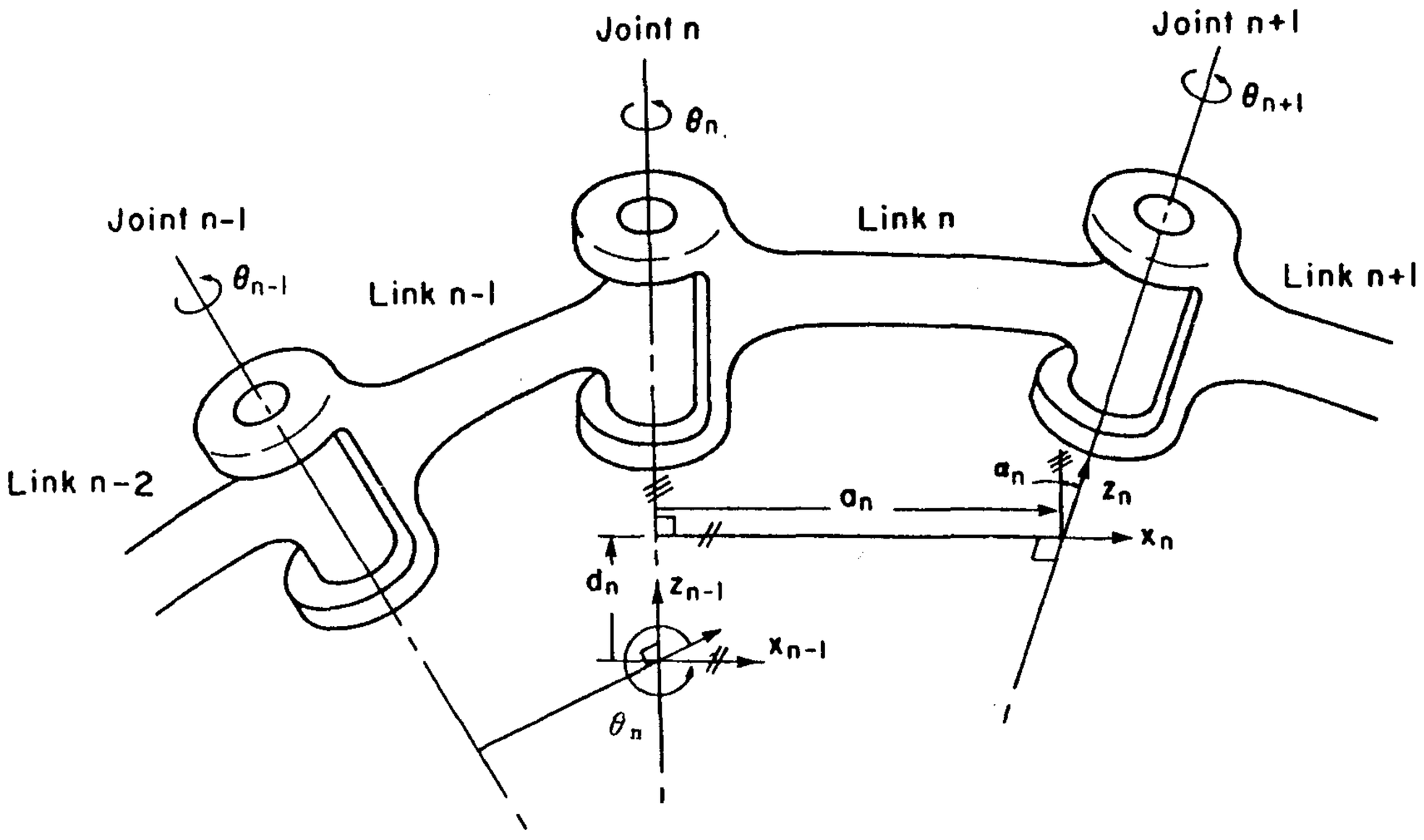


그림 2-1. Link Parameters θ, d, a and α

revolute joint 에서의 A_n matrix는 다음과 같다.

$$A_n = \mathrm{Rot}(Z,\theta)\ \mathrm{Trans}(0,0,d)\ \mathrm{Trans}(\alpha,0,0)\ \mathrm{Rot}(x,\alpha)$$

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

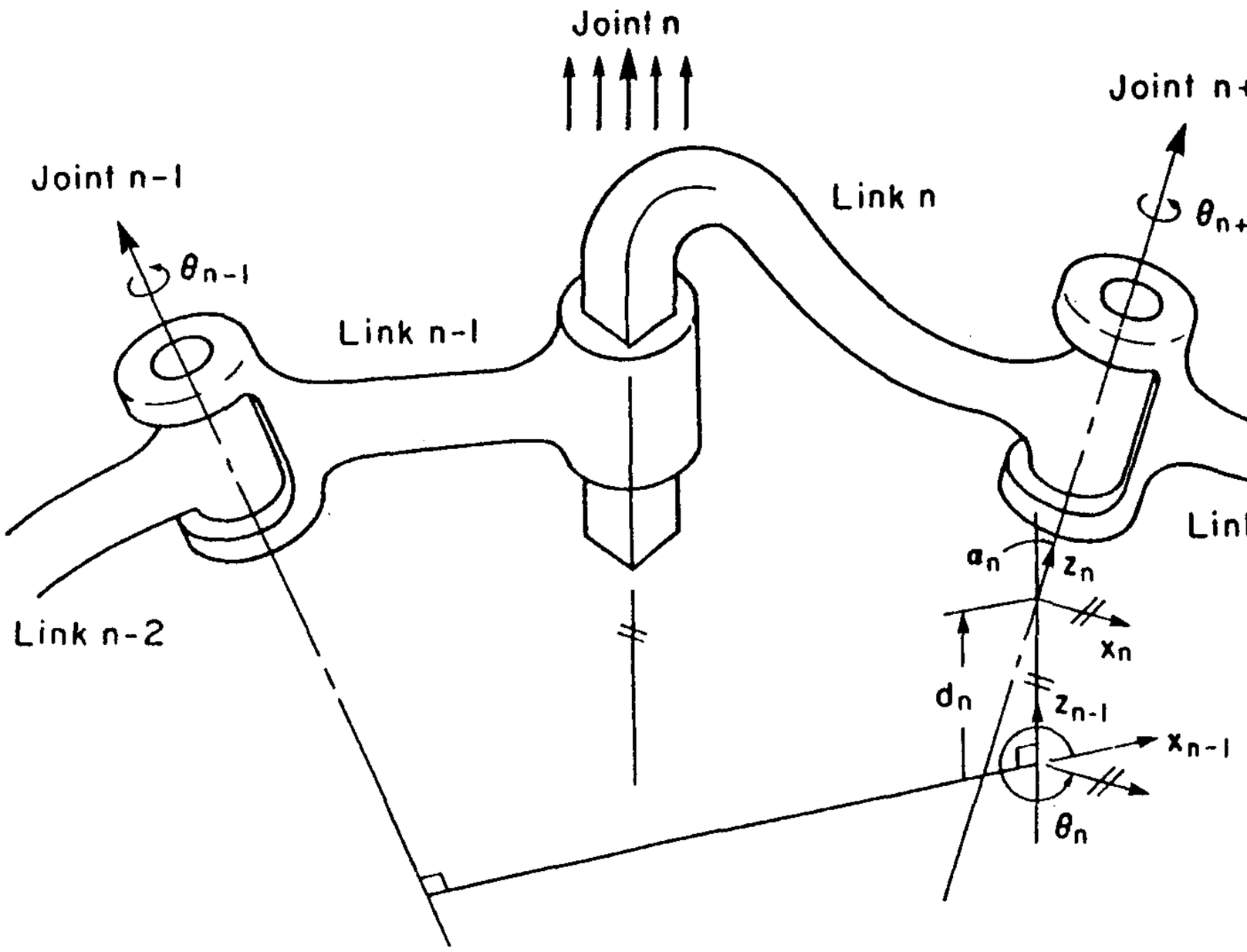


그림 2-2. Link Parameters θ, d and a for a Prismatic Joint

Prismatic joint 에서는 revolute joint 에서의 식에서 $a = 0$ 이므로

$$A_n = \mathrm{Rot}(Z,\theta)\ \mathrm{Trans}(0,0,d)\ \mathrm{Rot}(x,\alpha)$$

$$A_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & 0 \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\theta & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(2) FANUC ROBOT

1) FANUC Robot의 형상 및 작업영역

FANUC Robot는 5축로보트로서 크게 Control unit, mechanical unit, wrist의 세부분으로 나뉘어진다. Wrist는 A, E, F형의 세가지 종류가 있는데 이번 연구에서는 wrist F를 장착한 FANUC A1 Robot를 대상으로 삼았다.

자유도는 그림에서 보는 바와같이 Z축으로의 상하운동량인 'Z'와 Z축을 중심으로한 회전각도 'θ', 작업반경의 변화량인 'R' 그리고 wrist rotation 값인 α, wrist swing 값인 β로 구성되어 진다.

다음의 <그림 2-3>은 FANUC Robot의 전체형상을 나타내며 <그림 2-4>는 FANUC Robot의 작업공간과 제원을 나타낸다.

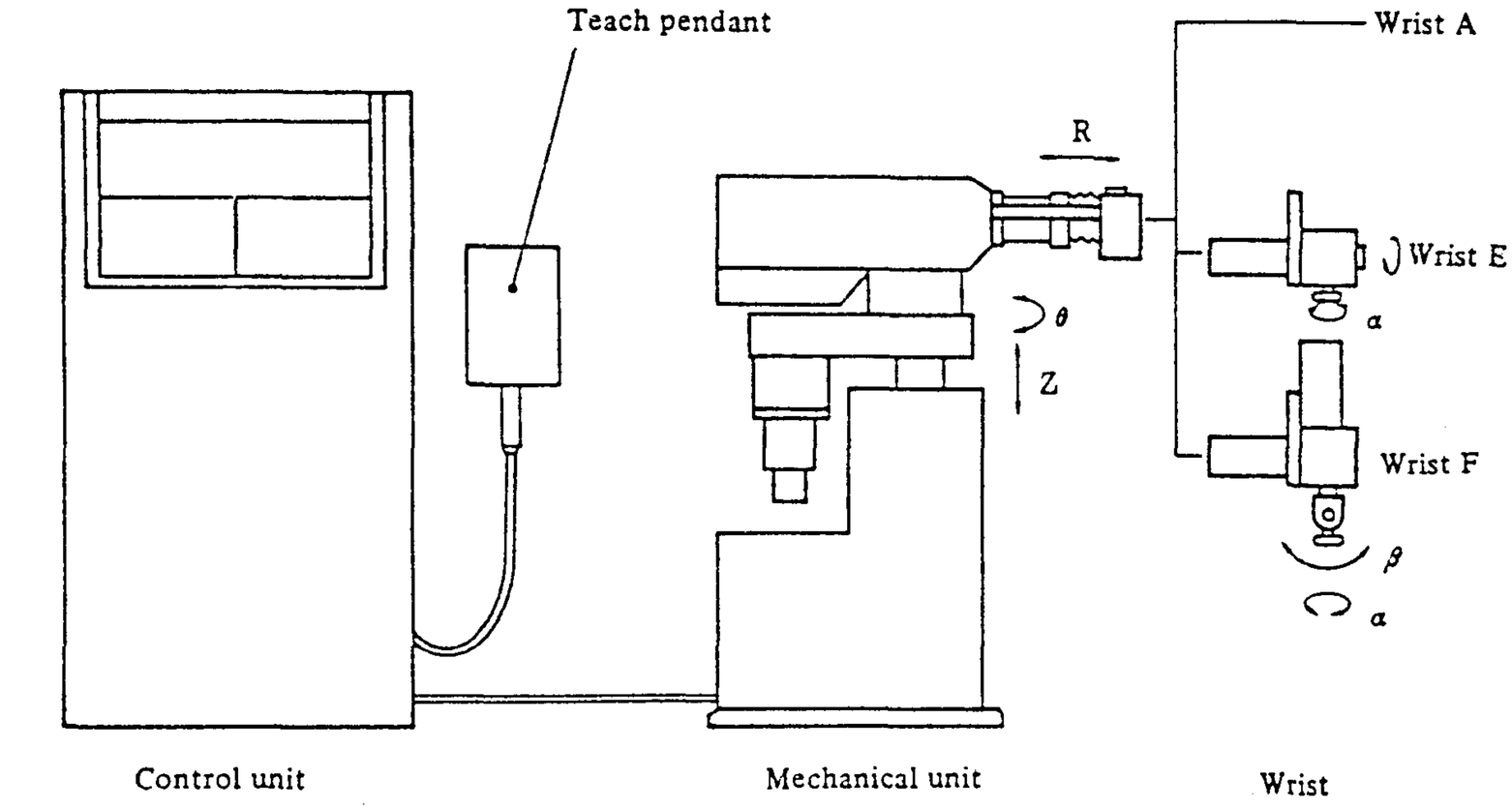


그림 2-3. FANUC Robot의 전체 형상

FANUC Robot의 작업공간은 다음과 같다.

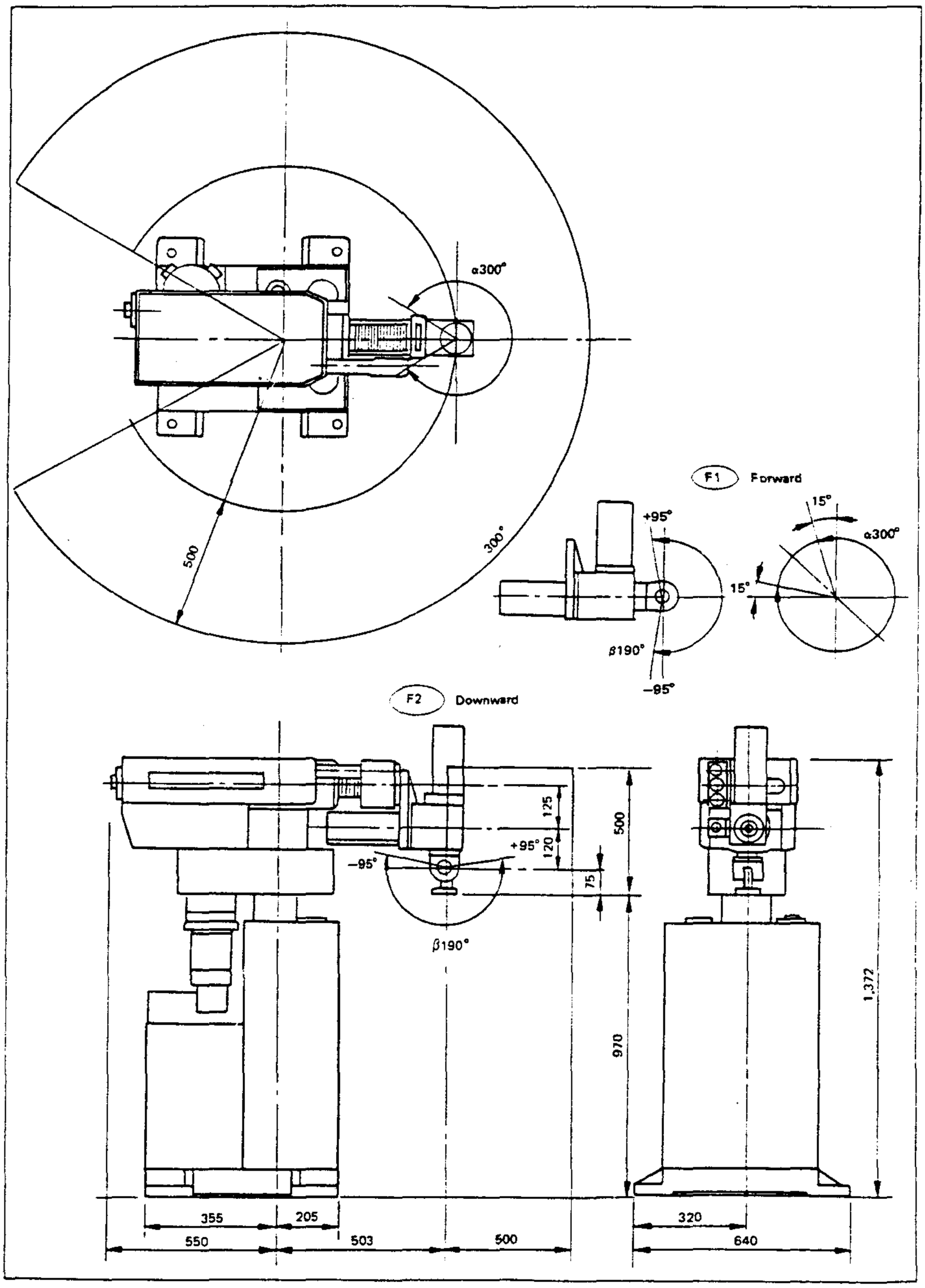


그림 2-4. FANUC Robot의 작업공간

2) Forward Transformation

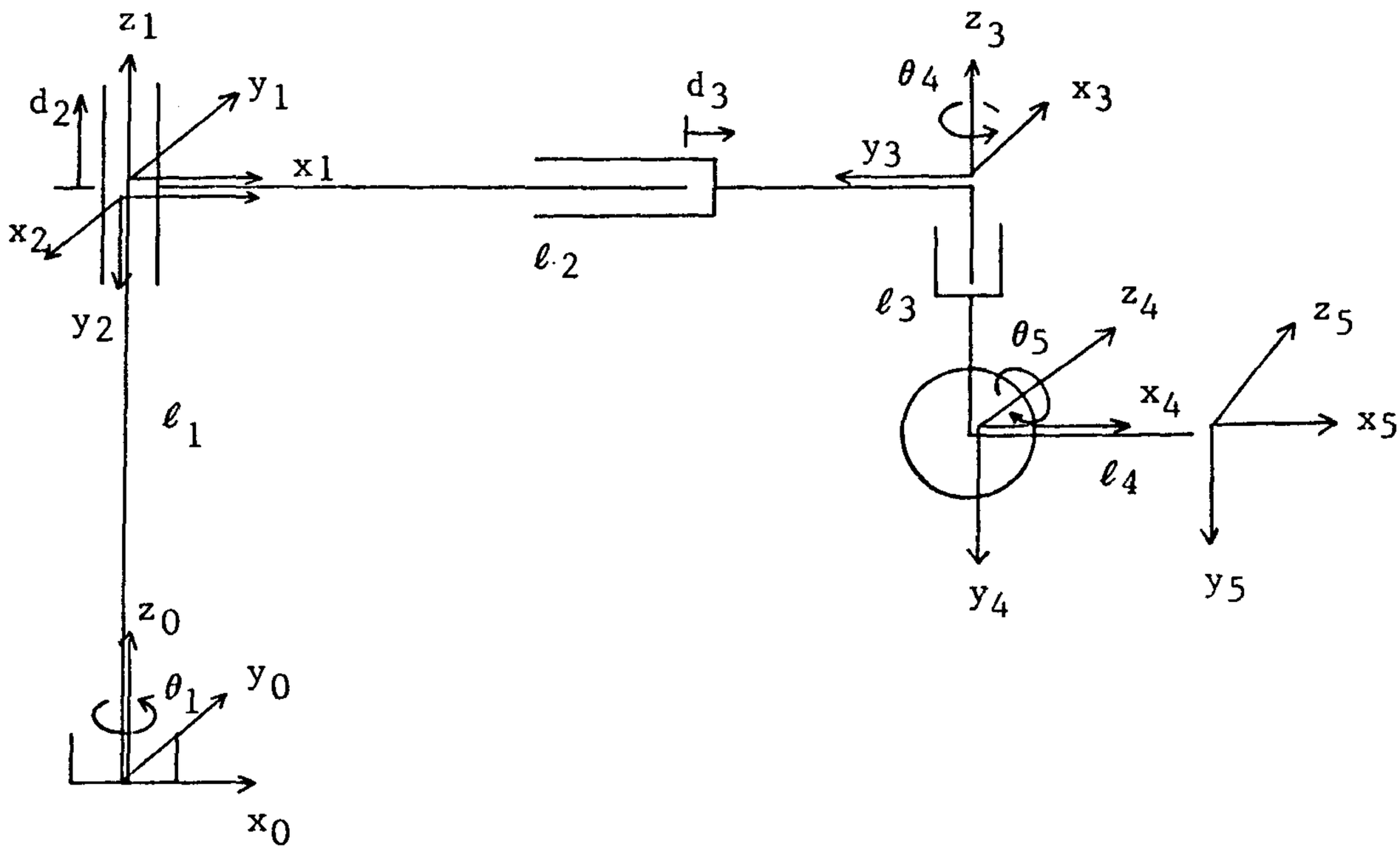


그림 2-5. FANUC Robot의 좌표축

$$A_0 = Rot(Z_0, \theta_1)\ Trans(Z_0, l_1)$$

$$A_1 = Rot(Z_1, -\frac{\pi}{2})\ Trans(Z_1, d_2)\ Rot(x_1, -\frac{\pi}{2})$$

$$A_2 = Rot(Z_2, \pi)\ Trans(Z_2, l_2+d_3)\ Rot(x_2, -\frac{\pi}{2})$$

$$A_3 = Rot(Z_3, -\frac{\pi}{2}+\theta_4)\ Trans(Z_3, -l_3)\ Rot(x_3, -\frac{\pi}{2})$$

$$A_4 = Rot(Z_4, \theta_5)\ Trans(x_4, l_4)$$

link	Variable	θ	α	a	d	cosα	sinα
1	θ_1	0	0	0	l_1	1	0
2	d_2	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	d_2	0	-1
3	d_3	π	$-\frac{\pi}{2}$	0	l_2+d_3	0	-1
4	θ_4	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	$-l_3$	0	-1
5	θ_5	0	0	l_4	0	1	0

