

## 가상현실을 이용한 로봇제어 학습용 시뮬레이터의 개발

서준형\*, 홍동기\*\*, 박주현\*, 이석규\*  
영남대학교 전기공학과\*, 대우중공업 선박해양기술연구소\*\*

### Development of Robot Control Simulator for Education using Virtual Reality

J.H. Seo\*, D.G. Hong\*\*, J.H. Park\*, S.G. Lee\*  
Dept. of Electrical Eng., Yeungnam University\*, Daewoo Heavy Industries\*\*

**Abstract** - 네트워크 환경의 보편화와 함께 원격강의와 가상실험에 대한 연구가 진행되고 있다. 로봇공학교육에 있어서 가상현실은 가상실험을 구현하기 위한 밑바탕이 되며 값싸고 위험부담없이 실제실험과 유사한 환경을 구축하기 위한 도구가 된다.

본 논문에서는 월드와이드웹 상에서 동작가능하며 다양한 궤적과 제어기를 사용자가 조작하면서 비교·학습할 수 있는 로봇제어 학습용 시뮬레이터를 개발한다. 가상로봇은 실제로 로봇과 유사한 특징을 가지며 개인값 설정에 따른 궤적추종을 로봇의 움직임으로 관찰할 수 있다.

## 1. 서론

최근 월드와이드웹(WWW)과 네트워크의 보편화로 인해 네트워크 환경을 이용한 원격강의의 연구가 활발히 진행되고 있다. 원격강의는 시간과 장소에 제약은 받는 기존의 강의에 비해 유연성과 더불어 다양성을 부여한다. 원격강의에 대한 연구가 진행됨에 따라 수업에서의 이해도를 높이기 위한 상호작용적인 가상실험의 필요성이 높아지고 있다.

로봇공학 교육에 있어 가상현실은 실제 실험을 하기에 어려운 부분들도 가상실험으로 구현하기 위한 도구가 되며 값비싼 로봇이나 작업을 모델링함으로써 위험부담없이, 또 실제 실험에서 관찰할 수 없는 각도에서도 실험·관찰할 수 있는 환경을 제공한다.

일반적으로 로봇 제어이론의 학습을 위해서는 먼저 기구학, 동역학, 궤적계획 등의 지식을 필요로 하는데 그 복잡도로 인하여 제어이론의 학습 후 프로그래밍언어를 이용하여 실제 로봇제어 시뮬레이션을 수행할 수 있는 단계에 이르기까지 오랜 시간을 필요로 한다. 기존의 여러 패키지들이 로봇제어와 연계하여 발표되어 왔으나 대부분 프로그래밍 언어를 위한 라이브러리이거나<sup>(1)</sup> 상용 수치해석 프로그램의 애드온형태이며<sup>(2),(3)</sup> 결과출력방식에 서 수치출력 혹은 그래프출력에 의존하고 있다.

따라서 본 논문에서는 WWW환경을 기반으로 실행가능하고 가상현실을 이용하여 학생들이 손쉽게 접근할 수 있는 로봇제어 학습용 시뮬레이터를 개발하였다. 가상현실을 이용하여 가상의 로봇 매니퓰레이터를 구성함과 다양한 기능을 제공함으로써 사용자의 궤적입력과 제어기선택 및 개인설정 에 따른 로봇움직임을 직접 시뮬레이터를 통하여 관찰할 수 있으며 여러 제어기의 성능과 동작을 전체적으로 비교·학습할 수 있다.

