

〈논 문〉 SAE NO. 953793

로봇 프로그래밍을 위한 GUI와 CAD의 응용

GIROB : Graphic User Interface for Robot Programming

김 재 정*
J. J. Kim

ABSTRACT

As international market pressures increase, product life cycles are becoming shorter at same time requirements for productivity and flexibility are becoming higher. Major improvement in robot programming user interface is central to increasing productivity and flexibility for automatic manufacturing environments. New computer technologies have the potential to accomplish this improvement.

In this study we have developed a system which proves that interactive computer graphics and geometric modeling have matured to the point where their artful application into an integrated system can result in radically new and powerful user interface for robot programming. With the graphic user interface environment the system efficiently utilizes the existing CAD data and gemetric modeling techniques for off-line programming and simulations. Also the system successfully generates robot control programs performing the desired tasks given through off-line programming.

주요기술용어 : CAD(컴퓨터 원용설계), Soild Modeling(고체형상 모델링), GUI(그래픽 사용자 접속 수단), Off-line Programming(오프라인 프로그래밍), 간섭검사

1. 서 론

제품의 가격경쟁력 강화를 위하여 제품생산시 생산성(productivity) 향상이 중요한 위치를 차지하고 따라서 산업현장의 생산라인이나 조립라

인의 자동화에 대한 필요성이 더욱 더 대두되고 있다. 이를 위하여 현재 국외는 물론 국내의 공장에는 많은 산업용 로봇이 생산라인에 활발히 투입되고 있다. 하지만 국내에서 사용중인 대부분의 로봇은 선진외국에서 설계제작된 것으로 그나마 국내에서 제작되는 로봇은 골격에 해당하는 몸체 부분이고 뇌에 해당하는 콘트롤러(control-

* 정회원, 한양대학교 기계설계학과 조교수

ler)나 프로그래밍 언어는 대부분을 외국산에 의존하고 있다. 그러므로 현재 로봇의 국산화를 위하여 많은 국내 대기업에서는 콘트롤러나 프로그래밍 언어개발에 박차를 가하고 있다. 본 연구는 이러한 로봇국산화 움직임에 발맞추어 로봇의 여러가지 연구분야중에서 중요한 문제의 하나인 로봇 프로그래밍 방법에 초점을 두었다.

로봇 프로그래밍에서 가장 많이 쓰이는 기존의 교시방법(teach by showing)에 따른 로봇 프로그래밍은 원하는 작업을 입력시키기 위해 로봇 매니플레이터의 모든 동작 하나하나를 지정해야 하므로 많은 시간과 노력을 필요로 한다. 더구나 작업경로나 작업환경이 기하학적으로 복잡한 경우에는 이 문제가 더욱 심각해진다. 또한 이러한 방식의 프로그래밍을 위해서는 생산라인을 중지시켜야 하므로 많은 경제적 손실도 수반하게 된다. 이런 문제점을 해결하기 위한 것으로 요즘 중요성을 인정받고 있는 방법이 바로 오프라인(off-line) 프로그래밍이다. 과거에는 오프라인 프로그래밍에 기술적인 제약이 많이 있었으나 근래들어 컴퓨터 기술과 컴퓨터 그래픽(computer graphic) 시뮬레이션(simulation) 그리고 GUI(graphic user interface) 기술의 눈부신 발전으로 충분한 여건이 갖추어졌다고 볼 수 있다.

로봇의 오프라인 프로그래밍 소프트웨어로서 외국에서 개발, 상용화된 시스템들이 현재 다수 시장에 나와 있다. 이들 시스템은 여러가지 목적을 가지고 다양한 기능을 제공하는 범용패키지(general-purpose package)들로서 상용패키지답게 훌륭한 사용자 인터페이스를 갖추고 있다. 그러나 이들 시스템들은 그들이 제공하는 일반적이고 다양한 기능에도 불구하고 실제로 어떤 특정한 영역의 작업을 수행할 때에는 항상 예기치 않은 문제에 봉착하여 그 유용성을 격감시키게

된다. 그 좋은 하나의 예로 데이터 호환성을 들 수 있겠다.

본 연구는 CAD 데이터와 컴퓨터 그래픽을 기반으로, 국내에서 개발중인 산업용 로봇들에게 오프라인 프로그래밍환경을 제공할 수 있는 전용시스템의 프레임워크(architecture) 구축을 목표로 하였다. 최근에는 제품의 설계 단계에서부터 CAD(computer-aided design)를 이용함에 따라 이들 CAD 데이터를 효율적으로 활용할 수 있는 오프라인 프로그래밍의 중요성이 더욱 강조되고 있다. 또한 전용시스템의 경우에는 기능의 다양성에서는 범용시스템에 뒤지지만 어떤 특정 영역에 특수하게 존재하는 여러 문제점들을 쉽게 해결해 줄 수 있다. 뿐만 아니라 특히 국내에서 프레임워크로 활용시 새로운 응용분야가 생길 때마다 필요한 기능들을 손쉽게 보강해 나갈 수 있는 장점도 있다. 이점은, 실제로 대부분의 국내 소프트웨어 사용자들이 외국에서 개발된 시스템을 사용하다 자기목적에 맞는 기능들을 추가시키고 싶어도 시스템의 폐쇄성으로 인하여 벽에 부딪힌다는 점에서 상당히 중요한 잇점으로 작용할 수 있다.

Fig.1에서는 이와 같은 목적을 위한 본 연구 접근방법의 전체적인 흐름을 간략히 나타내고 있다. 로봇과 작업공간(workcell)의 여러 물체들을 상용 모델러인 CATIA(computer aided three-dimensional interactive applications)로 모델링한 다음, 이들 기하학적 데이터(geometric data)들을 본 연구에서 프로토타입(prototype) 시스템으로 개발한 GIROB(graphic user interface for robot programming) 시스템으로 옮긴다. 그 다음 GIROB 시스템 내에서 이들 기하학적 정보들을 이용하여 로봇 매니플레이터의 동작을 시뮬레이션하면서 주어진 작업을 수행할



Fig.1 Overall flow of GIROB system

수 있는 로봇 프로그램을 오프라인으로 작성하고, 여기서 생성된 로봇제어 프로그램으로 실제 로봇을 구동하게 된다.

본 연구에서는, 기아기공의 협조를 얻어 일본 KAWASAKI 중공업의 Js-10 로봇을 구체적인 연구대상으로 삼았으나 GIROB의 접근방법은 물론 특정 기종의 로봇에 제한되지 않고 타기종의 로봇에도 일반적으로 적용될 수 있도록 설계되었다. Js-10 로봇은 7개의 링크(link)를 갖는 6관절 로봇으로서 용접(welding), 조립(assembly), 핸들링(handling) 등의 용도로 쓰이는 다목적용 로봇이다^{1), 2)}.

본 논문의 구성은 다음과 같다. 제 2 장은 CATIA를 이용하여 로봇을 모델링하고 이들 기하학적 정보를 변환하는 방법을 간단히 서술하고 Js-10 로봇의 역기구학(inverse kinematics)의 해법에 관해 논의한다. 제 3 장에서는 컴퓨터 그래픽을 이용한 로봇 프로그래밍의 큰 장점이라고 할 수 있는 간섭점사 알고리즘이 기술되어 있으며 제 4 장은 GIROB에서 제공되는 오프라인 프로그래밍의 환경 및 GIROB에 고유한 로봇제어 언어의 형태를 설명한 후 그것으로부터 특정 로봇을 제어하는 프로그램을 생성하는 후처리방법에 관해 서술한다. 그리고 제 5 장은 GIROB의 전체적인 구조와 각 기능들을 설명한 후 GIROB를 사용하여 실제 Js-10 로봇을 구동시킨 결과에 대해 기술하고 본 연구에 대한 최종적인 결론을 제 6 장에 서술하였다.

2. 로봇과 작업환경의 모델링

2.1 기하학적 모델링

본 연구에서는 로봇과 작업환경의 기하학적 정보를 모델링을 하기 위해 3차원 고체형상 모델링(solid modeling) 기능을 제공한 CATIA를 사용하여 물체들의 형상을 정의한 다음 이로부터 GIROB에 필요한 데이터를 추출하게 하였다. CATIA는 국내 제조업체들의 주종 CAD/CAM 시스템의 하나로써 프랑스의 DASSAULT SYSTEM에 의해 개발되어 IBM에서 판매하고

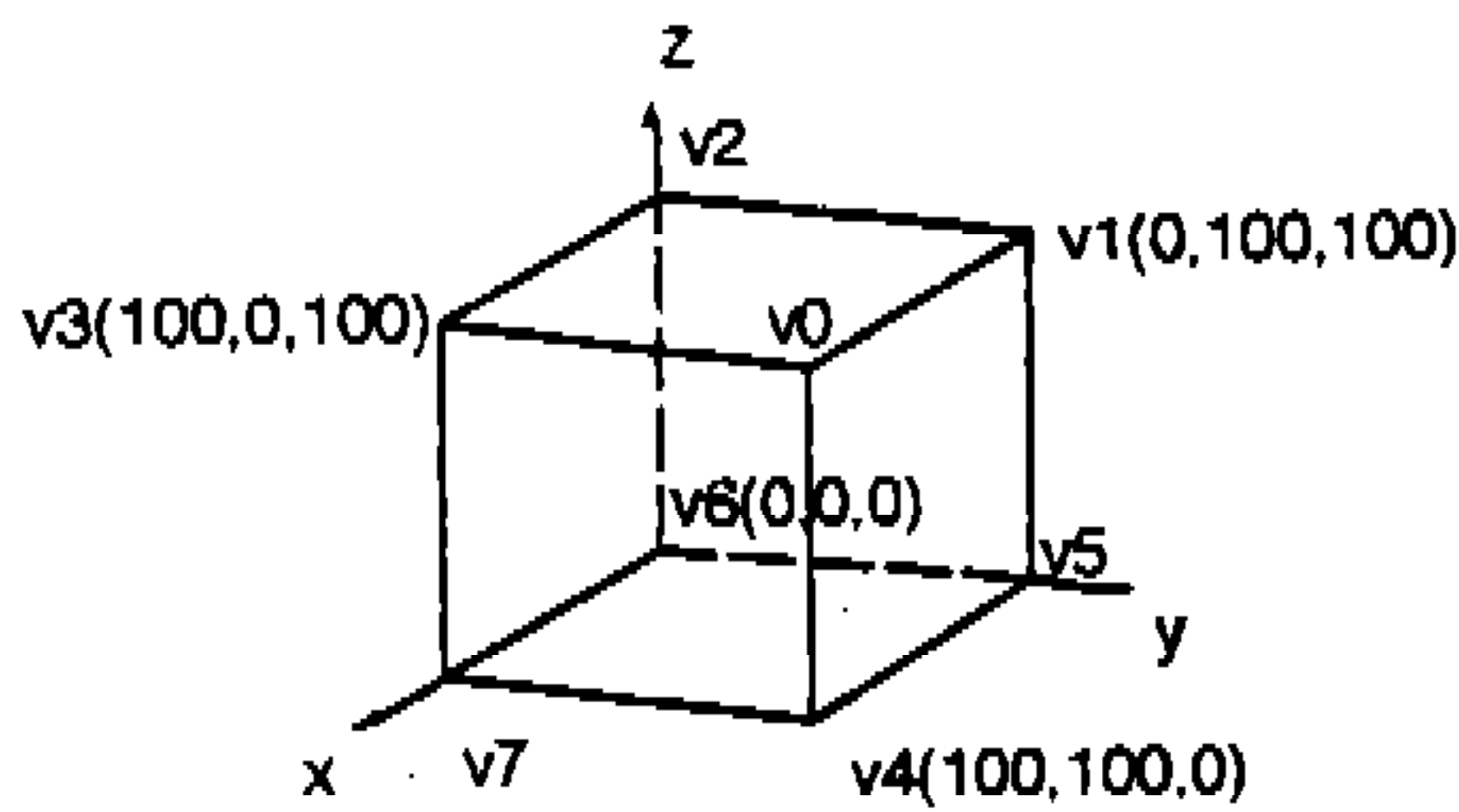
있다. CATIA는 그래픽 인터페이스(graphic interface)를 지원하는 기본모듈(base module)을 바탕으로 수십개의 응용모듈(application module)을 제공하며 특히 자유곡면(free surface)으로부터 NC 공구경로를 계산하는데 우수한 것으로 평가되어 우리나라의 자동차 및 항공관련 제조업체에 현재 약 5,000시트(seat)가 보급되어 있다.

