한국공작기계학회 '99년도 추계학술대회 논문집 pp.565~570

산업용 로봇의 3차원 오프라인 그래픽 시뮬레이터 개발

Development of a 3D Off-Line Graphic Simulator for Industrial Robot

이병국[1], 장영희[2], 김종수[3], 김용조[4], 한성현[4]

1. (주) 대우국민차
2. 부산대 메카트로닉스학과 대학원 (TEL : +82-51-1456)
3. 경남대 기계설계학과 대학원 (TEL : +82-551-249-2590)
4. 경남대 기계자동화공학부 (TEL:+82-551-249-2624, E-mail : shhan@kyungnam.ac.kr)

Abstracts

In this paper, we developed a Windows 95 version Off-Line Programming System which can simulate a Robot model in 3D Graphics space. 4 axes SCARA Robot (especially FARA SM5)was adopted as an objective model. Forward kinematics, inverse kinematics and robot dynamics modeling were included in the developed program. The interface between users and the OLP system in the Windows 95's GUI environment was also studied. The developing language is Microsoft Visual C++. Graphic libraries, OpenGL, by silicon Graphics, Inc. were utilized for 3D Graphics.

1. 서 론

현재 산업계는 생활향상에 따른 노동력의 고임금화와 소비시장의 다품종 소량생산 요구가 서로 상반되어서 상당한 진통을 겪고 있다. 주된 시장변화의 추세는 제품의 다양화와 제품 수명의 단축 등인 반면, 이를 만족시키기 위한, 작업의 빈번한 변화에도 적응할 수 있는 숙련된 기능공은 이전의 국내 사정과 달리 상당한 임금을 요구하고 있기 때문이다.

이로 인하여 근간 각 기업을 중심으로 생산성 향상을 위한 로봇의 도입을 통한 공장 자동화가 산업 전반에 걸쳐 진행되었었고, 또한 진행되고 있다. 대 부분의 로봇에 의한 자동화 교시작업이나, 변경된 시스템의 성능실험은 온라인 방식으로 수행되는 실정이어서, 현재 로봇을 통한 공장 자동화는 빈번한 작업환경 변화에 경쟁력 있게 대처하기보다는 오히려 시간과 인력을 더욱 소모하는 문제점으로 대두되고 있다.

이로 인하여 현재 각 공장의 로봇을 통한 자동화는 상당부분 포기되어 있는 상태이며, 산업계는 이에 대처하기 위하여 각종 생산 공장들을 중국과 동남아 등지로 이동시키고 있다.

앞서 거론된 바와 같이 현재 로봇을 통한 교시작업과 변경된 시스템의 성능 실험은 모두 온라인이다. 이러한 온라인 방식으로 교시, 성능 실험을 하기 위해서는 별도의 개발라인이 요구되며, 작업경험이 풍부한 전문가에 의하여 교시, 성능 실험이 이루어져야 한다.

또한 복잡한 작업의 경우에는 교시에 많은 시간이 소요되고, 시스템의 일부분의 변경에 따른 성능 평가 과정에서도 많은 시간과 비용이 소요되는 문제점이 있다.

프로그래밍을 위해서는 전체 시스템이 중단되어야 하며, 로봇의 오동작에 의한 안전사고의 위험성도 상당부분 내재되어 있다.[1]

이러한 문제점은 실제 작업과 유사한 환경을 가진 시뮬레이터인 오프라인 프로그래밍 시스템(Off-Line Programming System)을 이용함으로써 해결될 수 있다. OLP 시스템은 컴퓨터 그래픽스의 방법에 의하여 충분히 확장된 로봇 프로그래밍 언어로서 로봇에 직접 접근하지 않고도 로봇 프로그램을 개발할 수 있는 시스템을 지칭한다[2].

그러므로 개발되는 시스템을 이용하면 로봇의 가동중에도 동적 시뮬레이션이 가능하며, 작업교시, 궤적계획, 제어 알고리즘 등의 개발 및 성능평가를 소프트웨어로 수행할 수 있다.

따라서 오프-라인 방식으로 교시와 성능실험을 수행하면 생산라인의 작업중단도 방지할 수 있고, 동시에 작업에의 적용과 성능평가도 용이하게 된다[2][3].