- Transformation matrix

$$T_1 = \begin{bmatrix} \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ -\cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & -1 & 0 & d_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = A_0A_1A_2 = \begin{bmatrix} -\sin\theta_1 & -\cos\theta_1 & 0 & (l_2+d_3)\cos\theta_1 \\ \cos\theta_1 & -\sin\theta_1 & 0 & (l_2+d_3)\sin\theta_1 \\ 0 & 0 & 1 & l_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3 = A_0A_1A_2A_3 =$$

$$= \begin{bmatrix} -\sin(\theta_1+\theta_4') & 0 & -\cos(\theta_1+\theta_4') & (l_2+d_3)\cos\theta_1 \\ \cos(\theta_1+\theta_4') & 0 & -\sin(\theta_1+\theta_4') & (l_2+d_3)\sin\theta_1 \\ 0 & -1 & 0 & -l_3+l_1+d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

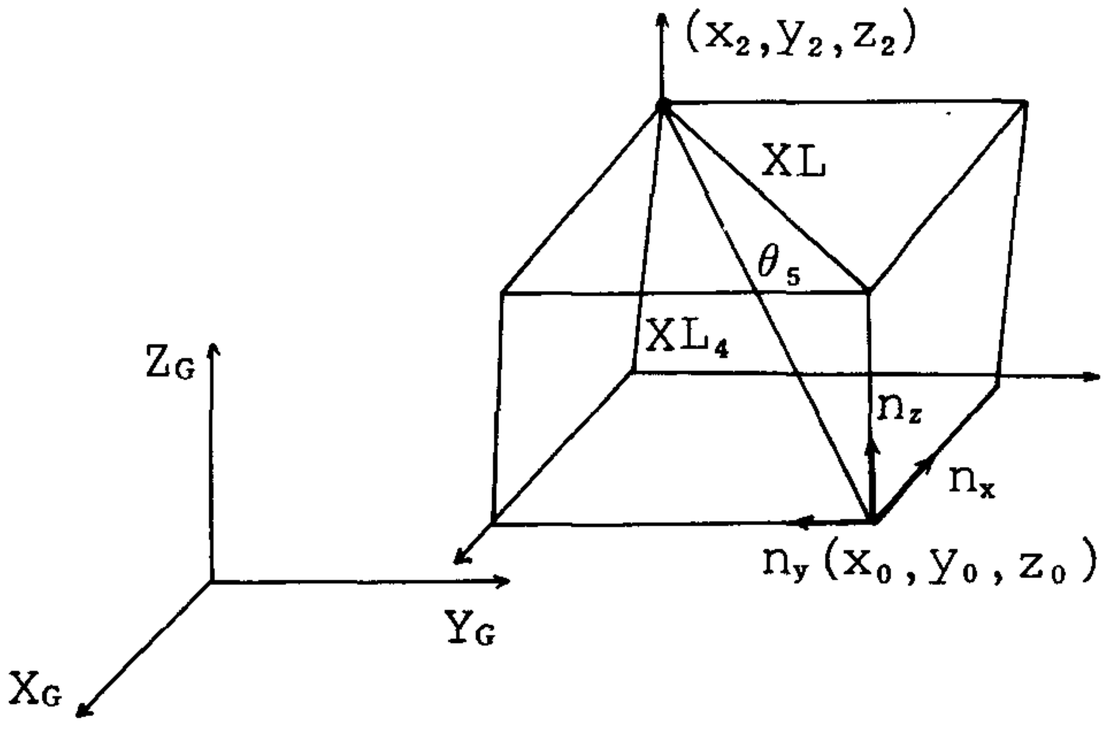
here, $\theta_4' = -\frac{\pi}{2} + \theta_4$

$$T_4 = \left[\begin{array}{ccc:c} -\cos\theta_5\sin(\theta_1+\theta_4') & \sin\theta_5\sin(\theta_1+\theta_4') & -\cos(\theta_1+\theta_4') & -\sin(\theta_1+\theta_4')l_4\cos\theta_5 \\ & & & +(l_2+d_3)\cos\theta_1 \\ \cos\theta_5\cos(\theta_1+\theta_4') & -\sin\theta_5\cos(\theta_1+\theta_4') & -\sin(\theta_1+\theta_4') & \cos(\theta_1+\theta_4')l_4\cos\theta_5 \\ & & & +(l_2+d_3)\sin\theta_1 \\ -\sin\theta_5 & \cos\theta_5 & 0 & -l_3+l_1+d_2 \\ & & & -l_4\sin\theta_5 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right]$$

3) Inverse transformation

input x_0, y_0, z_0 : surface 의 point data

$\bar{n} = [n_x\ n_y\ n_z]^T$: surface 의 unit normal vector



$$x_2 = x_0 + n_x \cdot XL_4$$

$$y_2 = y_0 + n_y \cdot XL_4$$

$$z_2 = z_0 + n_z \cdot XL_4$$

$$XL = \sqrt{n_x^2 + n_y^2}$$

그림 2-6. Wrist 부분의 inverse

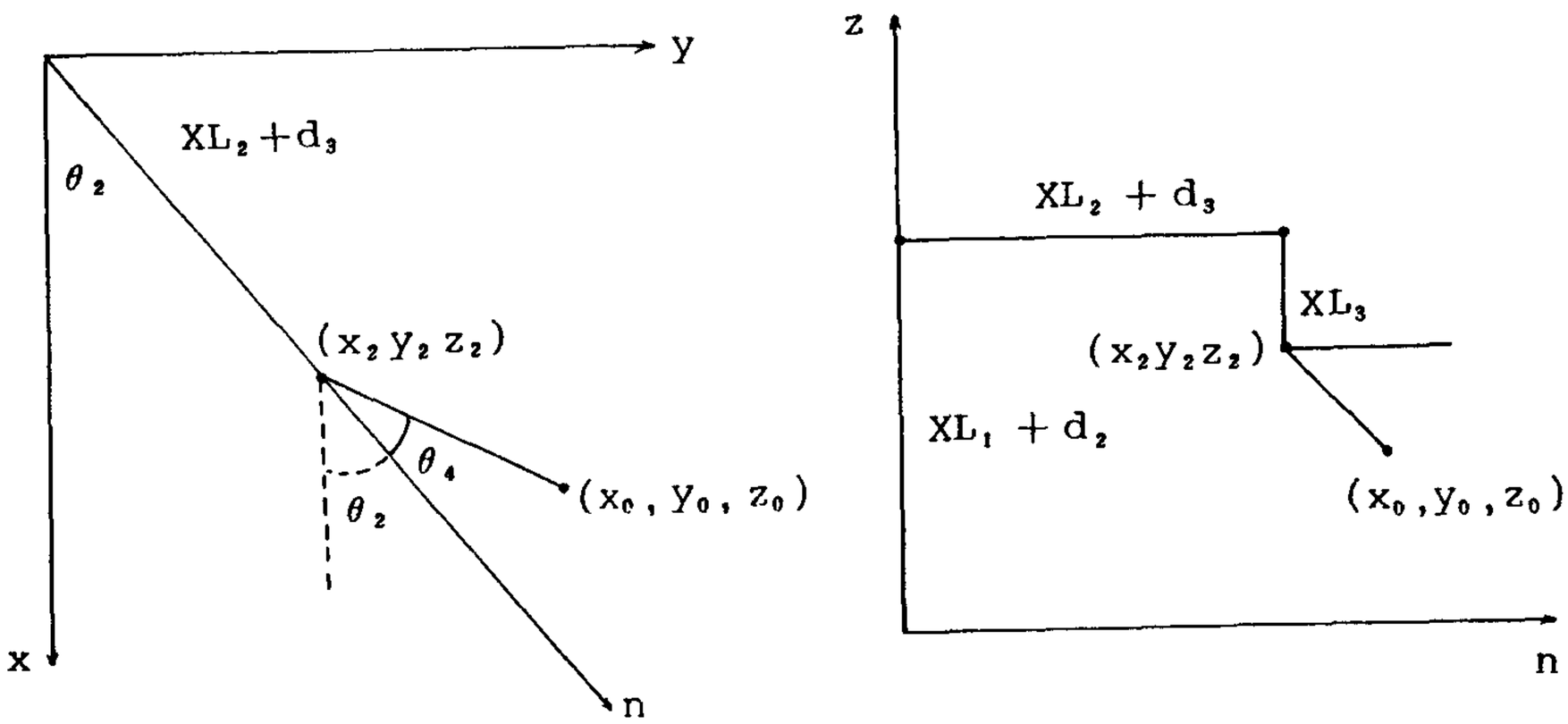


그림 2-7. arm의 inverse

$$\theta_2 = \mathrm{ATAN2}\left(\frac{y_2}{x_2}\right)$$

$$\theta_4 = \mathrm{ATAN2}\left(\frac{n_y}{n_x}\right) - \theta_2$$

$$\theta_5 = \mathrm{ATAN2}\left(\frac{n_z}{XL}\right) \quad \text{where } XL = \sqrt{n_x^2 + n_y^2}$$

$$d_2 = Z_2 - (XL1 - XL3)$$

$$d_3 = XXL - XL_2 \quad \text{where } XXL = \sqrt{x_2^2 + y_2^2}$$

(3) HYUNDAI Robot(NACHI)

1) 구 성

HYUNDAI Bobot HR-7000은 전동식 5축으로 구성된 Arc 용접용 robot 이다. HR-7000은 Robot 본체와 16 bit Micro Processor를 중심으로한 제어장치 및 Teaching Controller등으로 구성되어 있다.

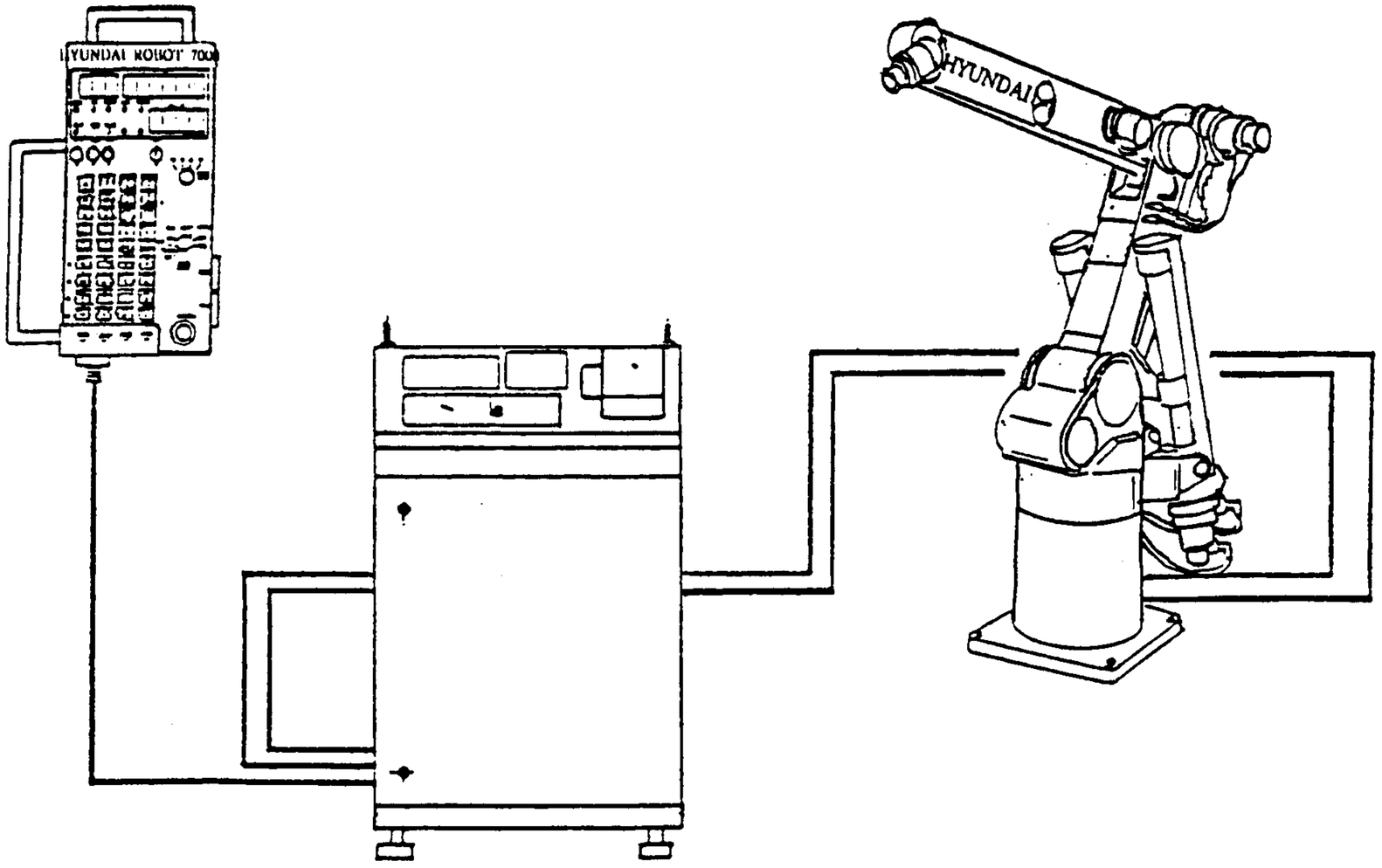


그림 2-8. HR-7000 의 기본구성

[표1-2]

축명칭	동 작	Teaching Controller Button	
S 축	Arm 선회 (좌,우회전)	Left	Right
V 축	Arm 상하 (상,하이동)	Up	Down
H 축	Arm 전후 (전,후이동)	Fwd	Back
R 축	Wrist 회전 (부분 회전)	R r	RL
B 축	Wrist 상하 (부분 상하)	B up	B Down