본 연구에서는 로봇과 작업환경의 모델링을 위하여 CATIA에서 제공되는 SOLID기능과 BOOLEAN 기능을 주로 사용하였으며 CATIA에서 모델링된 물체는 다각형(polygon)으로 근사(approximation)된 형체를 갖는다. 이렇게 모델링된 물체에 대한 기하학적 데이터를 외부와 인터페이스(interface)하여 다른 응용프로그램에서 사용할 수 있도록 하기 위해 CATIA에서는 특별한 기능을 제공해 준다. Geometry Interface(GI)라고 불리우는 이 기능은 GATGEO라고 하는 FORTRAN 서브루틴들로 구성되어 있으며 CATIA에서 제공되는 기하학적 요소들(점, 곡선, 물체)에 관련된 데이터 들을 첨가, 수정, 입, 출력등을 가능케 한다.

본 연구에서는 이들 GI 기능이 제공하는 CATGEO 서브루틴들을 이용하여 GIROB에서 필요로 하는 기하학적 데이터들을 추출할 수 있는 인터페이스 프로그램을 개발하였다. 이 인터페이스 프로그램을 이용하여 추출되는 정보는, GIROB에서 필요로 하는 와이어프레임(wireframe) 정보와 면(face), 면의 법선벡터(face normal vector)이며 텍스트(text) 파일형태로 출력된다. Fig.2는 CATIA에서 모델링된 정육면체(cube)로부터, 위에서 설명한 인터페이스 프로그램을 이용하여 기하학적 데이터를 생성시킨 예를 보여 주고있다. Fig.2에서 보는 바와 같이 인터페이스 프로그램을 통하여 CATIA로부터 추출되는 출력파일은 점테이블(vertex table)과 면테이블(face table), 면법선벡터테이블(face normal vector table)로 구성되는 3가지 종류의 기하학적 데이터를 저장하고 있다. 먼저 점테이블은 와이어프레임을 구성하는 데이터로서 점의 갯수와 각 점의 좌표값을 저장하고 있다. 다음으로 면테

이블은 물체간의 간섭검사에 필요한 정보인 면의 갯수와 각 면을 구성하는 다각형의 꼭지점들을 저장한다. 여기서 꼭지점은 점의 좌표값을 직접 갖지 않고 점테이블의 해당점에 대한 포인터(pointer)로 나타낸다. 그리고 마지막으로 물체의 음영표현(shading)에 필요한 각 면의 법선벡터가 면법선벡터테이블에 저장되어 있다.

GIROB은 이 파일을 입력 데이터로 하여 GIROB 내부의 물체에 대한 데이터 구조를 구성



| | | | | | |
|---|--------|--------|--------|-------------------------------|------------|
| 8 | | | | Number of Vertices | |
| 0 | 100.00 | 100.00 | 100.00 | Vertex table | |
| 1 | 0.00 | 100.00 | 100.00 | | |
| 2 | 0.00 | 0.00 | 100.00 | | |
| 3 | 100.00 | 0.00 | 100.00 | | |
| 4 | 100.00 | 100.00 | 0.00 | | |
| 5 | 0.00 | 100.00 | 0.00 | | |
| 6 | 0.00 | 0.00 | 0.00 | | |
| 7 | 100.00 | 0.00 | 0.00 | | |
| 6 | | | | Number of faces | |
| 0 | 0 | 1 | 0 | 4 | Face table |
| 7 | 4 | 0 | | 3 | |
| 1 | 0 | 1 | 1 | 4 | |
| 4 | 5 | 1 | | 4 | |
| 2 | 0 | 1 | 2 | 4 | |
| 0 | 1 | 2 | | 3 | |
| 3 | 0 | 1 | 3 | 4 | |
| 5 | 6 | 2 | | 1 | |
| 4 | 0 | 1 | 4 | 4 | |
| 6 | 7 | 3 | | 2 | |
| 5 | 0 | 1 | 5 | 4 | |
| 4 | 7 | 6 | | 5 | |
| 6 | | | | Number of face normal vectors | |
| 0 | 1.00 | 0.00 | 0.00 | Face normal vector table | |
| 1 | 0.00 | 1.00 | 0.00 | | |
| 2 | 0.00 | 0.00 | 1.00 | | |
| 3 | -1.00 | 0.00 | 0.00 | | |
| 4 | 0.00 | -1.00 | 0.00 | | |
| 5 | 0.00 | 0.00 | -1.00 | | |

Fig.2 An example of GIROB input file extracted from the cubic model created with CATIA

하게 된다. Fig.3은 Js-10 로봇과 공작물 및 테이블을 CATIA로 모델링한 후 인터페이스 프로그램을 이용하여 필요한 데이터를 추출하고 GIROB에서 이 로봇을 재구성한 형상을 와이어 프레임으로 보여주고 있다.

2.2 역기구학

로봇의 역기구학이란, 로봇에 부착된 공구(tool)가 지정된 위치(position)와 방향(orientation)에 도달 할 수 있도록 각 관절의 조인트 각도값을 구하는 것으로서 역기구학의 해가 존재하는 공간상의 영역을 로봇의 동작범위(workspace)라고 부른다. GIROB내에서 로봇을 프로그래밍 및 시뮬레이션하기 위해서는 역기구학 계산이 필요한데 로봇은 그 종류마다 서로 다른 조인트 수와 링크 파라미터(link parameter)들을 가지므로 매니플레이터가 운동하기 위한 역기구학 해도 서로 다르다^{3), 4)}. 따라서 각각의 로봇을 프로그래밍하고 시뮬레이션하기 위해서는 각 로봇의 종류에 따른 역기구학의 해를 저장하고 있어야 한다. 물론 최근의 활발한 연구결과로 임의의 조인트 수와 링크 파라미터를 갖는 로봇에 대한 역기구학의 해를 일반적으로 계산하는 방법도