이미 개발된 OLP 시스템에는 ROSI2, STAR, SILMA사에서 개발한 CimStation, Robot Simulation

Ltd.의 WORKSPACE, BYG Gsystem Ltd.의 GRASP, Tecnomatix Technologies Ltd.의 ROBCAD, 그리고 Deneb Robotics Inc.의 IGRIP등이 있다[4].

현재 대부분의 오프라인 스템은 Graphic User Interface가 지원 가능하도록 만들어지는 추세이며, 지금까지 개발된 대부분의 오프라인 프로그래밍 시스템은 워크스테이션에서 운용되어야 하고, 고가인 관계로 아직까지 보편화되지 못한 상태이다.

이에 본 연구에서는 스카라(SCARA)형 로봇에 대한 3차원 그래픽 시뮬레이터인 OLP 시스템을 현재 일반 PC의 OS인 Windows 95 버전으로 개발하였다.

전용 로봇 모델은 삼성 SCARA 로봇 SM5를 사용하며, 이 로봇에 대한 정기구학, 역기구학 해석을 프로그램에 이식하였으며, 로봇의 동역학을 모델링하여 프로그램에서 모의실험이 가능하게 하였다.

Windows 95의 Graphic User Interface 환경을 이용한 사용자와 OLP 시스템의 인터페이스에 대하여 연구하였으며, 적절한 궤적 계획과 제어방식 선택이 가능하도록 구성되어 있다.

개발 언어로는 Microsoft 사의 Visual C++을 사용하였으며[5], 실리콘 그래픽스사의 그래픽 라이브러리인 OpenGL을 사용하여 3차원 그래픽이 필요한 부분을 보강하였다[6].

2. 오프라인 프로그래밍 시스템이 갖추어야 할 요소

본 연구에서 개발되어진 프로그램은 편리한 사용자 인터페이스를 위하여 다양한 그래픽적 기능들을 보유한다. 이러한 기능들을 위해서는 3차원적 데이터베이스의 구축 및 그래픽적 알고리즘의 구축이 필요하게 된다.

특히 본 시뮬레이터는 PC에서 운용될 수 있도록 구축되고 있으므로 여기에 따른 문제점들을 해결해야만 한다.

은선 및 은면제거기술이나 음영처리기술은 PC에서 이용되기가 상당히 까다로우며, 시뮬레이터에 적용하기 위해서 기존의 알고리즘을 수정·보완하여 적용하였다.

3차원적 그래픽을 위한 알고리즘의 개발로 시뮬레이션 결과를 3차원 영상으로 볼 수 있게 된다. 이를 위해서 실제 좌표계에서 시각 좌표계로의 변환 및 원근투영 기술이 포함되었다.

그래픽을 이용한 시뮬레이터의 경우 가장 빈번히 사용되는 기능중의 하나가 시각위치 변경기능이므로 본 연구에서는 보다 편리하게 시각위치를 변경할 수 있는 기능을 갖추었다.

오프라인 프로그램내의 로봇과 다른 물체들에 대해 은선, 은면을 제거하고 음영 처리한 결과는 와이어 프레임으로 물체를 표현한 그림들에 비해 상당히 현실감을 높일 수 있음을 알 수 있다.

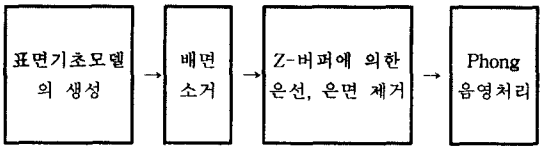
그러나 처리속도의 저하로 인하여 은선, 은면 제거와 음영처리된 상태로서 실시간 처리는 힘든 상황이고 사용자가 한 화면을 지정하여 음영 처리된 화면을 볼 수 있도록 프로그램이 구성되어 있다.

은선 및 은면의 제거를 위하여 스켄 라인 z-버퍼 방식을 채택하였고 개인용 컴퓨터 상에서 구현이 가능하게 하였다.

이 방식은 메모리 사용을 극단적으로 줄여 개인용 컴퓨터에서 실현이 가능하지만 실행속도의 저하를 감수해야만 한다.

음영처리 알고리즘으로는 Phong 음영처리 방식을 사용하였는데 이 방법은 면의 법선 벡터를 이용하여 면의 밝기를 계산하는 방식으로 우수한 음영처리 결과를 보여준다.