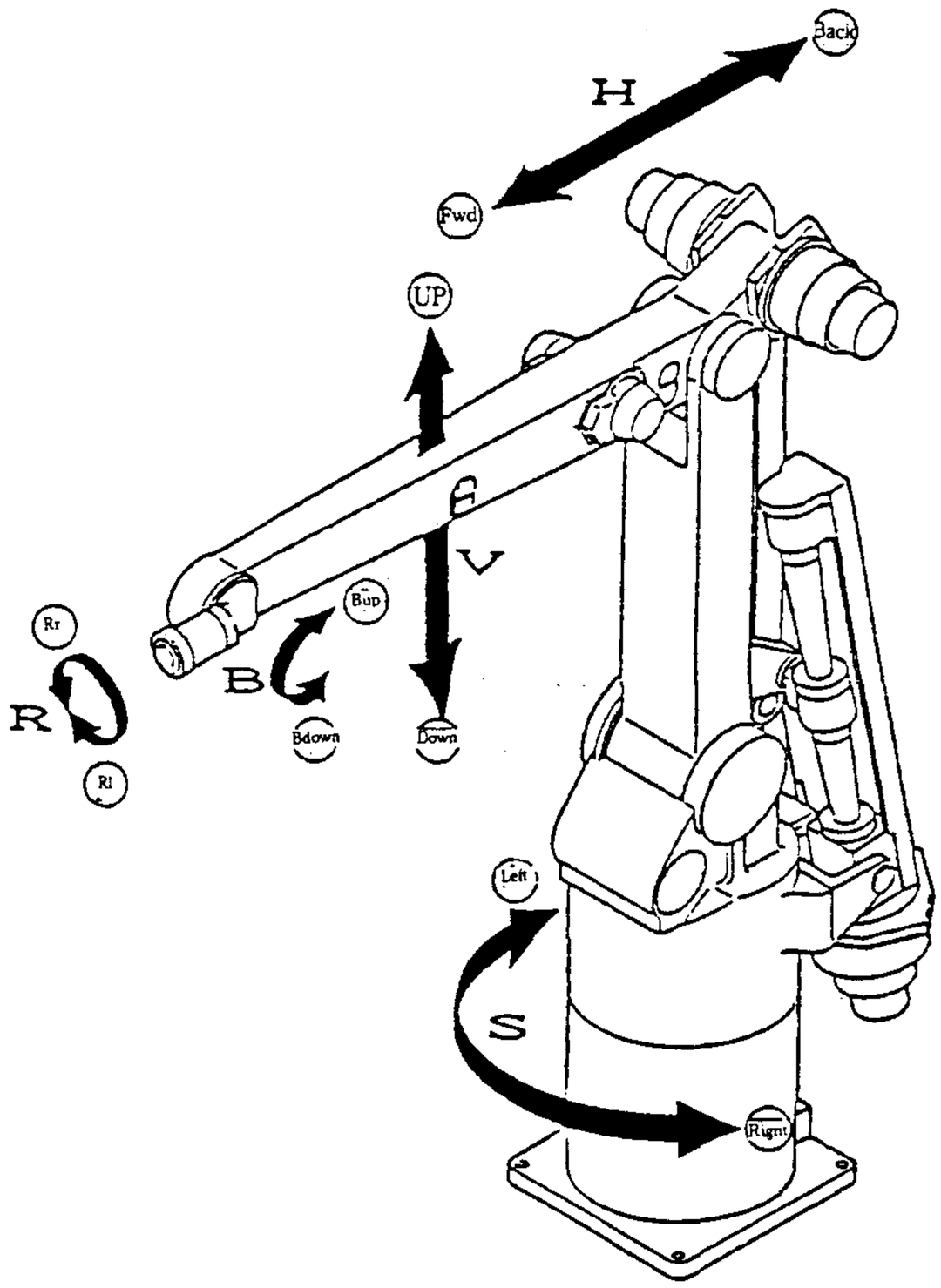


그림 2-9. 본체의 외관과 작동축

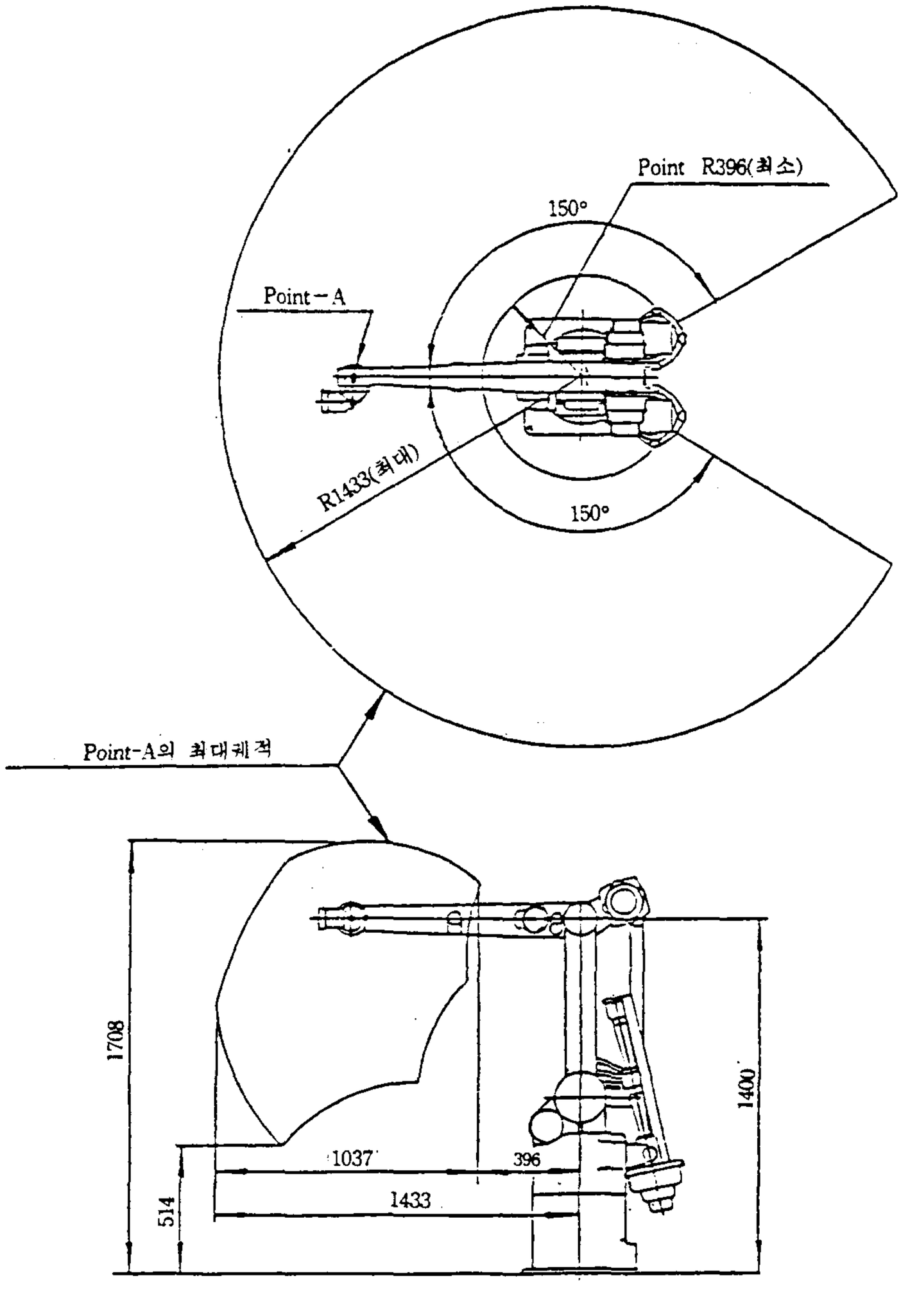


그림 2-10. 표준 동작범위

다음의 그림 2-8 는 HR-7000 Robot 의 기본구성에 대한 그림이다. HR-7000 robot 는 좌우회전을 해서 Arm 을 선회시키는 S 축과 Arm 을 상하이동 시키는 V 축, Arm 을 전후 이동 H 축, Wrist 를 회전시키는 R 축과 wrist 상하로 이동시키는 회전축 B 등의 5 개의 축으로 구성되어 있다.

다음의 그림 2-9 는 본체외관과 작동축을 설명한 그림이다. 그리고 그림 2-10 는 B 축 상의 점인 A 점의 최대괘적인 표준동작범위를 나타내는 그림이다.

2) Forward Transformation

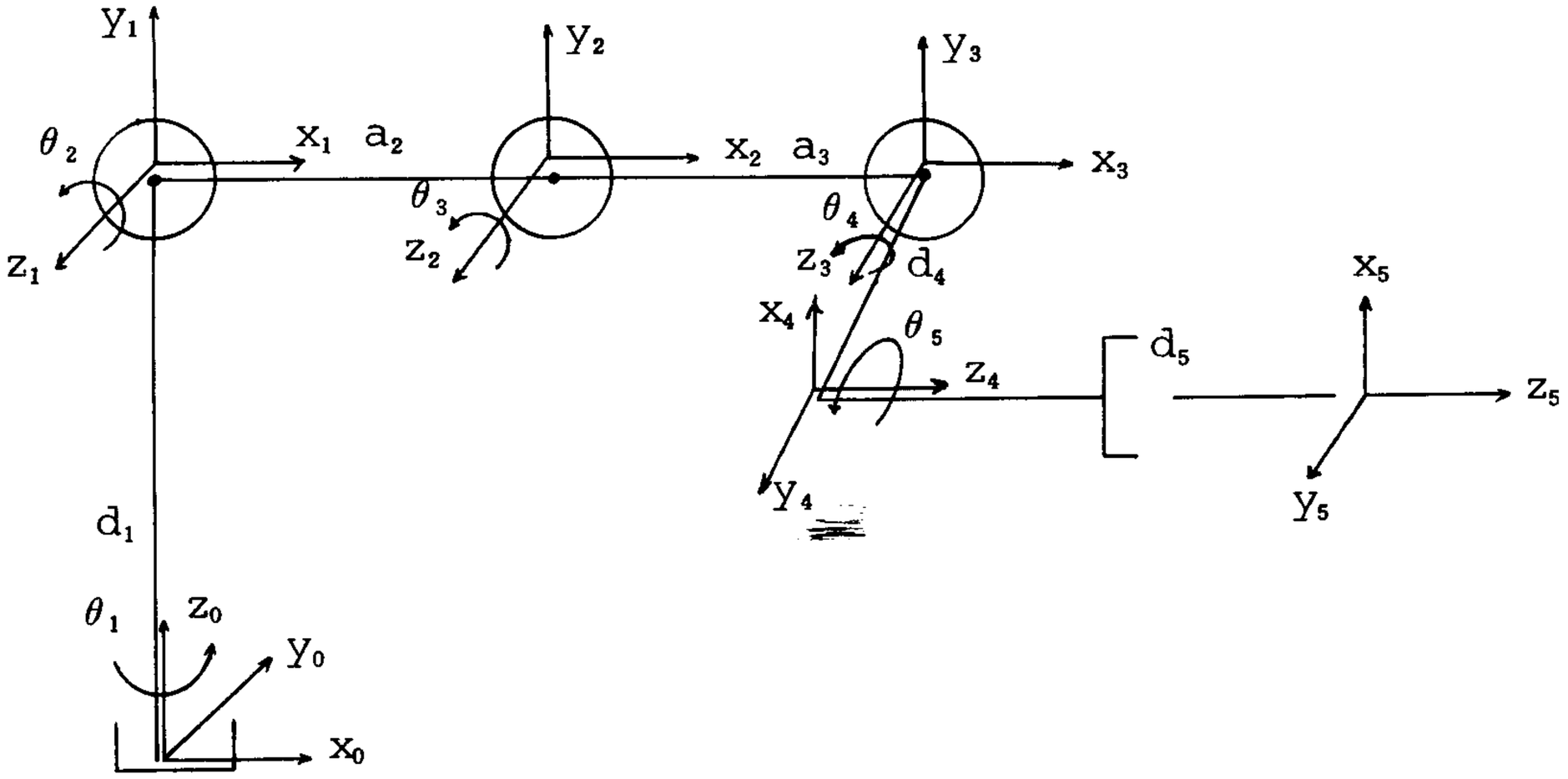


그림 2-11. HYUNDAI Robot 의 좌표축

Link	Variable	θ	α	a	d	cosα	sinα
1	θ_1	θ_1	$\frac{\pi}{2}$	0	d_1	0	1
2	θ_2	$\theta_2+\frac{\pi}{2}$	0	a_2	0	1	0
3	θ_3	$\theta_3-\frac{\pi}{2}$	0	a_3	0	1	0
4	θ_4	$\theta_4+\frac{\pi}{2}$	$\frac{\pi}{2}$	0	d_4	0	1
5	θ_5	θ_5	0	0	d_5	0	0

$A_0 = \mathrm{Rot}(Z_0, \theta_1)\ \mathrm{Trans}(Z_0, d_1)\ \mathrm{Rot}(x_0, \frac{\pi}{2})$

$A_1 = \mathrm{Rot}(Z_1, \theta_2+\frac{\pi}{2})\ \mathrm{Trans}(x_1, a_2)$

$A_2 = \mathrm{Rot}(Z_2, \theta_3-\frac{\pi}{2})\ \mathrm{Trans}(x_2, a_3)$

$A_3 = \mathrm{Rot}(Z_3, \theta_4+\frac{\pi}{2})\ \mathrm{Trans}(z_3, d_4)\ \mathrm{Rot}(x_3, \frac{\pi}{2})$

$A_4 = \mathrm{Rot}(Z_4, \theta_5)\ \mathrm{Trans}(Z_4, d_5)$

(4) Measurement

1) Hardware

KIST의 CAD/CAM실에서 개발중인 비접촉 3차원 측정기를 Kinematic library에 library화 하였다. 측정기의 특성은 공압 servo를 이용하여 3축을 구동시키고 1개의 축을 제어할 수가 있다.

Velocity control은 되지않고 position control만 가능한데 측정범위는 x축 y축으로 100 mm이고 z축으로는 200 mm까지이다.

각축의 최고속도는 3m/sec이고 위치제어오차는 0.01 ~ 0.02 mm이다. sensor의 특성은 파장 780 nm인 Laser를 이용한 비

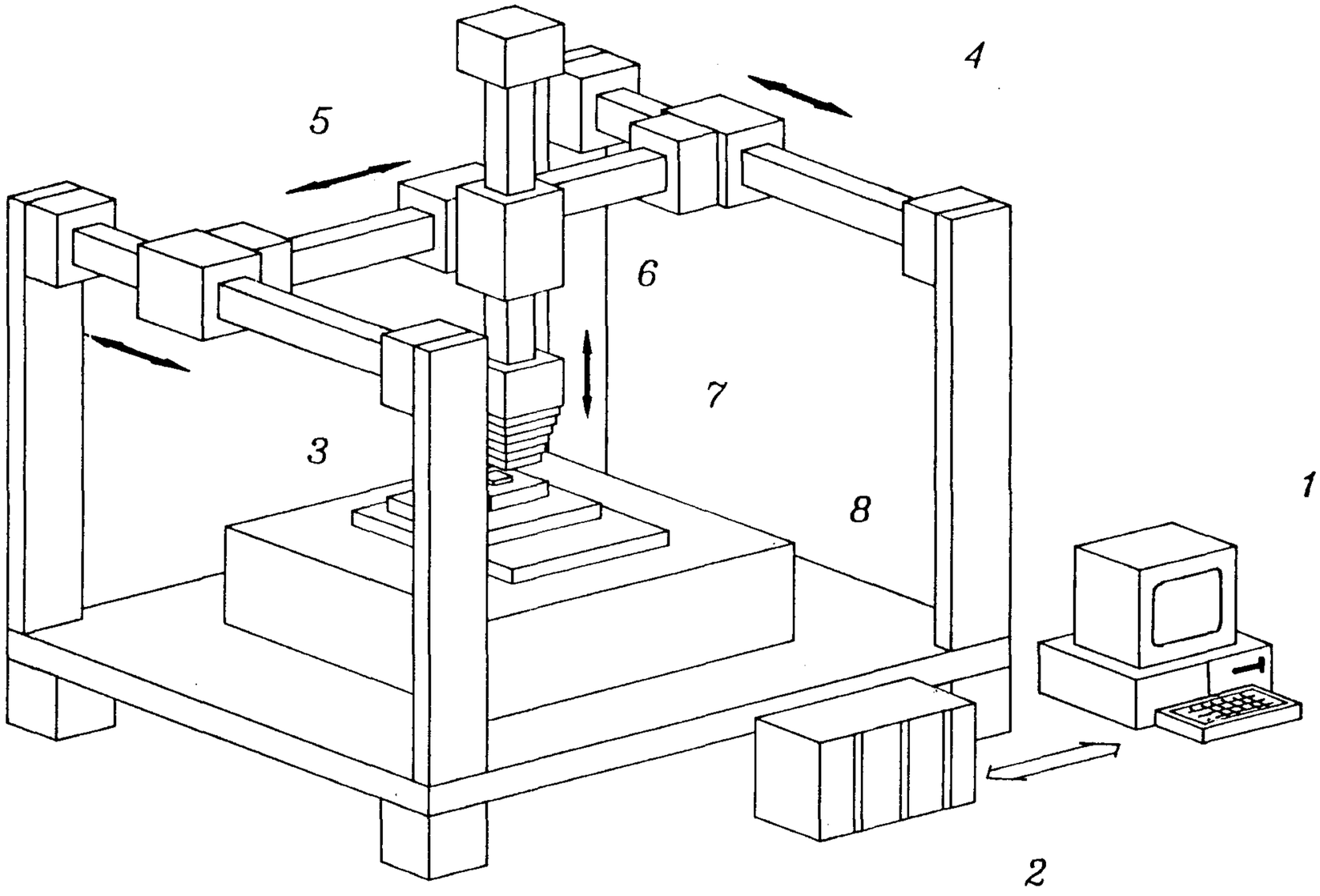


그림 2-12. 3차원 측정기의 형상

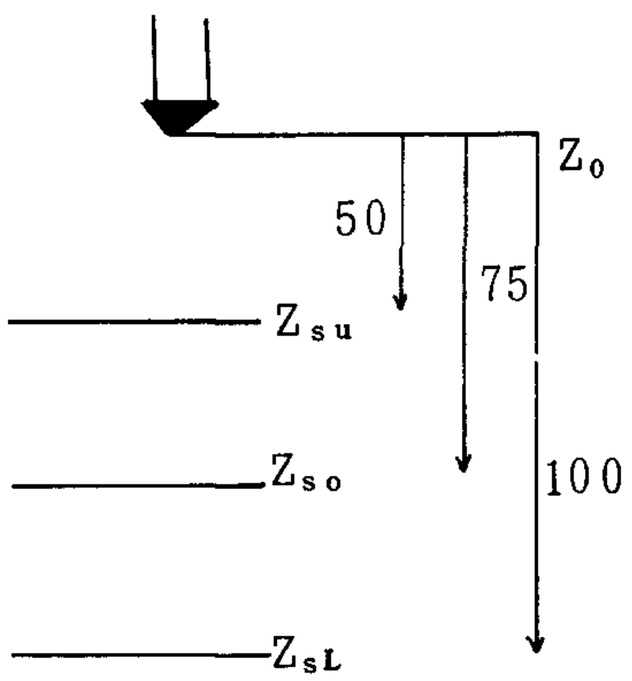
접촉 sensor 이 기준거리는 75 ± 1 mm 이고 측정거리는 ± 25 mm 이다.

다음 그림 2-12 는 3 차원 측정기의 형상을 나타낸 그림이다.

2) Forward Transformation

측정기의 Sensor 는 측정물로 부터 75 ㎜ 떨어진 곳을 기준 거리로 하며 기준거리에서 ± 25 ㎜까지의 범위에서 측정이 가능하다. 그러므로 측정범위를 벗어나면 기준거리를 ± 25 해주는 방법으로 측정이 행하여진다.