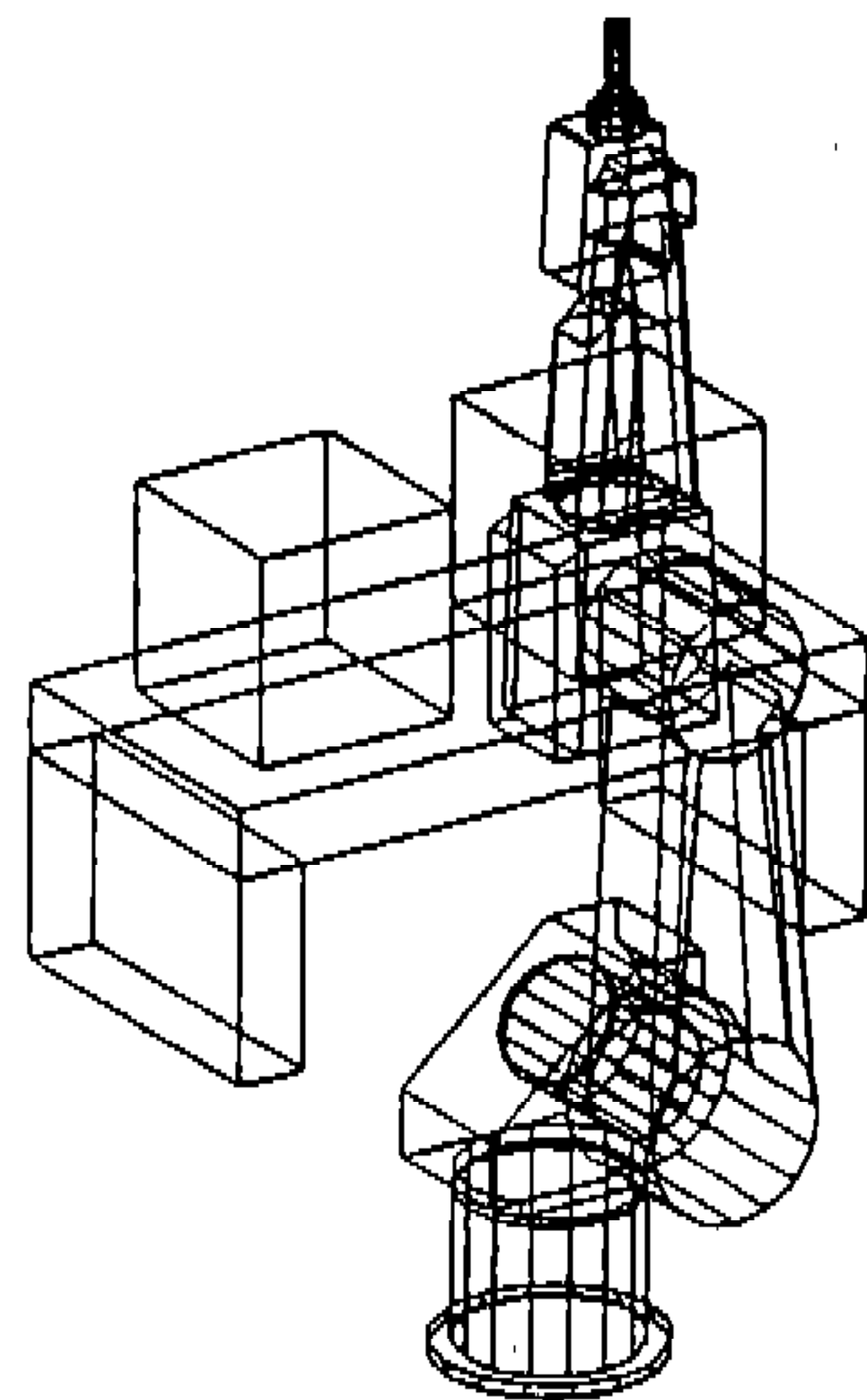


Fig.3 Js-10 robot and working environment displayed in wireframe

제시되고 있지만 이 경우에는 아주 많은 양의 변환행렬(transformation matrix) 곱셈을 수행해야 하기 때문에 계산시간이 많이 필요하고 따라서 GIROB과 같이 컴퓨터 그래픽을 이용하여 로봇의 동작을 3차원 공간에서 실시간(real time)으로 시뮬레이션하기에는 부적합하다고 할 수 있다. 따라서 GIROB이 일반적으로 응용되기 위해서는 대표적인 로봇제작회사의 여러 기종의 로봇에 대해 각각 역기구학의 해를 미리 구하여 이를 데이터 베이스화하여 라이브러리로서 구축해야

할 필요가 있다. GIROB은 우선 시험대상으로 삼았던 J_5-10 과 국내에서 개발중인 몇개의 로봇에 대해 역기구학의 해를 라이브러리로 구축하였으며 본 논문에서 유도과정은 생략하기로 한다. Fig.4는 GIROB상에서 J_5-10 로봇이 역기구학의 해를 구하며 자유곡선을 따라 움직이는 것을 보여주고 있다.

일반적으로 로봇의 역기구학의 해를 구하면, 로봇의 조인트 숫자, 링크 파라미터, 각 조인트의 허용동작범위(allowable range of motion)에 따라 여러가지 해가 존재하게 된다. 한예로 Fig.5는 같은위치와 방향에 도달할 수 있는 2가지해(ELBOW-UP/DOWN)를 보여주고있다. 따라서 이들 해 중에서 가장 적합한 해를 선택해야 할 필요가 있는데 대부분의 경우에는 이들 해 중에서 현재 로봇의 위치에서 가장 가까운 해(the closest one)를 선택하게 된다³⁾. 따라서 역기구학의 해를 구하는 과정에 있어 로봇 매니플레이터의 현재 위치값도 하나의 입력으로 작용한다. 이때 어느 해가 더 가까운 해인가를 결정하는데 있어서, 큰 조인트를 움직이는 것보다는 작은 조인트를 움직이게 선택되도록 가중치(weight)를 적용할 수도 있다. 그러나 만일 어떤 방해물이 로봇의 경로에 존재하여 가까운 해가 충돌을 일으키는 경우에는 먼 해가 선택되어야 한다. 따

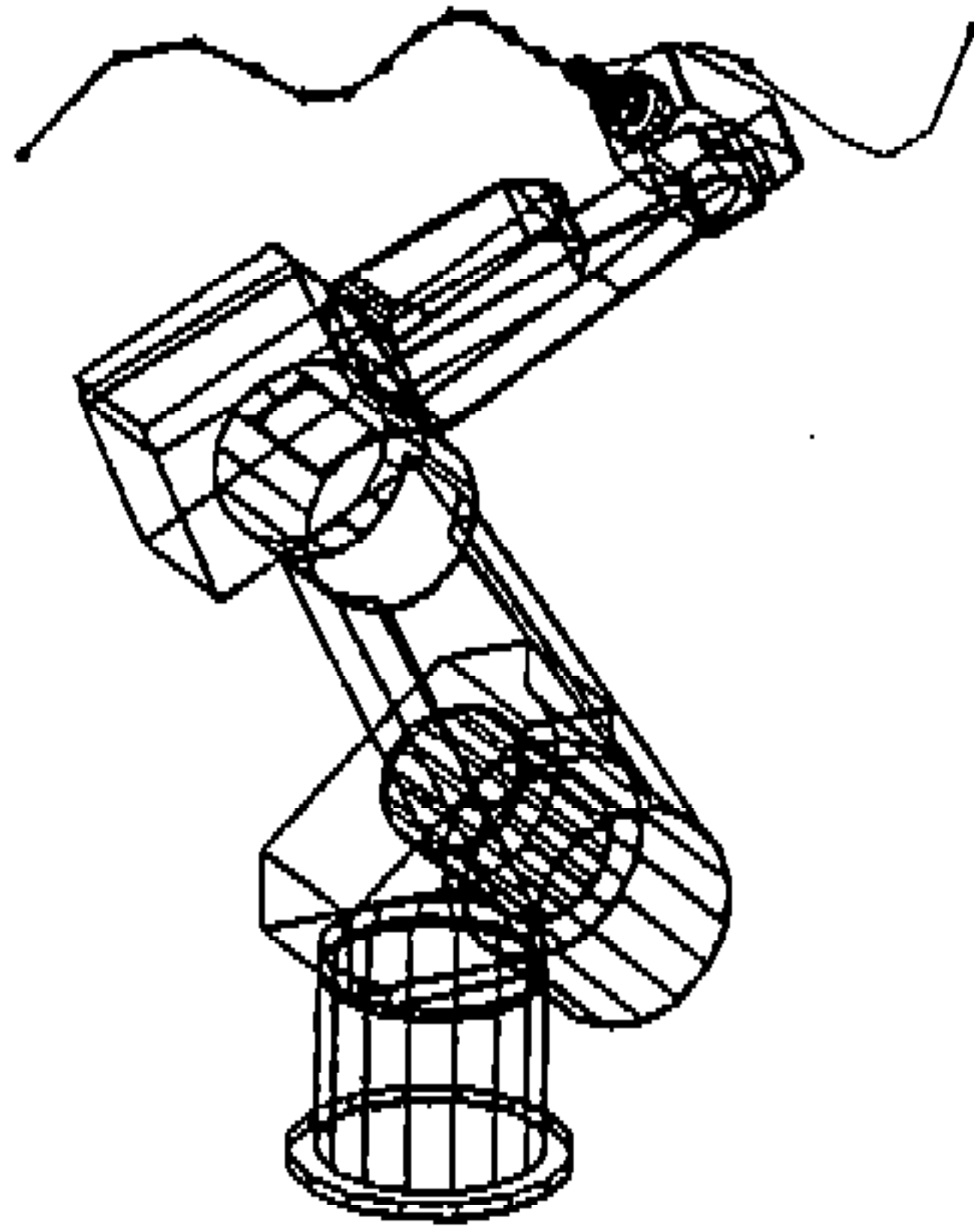


Fig.4 J_5-10 robot following Spline curve with inverse kinematics

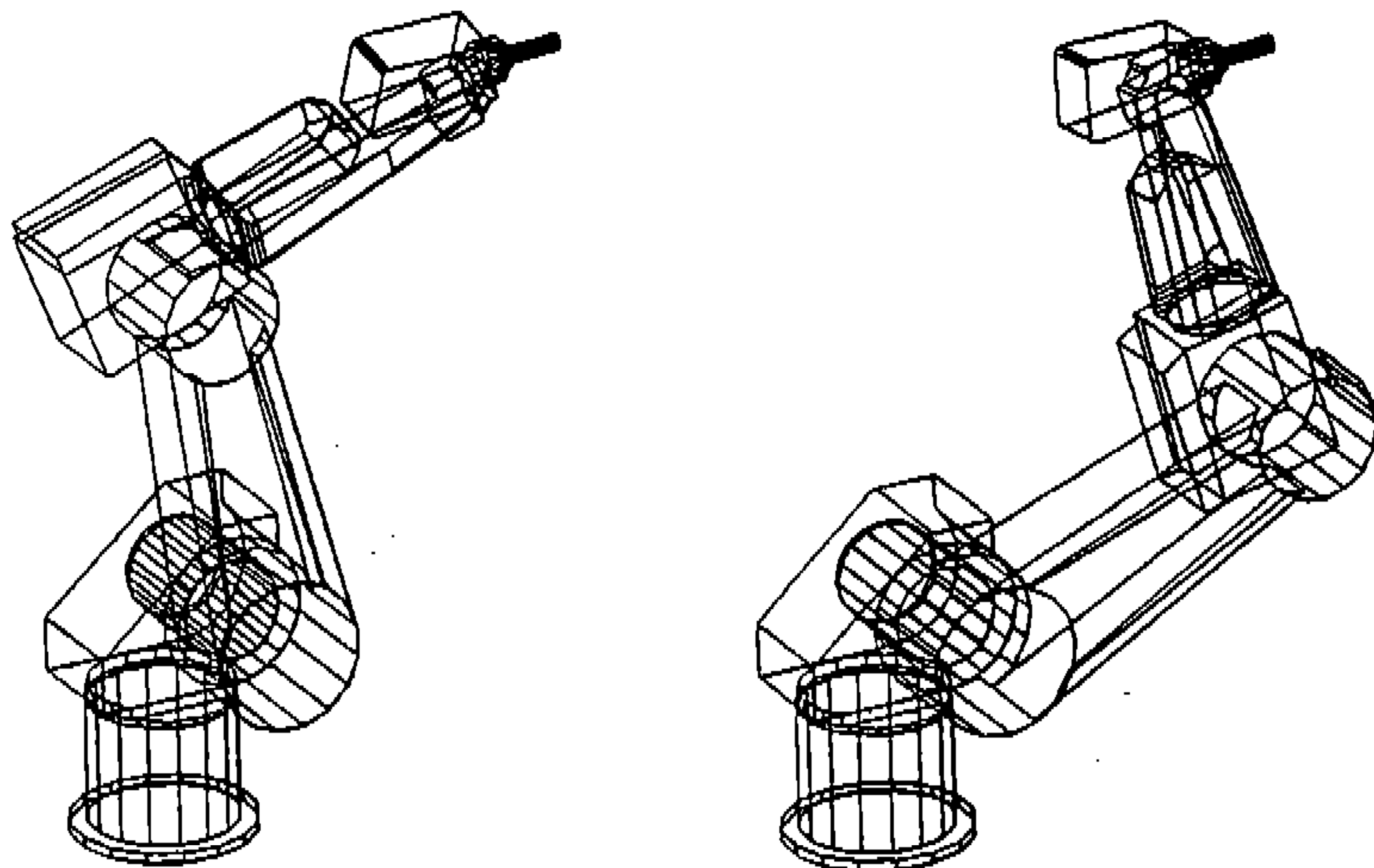


Fig.5 An example of two solutions : ELBOW-UP/ELBOW-DOWN

라서 일반적으로 모든 가능한 해를 일단 GIROB에서 구한 다음, 적절한 선택기준과 충돌 여부에 따라서 최종적인 해를 선택할 수 있도록 하였다.