아래의 그림은 은선, 은면 제거와 음영처리의 개략적인 흐름도를 보여준다. 먼저 표면기초모델(surface based model)을 생성하고 그후 시점으로부터 보이지 않는 배면을 소거한 후 z-버퍼법에 의한 은선, 은면 제거를 한 후 음영처리를 하는 순서로 진행된다.



OLP 시스템 개발시 꼭 포함되어야 할 사항으로 R. Bolles와 B. Roth는 다음의 5가지 요소들을 언급하였다[2]. 물론 그 외 고려되어야 할 여러 가지 요소가 있겠지만, 일단 본 논문에서는 Bolles와 Roth가 언급한 다음의 5가지 요소에 집중하여 프로그램을 개발하였다.

2.1 사용자 인터페이스

OLP 시스템은 사용자의 사용방법 습득 및 프로그램 데이터 관리가 쉽고 편해야 한다. 그 외에 로봇 시스템의 교시상자를 대체할 수 있는 컴퓨터상의 사용자의 사용자 인터페이스 방식 개발이 필요하다[2][3].

프로그램을 개발하면서 제시되었던 이러한 문제의 상당부분은 이미 Windows 95 버전으로 Upgrade 하면서 해결되었다. 이는 Windows 95가 기본적으로 사용자들의 편리를 위하여 GUI환경을 지원하기 때문이다[5]. OLP 시스템 Windows 95 버전에서는 OS에서 기본적으로 제공되는 여러 가지 대화상자 및 마우스 기능, 단축키 기능 등을 사용하여 프로그램 관리 및 사용방법을 쉽게 할 수 있었다. 결국 교시상자는 대화상자상에서 제공되는 여러 기능을 일부 변형 제작하여 최대한 쉽게 사용하게 하는데 집중했다.

2.2 3차원 모델링

컴퓨터 화면상에 제시되는 로봇 자체와, 공구, 작업 셀등이 모두 3차원 물체로 모델화되는 것을 요구한다. 또한 이 3차원 모델화된 화면상의 물체들은 애니메이션 기능을 가져야한다[2][3].

이 문제에 대해 개발된 프로그램에서는 실리콘 그래픽스사에서 제공하는 그래픽 라이브러리인 OpenGL를 이용하여 3차원 모델링을 구성하였다. 이 라이브러리의 사용으로 인하여 상당한 속도의 향상이 가능했으며, 고정밀도 애니메이션시 기존의 Win- dows용 GDI(Graphic Device Interface)에서 존재했던 약간의 깜박거림도 없앨 수 있었다[6].

2.3 기구학적인 에뮬레이션

3차원 시뮬레이션 되어진 세계를 유효하게 유지하기 위해서 필요하다. 대개 로봇의 정기구학과 역기구학을 프로그램에 포함시킴으로서 가능하게 된다[2][3].

SCARA 로봇 SM5의 정, 역기구학이 크게 어렵지 않으므로 개발된 프로그램에서는 수식을 풀어서 부함수(Subroutine)를 따로 작성하였다. 여기에 기존 로봇의 Link 길이 변화, 무게변화에 대처할 수 있도록 각종 Parameter 설정도 사용자가 할 수 있도록 하는 기능도 첨가시켰다.

2.4 경로 계획 에뮬레이션

오프라인 프로그래밍 시스템은 공간상에서 매니퓰레이터가 취하는 경로를 정확히 에뮬레이트 하여야만 한다. 이 에뮬레이션은 로봇의 작업을 설정하는 부분과 충돌을 정확히 예측하는 부분에서 필요하다[2][3].

이 문제에 대해 개발된 프로그램은 5개의 경로 계획 방식을 제공하여 선택하여 사용하도록 하고 있다. Windows 95의 대화상자를 이용한 가상 교시상자를 구성하여 경로계획을 지원한다.

2.5 동역학적 에뮬레이션

만약 오프라인 프로그래밍 시스템이 제어기의 경로계획 알고리즘을 잘 에뮬레이트하고, 실제 로봇이 매우 느린 속도로 원하는 경로를 쫓아간다면, 프로그램에서 모의되는 동작의 동적인 특성은 무시할 수가 있다. 그러나 아주 빠른 속도로 작업이 지시된 경우나 아주 높은 하중상태에서 경로를 추적하는데 발생하는 오류는 중요할 수 있다. 이러한 추적 오류를 시뮬레이트하기 위하여 정확한 동역학적 에뮬레이션이 필요하다[2][3].