## 2. 본론

시뮬레이터를 구성하기 위하여 본 논문에서는 Java언어와 Java3D API(Application Programming Interface)를 이용한다. Java언어는 운영체제 중립적인 언어로써 운영체제의 종류가 다르더라도 자바환경이 구현된 시스템에서는 실행파일의 수정없이 실행할 수 장점이 있으므로 인터넷관련 어플리케이션 제작에 유리하다. Java3D는 객체지향적인 구조를 가지는 고차원의 3D

그래픽 라이브러리로써 상호작용적인 3D 그래픽 프로그램을 빠르고 효율적으로 개발할 수 있도록 도와준다.<sup>(4),(5)</sup>

### 2.1 가상로봇의 구현

#### 2.1.1 대상로봇의 개요

가상로봇의 구현을 위하여 먼저 대상로봇을 SCARA 로봇으로 정하였다. <그림1>은 3D로 구현할 삼성 FARA SM3의 개략도이다. 기본적인 사양 및 구조는 SM3의 사양과 동일하게 맞추어 설계하였으나 3D좌표의 최소화를 위하여 많은 부분 간략화 되었다.

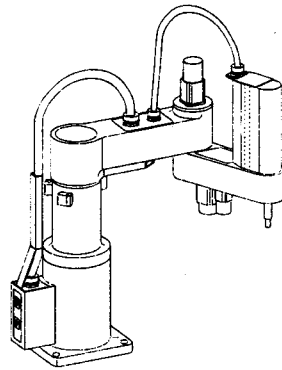


그림 1. SCARA ROBOT

스카라 로봇은 4축의 구조를 가지고 있다. 1축과 2축은 평행하게 운동하며 3축은 수직으로 직선운동을 한다. 4축의 경우에는 회전운동만을 한다. 본 논문에서는 계산 시간을 줄이고 모델링에서의 복잡도를 낮추기 위하여 3차원공간에서 말단장치(end-effector)의 위치를 결정하는 3축까지만을 구현하였다.

#### 2.1.2 SCARA 로봇의 모델링

3차원 공간상에 로봇을 구현하기 위해서는 먼저 로봇의 부분을 적절히 나누고 원점을 기준으로 각 부분을 모델링한다. 그 다음 프로그래밍 과정에서 적절한 연산을 통하여 각 부분들을 조립한다. 각 부분의 모델링을 하기 위한 방법으로는 크게 두가지로 나눌 수 있는데 첫번째는 그래픽틀을 이용하여 모델링한 후 VRML(Virtual Reality Modeling Language)파일이나 널리 알려진 3D포맷으로 저장한 후 프로그램내에서 실행시에 읽어들이 이용하는 방법이 있고, 다른 방법으로는 각각의 점을 직접 계산하여 Java3D의 문법으로 코딩하는 방법이 있다. 여기에서는 수행속도를 최대한 높이기 위하여 필요한 좌표를 자동생성하는 프로그램을 작성하여 후자의 방법으로 구현하였다.

<그림2>는 Java3D의 장면표시법으로 설계한 로봇의

개략도이다. Java3D에서 장면은 트리구조로써 나타낼 수 있으며 수개의 가지그래프(branch graphs)로써 이루어진다. 각각의 노드는 <그림2>에서 원형으로 표시되는 그룹과 삼각형으로 표시되는 리프노드로 나눌 수 있다. 그룹은 다른 그룹과 리프를 하나의 유닛으로 합치는 역할을 하며 리프는 하위노드를 가질 수 없는 노드로써 객체에 대한 정보를 포함한다. Shape3D는 좌표정보, 법선벡터 및 객체의 색상정보를 가지며 Transform3D는 회전, 이동변환을 위한 정보를 가진다.

그림에서 Locale의 오른쪽에 위치한 BranchGroup은 사용자의 위치와 방향을 포함하는 ViewPlatform이 설정되어 있으며 이의 의해 화면상에 적절하게 시각화된다. Locale의 왼쪽부분은 가상로봇의 좌표정보와 변환의 설정이다. 상위노드에서의 변환은 하위노드에도 적용된다. 따라서 로봇의 0축을 조립한 후 그 하위노드에서 0축의 높이만큼의 위치이동후 1축의 덧붙이는 방식으로 구현되었다. 점선으로 표시된 Transform3D 노드는 제어기에서의 입력값 혹은 조작패널의 입력에 의해 각 관절을 움직여 주기 위해 마련되었는데 로봇의 작업공간에 따른 한계값의 설정과 동역학과 기구학의 수치적인 부분의 구현으로 실제로로봇과의 차이를 줄였다.