3. 간섭검사

간섭(interference)이란 두 개의 물체가 같은 시간에 같은 공간을 점령하고 있는 현상을 말하는데 이러한 간섭의 존재 여부를 검사하는 작업을 간섭검사(interference test)라고 부른다⁵⁾. 간섭검사는 로봇의 경로 계획(robot path planning), NC 공구경로계획(NC tool path planning), 조립체 모델링, 조립 계획⁶⁾ 등에 응용된다.

로봇 운동의 시뮬레이션에서 간섭검사는 두가지 목적으로 사용된다. 첫번째는 로봇의 각 매뉴플레이터간의 간섭검사이다. 로봇에서 엔드이펙터(end-effector)가 도달해야 할 목표점이 주어졌을때 만약 그 목표점이 베이스 링크(base link)의 내부에 위치할 경우에는(실제로 로봇의 작업범위에 베이스 링크의 내부도 포함한다) 로봇이 운동하는 때 순간마다 충돌이 일어날 가능성이 있는 매뉴플레이터간에 간섭여부를 검사한다. 다음, 로봇의 각 매뉴플레이터를 운동시켜야 한다. 두번째는 운동하는 로봇의 각 매뉴플레이터와 외부 물체들 간의 간섭검사이다. 로봇이 주어진 작업을 수행하기 위하여 작업공간상에서 움직이는 경우에 외부 물체와 충돌이 일어날 수 있으므로 간섭검사를 통하여 충돌여부를 판단하면서 로봇을 운동시켜야 하며 또한 외부 물체와 충돌 없이 주어진 작업을 수행할 수 있도록 작업경로를 찾는 작업경로계획(collision-free path plan)이 필요하다.

간섭검사란 수학적인 관점에서는 두 물체 간에 교집합(intersction)을 찾는 문제이다⁷⁾⁸⁾⁹⁾. 간섭검사 알고리즘은 간섭검사의 대상이 되는 물체의 표현방식에 따라 여러가지 방법이 있을 수 있다. 본 연구에서는 대상 물체를 CATIA로 모델링하면서, 다각형을 기본으로하는 다면체들(polyhedra)로 표현하므로 두 다면체에 대한 간섭검사 알고리즘을 개발하였다. 일반적으로 정지되어 있

는 두 물체간의 간섭검사를 위해서도 많은 계산량이 요구되는데 특히 로봇 시뮬레이션처럼 연속적으로 위치가 변하는 물체들간의 간섭검사는 막대한 계산량을 필요로 한다. 그러므로 본연구에서는 간섭검사 알고리즘의 계산효율을 높이기 위하여 i)최소-최대 박스(min-max box)를 이용한 두물체들간의 간섭가능성 결정, ii)오버랩 박스(overlap box)를 이용한 간섭후보면의 선택, iii)간섭후보면들 간의 간섭검사와 같이 세 단계로 구성하였는데 각단계에 대한 설명을 아래에 간단히 기술한다.

3.1 최소-최대박스를 이용한 간섭가능성 결정

본 연구에서 개발된 간섭검사 방법은 기본적으로 두 물체의 면과 면 사이의 간섭을 검사하기 때문에, 모든 면들을 대상으로 간섭검사를 할 경우에는 많은 계산을 필요로하게 된다. 또한 대개의 경우에 있어서는 두 물체 간에 간섭이 일어나지 않는 경우가 보통이므로 이런 경우에도 두 물체의 모든 면을 대상으로 간섭검사를 하는 것은 계산의 낭비를 가져오게 된다. 따라서 간섭검사의 효율을 증대시키기 위하여는 두 물체 간에 간섭이 일어날 수 있는 가능성 여부만을 간단하고 빠른 방법으로 먼저 판단할 필요가 있다. 이를 위해 도입된 것이 최소-최대박스를 이용한 간섭

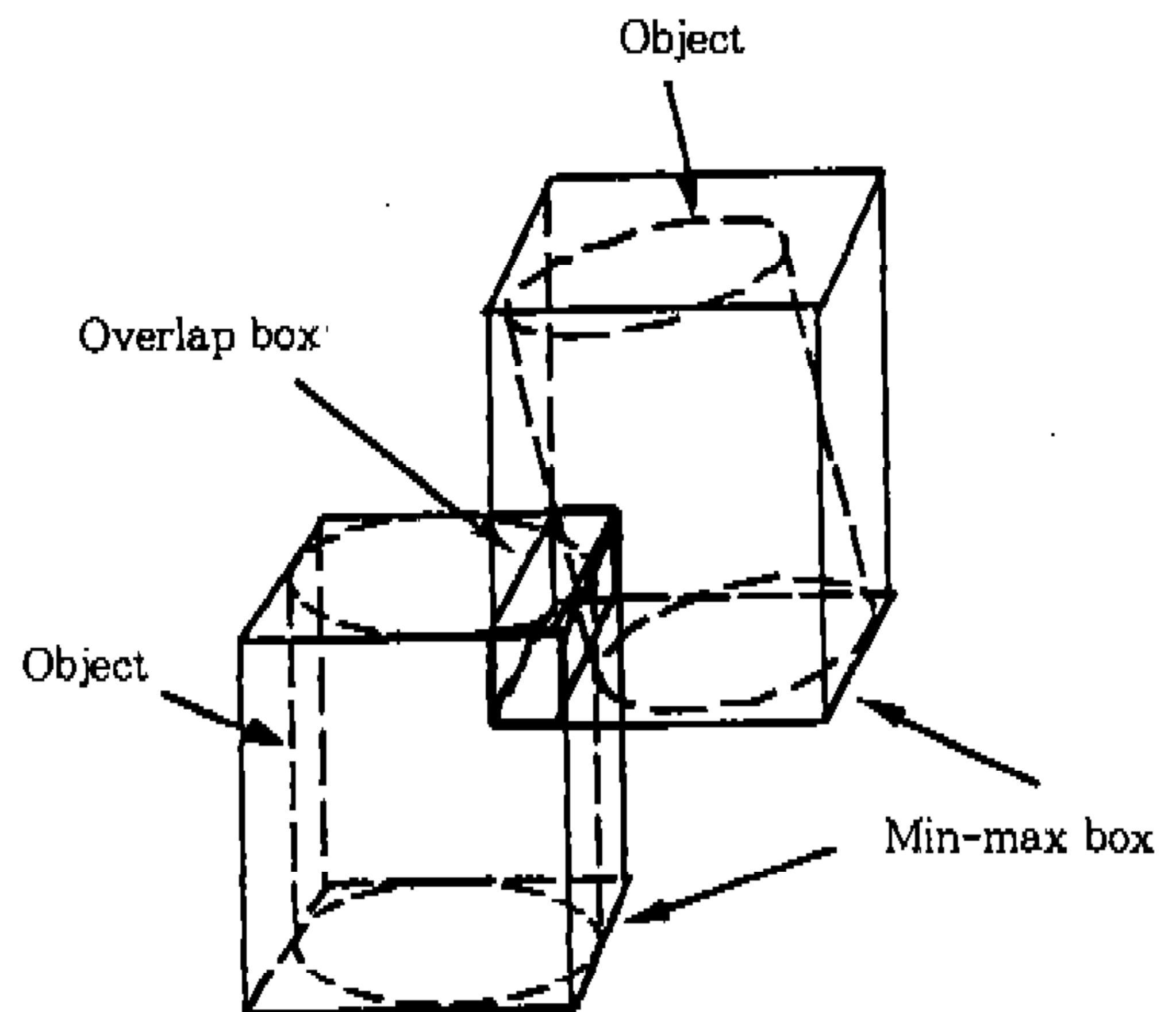


Fig.6 Overlap box generated from two min-max boxes

가능성 여부의 계산이다.

최소-최대박스란 Fig.6에서 보는 바와 같이 주어진 물체를 포함할 수 있는 최소인 동시에 최대인 직각육면체(cuboid)를 말하는데 다면체의 각 꼭지점의 좌표값으로부터 쉽게 계산할 수 있다. 간섭검사의 대상인 두 물체로부터 최소-최대 박스를 구성하고 이들 두 직각육면체를 이용하여 면-면 간섭검사를 수행하면 $6C_1 \times 6C_1 = 36$ 번의 검사로 간섭 가능성 여부를 알 수 있다. 만일 최소-최대박스를 이용한 간섭검사 결과 간섭이 일어나지 않으면 두 물체 사이에는 간섭이 전혀 없다고 판단할 수 있고 그렇지 않으면 실제 두 물체 간에 간섭이 생길 가능성이 있으므로 아래의 한단계 더 정확한 검사를 통하여 간섭 여부를 판단해야 한다.

3.2 오버랩 박스를 이용한 간섭후보면의 선택

최소-최대박스를 이용하여 간섭검사를 수행한 결과 간섭의 가능성이 있다고 판단되는 경우에는 물체들의 모든 면에 대하여 직접 간섭검사를 수행함으로써 실질적인 간섭 여부를 알 수 있으나 이 경우 역시 많은 계산을 필요로 한다. 따라서 본 연구에서는 간섭이 일어날 수 있는 후보면들을 선택하여 그 후보면들에 대해서만 간섭검사를 수행하여 간섭검사에 필요한 계산량을 줄일 수 있도록 하였다.