이 문제에 대해 개발된 프로그램은 비 보존계에 대한 Lagrange-Euler 공식을 이용한 SCARA Robot의 동역학 방정식을 구했고, C++로 코딩하여 부함수화 하였다. 구해진 동역학 방정식에는 로봇 매니퓰레이터를 구동하는 서보계인 DC 모터에 대한 운동방정식도 함께 고려하여 보다 실제시스템에 가까운 모델링을 하고자 하였다.

3. 개발된 OLP 시스템의 구성 및 기능

3.1 OLP 시스템의 전체 구성

그림 1은 개발된 OLP 시스템의 전체 구성을 나타낸다. OLP 시스템은 크게 Setting 및 Teaching, 시뮬레이션, 성능 평가의 3부분으로 크게 나뉠 수 있다. Setting 및 Teaching 부분에서는 Parameter 설정, 경로점 교시, 궤적 생성 등이 수행되며, 시뮬레이션 부분에서는 주어진 여러 제어 알고리즘에 의한 동역학 및 제어 시뮬레이션을 할 수 있고, 성능평가 부분에서는 수치 시뮬레이션 데이터에 의한 3차원 애니메이션과 제어알고리즘의 추종 성능 검증이 수행된다.

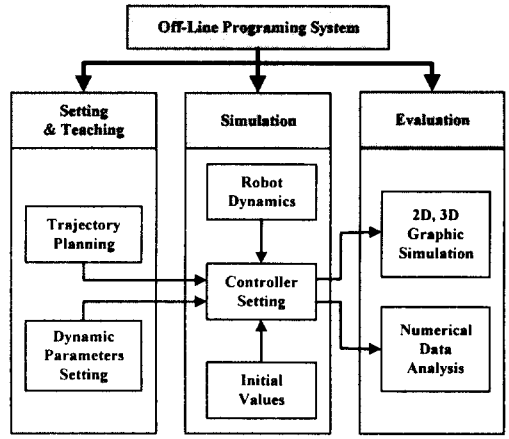


그림 1. 오프라인 프로그래밍 시스템의 구성

3.2 OLP 시스템의 시작화면

프로그램을 시작시키면 그림 2와 같이 풀다운 방식의 메뉴를 기본 틀로 하는 초기창이 뜨며, 이 초기창의 클라이언트 구역에는 스카라 로봇이 그려지도록 구성되어 있다. 이때 스카라 로봇은 모든 Joint 값이 0인 상태로 기본자세를 취하게 된다.

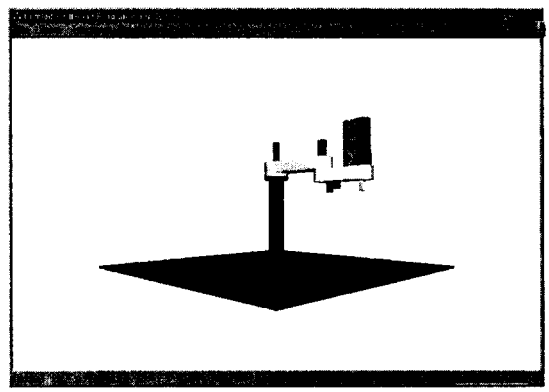


그림 2. OLP 시스템의 초기창 모습

3.3 OLP 시스템의 메뉴 구성

개발된 OLP 시스템의 풀다운 메뉴 구성은 다음의 표 1과 같으며, 이 메뉴중 실행 메뉴(**Execution**)는 DSP 보드를 이용한 오프라인 프로그래밍 시스템과 실제 스카라 로봇의 연결을 위한 것으로서 아직 기능을 연구 중에 있다.