그러므로 surface 표면의 data X_m, Y_m, Z_m의 값에서부터 sensor 의 좌표 X_0, Y_0, Z_0 를 구하기 위해서도 다음의 식을 만족해야 한다.



$$X_0 = X_m$$

$$Y_0 = Y_m$$

$$Z_0 =$$

i) $Z_m > Z_{su}$ $Z_0 = Z_0 + 25$

ii) $Z_m < Z_{sL}$ $Z_0 = Z_0 - 25$

그림 2-13. 측 정 범 위

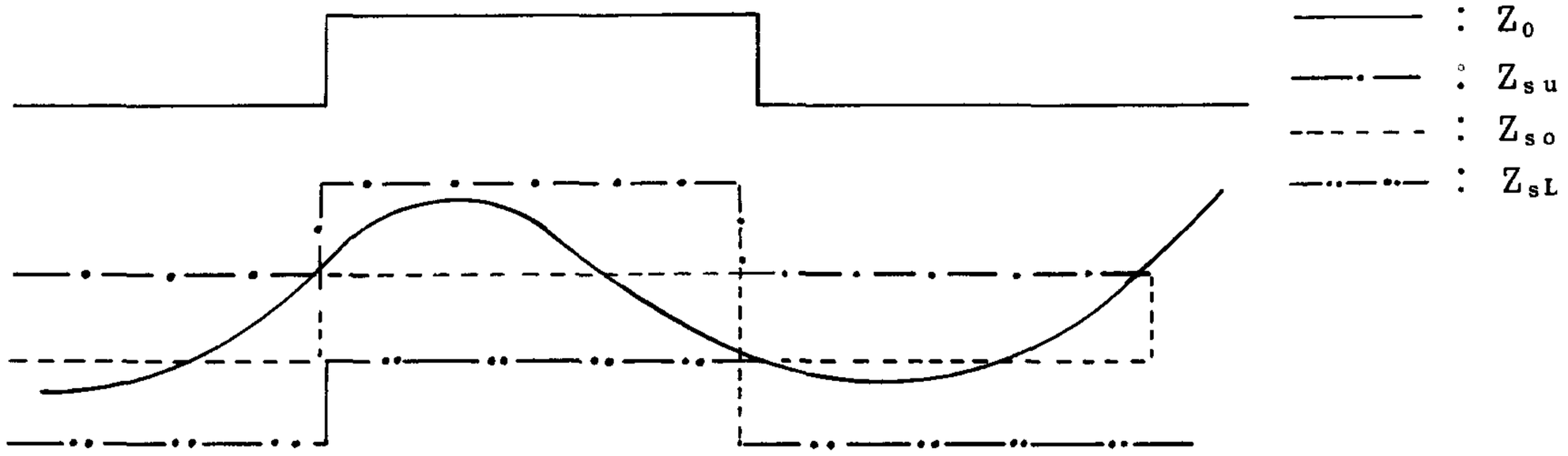


그림 2-14. 측 정 경 로

※ transformation matrix는 다음과 같다.

$$T_1 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline X_0 & 0 & 0 & 1 \end{array}\right]$$

$$T_2 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline x_0 & y_0 & 0 & 1 \end{array}\right]$$

$$T_3 = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline x_0 & y_0 & z_0 & 1 \end{array}\right]$$

3. MENU

o SEL-ROBO : main menu
o SEL-SURF
o MOV-SURF
o GEN-PATH
o SIMULATN
o OPTION
o TEST

(1) SEL-ROBO < select kinematic >

작업에 사용할 Robot kinematic의 이름을 display 해주고, 그 중 하나를 선택하면 IROBO가 결정되고 각 transformation의 index가 된다. Robot를 그린다.

o submenu

FANUC	IROBO = 1
HYUNDAI	IROBO = 2
MEASURE	IROBO = 3

(2) SEL-SURF < SELECT SURFACE >

Surface model library에서 surface를 선택하는 기능

(3) MOV-SURF < move surface >

기공을 할 surface를 file로 부터 읽어온 후에 Robot의 작업영역내에 정치시키기 위하여 화면을 4등분한다. 3개의

window는 3각도법에 의한 robot의 형상과 surface의 형상 및 robot의 운동범위를 표시하고, 나머지 1개의 window에는 1사분면에 eye coordinate를 둔 형상을 표시한다.

menu를 선택하여 x,y,z 방향으로의 translation과 Z축을 중심으로한 rotation을 하게 함으로써 작업영역내에 surface를 정치시킨다. 이때 translation양과 rotation양을 나타내는 message가 계속 나타나며 surface는 4개의 window에서 동시에 움직이게 된다.

- Submenu

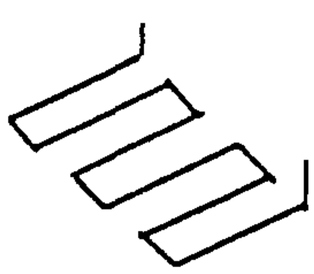
TRANS X
TRANS Y
TRANS Z
ROT Z

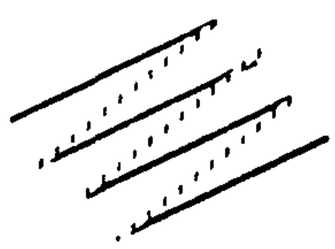
(4) GEN-PATH < generate path >

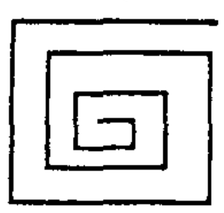
path의 종류가 display 되면 path의 종류를 결정하여 IPATH를 정한다. surface 상에 tool path를 그린다.

submenu

TRIPPATH	IPATH = 1
RETURN	IPATH = 2
SPIRAL	IPATH = 3







(5) SIMULATN

surface의 각 point들의 좌표와 법선 벡터가 기억되어 있는 file을 open하여 그 path를 따라가면서 joint restriction을 check한다. 만일 restriction에 걸리면 error message를 내면서 tip 끝을 따라 긋던 선의 색깔을 달리하게 된다.

- error message : 'nth joint restriction error
- submenu

  STEP : dialogue box에서 " input scanning step size "라는 message를 내고 step size를 input 받고 n번 step을 실행하고 정지한다.

  Cycle : 처음부터 마지막까지 연속 수행하면서 error시 tip line의 색깔을 바꾼다.

  break : 작동중 일시 stop 기능

  continue : Stop 후 재 진행기능

(6) OPIION

Kinematic과 tool, surface등을 보이게 할 것인지 안 보이게 할 것인지를 정하여 주며 각 segment의 색깔을 지정하여 준다.

- submenu

  KINEMA | TOOL | SURFACE | TRACE

— sub-submenu

ON — 지정한 segment를 보이게 한다.

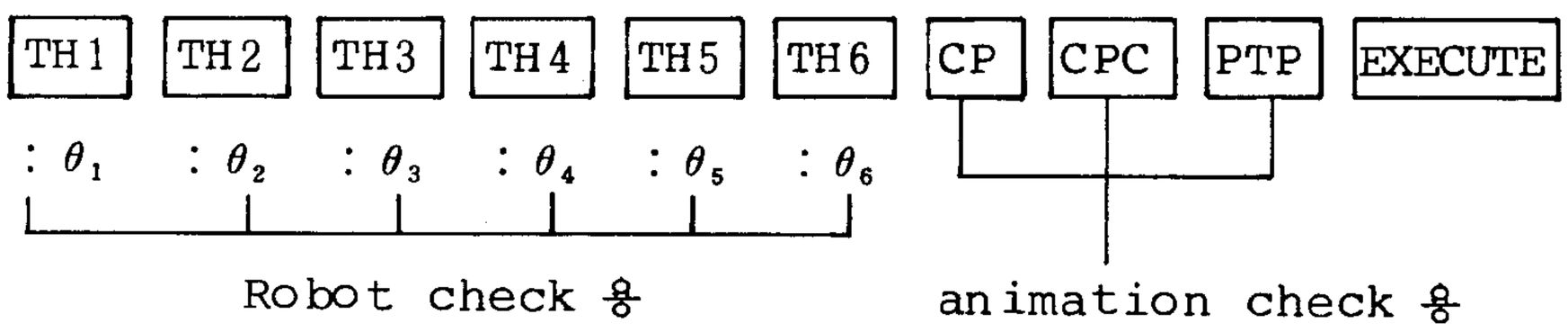
OFF — 지정한 segment를 안보이게 한다.

COLOR — 지정한 segment의 색깔을 바꾸어 준다.

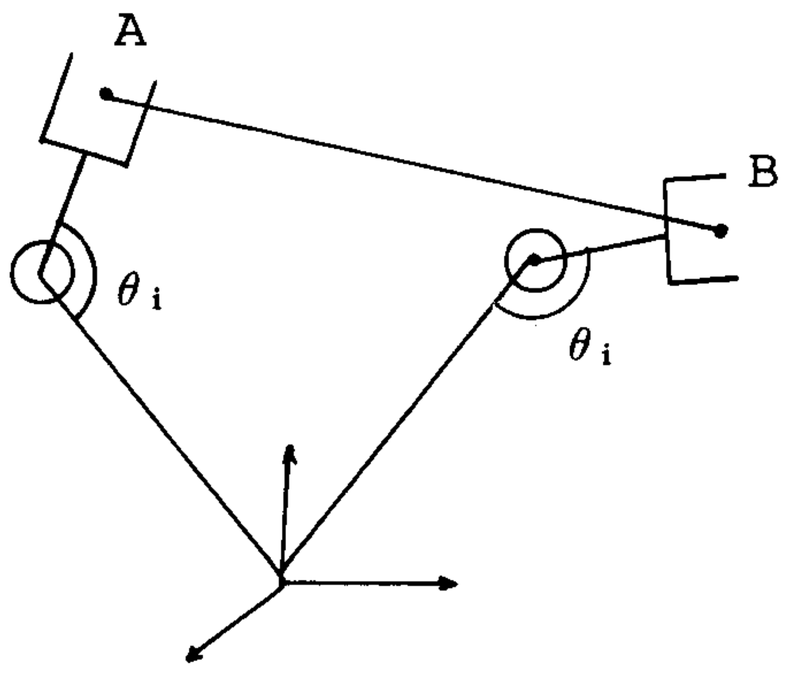
(7) TEST

Kinematic의 각 변수 값에 따른 animation을 Test 해보고 임의의 두 point A,B 사이를 움직일때의 운동형태를 결정하고 Animation 한다. 운동형태에는 CP, CPC, PTP 가 있다.

— submenu

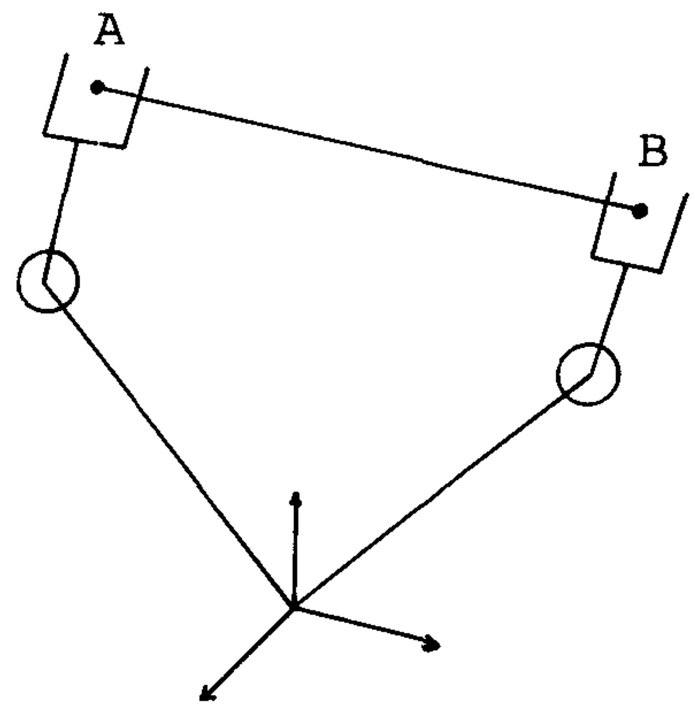


1) CP [continuous path]



: A점에서 B점으로 가는 동안 직선운동을 하며 4,5,6 번째 관절의 variable이 변하지 않는다. 즉 wrist가 고정된채로 움직인다.

2) CPC [CP with constant orientation]

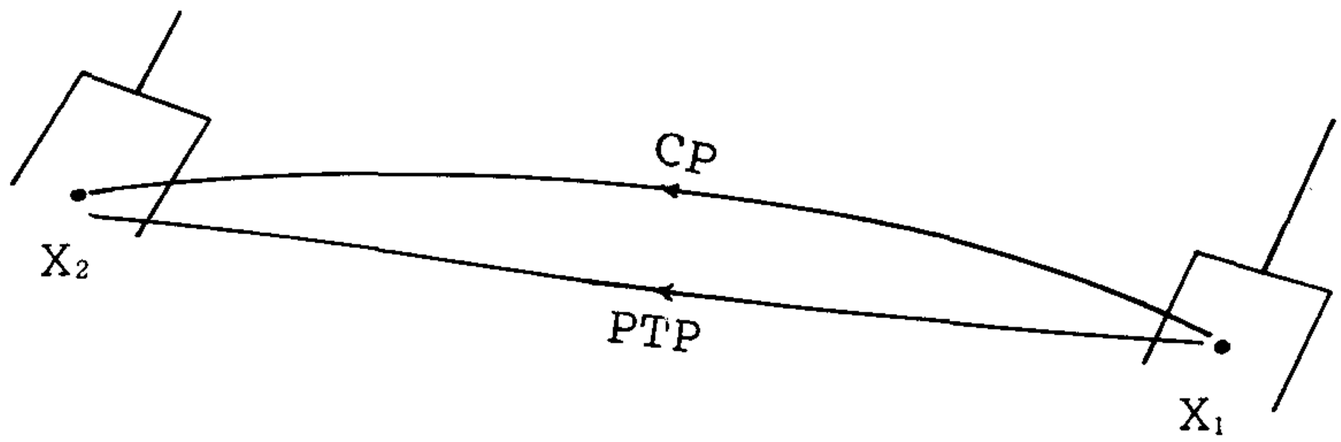


: A점에서 B점으로 가는 동안 직선운동을 하며 Global coordinate system에 대해 일정한 orientation 을 유지한다.

3) PTP [point to point]

START PT

$\underline{X_1} = [X_1, Y_1, Z_1, r_1, p_1, y_1]^T$



END PT

$X_2 = [X_2, Y_2, Z_2, r_2, p_2, y_2]^T$

• PTP 에서 중간 path 는 robot kinematics 의 path 및 configuration 과는 무관하고 tool 의 orientation 과도 무관하다.

X_1 에 대한 $\underline{q}_1$ 값

$$q_1 = [\ \theta_1^1,\ \theta_2^1,\ \cdots\cdots,\ \theta_n^1\ ]^T$$

X_2 에 대한 $\underline{q}_2$ 값

$$q_2 = [\ \theta_1^2,\ \theta_2^2,\ \cdots\cdots,\ \theta_n^2\ ]^T$$

$$\underline{q}_2 - \underline{q}_1 = [\ \triangle\theta_1^{12},\ \triangle\theta_2^{12},\ \cdots\cdots,\ \triangle\theta_n^{12}\ ]^T$$

정의된 기본 increment step 으로 나누면 step 이 결정된다.

$$\frac{q_2 - q_1}{max\ inc} = [\ n_1,\ n_2,\ \cdots,\ n_n\ ]^T$$

이 중에서 가장 큰 값을 step number(SN_r)로 결정한다면 1 step 을 위한 q값의 increment 양 $\triangle q$ 는 다음과 같다.

$$\triangle q = \frac{q_2 - q_1}{SN_r} = [\ \frac{\triangle\theta_1}{SN_r},\ \frac{\triangle\theta_2}{SN_r},\ \cdots\cdots,\ \frac{\triangle\theta_n}{SN_r}\ ]^T$$