본 연구에서 개발한 간섭후보면의 선택은 오버랩 박스의 구성과 오버랩 박스로부터 간섭후보면 선택이라는 두 단계로 나누어진다. 먼저 오버랩 박스의 구성은 간섭이 일어난 두 최소-최대박스를 이용하여 간섭이 일어날 가능성이 있는 영역을 포함하는 직육면체를 구하는 것이다. Fig.6의 굵은선은 두 최소-최대박스로부터 얻은 오버랩 박스를 나타내는데, 이는 먼저 오버랩 박스의 x좌표의 최소값은 두 최소-최대박스로부터 얻은 두 개의 최소 x값들 중에서 최대값을 취하여 정하고, x좌표의 최대값은 두 최소-최대박스로부터 얻은 두 개의 최대 x값들 중에서 최소값을 취하여 구한다. 오버랩 박스의 나머지 y, z좌표값들도 같은 방법으로 결정된다. 다음, 이러한 과정을 거쳐 구한 오버랩 박스와 실제의 두 물체

사이에 간섭이 일어날 수 있는 간섭후보면들을 찾아 내는데 간섭후보면은 물체의 모든 면들 중에서 그 면의 일부분이라도 오버랩 박스와 간섭하는 면을 말한다. 여기서 오버랩 박스와 물체의 면과의 간섭 여부는 그 면의 각 꼭지점들과 오버랩 박스와의 인-아웃 테스트(in/out test)를 통하여 간단히 알 수 있다^{8),9)}. 만일 두 물체의 모든 면들 중에서 오버랩박스와 간섭하는 면이 없으면 두 물체 간에는 간섭이 전혀 일어나지 않는 것이며 그렇지 않으면 간섭 후보면들끼리 아래의 면-면 간섭검사를 수행함으로써 최종적인 간섭 여부를 판단할 수 있다.

3.3 간섭후보면들 간의 간섭검사

두 간섭후보면들 간의 간섭 여부를 최종적으로 판단하기 위해서는 우선 간섭검사의 대상이 되는 두 면을 기준면(base face)과 검사면(test face)으로 나누어야 한다. 예를들면 Fig.7에서와 같이 두 면 중에서 임의의 한 면 A를 선택하여 이를 기준면이라 하고 나머지 B면을 검사면으로 정의한다. 그리고 난 후 이 두면을 간섭검사공간(interference test space)인 좌표계 O-uvw로 변환해야 하는데 간섭검사를 수행할 좌표계 O-uvw는 다음과 같이 결정한다. 우선 기준면위의 임의의 한 꼭지점을 원점 O로 삼고 이 점을 포함하는 한 변을 u축으로 정한다. 다음, 이와 수직이고 그 면을 포함하는 평면 상에 존재하는 v축을 정하고 난후 w축은 $u \times v$ 로 계산한다. 이와 같

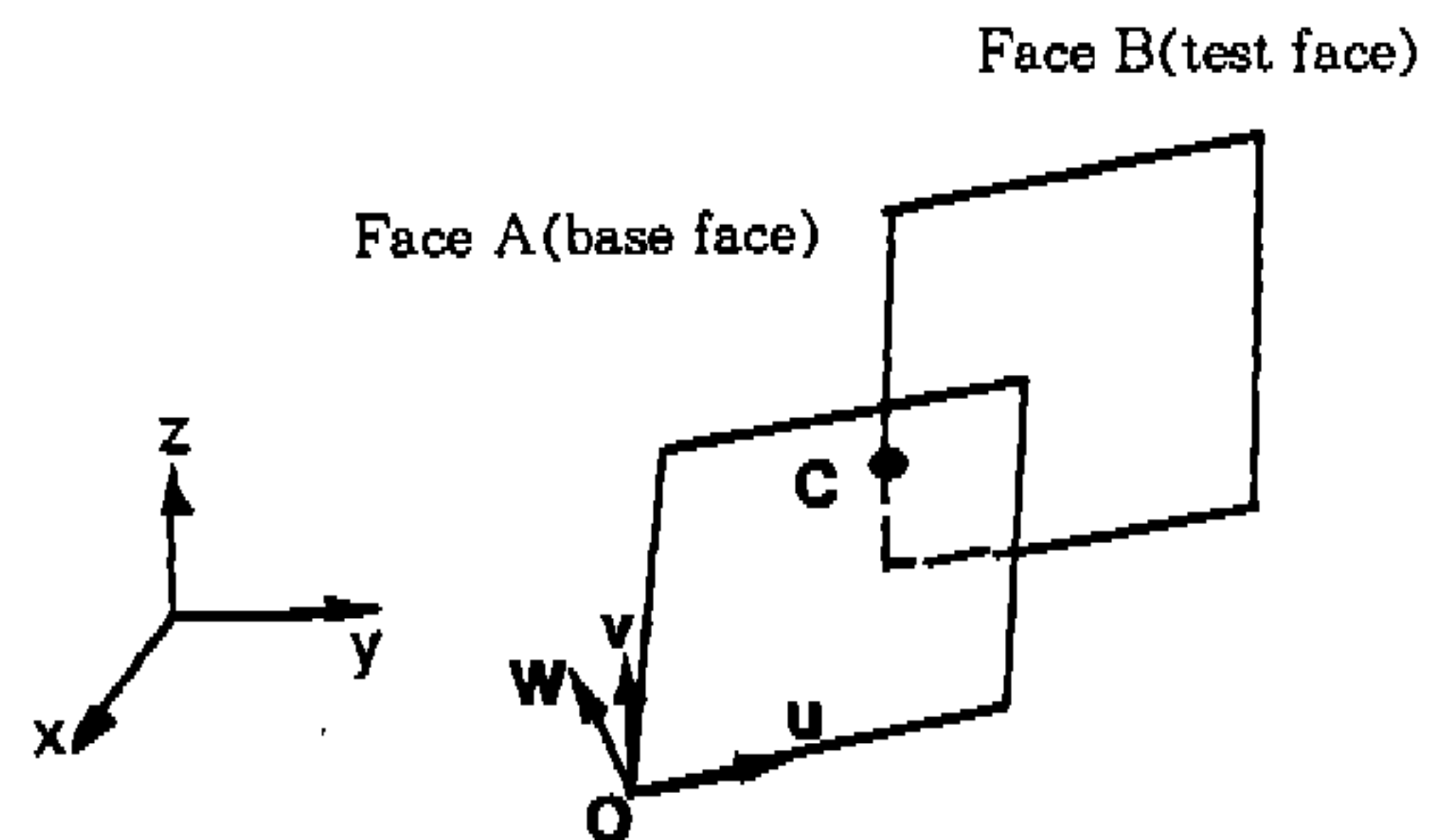


Fig.7 Test space O-uvw

이 결정된 O-uvw 좌표계로 간섭 대상 면들을 변환함으로써 두 면에 대해 간섭검사를 수행할 준비 작업은 끝난다.

마지막으로 간섭검사 공간 O-uvw 상에 위치한 두 면에 대한 간섭검사는 두 면 간에 교집합이 존재하는지를 판단함으로써 이루어진다. 즉 기준면 A를 포함하고 있는 평면과 검사면 B의 각 변들 사이에서 교점 C를 구한 후 만일 교점이 존재하지 않으면 두 면 사이에 간섭이 전혀 일어나지 않는 것으로 결론지을 수 있고 그렇지 않으면 교점 C가 기준면 A의 내부에 존재하는지 여부를 검사하여 최종적으로 간섭여부를 결정짓는다.

Fig.8은 GIROB을 이용, 각 매니플레이터와 주변물체들 간에 간섭검사의 예를 보여 주고 있다. 그림에서 붉은 선으로 표시된 것은 간섭이 일어난 면을 나타내고 있다.

4. 오프라인 프로그래밍(Off-line programming)

오프라인 프로그래밍을 제공하는 시스템의 중요한 목적 중에 하나는 로봇 프로그래밍 작업을 특정 기종의 로봇에 구애받지 않고 일반적이고 독립적인 방법으로 수행 하고자 하는데 있다. 이를 위해 오프라인 프로그래밍 시스템은 다양한 로봇에 적용될 수 있도록 표준화된 프로그래밍 도구들을 제공해야 하며, 또한 완성된 로봇 제어

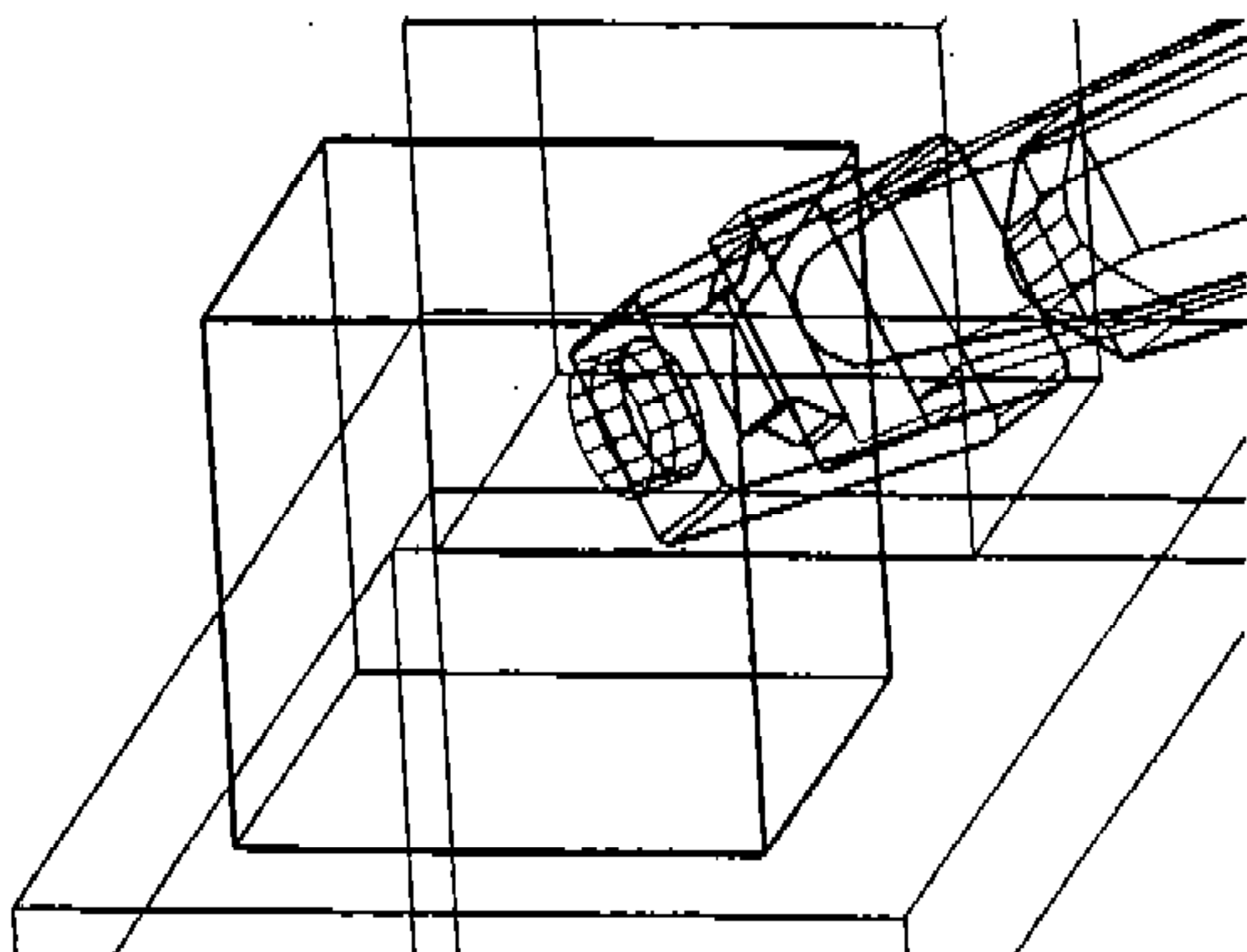


Fig.8 An example of interference test

프로그램이 특정 로봇의 제어언어와도 무관하게 유지되어야 한다. 따라서 GIROB에서는 자신의 오프라인 프로그래밍 환경을 써서 완성된 로봇 프로그램을 GIROB 고유의 형태(이하 이것을 GIROB 로봇 프로그램이라 지칭한다)로 저장될 수 있도록 구조 및 형태를 개발하였다. 또한 일반적인 로봇 제어언어들의 특성을 분석하여, GIROB 로봇 프로그램이 후처리에 의해 이들 언어로 변환될 때 필요한 정보를 모두 포함할 수 있도록 하였다.