표 3. OLP 시스템의 메뉴 구성

FILE	SETUP	TEACHING	SIMULATION
New	Specification	New	Control Type
Open	Work Range	Open	Run
Save..	Max. Velocity	Trajectory	Evaluation
Save as	Load		
Exit			

EXECUTION	VIEW	HELP
On-Line	Change Posture	About OLPS
Verification	Change View	
Comparison	Wire Frame	

3.3.1 파일 메뉴 및 관리

오프라인 프로그래밍 시스템은 교시점에 관한 정보, 표준 궤적 데이터, 시뮬레이션 궤적 데이터등 모든 정보를 데이터 파일로 관리한다. 각종 정보가 저장되어 있는 파일들은 개발시 선정된 몇 가지 고유한 확장자를 가지게 되며, 작업이 수행되기 전에 파일메뉴에서 새롭게 만들거나 기존의 작성된 파일을 열어야 한다. 그림 3은 파일 메뉴에서 Open 명령으로 확장자 inf 파일을 열려하는 모습이다.

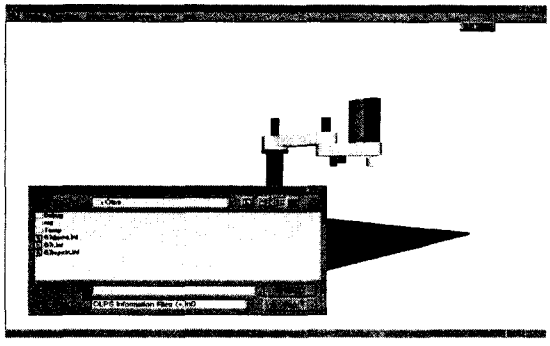


그림 3. FILE 메뉴의 Open 명령 수행

3.3.2 Setup 메뉴

Setup 메뉴에서는 작업을 수행하기에 앞서 링크별 길이 및 질량, 작업 영역, 최대속도, 그리고 하중의 종류 등을 지정할 수 있다. 링크별 길이 및 질량지정 메뉴(**Specification**)에서는 각 링크의 길이를 mm단위로, 그리고 질량을 g단위로 지정할 수 있다. 작업영역지정 메뉴(**Work Range**)에서는 각 관절의 최대 운동범위를 지정할 수 있다. 최대속도지정 메뉴(**Max. Velocity**)에서는 관절별로 최대 속도의 한계를 지정할 수 있으며 이 모든 조건들은 궤적 계획 시에 제한 조건으로 이용된다. 각 메뉴 선택 시는 해당하는 정보의 수정을 위하여 대화상자가 뜨게 되며, 이 상태에서 키보드나 마우스를 이용하여 원하는 수치들을 기입한다. 그림 4는 이 대화상자들에서 데이터를 설정하는 모습이다.

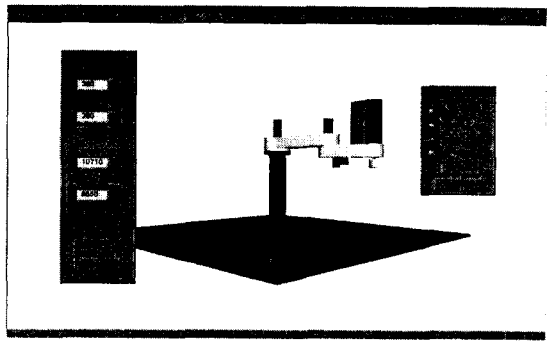


그림 4. Setup 메뉴의 Specification과 Payload 대화 상자

3.3.3 교시 메뉴 및 궤적 계획

교시는 교시 메뉴(**Teaching**)에서 대화 상자를 통해 교시 정보를 저장할 교시 파일을 만들고, 여러 교시정보를 입력한 후, 교시 파일을 저장함으로써 이루어진다. 입력하여야 할 교시 정보로는 관절 좌표계나 절대 좌표계에서의 로봇 경유점, 통과 시간, 그리고 경유점 사이의 궤적 계획법이 있다. 관절 좌표계에서의 로봇 경유점 입력은 네 관절의 위치를 각각 입력함으로써 이루어지고, 절대 좌표계에서의 로봇 경유점 입력은 로봇 말단 효과 장치의 3차원 좌표와 회전 각도를 각각 입력함으로써 이루어진다. 통과시간은 초 단위로 입력하여야 하고 궤적 계획법으로는 Cubic Spline법, LFPB(Linear Function with Parabolic Blend)법, B-Spline법, 직선 보간법, 원호 보간법등이 준비되어 있어 이중에서 한가지를 선택한다.