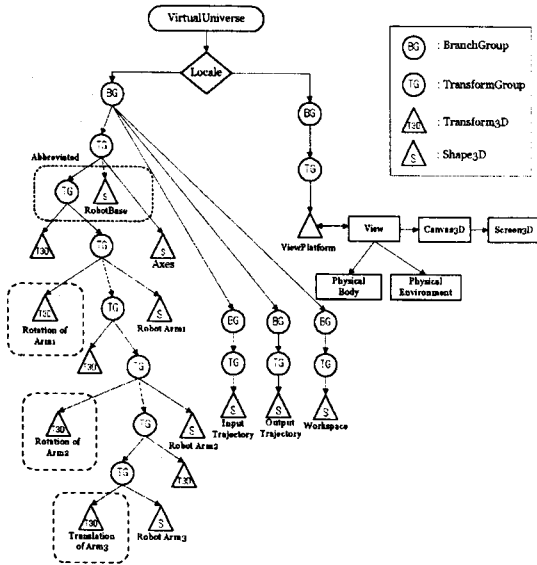


그림 2. 가상로봇의 장면개략도

## 2.2 시뮬레이터의 구조

본 논문에서 개발할 시뮬레이터는 여러 종류의 입력제어와 제어를 수용할 수 있어야 한다. 따라서 프로그램의 유연한 구현을 위하여 일반적인 제어시스템의 구조를 수정하여 <그림3>에 보인 것과 같이 클래스로 구성하였다.

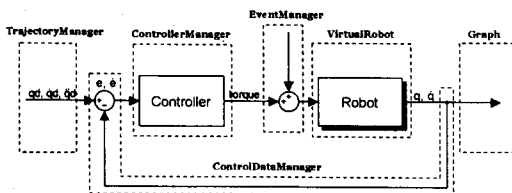


그림 3. 일반적인 제어시스템 구성과 클래스 구성

<그림4>는 <그림3>에서 보인 클래스들간의 자세한 관계도이며 각 클래스들의 주요한 기능은 다음과 같다.