본 연구에서는 오프라인 프로그래밍을 적용하는 실험대상으로 Js-10 로봇을 선정하였으므로 아래의 GIROB 로봇 프로그램 환경에 관한 설명은 편의상 Js-10 로봇을 구동하는 AS언어를 이용하기로 한다²⁾.

4.1 로봇 프로그래밍

우선 GIROB에서 로봇 프로그래밍은 명령어 하나의 단위로 이루어지거나 몇몇 명령어가 조합되어 이루어진다. 여기서 명령어는 i) 속도, 가속도, 정확도 등의 동작특성을 지칭하는 것과, ii) 로봇의 동작자체를 지정하는 것으로 크게 나누어진다.

먼저 로봇의 동작특성을 지정하는 명령들은 여러 로봇제어 언어를 분석한 결과 일반적으로 다음과 같은 형태를 지닌다.

Instruction numerals [option]

여기서 instruction은 로봇의 동작특성을 지정하는 명령이고 numerals은 그 특성의 값을 나타내는 수치이며, 그리고 option은 이 특성에 관련된 선택사항을 나타낸다. Js-10 로봇의 경우에는 이 선택사항으로서 ALWAYS가 주어지는데 이것은 이 특성이 바로 다음 명령에만 국한되는 것인지 아니면 그아래로 계속 영향을 미치는가를 표현한다.

이런 동작특성들에 대한 정보를 저장하려면 그 정보가 무엇에 관련되는지를 표현하는 플래그(flag)과 그에 따른 수치값, 그리고 옵션을 함께 기억해야 하는데 Table 1에 이러한 명령들에 대하여 GIROB 로봇 프로그램에서 사용되는 플래그(flag)을 나타내었다. Table 1에서 numerals은

그 특성의 값을, option의 A는 여기서 지정되는 특성들이 이후의 모든 동작들에 영향을 미친다는 것을 나타낸다. 이와 같은 로봇의 동작특성을 나타내는 명령들을 표현하기 위해서 GIROB 로봇 프로그램은 다음과 같은 형태로 구성된다.

Flag numerals [option]

다음, 로봇의 운동방법은 조인트보간법(joint-interpolation method)과 선형보간법(linear-interpolation method) 두가지로 구분되는데 조인트보간법이란 어떤 특별한 제한없이 각 조인트에서 움직여야 할 변위만큼 운동하는 방법이고 선형보간법은 시작점과 목표점 사이를 직선 경로를 따라 움직이도록 하는 방법이다. 이러한 로봇의 동작자체를 지정하는 명령들은 여러 로봇제어 언어에서 일반적으로 다양한 형태의 하이-레벨(high-level) 명령어들로 구성된다. 하지만 이러한 하이-레벨 명령어들은 실질적으로 가장 기본적인 명령어 몇개로 모두 표현이 가능하다. 예를 들면 J_s-10 로봇의 AS언어에도 여러가지 하이-레벨 명령어(approach, depart, draw 등) 들이 존재하지만 이들 명령어들은 위에서 설명한 두개의 가장 기본적인 동작을 지정하는 명령어로 모두 표현이 가능하다.

GIROB에서는 따라서 이런 하이-레벨 기능들

Table 1 Attributes of robot motion and their corresponding flags

| Instruction | Flag | Option |
|--------------|-------|--------|
| speed | SPEED | A |
| accuracy | ACCUR | A |
| acceleration | ACCEL | A |
| deceleration | DECEL | A |

Table 2 Robot motion method and their flags

| Instruction | Flag |
|----------------------|------|
| joint interpolation | JM |
| linear interpolation | LM |

도 모두 지원하되 GIROB 로봇 프로그램의 구조가 간단해 지도록 이들 동작 자체는 가장 기본적인 명령어로만 표현되도록 로봇의 운동방법과 목표점에 도달할 수 있는 각 조인트의 변위를 지정하여 로봇의 동작을 표현한다. Table 2는 로봇의 운동방법과 이를 GIROB 로봇 프로그램에서 표현하기 위한 플래그를 나타내고 있다. GIROB 로봇 프로그램에서 로봇 동작을 표현하는 방법은 다음과 같은 형태로 구성된다.

Flag numerals

여기서 numerals는 로봇의 각 조인트의 변위를 나타내는 일련의 숫자로서 6축 로봇인 경우는 6개의 숫자로 다음과 같이 구성된다.

Numerals=jt1 jt2 jt3 jt4 jt5 jt6

한편 위에서 설명한 것 외에도 GIROB에는 특수한 동작 명령들을 처리하기 위한 많은 명령들이 존재하지만 지면관계상 생략하기로 한다.

4.2 후처리

일반적으로 로봇들은 제작회사에 따라 각기 다른 종류의 로봇 제어 언어를 사용하고 있으므로 실제로 로봇을 구동하기 위해서는 GIROB에서 오프라인 프로그래밍된 로봇 프로그램으로부터 특정 기종의 로봇을 제어하는 로봇 프로그램을 생성할 수 있도록 하는 후처리 작업이 필요하다. 종류가 다른 로봇 제어언어는 그 특수성에 의해서 다른 후처리 프로그램들이 필요하지만 이들 후처리의 접근방법은 동일하다고 할 수 있다. 따라서 GIROB에서는 AS언어를 위한 후처리 모듈만 구현하였지만, 이의 경험을 살리면 다른 후처리 모듈도 손쉽게 개발될 수 있다.

Table 3은 위에서 설명한 GIROB 로봇 제어 프로그램으로부터, AS언어의 로봇 제어프로그램으로 생성할 수 있는 일대일 대응관계에 대해서 보여주고 있다. 로봇의 동작을 지정하는 명령들을 일대일 대응으로 처리할때 주의 할 점은 예를 들면 AS언어는 로봇의 위치를 지정하기 위해 각 조인트의 변위를 사용할 경우 이를 조인트 변위값(joint displacement value)이라하여 로봇 콘트롤러 내부에서 나름의 방법으로 처리하므로, 조인트의 변위들을 AS언어의 조인트변위값으로

Table 3 Conversion of motion attributes and motion instructions from GIROB to AS language

| GIROB robot control instruction | | AS language | |
|---------------------------------|-------------------------|-------------|---------------------------------------|
| SPEED | numeral [A] | SPEED | numeral [ALWAYS] |
| ACCUR | numeral [A] | ACCURACY | numeral [ALWAYS] |
| ACCEL | numeral [A] | ACCEL | numeral [ALWAYS] |
| DECEL | numeral [A] | DECEL | numeral [ALWAYS] |
| JM | jt1 jt2 jt3 jt4 jt5 jt6 | JMOVE | #PPOINT(jt1, jt2, jt3, jt4, jt5, jt6) |
| LM | jt1 jt2 jt3 jt4 jt5 jt6 | LMOVE | #PPOINT(jt1, jt2, jt3, jt4, jt5, jt6) |

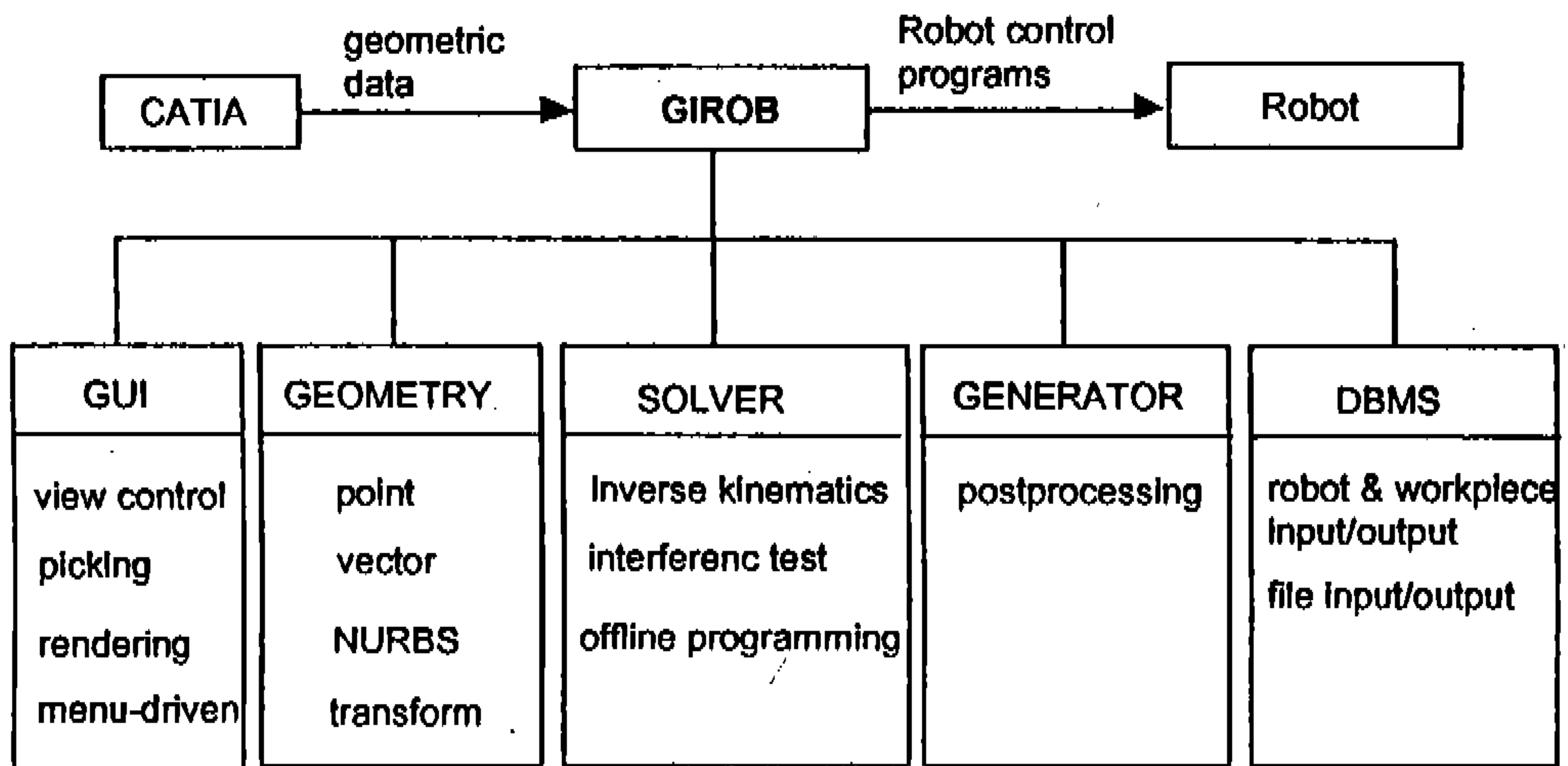


Fig.9 Architecture of GIROB system

변환해 주는 함수 #PPOINT(...)를 사용해야 한다. 이는 각 로봇제어 언어들이 가지고 있는 특수성을 보여주는 좋은 예라 하겠다.