4. 결 과

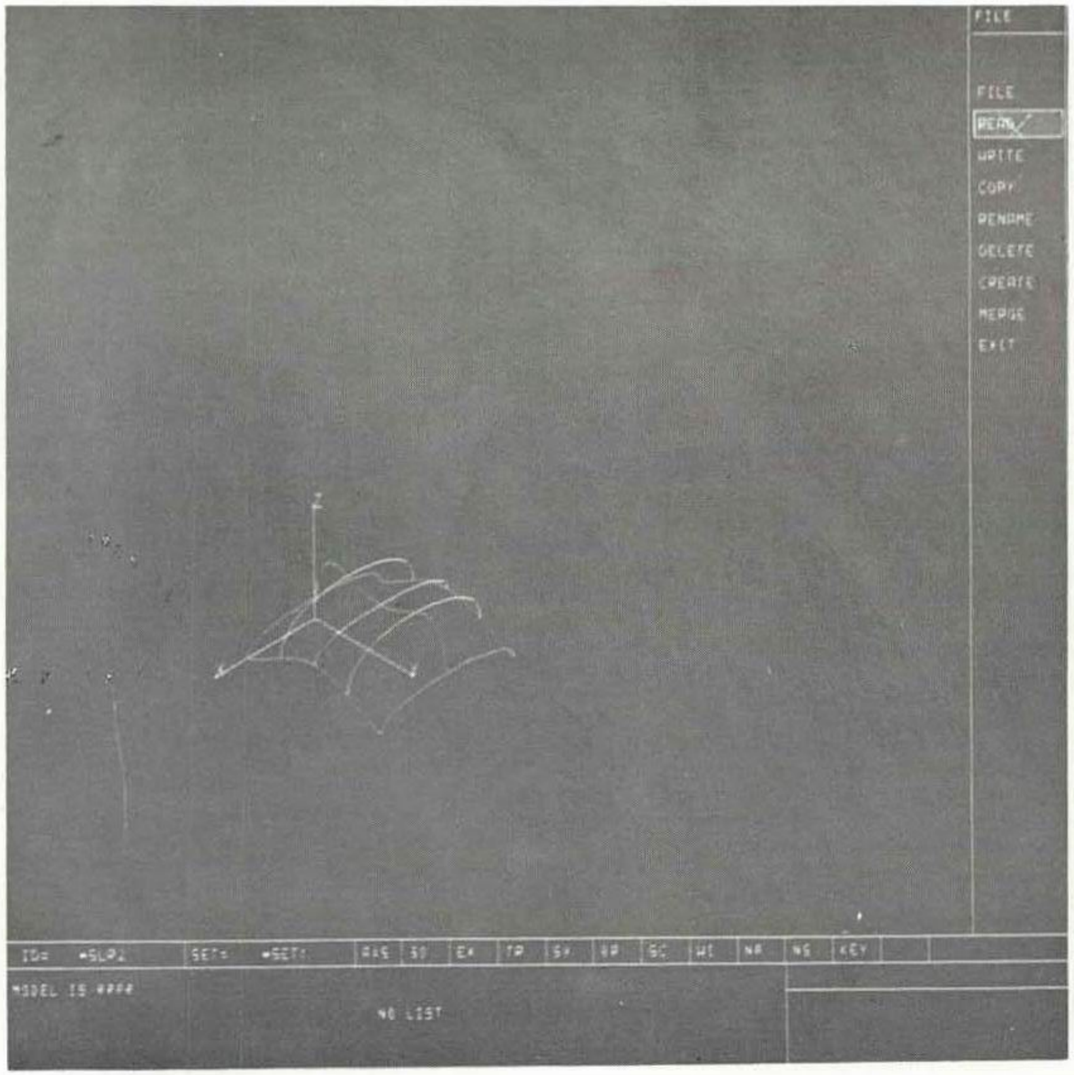


그림 4-1. surface library 에서 surface를 선택

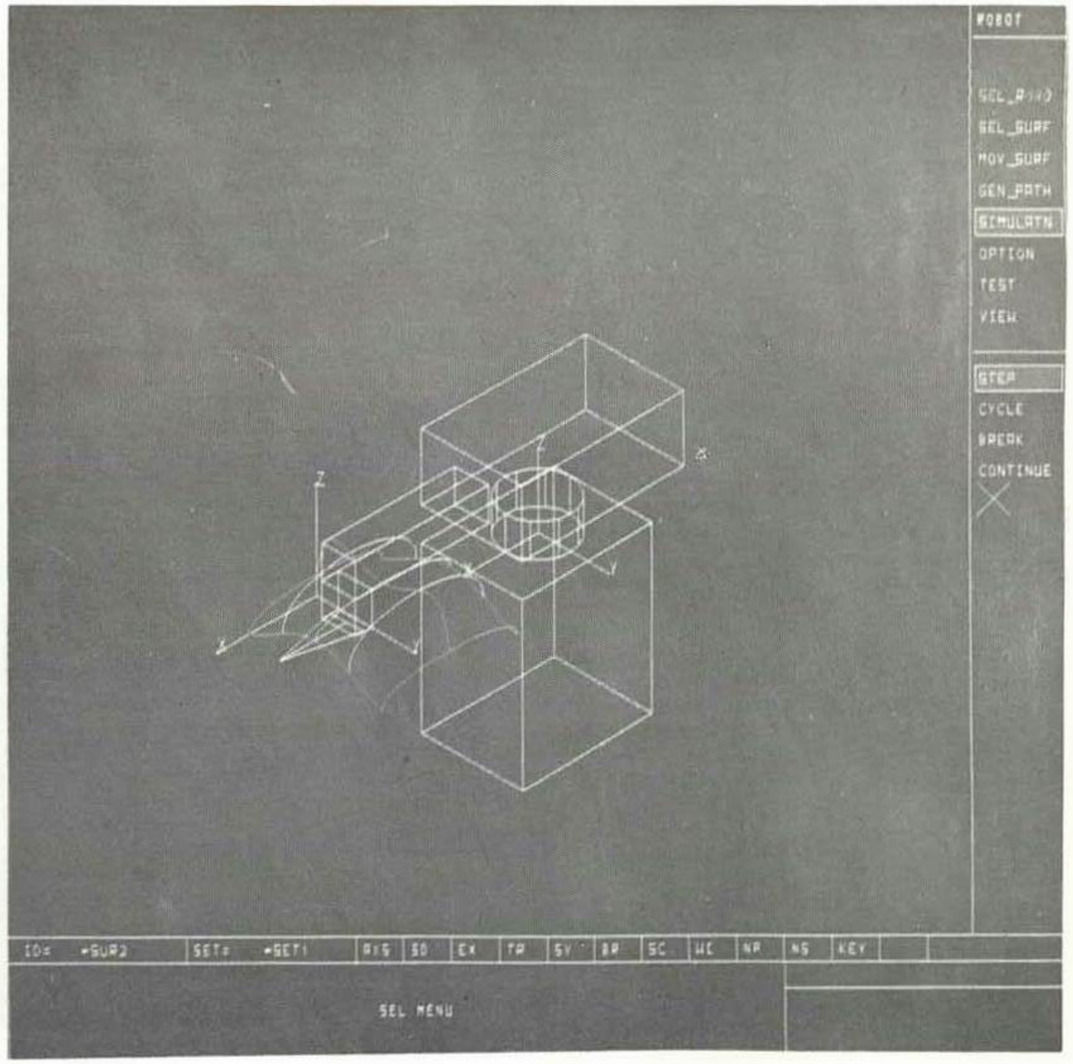


그림 4-2. SEL-ROBO 에서 FANUC Robot 선택

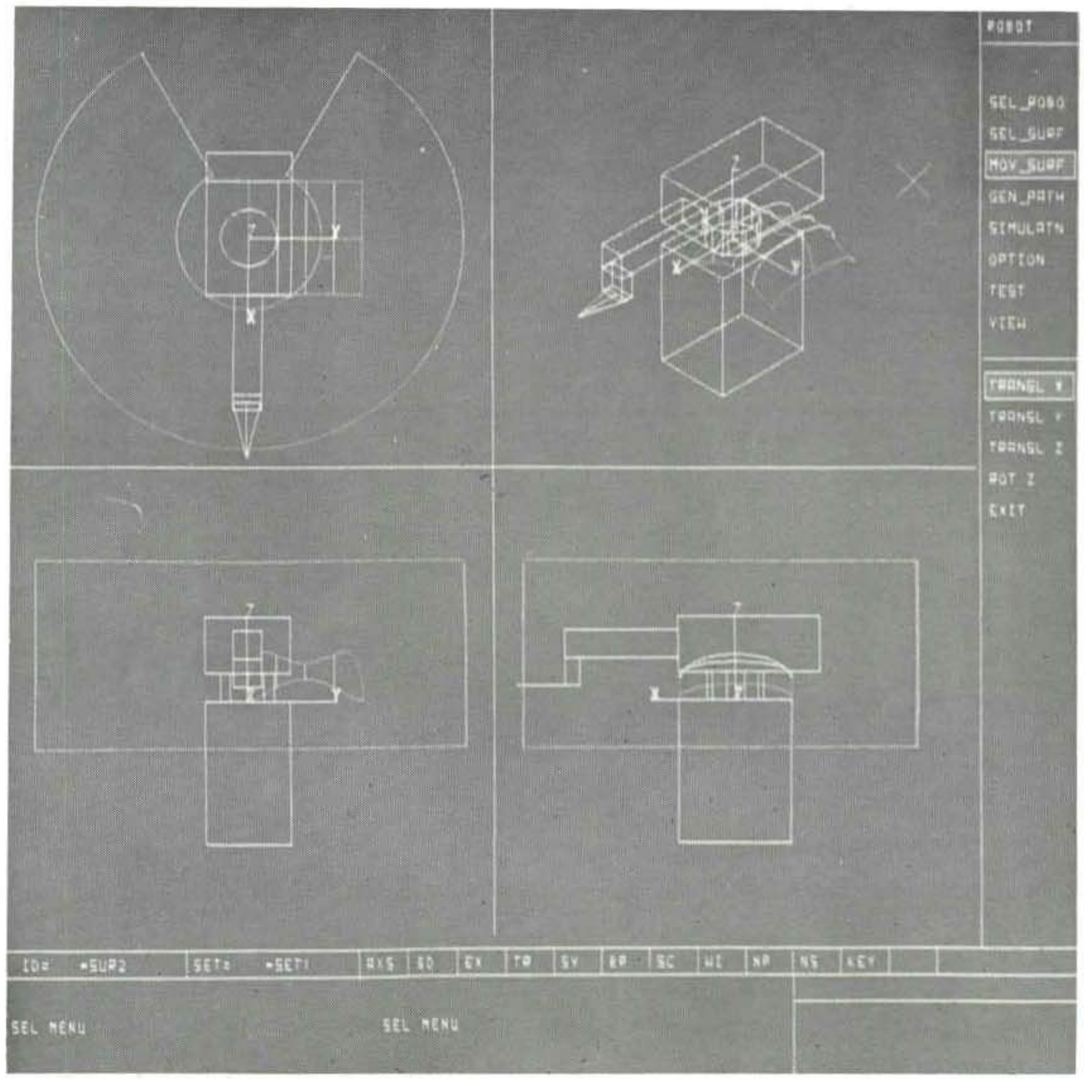


그림4-3. MOV-SURF의 초기모습

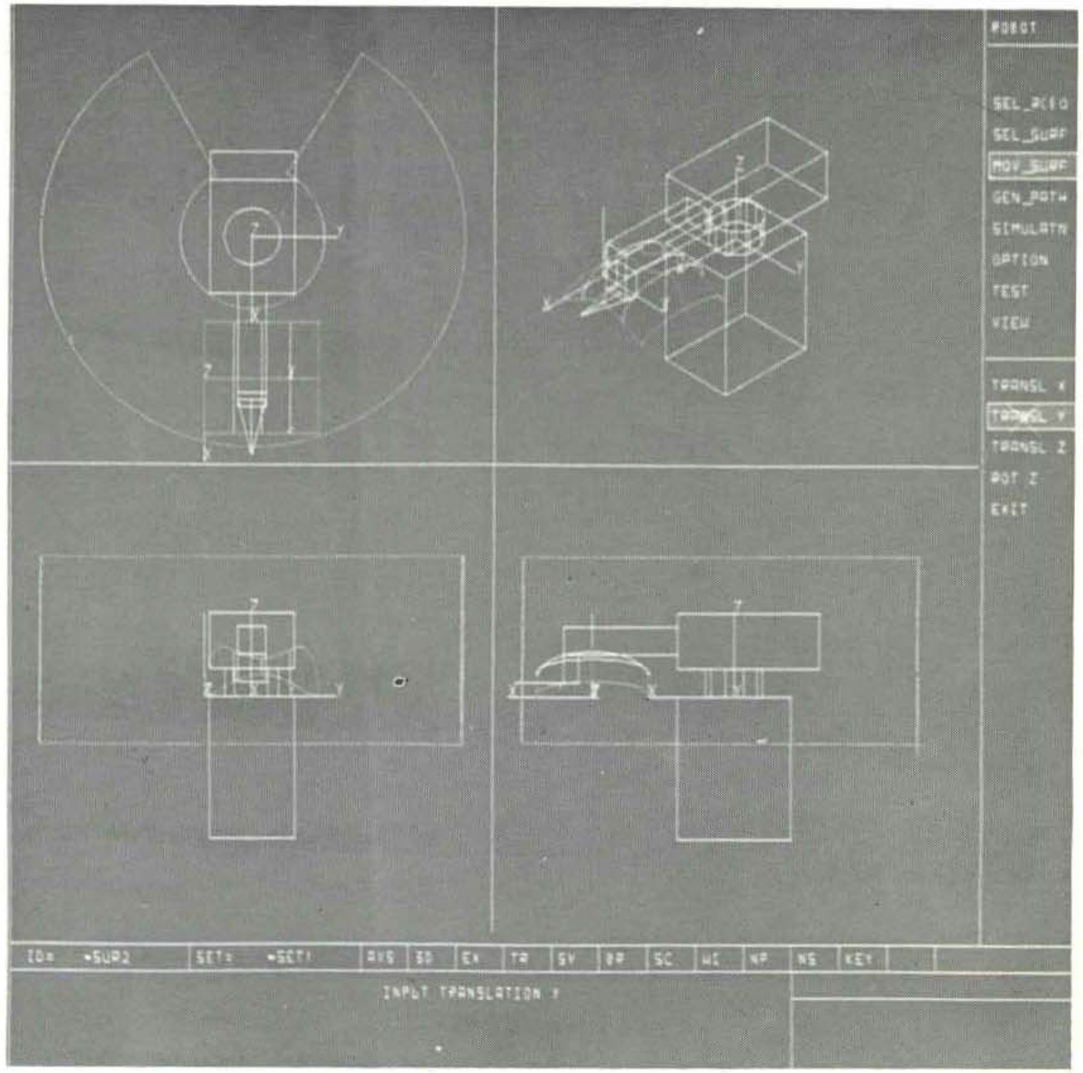


그림4-4. MOV-SURF의 최종모습

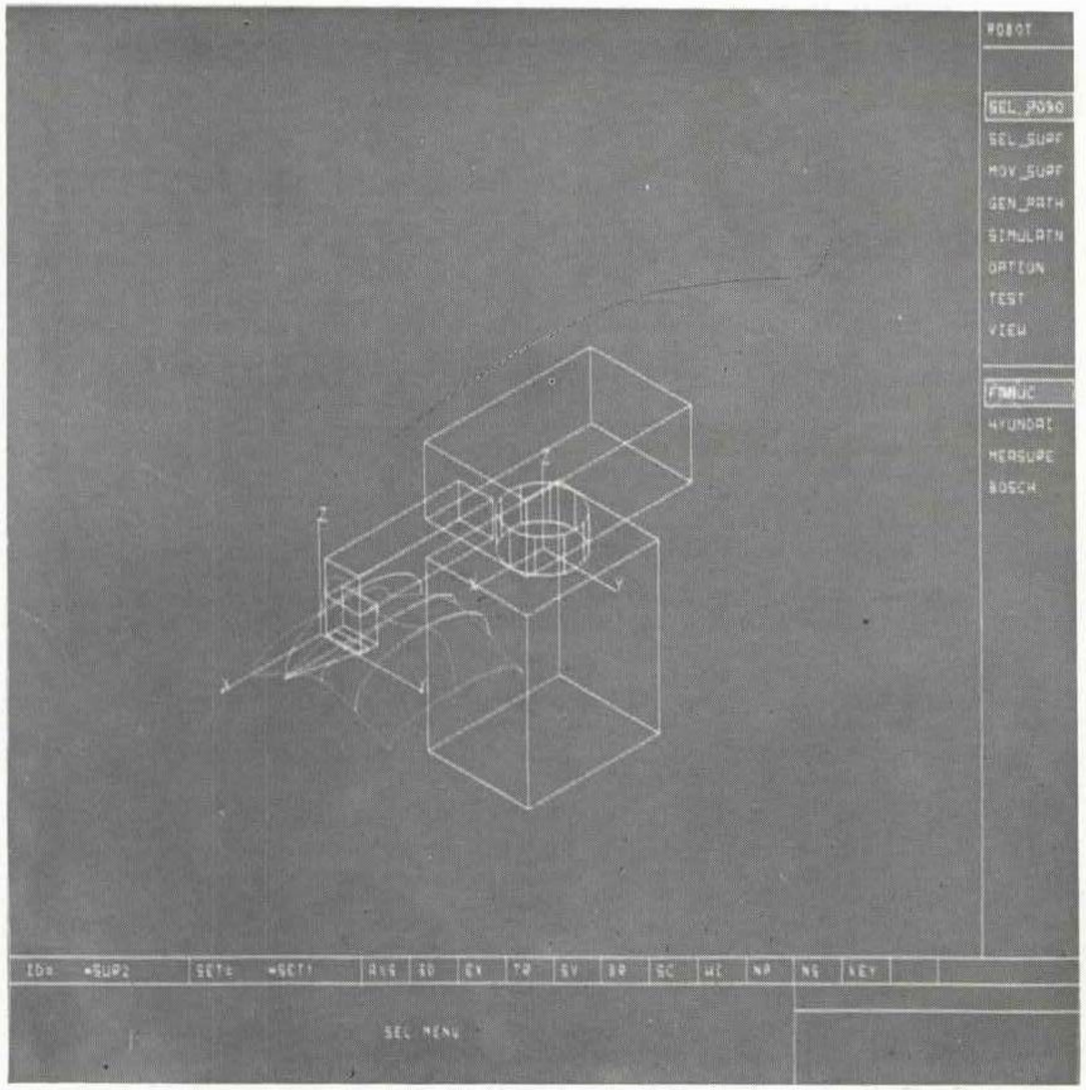


그림 4-5. MOV-SURF 에서 EXIT한 후의 모습