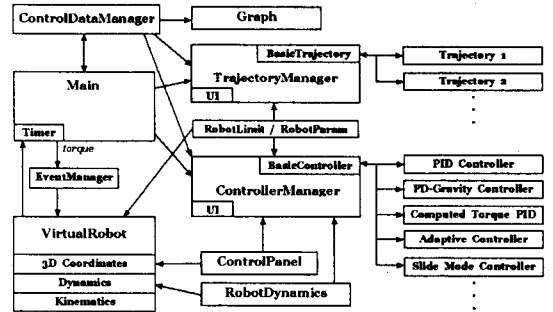


그림 4. 클래스간의 관계도

- RobotLimit: 로봇의 작업공간의 설정값으로써 가상로봇이 작업공간이 벗어나지 못하도록 제한하기 위한 참조값으로 이용되며 제적관리자에서는 생성된 제적의 유효성 검사에 이용된다.
- VirtualRobot: 가상로봇의 구현클래스. 3차원 좌표정보, 동역학정보, 기구학정보를 담고 있다. 입력토크를 받아 각 링크의  $q, \dot{q}$ 를 출력하는 기능을 가지고 있다.
- ControlDataManager(CDM): 각 계산시스템마다 입·출력제적의 직교좌표값, 입·출력 관절좌표값, 에러, 입력토크 등 시뮬레이터의 데이터를 총괄하여 저장하기 위한 클래스.
- TrajectoryManager: 구현된 입력제적의 관리자. 입력제적은 기본적인 제적생성의 메소드를 제공하는 BasicTrajectory를 확장하여 구현한다. 각각의 제적은 제적의 설정을 위한 UI(User Interface)를 가지고 있다. 이는 제적관리자 내의 UI와 함께 조립되어 제적설정시 사용자의 세부설정을 위해 메인루프로 보내진다.
- ControllerManager: 구현된 제어기의 관리자. 각각의 제어기는 BasicController의 확장으로 구현된다. 제어기는 필요에 따라 로봇의 동역학에 관한 메소드를 지원하는 RobotDynamics 클래스의 참조할 수 있다. 제적관리자와 동일하게 각 제어기는 개인설정 등을 위한 UI를 내장하고 있으며 이는 제어기에 선택시 제어기관리자에 의하여 제어기관리자의 UI 함께 조립되어 메인루프로 보내진다.
- EventManager: 외란이나 부하 등을 모델링하기 위하여 구현되었으며 메인루프에서 입력토크를 받아 적절한 연산후 결과를 가상로봇으로 넘겨게 된다.
- ControlPanel: 버튼조작으로 로봇링크를 움직이기 위한 클래스. 로봇의 초기위치를 사용자로부터 입력받기 위해 이용할 수 있다.
- Main: 초기화작업을 수행하며 CDM의 참조값을 제적관리자와 제어기관리자로 넘겨준다. 또한 각각의 클래스에서 UI를 받아 셋업화면을 생성하며 사용자의 동작신호에 맞추어 타이머를 동작시키고 타이머의 주기마다 제어기관리자로부터 입력토크값을 받아 이벤트관리자로 넘긴다. 이벤트관리자에서 출력된 토크값에 의해 가상로봇이 동작하게 된다.

### 2.3 시뮬레이터의 동작

〈그림5〉는 웹브라우저 상에서 실행된 시뮬레이터의 화면이다. 웹브라우저상에는 가상로봇만이 보이고 다른 설정은 외부의 프레임으로써 실행된다. 웹브라우저의 좌측은 컨트롤패널과 메인프레임이 위치해 있는데 메인프레임에서 궤적과 제어기를 선택하고 실행함으로써 가상로봇을 동작시킬 수 있다.

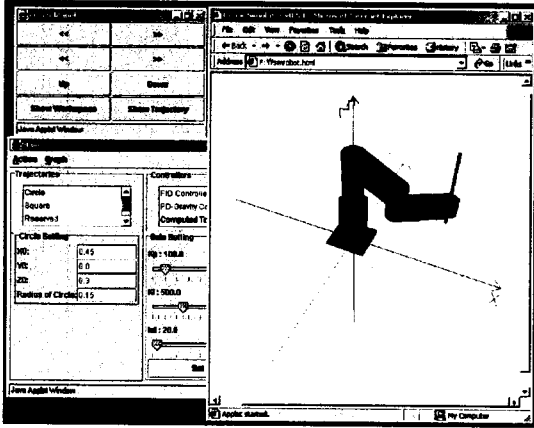


그림 5. 웹브라우저 상에서 실행된 시뮬레이터의 화면

〈그림6〉은 작업공간(Workspace)을 표시한 상태에서 다양한 각도에서 캡처한 그림이다. 마우스를 통하여 가상로봇 상에서 조작함으로써 로봇의 다른 각도에서 관찰할 수 있다.

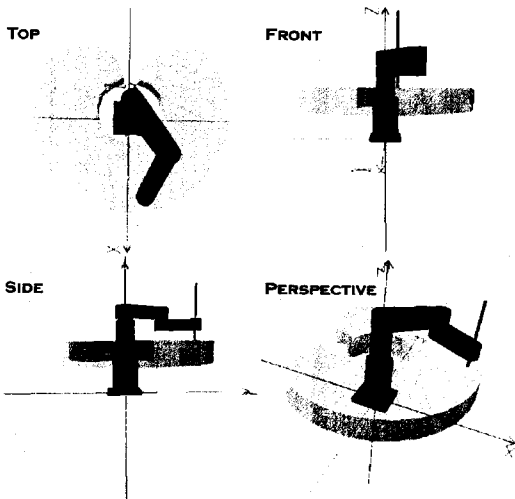


그림 6. 여러 관점에서 바라본 가상로봇

〈그림7〉은 원궤적에 대하여 계산된 토크방법을 선택하여 시뮬레이터를 동작시킨 후 4장의 화면으로 캡처한 화면이다.