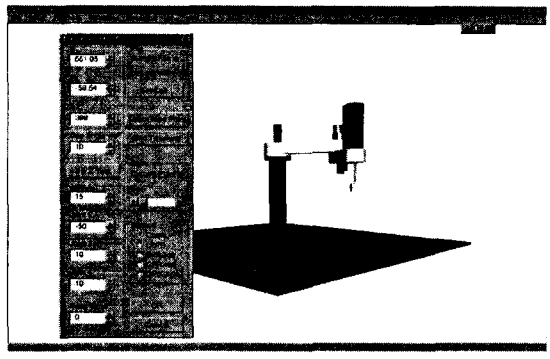


그림 5. 교시 대화 상자에서의 경유점 정보 입력

그림 5는 실제 로봇의 교시상자를 대체하는 OLP 시스템의 교시 대화상자와, 교시 대화상자에서 설정 값을 변경하였을 때, 자동적으로 자세를 바꾸는 로봇의 모습을 보여주고 있다.

3.3.4 동적 시뮬레이션 및 성능 예상

시뮬레이션 메뉴(Simulation)에서는 궤적 계획에 이은 동역학 및 제어 시뮬레이션을 통해 수행하고자 하는 작업을 3차원과 2차원 그래픽으로 컴퓨터 화면상에서 미리 볼 수 있고 이에 이은 시뮬레이션 궤적 확인에 의해 제어 추종 성능 평가가 가능하므로 로봇에 적용될 제어 알고리즘 교시의 적합성을 판단할 수 있다. 시뮬레이션에 적용될 알고리즘으로는 기존 산업용 제어에 널리 이용되는 비례-미분(PD)제어, 로봇의 동역학 해석에 기초한 계산 토크(computed torque)제어, 그리고 가변 구조 제어 이론을 근간으로 하는 슬라이딩 모드(sliding mode)제어가 준비되어 있는데 이 중 하나를 선택할 수 있다. 다음의 그림 6은 시뮬레이션 메뉴(Simulation) 에서 제어방식 선택을 하는 모습이며, 그림 7은 주어진 제어방식과 궤적에 따라 로봇이 동적 시뮬레이션을 하고 있는 장면이다.

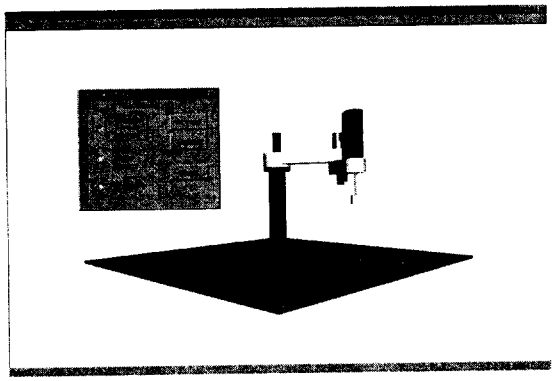


그림 6. 제어방식 선택

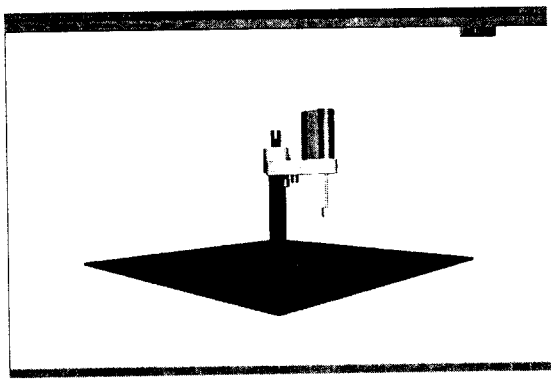


그림 7. 3차원 그래픽 공간상에서의 시뮬레이션

3.3.5 그 외의 기능

그 외의 메뉴에서는 원하는 자세로 로봇 이동, 시각위치 변경, 와이어 프레임 모델로 처리 등이 가능하다. 그림 8은 자세 변경된 상태에 시각위치 변동까지 포함된 모습이다. 특히 시각위치 변경은 마우스로 자유자재로 가능하다