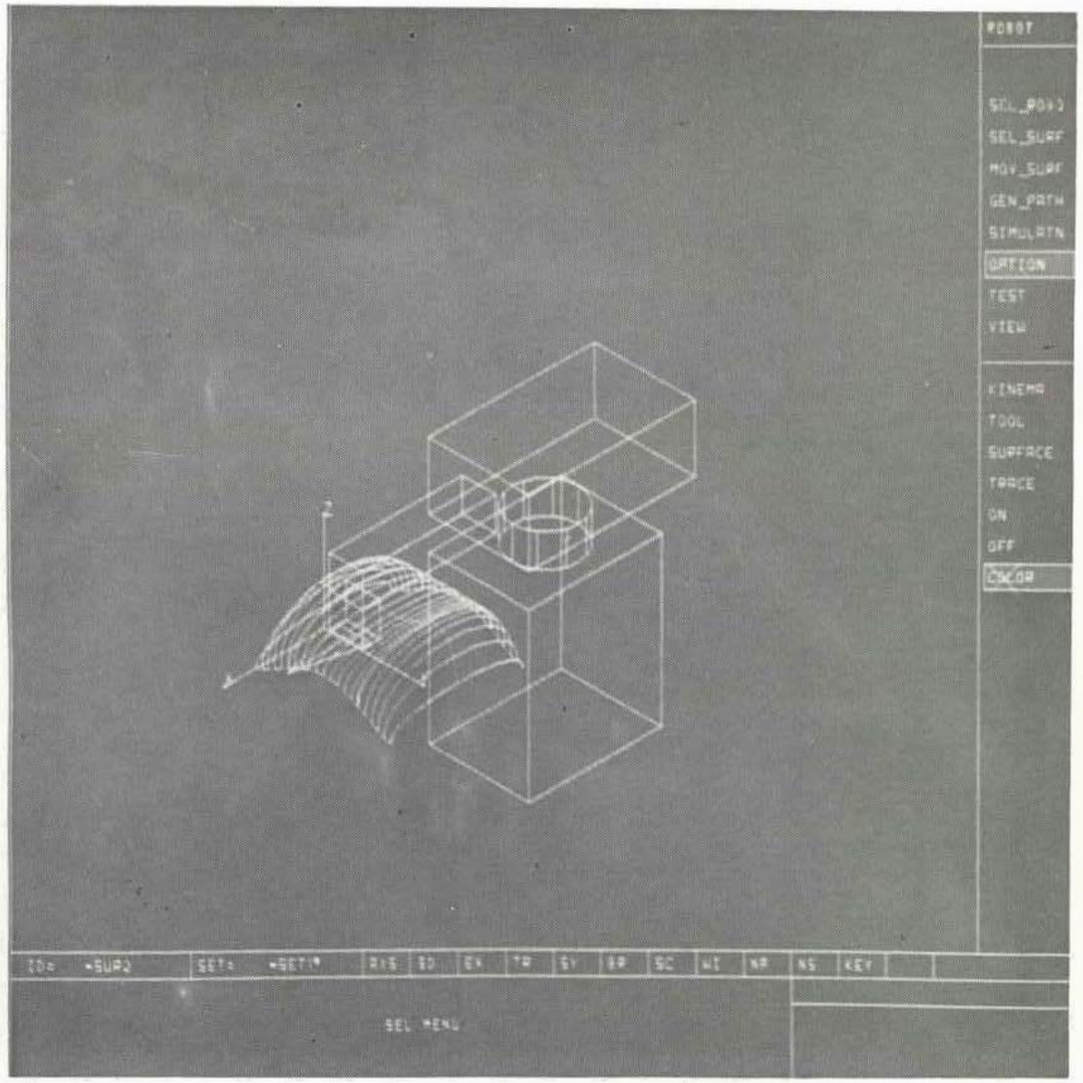


그림4-6. GEN-PATH를 한 후의 모습

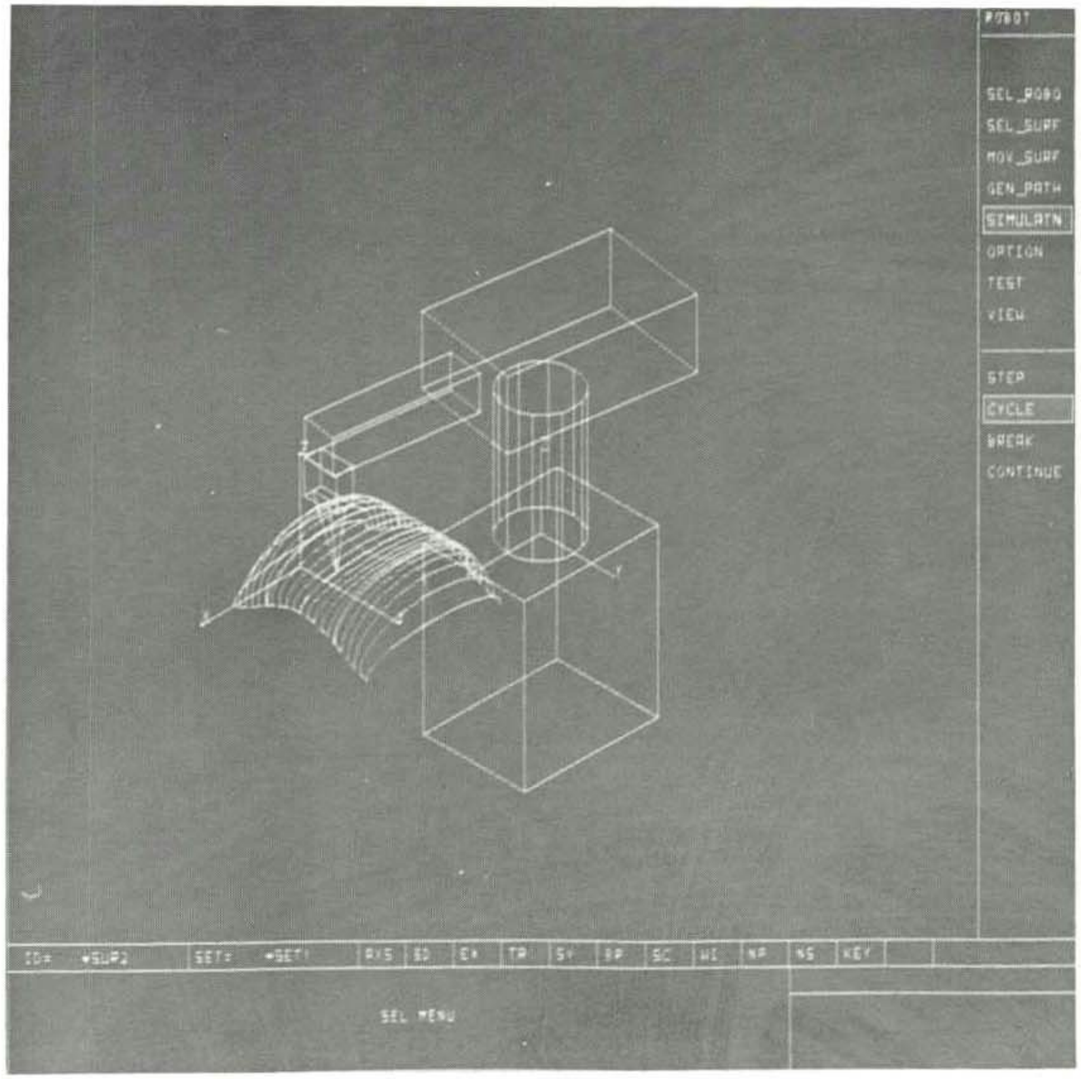


그림 4-7. SIMULATN에서 CYCLE 수행중

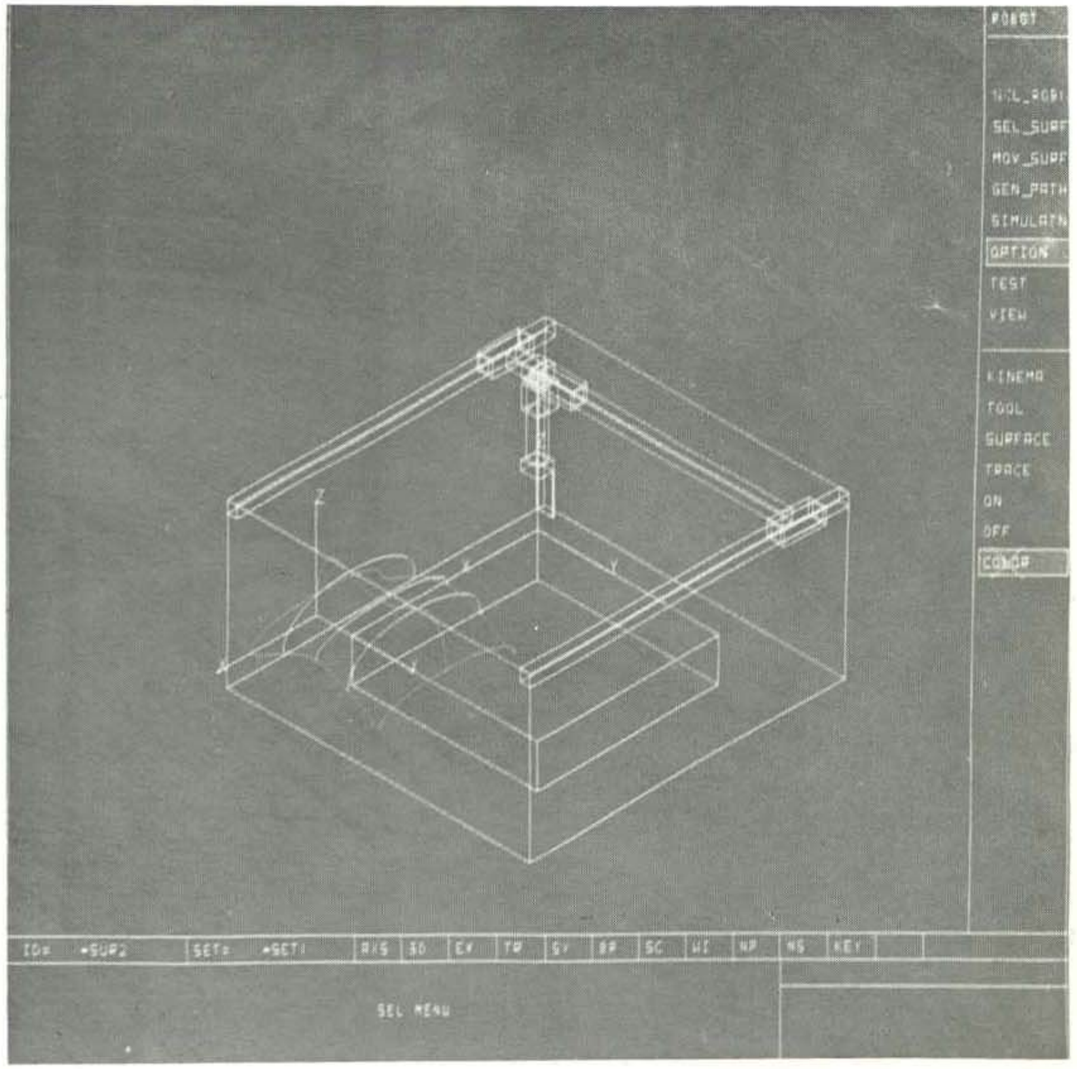


그림 4-8. SEL-ROBO에서 MEASURE 선택

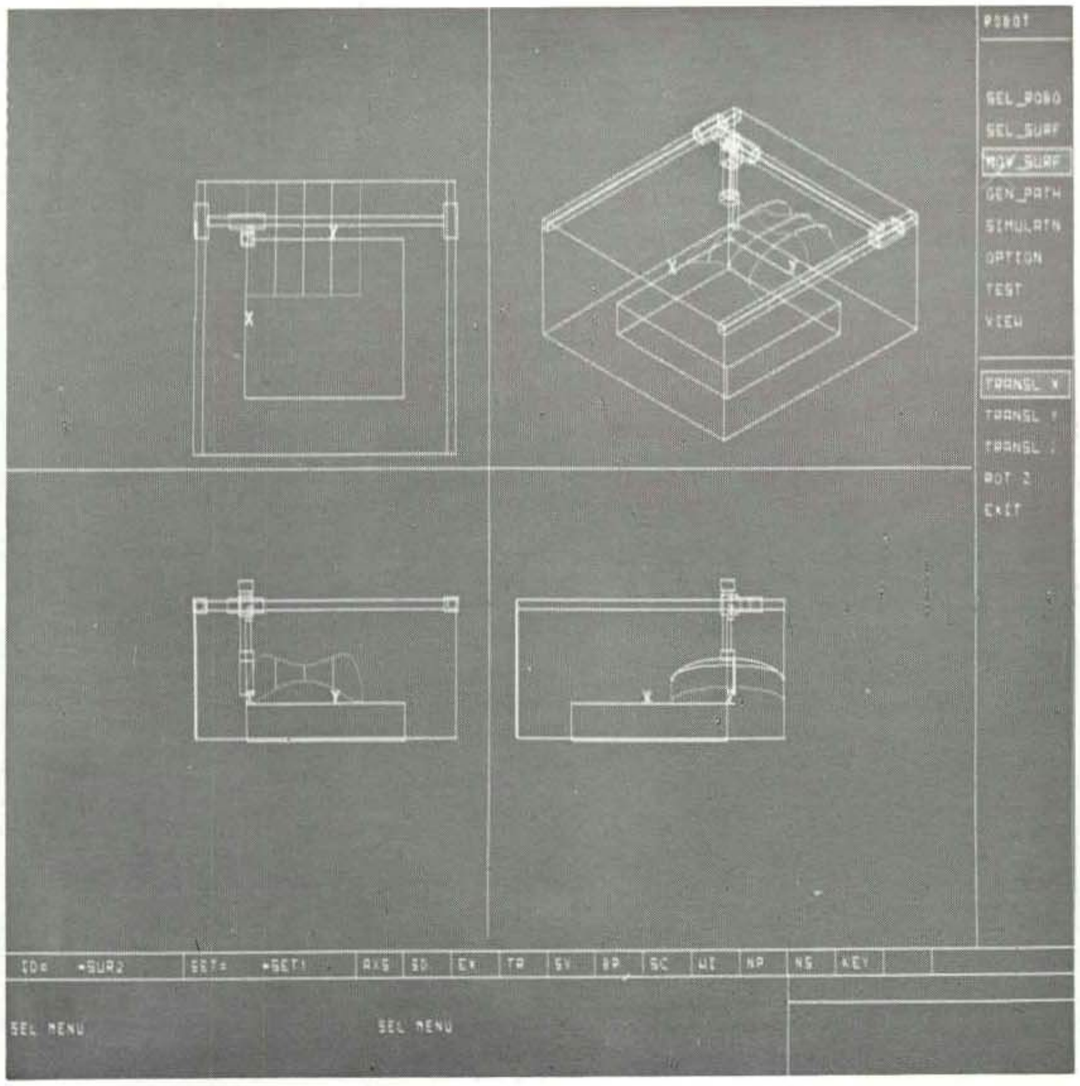


그림4-9. MOV-SURF 초기모습

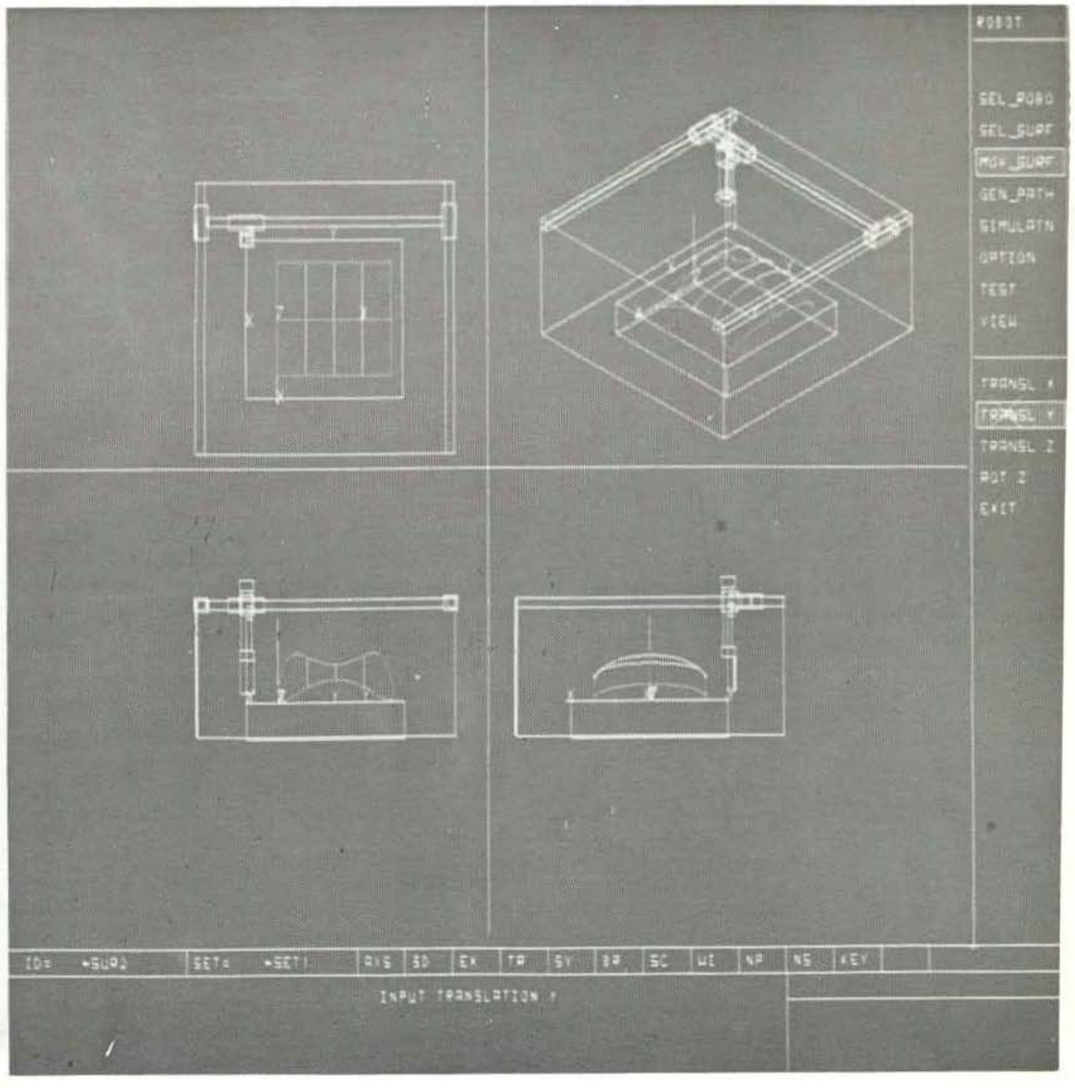


그림4-10. MOV-SURF의 최종모습

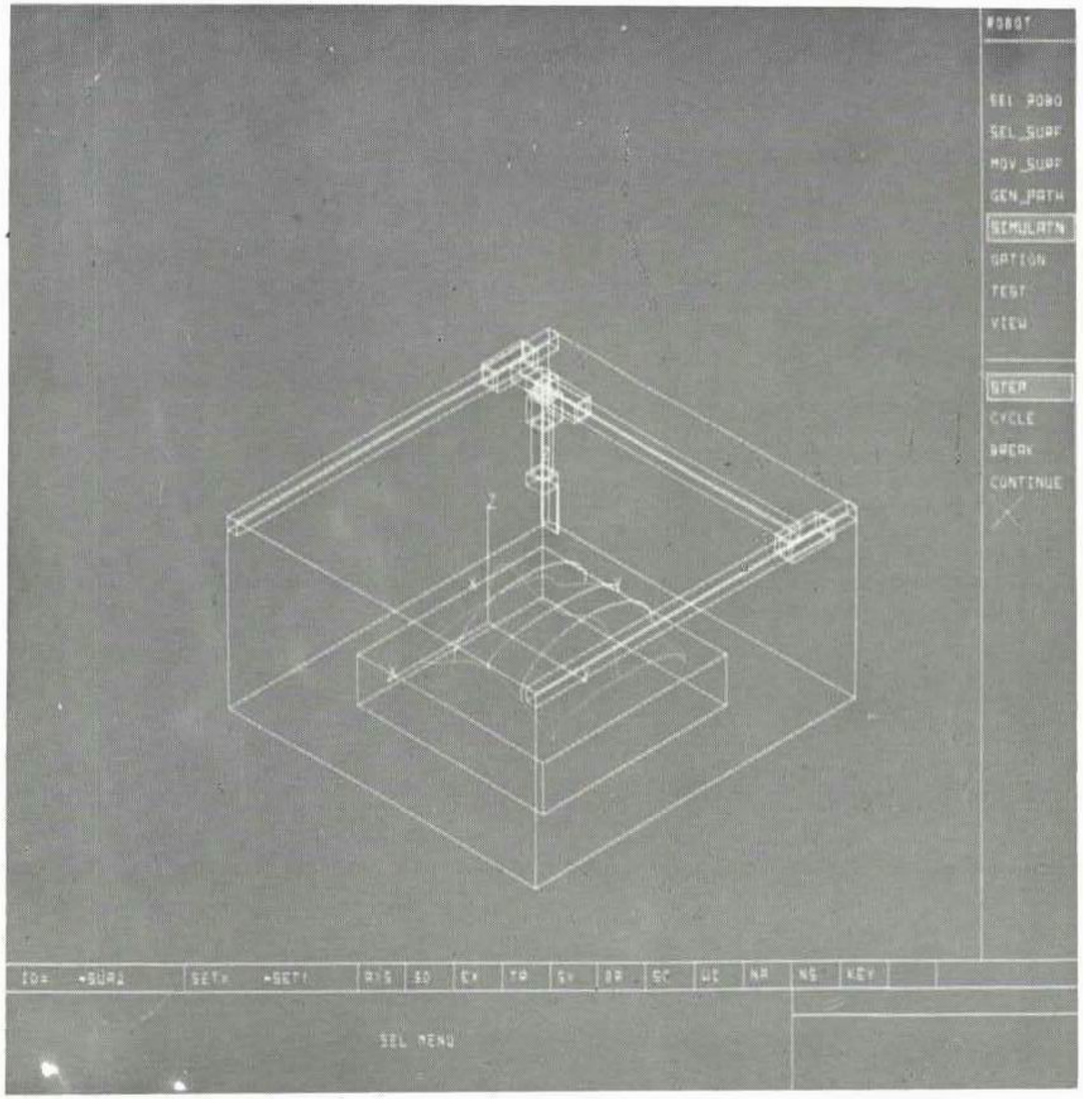