5. GIROB 시스템의 구현 및 적용예

5.1 시스템의 구현

GIROB 시스템은 앞에서 설명한 연구결과들을 통합하여 IBM사의 RS/6000 워크스테이션 상에서 Silicon Graphic의 GL 그래픽 라이브러리¹⁰⁾(library)를 기본으로 하여 개발되었으며 시스템의 구성은 Fig.9에서 보는 바와같이 크게 개념적으로 GUI, GEOMETRY, SOLVER, GENERATOR, DBMS 다섯 개의 부분으로 나누어진다.

GUI 기능은 시스템 사용자의 편의를 위해 컴퓨터 그래픽의 장점을 활용하여 여러가지 작업을 처리할 수 있도록 하는 것으로 여기에는 컴퓨터 상의 물체를 여러 각도에서 볼 수 있도록 해주는 view 조정기능, 그래픽 물체들을 선택할 수 있는 picking 기능, 그래픽 물체들을 컴퓨터 화면 상에 그려주는 기능등의 여러가지 부속기능들을 메뉴방식으로 제공해 주는 기능들이 있다. 이 GUI 기능은 사용자로 하여금 GIROB을 배우고 사용하기 쉬운 인터페이스가 되도록 한다.

GEOMETRY 기능은 물체를 모델링하거나 로봇의 작업경로를 입력하기 위한 것으로 NURBS (Non-Uniform Rational B-Splines)를 이용하여 직선, 자유곡선들을 모델링 할 수 있으며, 로

봇의 엔드이펙터의 위치와 방향을 지정하기 위해 점과 벡터를 생성할 수 있다¹¹⁾. 또한 생성된 모델을 회전, 이동시키기 위한 여러가지 변환기능도 제공된다. Fig.10은 GIROB 시스템에서 제공되는 기능을 써서 점, 선분, 원, 원호, 자유곡선 등의 기하학적 요소(entity)들을 모델링 한 예이다.

SOLVER는 로봇의 운동을 시뮬레이션하기 위해서 역기구학의 해를 구하며, 물체 간의 충돌여부를 확인하기 위해 간섭검사를 하고 오프라인 프로그래밍을 수행하는 등의 계산기능을 담당하는 GIROB의 핵심부분이라고 할 수 있다.

GENERATOR는 GIROB 고유의 표준화된 로봇 프로그램을 후처리하여 특정 기종의 로봇을 제어할 수 있도록 각각의 로봇 기종에 맞는 로봇 제어언어를 생성하는 곳이다. 이를 위해 로봇의 기종에 따라 각기 다른 후처리기(postprocessor)가 필요하다.

DBMS(Data Base Management System)는 GIROB에서 다루는 여러 데이터들을 관리하는 곳이다. 여기에는 CATIA로부터 모델링 된 로봇이나 공작물들에 관한 데이터를 인터페이스 프로그램을 이용, 변환하여 읽어들이거나 이를 데이터를 GIROB 시스템 고유의 형태로 저장하고 복구하는 기능, 본 시스템에서 제공되는 NURBS 데이터들을 저장, 복구하는 기능, GIROB 로봇

제어프로그램을 저장, 복구하는 기능 등이 포함된다.

5.2 시스템의 적용예

GIROB 시스템을 사용하여 Js-10 로봇을 대상으로 실제로 오프라인 프로그래밍을 수행해 보았다. 프로그래밍 예로서는 로봇을 이용하여 Fig.11의 글씨를 쓰는 것을 선택하였는데 여기서 글씨는 본 시스템에서 제공되는 NURBS를 이용하여 모델링하였다. Fig.12는 R글자의 일부분을 본 시스템에서 제공하는 스플라인 운동(spline