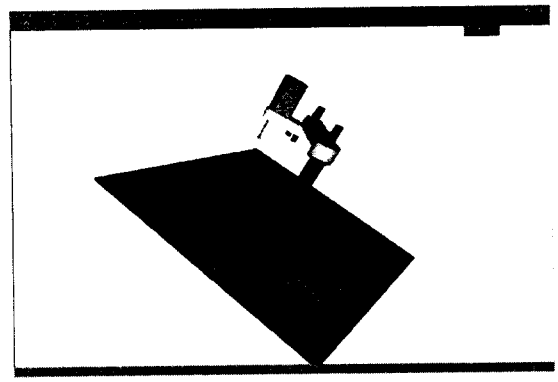


그림 8. 로봇의 자세, 시각위치 변경

그림 9는 와이어 프레임 모델로 옵션이 선택된 후의 모습이다. 이러한 모델에서도 앞에서 수행하였던 모든 작업이 동일하게 수행가능하며, 시뮬레이션도 할 수 있다.

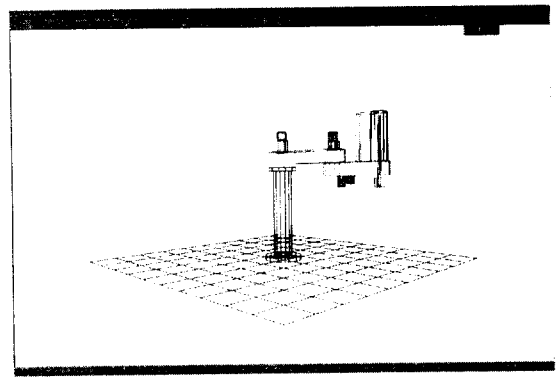


그림 9. 와이어 프레임 모델

4. 프로그램 개발 환경

개발된 오프라인 프로그래밍 시스템은 Microsoft Visual C++ 버전 4.0과 OpenGL을 이용하여 코딩되었다. Visual C++은 Window 프로그램을 쉽고 구조적으로 제작할 수 있는 MFC와 각종 Wizard(기본적인 Source를 자동으로 코딩하여주는 기능)를 제공하므로, 개발된 프로그램은 차후 다른 여러 사용자들에 의해서 간단히 확장되거나 이식될 수 있다. 각종 Visual

Editor들은 원하는 대화상자나, 메뉴 바를 간단히 제작할 수 있도록 한다[5]. OpenGL은 새로운 산업용 3D그래픽 표준으로 등장하고 있는 Library이다[6]. Windows 상에서 빠른 3D 그래픽 능력과 간단한 프로그래밍 명령어로 프로그램 개발에 유용히 활용된다.

5. 결 론

본 연구에서는 PC의 기본 OS인 Windows 95 환경하에서 운용되는 4축 스카라 로봇에 대한 3차원 그래픽 시뮬레이션 툴을 개발하였다. 3차원 그래픽, 궤적계획, 기구학, 동역학, 제어 알고리즘등이 부가되고, 그래픽 사용자 인터페이스(GUI)에 기초한, 메뉴바 및 대화상자들로 구성되는 OLP 시스템을 개발하였으며, 이를 통해 로봇의 작동과정과 구성요소의 특성을 화면상에서 평가할 수 있도록 하였다.

기존의 그래픽 공간상의 시뮬레이션과 실제 로봇의 동작과 연결시킬 수 있는 기능이 앞으로 더 연구되어야 하겠다.

참고문헌

[1] 박민조, 손권, 안두성, "PC에서 운용되는 스카라형 로보트의 오프-라인 프로그래밍 시스템," *대한 기계학회 논문집*, 제19권, pp. 568-579, 1995.

[2] R. Bolles, B. Roth, *International Symposiums of Robotics Research,*. MIT Press, Cambridge, MA, 1988.

[3] Craig, J. J., *Introduction to Robotics Mechanics and Control*, Second edition Addison-Wesley, New York, 1989.

[4] G. Wittenberg, "Developments in Offline Programming : An Overview," *Industrial Robot*, Vol.22, No.3, pp. 21-23, 1995.

[5] David J. Kruglinski, *Inside Visual C++ 4*, Microsoft Press, 1996.

[6] Ron Fosner, *OpenGL Programming for Windows 95 and Windows NT* , Addison-Wesley Developers Press, 1997.