그림 4-11. MOV-SURF에서 나왔을때의 모습

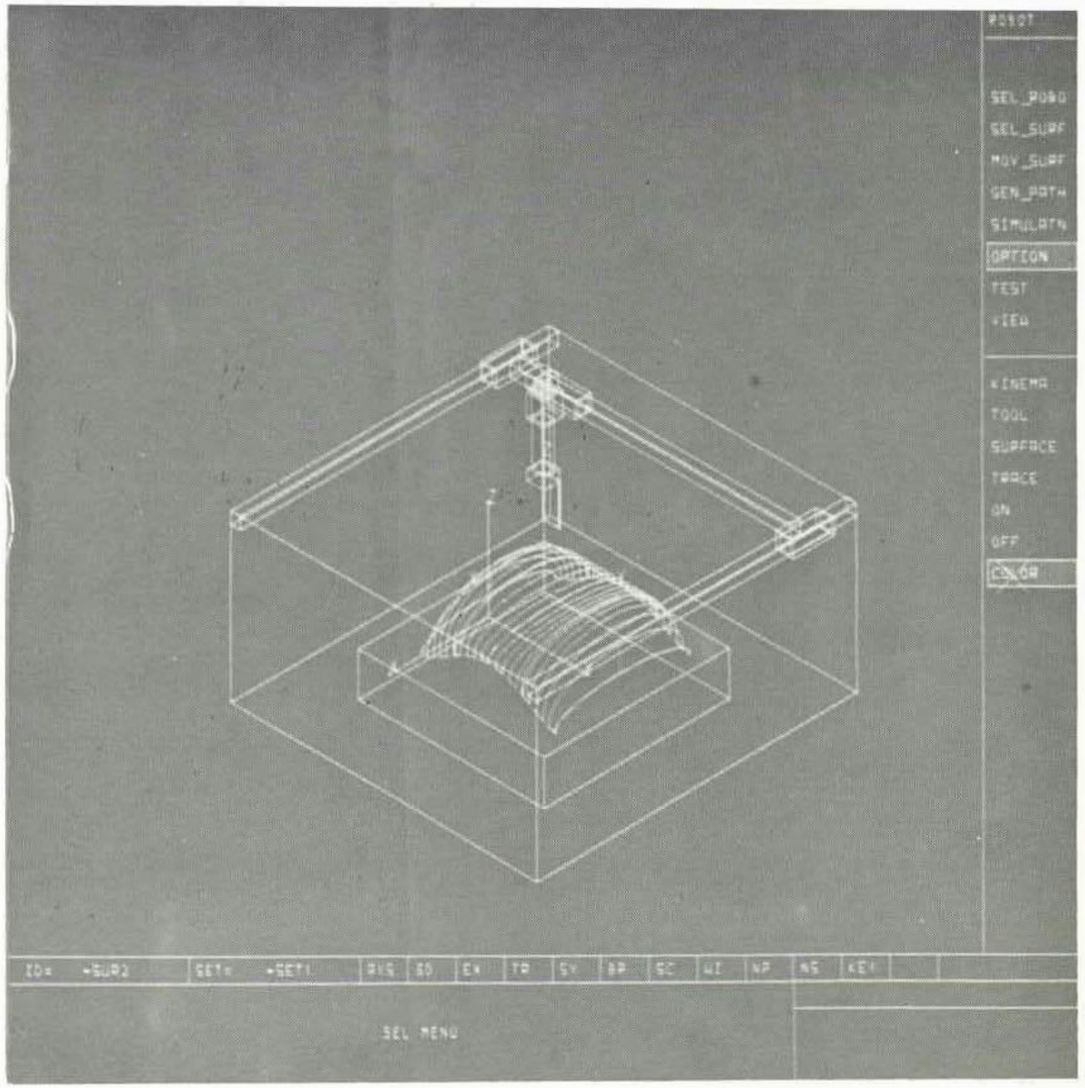


그림 4-12. GEN-PATH를 한후의 모습

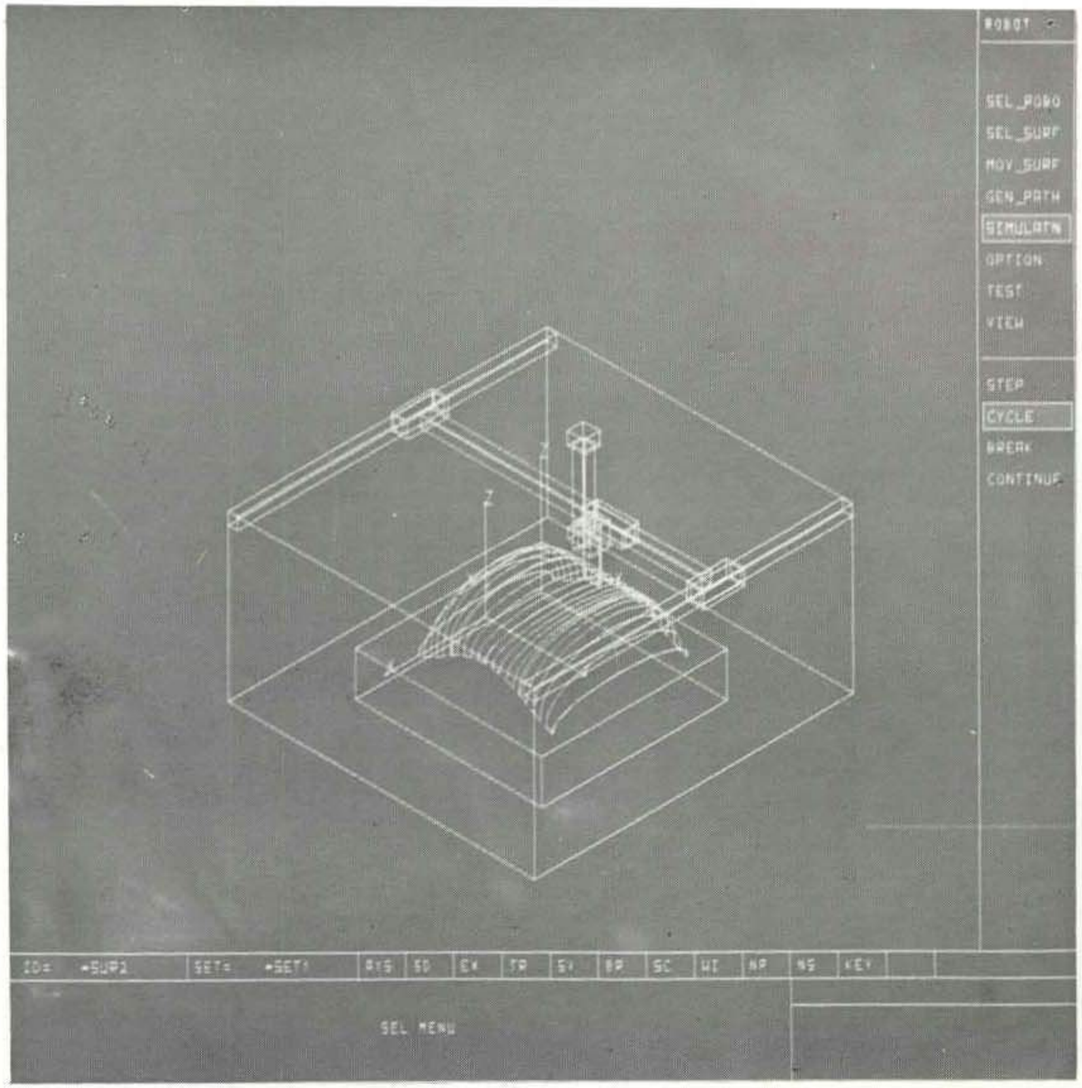


그림 4-13. SIMULATN

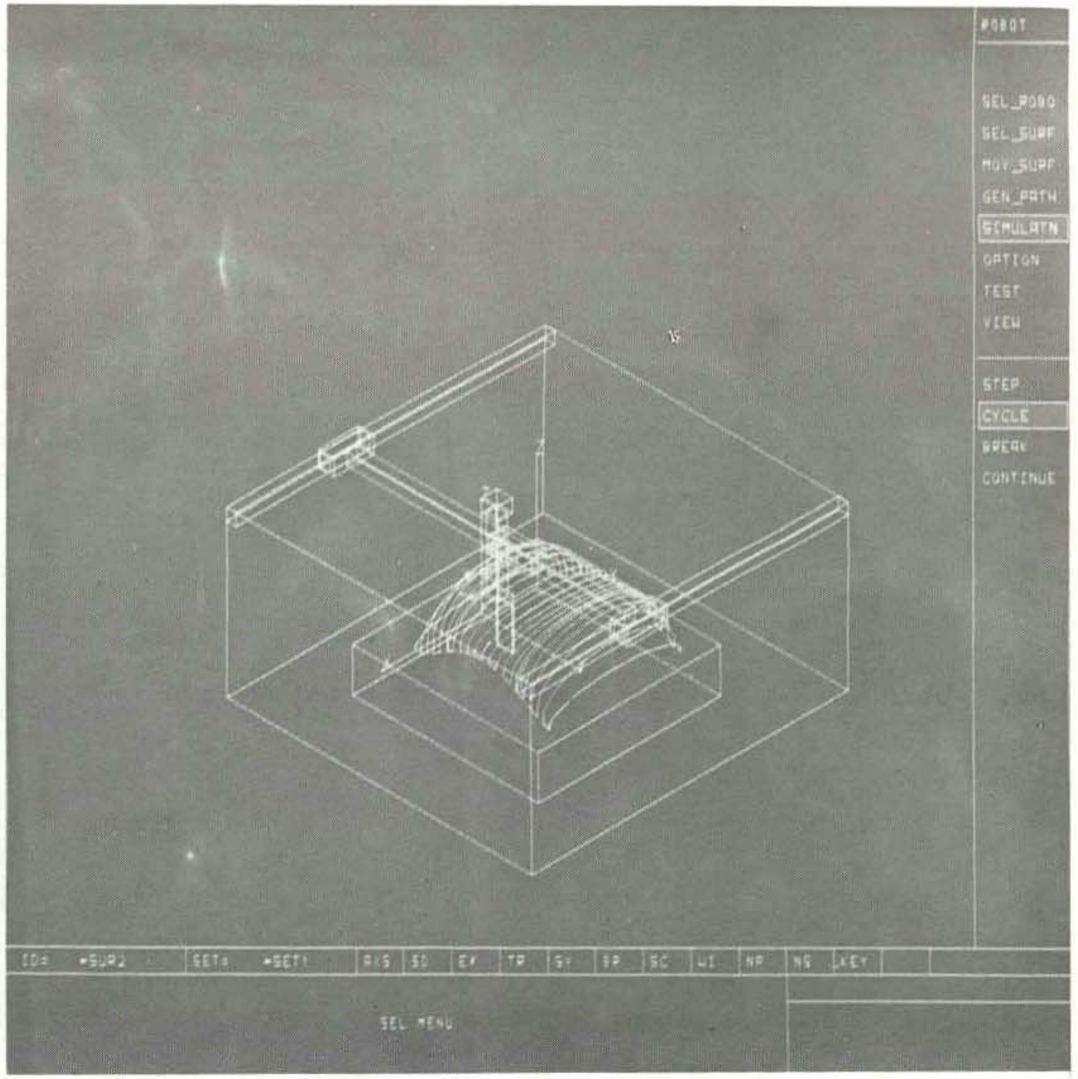


그림 4-14. SIMULATN

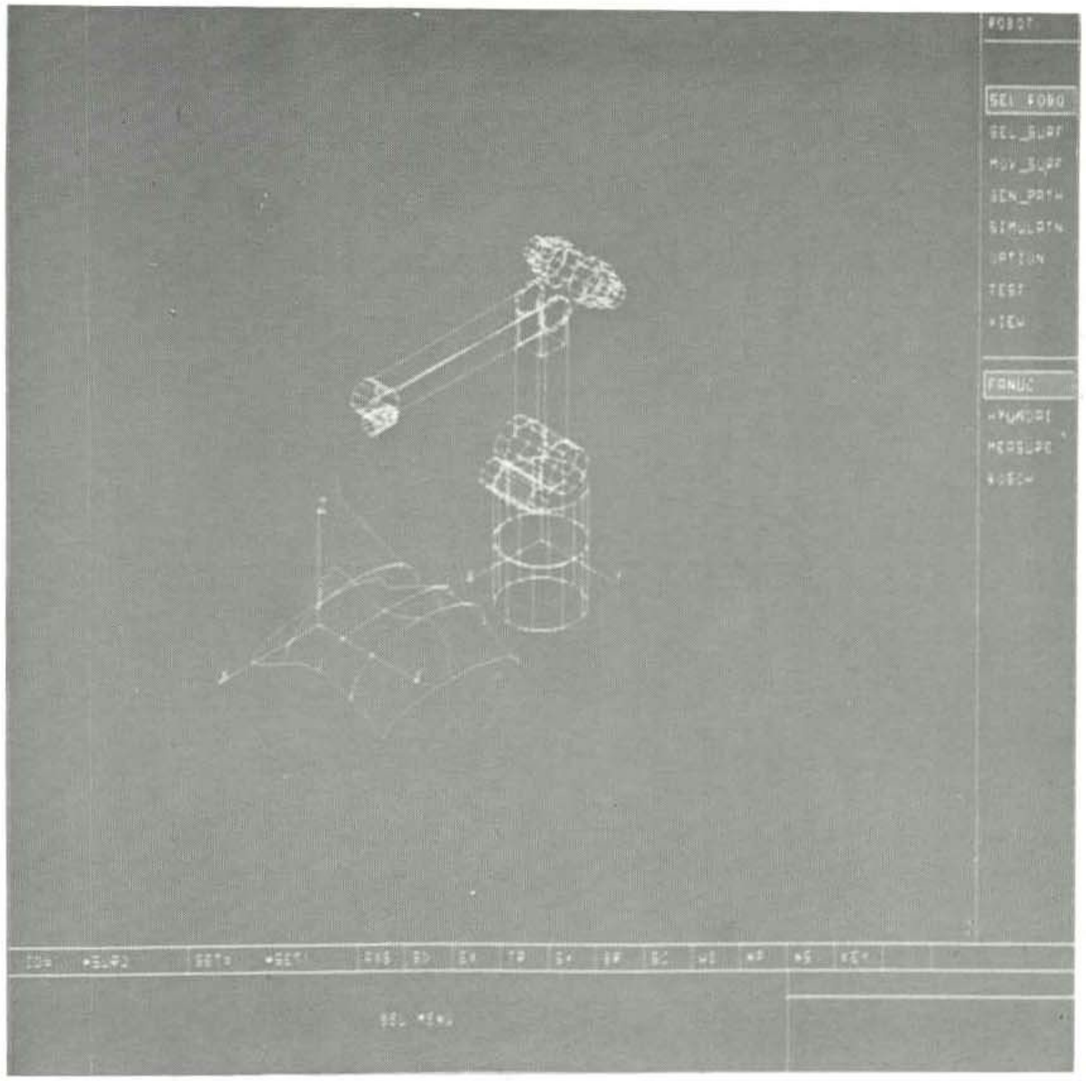


그림 4-15. HYUNDAI ROBOT

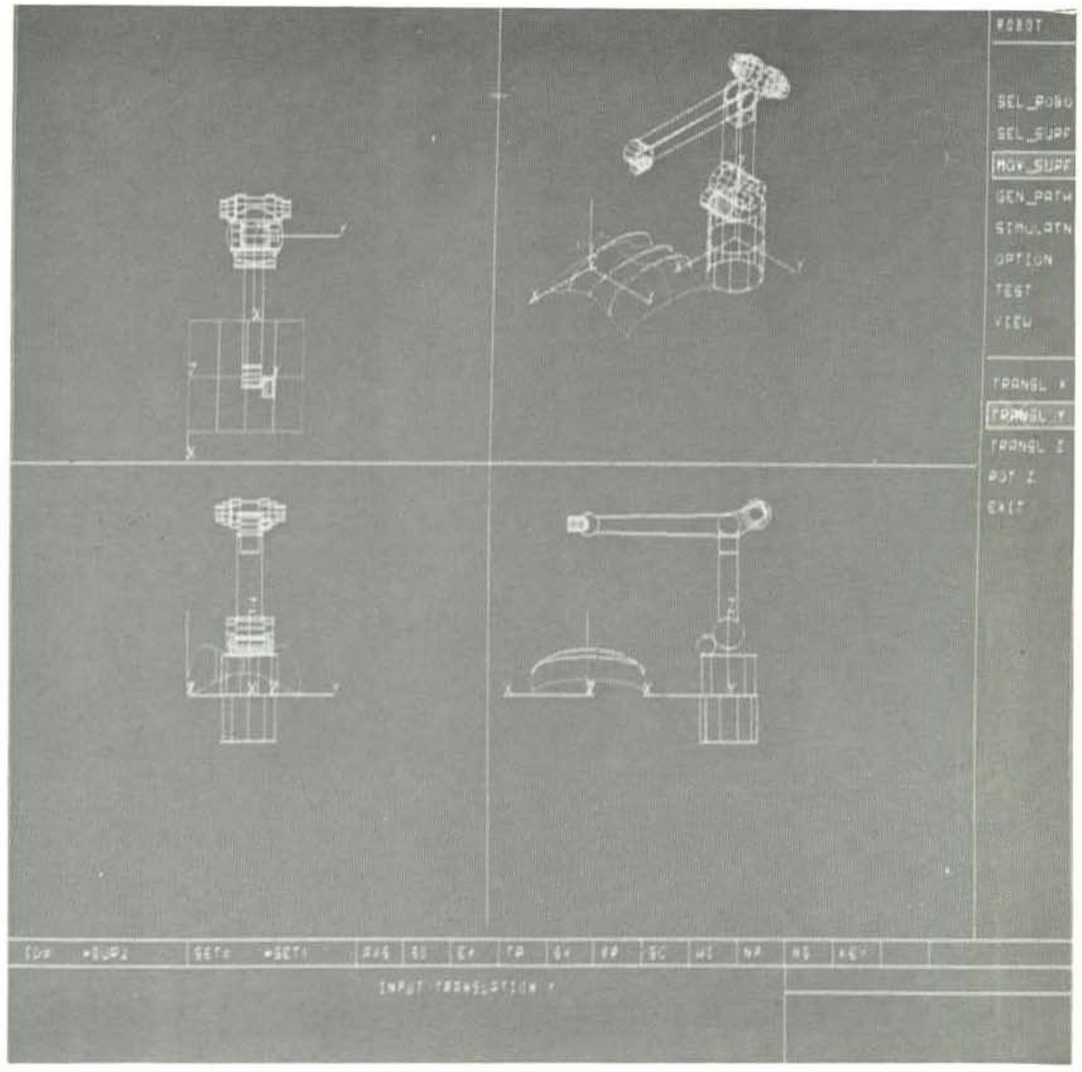
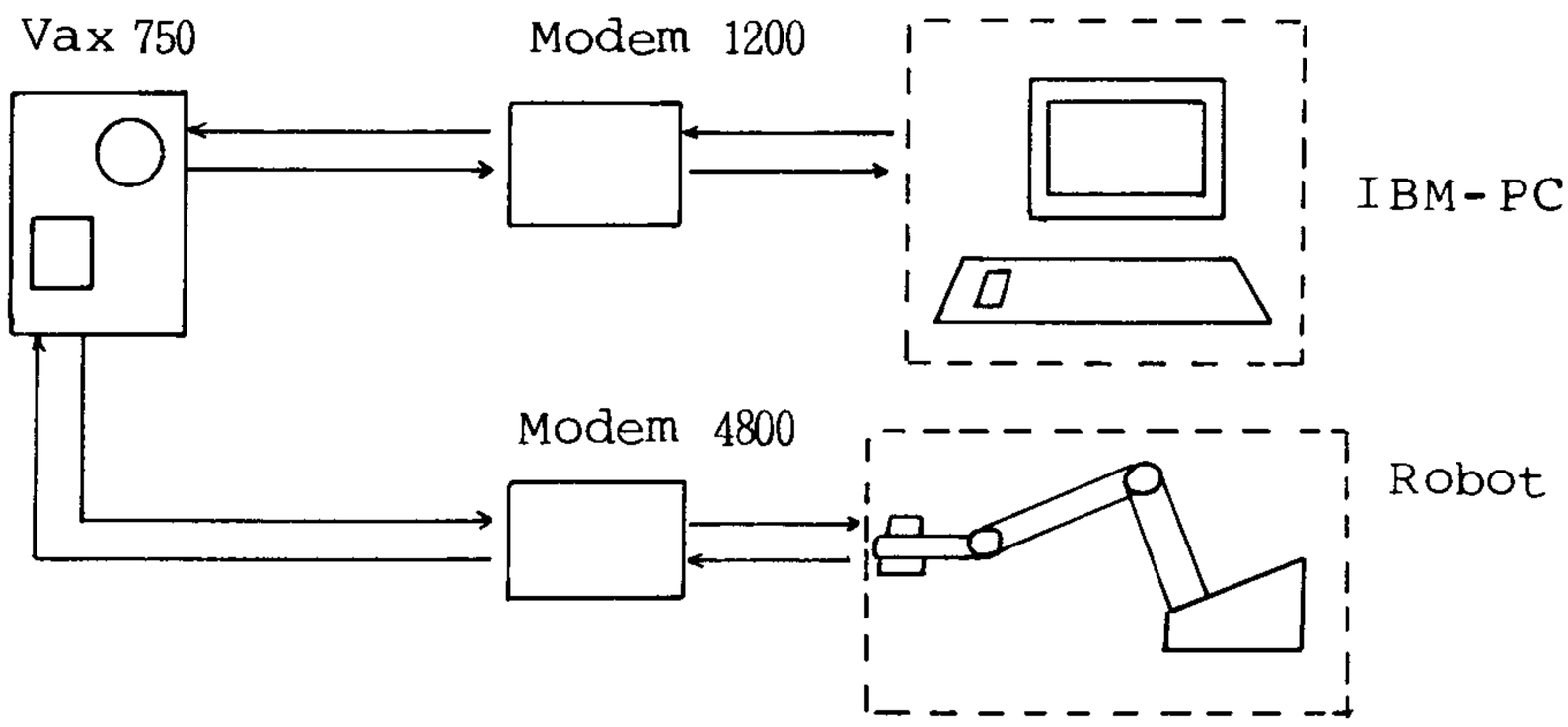