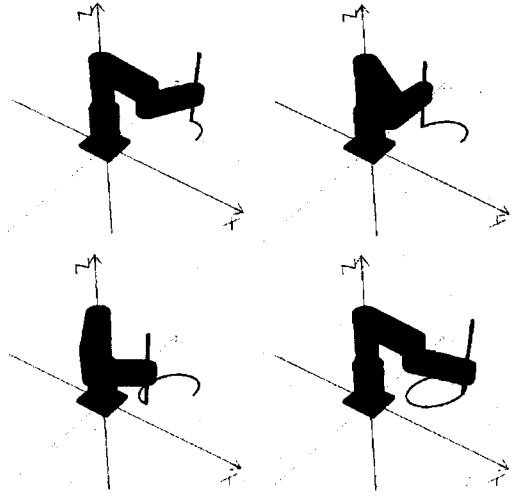


그림 7. 원궤적에 대한 시뮬레이터의 동작화면

### 3. 결론 및 향후 과제

본 논문에서는 가상현실과 자바를 이용하여 월드와이드웹 환경에서 동작하는 로봇제어 학습용 시뮬레이터를 개발하였다. 이 시뮬레이터는 원궤적의 또는 일반수업시의 여러 제어기에 대한 이해를 돕고 각 제어기의 장단점 등을 비교학습하는 보조자료로써 충분히 활용가능하다.

문제점으로 3차원 그래픽을 위한 연산과 제어시스템의 연산을 위해 일반적인 어플리케이션이 동작하는 환경보다 훨씬 더 높은 사양의 PC와 그래픽 카드를 요구하는 문제점이 있지만 이는 현재의 PC환경의 발전속도를 볼 때 곧 해결될 것이다.

향후 과제로 빠른 속도를 위한 최적화와 더불어 교육 효과의 극대화를 위해 이론적 설명을 위한 도움말이 제공되어야 하겠고, 각 제어기를 특징화할 수 있는 그래픽 출력이 지원되어야 하겠다. 또한 스카라로봇뿐만 아니라 다양한 로봇의 모델을 지원하여야겠고, 새로운 제어기를 온라인에서 직접 설계해 볼 수 있도록 자체문법의 도입을 할 수 있다. 또한 비전과 원궤제어 등을 연계하여 월드와이드웹 상에서 가상의 로봇의 움직임과 실제 로봇의 움직임을 비교하는 방식도 고려할 수 있겠다.

#### (참 고 문 헌)

- [1] Gourdeau, R., "Object-Oriented Programming for Robotic Manipulator Simulation", IEEE Robotics & Automation Magazine, Sep. 1997.
- [2] Nethery, J.F. & Spong, M.W., "Robotica: A Mathematica Package for Robot Analysis", IEEE Robotics & Automation Magazine, March 1994.
- [3] Corke, P.I., "A Robotics Toolbox for MATLAB", IEEE Robotics & Automation Magazine, March 1996.
- [4] Sowizral, H.A. & Deering, M.F., "Projects in VR: The Java 3D API and Virtual Reality", IEEE Computer Graphics and Applications, May/June 1999.
- [5] Sun Microsystems, Java 3D™ API Specification v1.2, Sun Microsystems, April 2000.
- [6] Lewis, F.L., Abdallah, C.T., & Dawson, D.M., Control of Robot Manipulators, Macmillan, 1993.
- [7] Sciacivco, L. & Siciliano, B., Modeling and Control of Robot Manipulators, McGraw-Hill, 1996.