Fig.11 Letters modeled with NURBS

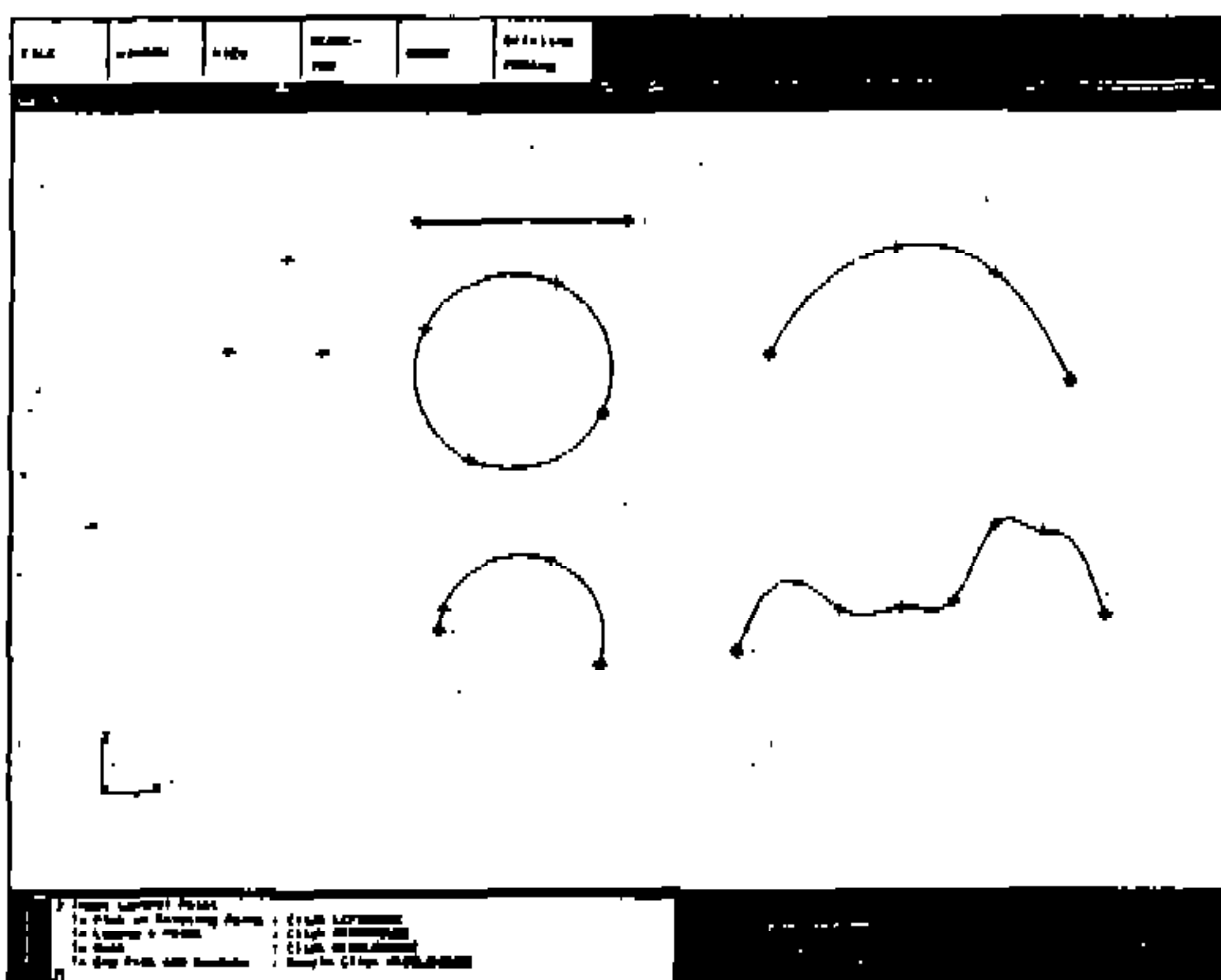


Fig.10 Geometric entities created with GIROB system

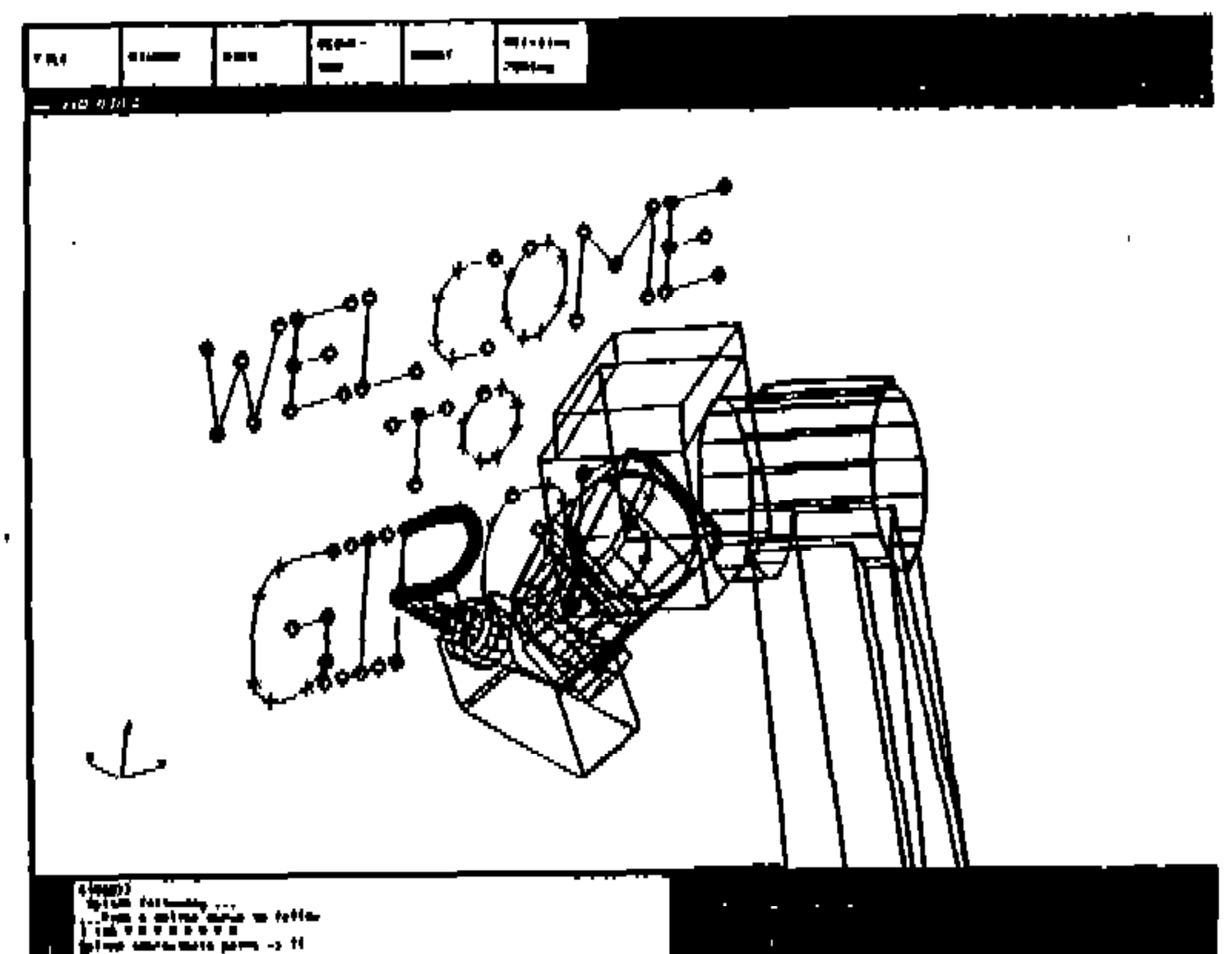


Fig.12 Off-line programming

```

SPEED 30 ALWAYS
ACCURACY 10 ALWAYS
LMOVE #PPOINT (-61.95, 29.28, -102.91, 24.40, 44.86, -107.83)
DELAY 1
LMOVE #PPOINT (-60.20, 35.85, -91.36, 24.20, 39.77, -109.05)
DELAY 1
LMOVE #PPOINT (-59.20, 39.47, -93.23, 20.47, 44.56, -104.89)
DELAY 1
LMOVE #PPOINT (-58.19, 36.20, -92.82, 20.42, 40.85, -105.73)
DELAY 1
LMOVE #PPOINT (-57.17, 38.96, -94.21, 17.50, 44.53, -102.67)
DELAY 1
LMOVE #PPOINT (-56.15, 34.75, -93.33, 17.72, 39.44, -103.86)
DELAY 1
LMOVE #PPOINT (-57.46, 28.10, -104.91, 17.94, 44.44, -103.02)
LMOVE #PPOINT (-56.31, 27.87, -105.32, 16.29, 44.35, -101.80)
LMOVE #PPOINT (-55.11, 34.53, -93.73, 16.07, 39.37, -102.55)
DELAY 1
LMOVE #PPOINT (-51.97, 33.99, -94.67, 11.07, 39.19, -98.63)
DELAY 1
LMOVE #PPOINT (-52.81, 27.29, -106.28, 11.25, 44.13, -98.12)
LMOVE #PPOINT (-56.31, 30.09, -106.09, 15.49, 47.24, -100.66)
LMOVE #PPOINT (-55.11, 36.47, -94.47, 15.22, 41.96, -101.44)
DELAY 1
LMOVE #PPOINT (-53.02, 36.11, -95.14, 12.07, 41.89, -99.04)

```

Fig.13 Portion of the robot control program writing the letter in Fig.11

motion) 기능을 써서 오프라인 프로그래밍하는 것을 보여주고 있으며 Fig.13은 이 오프라인 프로그램을 포스트프로세싱하여, J_s-10 로봇을 위해 AS언어로 작성된 로봇 제어프로그램의 일부를 보여주고 있다. 이 로봇 제어프로그램을 J_s-10 로봇에 다운-로드 하여 실제로 로봇을 구동하면서 그 동작을 검증하였고, Fig.14는 이 제어프로그램을 이용 로봇이 글씨를 쓰는 모습을 보여주고 있다.

6. 결 론

GUI와 CAD의 기술을 응용 로봇의 오프라인 프로그래밍 환경을 제공하는 GIROB 시스템이 국내의 기술로 개발되었다. GIROB 시스템은 아크용접, 점용접 등의 용접작업, 부품을 이동, 조립시키는 핸들링(handling)작업 그리고 도장작업을 주요한 응용분야로 삼고있다. 실제로 로봇의 사용분야 중에서 용접, 조립 및 도장작업이 전체수요의 70~80%를 차지하고 있다. GIROB을 사용함으로써 얻을 수 있는 기대효과는 다음과 같다.

- 1) 로봇 프로그래밍 작업의 질적 향상 및 비용절감
GIROB 시스템은 컴퓨터 그래픽을 이용하여 실제 외부세계(real world)를 컴퓨터



Fig.14 J_s-10 writing the letters "WELCOME TO GIROB"

상에 구현하여 시각화 함으로써 사용자가 실제 작업환경에 접근할 필요없이 로봇을 프로그래밍할 수 있고 로봇의 동작을 시뮬레이션을 통해 상세히 분석하여 잠재적인 문제점들을 해결할 수 있으므로 프로그래밍 작업의 질적 향상과 생산라인의 설계에 소요되는 비용을 절감할 수 있다.

2) CAD 데이터의 활용

일반 제조업체에서는 설계시 CATIA 등의 CAD 시스템의 활발한 사용으로 인하여 많은 부품 또는 공작물들이 이미 CAD 데이터화 되어있다. 그러므로 이 공작물에 대한 CAD 데이터를 직접 활용함으로써 로봇의 동작경로를 정확하게 프로그래밍할 수 있고 로봇 프로그래밍에 있어서의 프로그래밍/테스트/최적화를 실현할 수 있다. 더구나 로봇의 작업경로가 기하학적으로 복잡한 경우는 더욱 더 큰 부가가치를 부여할 수 있다.

3) 국산 로봇에 효과적인 응용

현재 국내 다수의 대기업들은 사용중인 외국산 로봇을 자사가 개발중인 로봇으로 대체할 계획을 가지고 로봇연구와 개발에 많은 노력을 기울이고 있다. 그러므로 GIROB의 개방성을 활용, 개발중인 로봇들의 프로그래밍 환경으로 실험 사용 함으로

서 개발기간의 단축을 가져 올 수 있다. 더구나 대부분의 이 기업들은 CATIA를 CAD 모델러로 사용하고 있으므로 신속하고 효과적으로 GIROB을 사용 할 수 있을 것이다.

미래에 좀더 유용한 시스템이 되기위해 GIROB 시스템에 추가로 연구되어야 할 사항은 다음과 같다. 현재의 GIROB은 한대의 로봇에 대한 프로그래밍 및 시뮬레이션만을 고려하고 있는데 실제 생산현장에서는 여러대의 로봇이 동시에 작업을 하므로 여러대의 로봇에 대한 프로그래밍 및 시뮬레이션을 고려해야 한다. 여러대의 로봇을 프로그래밍하고 시뮬레이션하는 것도 기본적으로 하나의 로봇과 다를 바 없으므로 이미 구현된 기능들을 이용하여 확장이 가능하다. 그러나 이때 주의해야 할 것은 여러대의 로봇이 동시에 동작하므로 시뮬레이션할때 각 로봇의 움직임을 다른 로봇의 움직임과 동기화(synchronization)시켜야 하고 또한 로봇과 로봇, 로봇과 주변기기와의 I/O신호를 처리할 수 있는 방안에 대한 추가적인 연구가 필요하다.

또한 일반적으로 오프라인 프로그래밍에서 가장 큰 문제점으로 지적되는 것은 컴퓨터 내부 모델과 외부 현실 세계의 물체와의 매칭(matching) 문제이다. 그러나 오프라인 프로그래밍된 응용프로그램을 다운-로드 한 후에 부품이나 픽스처(fixture) 등의 위치를 전통적인 교시 방법에 의해 다시 지정함으로써 이런 문제점을 어느 정도 줄일 수 있다. 이와 같이 비록 완전한 작업이 오프라인으로 프로그래밍되지 않는 못한다 하더라도 응용프로그램의 상당한 부분이 오프라인으로 이루어지는 것이 가능하고 이것만으로도 실제 작업에 적용시 경제적으로 상당한 잇점이 된다고 할 수 있겠다.

후 기

본 연구는 한국 IBM의 재정적인 지원과 기아 자동차 및 기아기공의 기술적인 협조에 의해 수행 되었습니다. 관계자 여러분께 심심한 감사의

뜻을 표합니다. 또한 GIROB 시스템 구현에 공헌한 MIT의 이상찬박사와 서울대학교의 한영현 학생께 감사드립니다.

참 고 문 헌

1. KAWASAKI Heavy Industries, LTD. Robot Division. "KAWASAKI Operation Manual, A series for KAWASAKI J_s-10", 1990
2. KAWASAKI Heavy Industries, LTD. Robot Division, "AS Language Reference Guide, KAWASAKI Robot Controller A Series", 1990
3. Asada, H. and Slotine, J. J., "Robot Analysis and Control", John-Wiley and Sons, 1985
4. Paul, R. "Robot Manipulators", MIT Press, 1982
5. Boyse, J., "Interference Detection Among Solids and Surfaces", Commun. ACM, Vol.22, pp.3-9, 1979
6. Kim, J. and Gossard, D., "Reasoning on the Location of Components for Assembly Packaging", Transactions of the ASME Journal of Mechanical Design, Vol.113, December, pp.402-407, 1991
7. Mantyla, M., "An Introduction to Solid Modeling", Computer Science Press, 1987
8. Mortenson, M., "Geometric modeling", John-Wiley and Sons Mechanical Design, Vol., 1985
9. 이상찬, "자동차 외형설계를 위한 CAD시스템의 개발", 박사학위논문. 서울대학교 대학원 기계설계학과, 1992
10. Silicon Graphics, Inc., "Graphics Library Programming Guide", 1992
11. Farin, G., "Curves and Surfaces for Computer Aided Geometric Design : A Practical Guide", Academic Press, Inc., 1990