그림 4-16. HYUNDAI ROBOT의 MOV-PATH

제 3 절 Computer 와 Robot 의 Interface

1. VAX 750 과 FANUC Robot 의 Communication
(IBM-PC 를 VAX 750 의 monitor 로 사용)

(1) general



(Vax 750 — IBM-PC
  Vax 750 — Robot

Vax 750 과 Robot 의 communication 이 목적이고 IBM-PC 는 단지 Vax 750 의 Terminal 로 사용(monitor 로 이용)

(2) Modem

Computer system 에서는 remote terminal 간에 digital data 를 주고 받고 하기위해 전화선을 이용한 Modem 을 사용한다.

FANUC Robot 의 transmission spec 상 4800 Modem 을 이

용한다.

사용하고 있는 Modem의 connection 부는 다음과 같다.

1) Modem 1200(IBM-PC)

PW	TD	RD	DS	CO	TS	TST
○	○	○	○	○	○	▭

(view from the front)

PW ; 전원

TD ; transmitted data

RD ; Received data

※ 정상상태에서는 PW, DS, CO에 불이 켜져야 된다.

TST를 눌러 모든 lamp에 불이 들어오면 정상

2) Modem 4800(Robot)

PW	TD	RD	RTS	CTS	DCR	ERR	TST
□	□	□	□	□	□	□	□

(view from the front)

PW ; 전원

TD ; Transmitted data

RD ; Received data

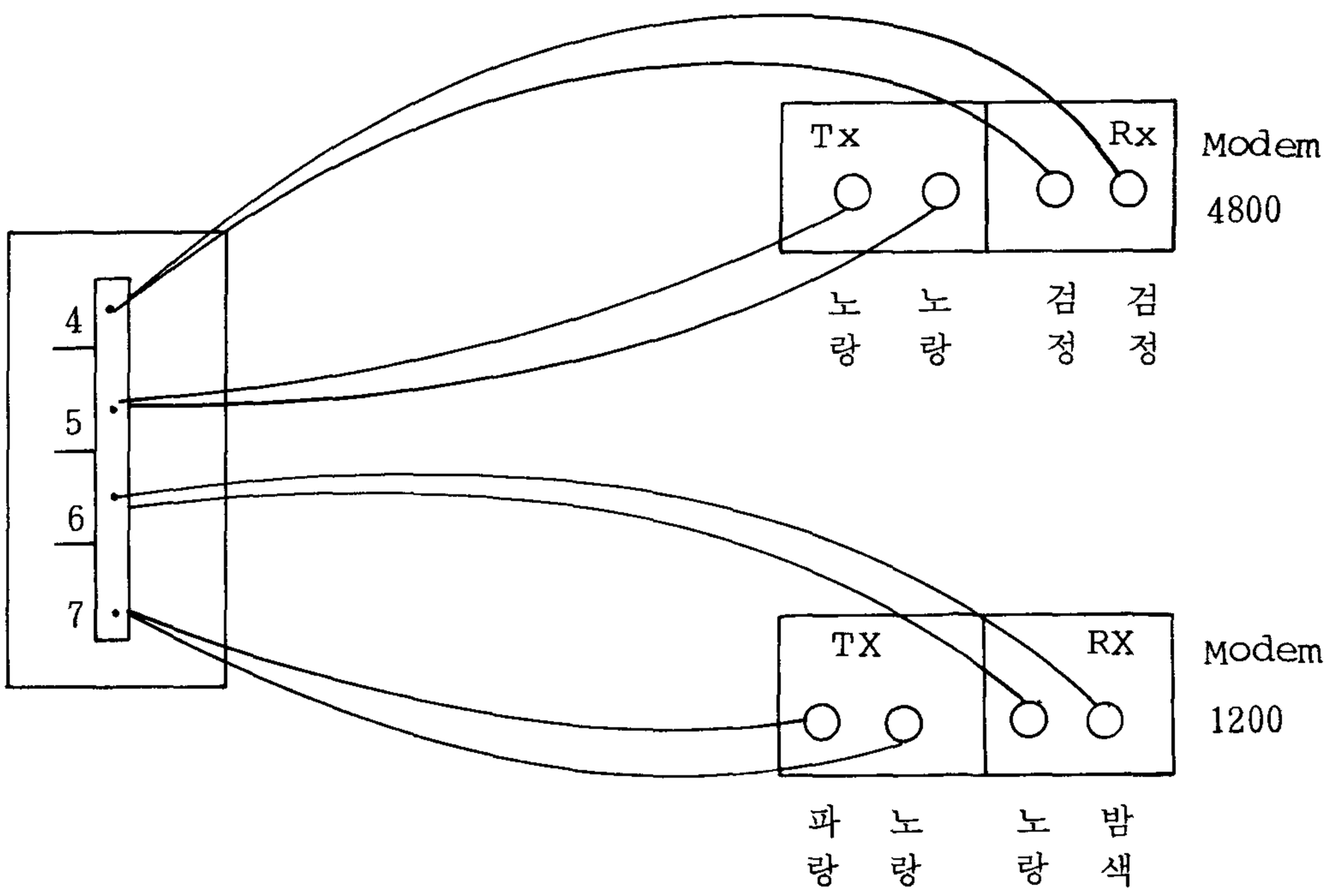
RTS ; Request to send

CTS ; Clear to send

※ 정상상태에서는 PW, RTS, CTS, DCR에 불이 들어와야 한다.

(3) Connection

Vax 750 line이 4, 5, 6, 7번이 들어와 있으면 각 번은 2line으로 되어 있다.



※ 5번 line은 극성이 있으므로 Label대로 설치

(4) Intercommunication program

Communication Program이 필요하며, Pflink directory 내에 IC program이 이것이다.

C > pflink > ic

F9 ↓

port option에서 band rate, character bit수, parity stop bit수, Communication Port를 지정해 준다.

Modem 1200을 사용시는(IBM-PC) 각각 1200, 7bit, odd, 1, com2로 setting 한다.

Modem 4800을 사용시는(Robot) 각각 4800, 8bit, odd 1bit, com2로 setting 한다.

F10 ↓↓

2. RS232c을 이용한 Robot(FANUC)와 IBM-PC 간의 Interface

(1) Procedure

- RS232c의 각핀의 연결(Robot-computer); Robot controller의 CLI port 이용
- Interfacing program(c languge 이용); offline program transfer(Robot → computer)

(2) Interfacing program

1) Robot transmission

- 1 start bit
- 1 stop bit
- 8 bits information
- odd parity
- baud rate-4800

2) INS8250 Registers discription

IBM asynchronous Communications Adapter의 INS

8250 LSI chip은 많은 accessible register가 있는데 이를 이용하여 Robot와 data를 주고 받고 한다.

① Modem control Register ;

Bit0(DTR)=1 ;데이터단말 ready를 on

Bit1(RTS)=1 ;송신요구(PC가 Robot에게)를 on

Bit4(loop) ; 1이면 loopback test를 할 수 있다.

적용 0로 setting

MCR= 0X03

② Modem status register(MSR)

Robot에서 PC로의 control line의 현상태를 보여준다.

③ Line control Register

Asynchronous data comnunication의 format 설정

Bit 0와 Bit 1 ; Bit 0 = 1, & Bit 1 = 1로 setting

하여 8bits serial chracter information

Bit2 = 0 ; stop bit을 1로 setting

Bit3 = 1 ; parity enable

Bit4 = 0 ; odd parity bit로 설정

Bit5 = 0 ; stick parity

Bit6 = 0 ; set break disabled-terminal 상에 ala-

rm 표시

Bit 7 = 0 ; receiver buffer, transmitting hold

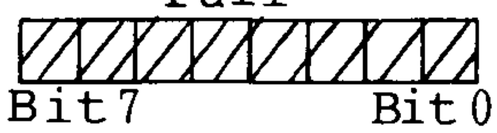
register, interrupt enable Register

access

∴ LCR ← OXOB

④ Line Status Register

data 전송의 status information을 제공하여 준다.

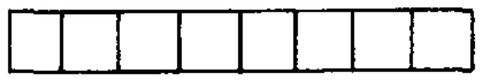
Bit 0 ; 

Bit0(Data Ready)=1 이면 complete incoming character가 전송됨을 알림

Bit 0 = 1 일때 읽어야 한다.

Bit1 ; overrun error

Bit2 ; parity error-LCR의 Bit5=1로하면 parity error가 생김.

Bit5와 Bit6  Transmitter Holding Register empty

Transmitter Shift register empty

각 Register가 empty이면 1로 setting 되는데, 이때 information을 보낼 수 있다.

Bit5 = 1, & Bit6 = 1 일때 information을 보내야한다.

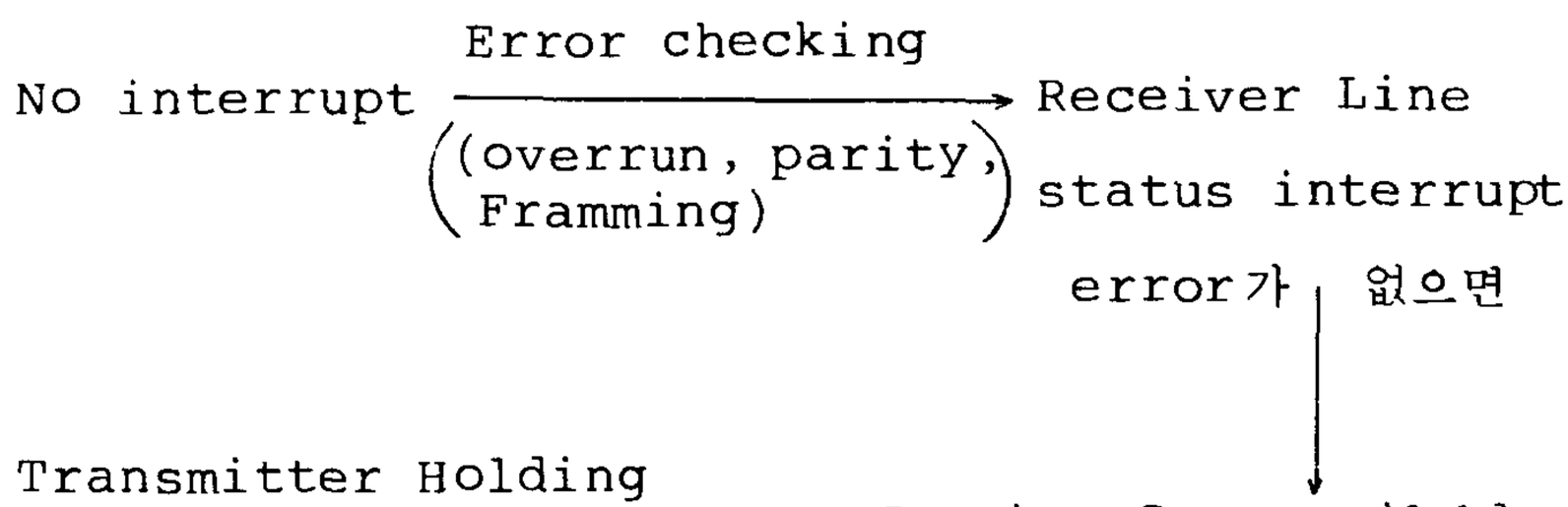
⑤ Interrupt Enable Register(IER)

INS 8250의 4종류의 interrupt(Data available interrupt, Transmitter Holding Register Empty, Receive line status, Modem status)을 enable시킨다

∴ IER=OXOF로 setting

⑥ Interrupt Identification Register

각 Interrupt mode(4가지중)를 실행하기 위해서 chip-select time 동안에 addressed 되어 priority에 의해 수행된다.



⑦ Buffer Register

received, transmitter character를 실어 놓는다.

※ 각 Status Register는 read하면 Setting 된다.

3) Interfacing program

filename ; RS. C(Lattice C)

compiler ; CLS RS

linker ; links RS

3. Robot와 PC와의 communication

(1) 프로그램 개요

Program SEND2.C는 Robot와 PC간의 communication을 위한 것으로 Robot에서 program을 receive하는 GetRho( ) 부분과 PC에서 Robot에 transmit 하는 Send Rho( ) 부분으로 구성되어 있다.

program name : SEND2.C

programming language : C

compiler : msc

(2) INCLUDE FILES

STDIO.H : msc에서 제공하는 file. file i/o에 관여

STDLIB.H : msc에서 제공하는 file. dos command 사용

declare.H : SEND2.C의 defined constants, macros

(3) VARIABLES

job	int	menu No.

(4) SUBROUTINES

GetRho( ) : Robot에서 보내는 program을 받아 file에 저장

SendRho( ): PC에서 Robot로 program을 보내는 routine

SendACK( ) : TCC(02)를 보낸다.

GetACK( ) : TCC(02)를 받는다.

SendCode( ):offline transfer code, TCC(82)를 보낸다

(5) ALGORITHM

1. menu를 선택한다. menu = 1 이면 2., 아니면 3.

2. GetRho

1) file을 open한다.

2) program send code, TCC(80)을 받는다.

3) ACK, TCC(02)를 보낸다.

4) program No.와 point No.를 받는다.

5) ACK,TCC(02)를 보낸다.

6) 한 block씩 받아들여 file에 저장한다.

7) return

3] SendRho

1) 기존의 file을 open한다.

2) file의 내용을 block에 대입한다.

3) program reception code, TCC(81)을 받는다.

4) ACK를 보낸다.

5) Get program No.

6) offline transfer code, TCC(82)를 보낸다.

7) Get ACK

8) file length를 보낸다.

9) Get ACK

10) block을 보낸다.(23bytes씩)

11) retun

4] END

4. Sensor와 Robot와의 communication

(1) 프로그램 개요

Program SENSOR.C는 Robot와 Sensor의 data transfer를 위한 program으로, sensing되어 들어온 offsetdata를 Robot에 전송하여 위치보정을 하도록 한다.

1) program name : SENOR.C

2) programming language : C

3) compiler : msc

(2) INCLUDE FILES

STDIO.H : msc에서 제공하는 file. file i/o에 사용

STDLIB.H : msc에서 제공하는 file. dos command 사용

rspatch.C : analog input을 위한 file

(3) VARIABLES

j	long	error counter
i, cnt, k, kk	int	counters
num	int	G46 code 갯수
data[3]	char	informing G46(TCC AO)

(4) SUBROUTINES

GetCode( ) : Robot에서 보내는 TCC code를 받는다.

int Send Offset(int) : Robot에 offset data를 보낸다.

Offset( ) : sensing한 값을 얻는다.

(5) ALGORITHM

1) Sensing point 갯수를 입력한다.

2) 갯수만큼 아래과정을 반복한다.

3) Informing G46, TCC(AO)를 Robot에서 받는다.

4) Informing the kind of sensor, TCC(B8)을 보낸다.

5) offset data request, TCC(A4)를 받는다.

6) sensing한 offset data를 보낸다.

7) end of transfer, TCC(AF)를 받는다

8) 결과를 화면에 display한다.

9) END

제 3 장 결 론

가. CAD system과 PROPS를 접속하여 CAD system에서 design된 surface를 그대로 가져와 사용할 수가 있으며 Robot kinematics를 graphic library 화하여 surface 배치 수정 및 path generation 및 Animation을 통하여 가공작업을 위한 로보트 운동을 simulation 할 수 있게 되었다.

나. Denavit-Hartenberg Transformation form 에 의해 여러 Robot Kinematic을 일반적인 형식으로 library화 하였다.

다. 금형 가공의 공정들을 menu로 만들어서 expert system을도입 손쉽게 interactive한 작업을 할 수 있게 하였다.

라. Robot과 PC, Robot과 VAX 와의 interface가 이루어졌다.

마. 차후의 연구 목표는 로보트 calibration S/W의 개발 및 실현 그리고 expert system을 이용한 Robot Program Generator의 완성을 통한 전체 off-line programming system을 정립하는데 있다.

이를 위해서 더 실제적인 tool path generation 과 expert system을 이용한 가공조건의 결정 및 user interface를 위한 window가 개발되어야 한다.

Tool path generation 과정에는 tool의 교환과정 및 robot과 surface, tool과 surface, robot과 tool과의 collision check 과정이 포함될 것이다.

그리고 여러개의 patch로 구성된 surface의 경우 optimal한 tool path를 generation하는 과정도 고려되어야 한다.